
Nyoka Documentation

Release 3.1.0

maintainer

Jul 03, 2020

Contents

1	PMML44 module	1
2	PMML44Super module	3
3	arima module	305
4	exponential_smoothing module	307
5	keras_model_to_pmml module	309
6	lgb_to_pmml module	311
7	metadata module	313
8	pre_process module	315
9	skl_to_pmml module	317
10	xgboost_to_pmml module	319
	Python Module Index	321
	Index	323

CHAPTER 1

PMML44 module

CHAPTER 2

PMML44Super module

```
class PMML44Super.AR (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='AR', namespacedef="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='AR')

    exportChildren (outfile, level, namespace_="", name_='AR', fromsubclass_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='AR')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Array ()

    get_Extension ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Array (Array)

    set_Extension (Extension)

    subclass = None
```

```
    superclass = None
    to_etree (parent_element=None, name_='AR', mapping_=None)
class PMML44Super.ARDSquaredExponentialKernel (description=None, gamma='1', noiseVariance='1', Extension=None, Lambda=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    add_Lambda (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ARDSquaredExponentialKernel', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ARDSquaredExponentialKernel')
    exportChildren (outfile, level, namespace_="", name_='ARDSquaredExponentialKernel', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='ARDSquaredExponentialKernel')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Lambda ()
    get_description ()
    get_gamma ()
    get_noiseVariance ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_Lambda_at (index, value)
    replace_Extension_at (index, value)
    replace_Lambda_at (index, value)
    set_Extension (Extension)
    set_Lambda (Lambda)
    set_description (description)
    set_gamma (gamma)
    set_noiseVariance (noiseVariance)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='ARDSquaredExponentialKernel', mapping_=None)
```



```

    validate_REAL_NUMBER (value)
class PMML44Super.ARIMA (RMSE=None, transformation='none', constantTerm='0', prediction-
                        Method='conditionalLeastSquares', Extension=None, NonseasonalCom-
                        ponent=None, SeasonalComponent=None, DynamicRegressor=None,
                        MaximumLikelihoodStat=None, OutlierEffect=None)
Bases: PMML44Super.GeneratedsSuper
    add_DynamicRegressor (value)
    add_Extension (value)
    add_OutlierEffect (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ARIMA', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ARIMA')
    exportChildren (outfile, level, namespace_="", name_='ARIMA', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ARIMA')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_DynamicRegressor ()
    get_Extension ()
    get_MaximumLikelihoodStat ()
    get_NonseasonalComponent ()
    get_OutlierEffect ()
    get_RMSE ()
    get_SeasonalComponent ()
    get_constantTerm ()
    get_predictionMethod ()
    get_transformation ()
    hasContent_ ()
    insert_DynamicRegressor_at (index, value)
    insert_Extension_at (index, value)
    insert_OutlierEffect_at (index, value)
    replace_DynamicRegressor_at (index, value)
    replace_Extension_at (index, value)
    replace_OutlierEffect_at (index, value)
    set_DynamicRegressor (DynamicRegressor)

```

```
set_Extension (Extension)
set_MaximumLikelihoodStat (MaximumLikelihoodStat)
set_NonseasonalComponent (NonseasonalComponent)
set_OutlierEffect (OutlierEffect)
set_RMSE (RMSE)
set_SeasonalComponent (SeasonalComponent)
set_constantTerm (constantTerm)
set_predictionMethod (predictionMethod)
set_transformation (transformation)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ARIMA', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.ARMABPart (constant='0', p=None, q=None, Extension=None, AR=None,
                             MA=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ARMAPart', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ARMAPart')
    exportChildren (outfile, level, namespace_="", name_='ARMAPart', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ARMAPart')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_AR ()
    get_Extension ()
    get_MA ()
    get_constant ()
    get_p ()
    get_q ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
```

```

set_AR (AR)
set_Extension (Extension)
set_MA (MA)
set_constant (constant)
set_p (p)
set_q (q)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ARMAPart', mapping_=None)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)

class PMML44Super.AbsoluteExponentialKernel (description=None, gamma='I', noiseVariance='I', Extension=None, Lambda=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    add_Lambda (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='AbsoluteExponentialKernel', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='AbsoluteExponentialKernel')
    exportChildren (outfile, level, namespace_="", name_='AbsoluteExponentialKernel', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='AbsoluteExponentialKernel')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Lambda ()
    get_description ()
    get_gamma ()
    get_noiseVariance ()
    hasContent ()
    insert_Extension_at (index, value)
    insert_Lambda_at (index, value)
    replace_Extension_at (index, value)

```

```
replace_Lambda_at (index, value)
set_Extension (Extension)
set_Lambda (Lambda)
set_description (description)
set_gamma (gamma)
set_noiseVariance (noiseVariance)
subclass = None
superclass = None
to_etree (parent_element=None, name_='AbsoluteExponentialKernel', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.Adadelta (learningRate=None, rho=None, decayRate=None, epsilon=None, Ex-
                             tension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Adadelta', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Adadelta')
    exportChildren (outfile, level, namespace_="", name_='Adadelta', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Adadelta')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_decayRate ()
    get_epsilon ()
    get_learningRate ()
    get_rho ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_decayRate (decayRate)
    set_epsilon (epsilon)
    set_learningRate (learningRate)
```

```

    set_rho(rho)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Adadelta', mapping_=None)
    validate_REAL_NUMBER(value)
class PMML44Super.Adagrad(learningRate=None, decayRate=None, epsilon=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace="", name_='Adagrad', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace="", name_='Adagrad')
    exportChildren(outfile, level, namespace="", name_='Adagrad', fromsubclass_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='Adagrad')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_decayRate()
    get_epsilon()
    get_learningRate()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_decayRate(decayRate)
    set_epsilon(epsilon)
    set_learningRate(learningRate)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Adagrad', mapping_=None)
    validate_REAL_NUMBER(value)
class PMML44Super.Adam(learningRate=None, beta_1=None, beta_2=None, decayRate=None, epsilon=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

```

```
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Adam', namespacesdef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Adam')
exportChildren (outfile, level, namespace_="", name_='Adam', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Adam')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_beta_1 ()
get_beta_2 ()
get_decayRate ()
get_epsilon ()
get_learningRate ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_beta_1 (beta_1)
set_beta_2 (beta_2)
set_decayRate (decayRate)
set_epsilon (epsilon)
set_learningRate (learningRate)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Adam', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.Adamax (learningRate=None, beta_1=None, beta_2=None, decayRate=None, ep-
                        silon=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
```

```
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Adamax', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Adamax')
exportChildren (outfile, level, namespace_="", name_='Adamax', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Adamax')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_beta_1 ()
get_beta_2 ()
get_decayRate ()
get_epsilon ()
get_learningRate ()
hasContent _ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_beta_1 (beta_1)
set_beta_2 (beta_2)
set_decayRate (decayRate)
set_epsilon (epsilon)
set_learningRate (learningRate)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Adamax', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.Aggregate (field=None, function=None, groupField=None, sqlWhere=None, Ex-
                             tension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Aggregate', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Aggregate')
```

```
exportChildren (outfile, level, namespace_="", name_='Aggregate', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Aggregate')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_field()
get_function()
get_groupField()
get_sqlWhere()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_field (field)
set_function (function)
set_groupField (groupField)
set_sqlWhere (sqlWhere)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Aggregate', mapping_=None)
validate_FIELD_NAME (value)

class PMML44Super.Alternate (AnyDistribution=None, GaussianDistribution=None, PoissonDistri-
                             bution=None, UniformDistribution=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Alternate', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Alternate')
    exportChildren (outfile, level, namespace_="", name_='Alternate', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Alternate')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_AnyDistribution()
```



```

get_GaussianDistribution()
get_PoissonDistribution()
get_UniformDistribution()
hasContent_()
set_AnyDistribution(AnyDistribution)
set_GaussianDistribution(GaussianDistribution)
set_PoissonDistribution(PoissonDistribution)
set_UniformDistribution(UniformDistribution)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Alternate', mapping_=None)
class PMML44Super.Annotation(Extension=None, valueOf_=None, mixedclass_=None, content_=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='Annotation', namespacesdef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='Annotation')
    exportChildren(outfile, level, namespace_="", name_='Annotation', fromsubclass_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='Annotation')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_valueOf_()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_valueOf_(valueOf_)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Annotation', mapping_=None)

```

```
class PMML44Super.AnomalyDetectionModel (modelName=None, functionName=None, algo-  
rithmName=None, algorithmType=None, Min-  
ingSchema=None, Output=None, LocalTransfor-  
mations=None, ParameterList=None, ModelVerifi-  
cation=None, AssociationModel=None, Bayesian-  
NetworkModel=None, BaselineModel=None,  
ClusteringModel=None, DeepNetwork=None,  
AnomalyDetectionModel_member=None,  
GaussianProcessModel=None, GeneralRe-  
gressionModel=None, MiningModel=None,  
NaiveBayesModel=None, NearestNeighbor-  
Model=None, NeuralNetwork=None, Re-  
gressionModel=None, RuleSetModel=None,  
SequenceModel=None, Scorecard=None, Support-  
VectorMachineModel=None, TextModel=None,  
TimeSeriesModel=None, TreeModel=None,  
Extension=None)  
  
Bases: PMML44Super.GeneratedsSuper  
  
add_Extension (value)  
  
build (node)  
  
buildAttributes (node, attrs, already_processed)  
  
buildChildren (child_, node, nodeName_, fromsubclass_=False)  
  
export (outfile, level, namespace_="", name_='AnomalyDetectionModel', namespacedef_="",  
pretty_print=True)  
  
exportAttributes (outfile, level, already_processed, namespace_="",  
name_='AnomalyDetectionModel')  
  
exportChildren (outfile, level, namespace_="", name_='AnomalyDetectionModel', fromsub-  
class_=False, pretty_print=True)  
  
exportLiteral (outfile, level, name_='AnomalyDetectionModel')  
  
exportLiteralAttributes (outfile, level, already_processed, name_)  
  
exportLiteralChildren (outfile, level, name_)  
  
static factory (*args_, **kwargs_)  
  
get_AnomalyDetectionModel ()  
  
get_AssociationModel ()  
  
get_BaselineModel ()  
  
get_BayesianNetworkModel ()  
  
get_ClusteringModel ()  
  
get_DeepNetwork ()  
  
get_Extension ()  
  
get_GaussianProcessModel ()  
  
get_GeneralRegressionModel ()  
  
get_LocalTransformations ()  
  
get_MiningModel ()
```

```
get_MiningSchema ()
get_ModelVerification ()
get_NaiveBayesModel ()
get_NearestNeighborModel ()
get_NeuralNetwork ()
get_Output ()
get_ParameterList ()
get_RegressionModel ()
get_RuleSetModel ()
get_Scorecard ()
get_SequenceModel ()
get_SupportVectorMachineModel ()
get_TextModel ()
get_TimeSeriesModel ()
get_TreeModel ()
get_algorithmName ()
get_algorithmType ()
get_functionName ()
get_modelName ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_AnomalyDetectionModel (AnomalyDetectionModel)
set_AssociationModel (AssociationModel)
set_BaselineModel (BaselineModel)
set_BayesianNetworkModel (BayesianNetworkModel)
set_ClusteringModel (ClusteringModel)
set_DeepNetwork (DeepNetwork)
set_Extension (Extension)
set_GaussianProcessModel (GaussianProcessModel)
set_GeneralRegressionModel (GeneralRegressionModel)
set_LocalTransformations (LocalTransformations)
set_MiningModel (MiningModel)
set_MiningSchema (MiningSchema)
set_ModelVerification (ModelVerification)
set_NaiveBayesModel (NaiveBayesModel)
```

```
set_NearestNeighborModel (NearestNeighborModel)
set_NeuralNetwork (NeuralNetwork)
set_Output (Output)
set_ParameterList (ParameterList)
set_RegressionModel (RegressionModel)
set_RuleSetModel (RuleSetModel)
set_Scorecard (Scorecard)
set_SequenceModel (SequenceModel)
set_SupportVectorMachineModel (SupportVectorMachineModel)
set_TextModel (TextModel)
set_TimeSeriesModel (TimeSeriesModel)
set_TreeModel (TreeModel)
set_algorithmName (algorithmName)
set_algorithmType (algorithmType)
set_functionName (functionName)
set_modelName (modelName)
subclass = None
superclass = None
to_etree (parent_element=None, name_='AnomalyDetectionModel', mapping_=None)
validate_ALGORITHM_TYPE (value)
validate_MINING_FUNCTION (value)
class PMML44Super.Anova (target=None, Extension=None, AnovaRow=None)
    Bases: PMML44Super.GeneratedsSuper
    add_AnovaRow (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Anova', namespacesdef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Anova')
    exportChildren (outfile, level, namespace_="", name_='Anova', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Anova')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_AnovaRow ()
```

```

get_Extension()
get_target()
hasContent_()
insert_AnovaRow_at(index, value)
insert_Extension_at(index, value)
replace_AnovaRow_at(index, value)
replace_Extension_at(index, value)
set_AnovaRow(AnovaRow)
set_Extension(Extension)
set_target(target)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Anova', mapping_=None)
validate_FIELD_NAME(value)
class PMML44Super.AnovaRow(type_=None, sumOfSquares=None, degreesOfFreedom=None,
                           meanOfSquares=None, fValue=None, pValue=None, Extension=None)
Bases: PMML44Super.GeneratedSuper
add_Extension(value)
build(node)
buildAttributes(node, attrs, already_processed)
buildChildren(child_, node, nodeName_, fromsubclass_=False)
export(outfile, level, namespace="", name_='AnovaRow', namespacedef="", pretty_print=True)
exportAttributes(outfile, level, already_processed, namespace="", name_='AnovaRow')
exportChildren(outfile, level, namespace="", name_='AnovaRow', fromsubclass_=False,
               pretty_print=True)
exportLiteral(outfile, level, name_='AnovaRow')
exportLiteralAttributes(outfile, level, already_processed, name_)
exportLiteralChildren(outfile, level, name_)
static factory(*args_, **kwargs_)
get_Extension()
get_degreesOfFreedom()
get_fValue()
get_meanOfSquares()
get_pValue()
get_sumOfSquares()
get_type()
hasContent_()

```

```
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_degreesOfFreedom (degreesOfFreedom)
set_fValue (fValue)
set_meanOfSquares (meanOfSquares)
set_pValue (pValue)
set_sumOfSquares (sumOfSquares)
set_type (type_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='AnovaRow', mapping_=None)
validate_NUMBER (value)
validate_PROB_NUMBER (value)
class PMML44Super.AntecedentSequence (Extension=None, SequenceReference=None,
                                      Time=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='AntecedentSequence', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="",
                     name_='AntecedentSequence')
    exportChildren (outfile, level, namespace="", name_='AntecedentSequence', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='AntecedentSequence')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_SequenceReference ()
    get_Time ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
```

```

    set_SequenceReference (SequenceReference)
    set_Time (Time)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='AntecedentSequence', mapping_=None)
class PMML44Super.AnyDistribution (mean=None, variance=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='AnyDistribution', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='AnyDistribution')
    exportChildren (outfile, level, namespace_="", name_='AnyDistribution', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='AnyDistribution')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_mean ()
    get_variance ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_mean (mean)
    set_variance (variance)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='AnyDistribution', mapping_=None)
    validate_REAL_NUMBER (value)
class PMML44Super.Application (name=None, version=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)

```

```
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Application', namespacesdef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Application')
exportChildren (outfile, level, namespace_="", name_='Application', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Application')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_name ()
get_version ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_name (name)
set_version (version)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Application', mapping_=None)
class PMML44Super.Apply (function=None, mapMissingTo=None, defaultValue=None, invalid-
                        ValueTreatment='returnInvalid', Extension=None, FieldRef=None,
                        Constant=None, NormContinuous=None, NormDiscrete=None, Dis-
                        cretize=None, MapValues=None, TextIndex=None, Apply_member=None,
                        Aggregate=None, Lag=None)
Bases: PMML44Super.GeneratedsSuper
add_Aggregate (value)
add_Apply (value)
add_Constant (value)
add_Discretize (value)
add_Extension (value)
add_FieldRef (value)
add_Lag (value)
add_MapValues (value)
add_NormContinuous (value)
add_NormDiscrete (value)
add_TextIndex (value)
```



```
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Apply', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Apply')
exportChildren (outfile, level, namespace_="", name_='Apply', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Apply')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Aggregate ()
get_Apply ()
get_Constant ()
get_Discretize ()
get_Extension ()
get_FieldRef ()
get_Lag ()
get_MapValues ()
get_NormContinuous ()
get_NormDiscrete ()
get_TextIndex ()
get_defaultValue ()
get_function ()
get_invalidValueTreatment ()
get_mapMissingTo ()
hasContent_ ()
insert_Aggregate_at (index, value)
insert_Apply_at (index, value)
insert_Constant_at (index, value)
insert_Discretize_at (index, value)
insert_Extension_at (index, value)
insert_FieldRef_at (index, value)
insert_Lag_at (index, value)
insert_MapValues_at (index, value)
insert_NormContinuous_at (index, value)
insert_NormDiscrete_at (index, value)
```

```
insert_TextIndex_at (index, value)
replace_Aggregate_at (index, value)
replace_Apply_at (index, value)
replace_Constant_at (index, value)
replace_Discretize_at (index, value)
replace_Extension_at (index, value)
replace_FieldRef_at (index, value)
replace_Lag_at (index, value)
replace_MapValues_at (index, value)
replace_NormContinuous_at (index, value)
replace_NormDiscrete_at (index, value)
replace_TextIndex_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
set_defaultValue (defaultValue)
set_function (function)
set_invalidValueTreatment (invalidValueTreatment)
set_mapMissingTo (mapMissingTo)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Apply', mapping_=None)
validate_INVALID_VALUE_TREATMENT_METHOD (value)
class PMML44Super.ArrayType (n=None, type_=None, valueOf_=None, mixedclass_=None, content_=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
```

```

export (outfile, level, namespace_="", name_='ArrayType', namespacedef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='ArrayType')
exportChildren (outfile, level, namespace_="", name_='ArrayType', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='ArrayType')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_n ()
get_type ()
get_valueOf_ ()
hasContent_ ()
set_n (n)
set_type (type_)
set_valueOf_ (valueOf_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ArrayType', mapping_=None)
validate_INT_NUMBER (value)
class PMML44Super.AssociationModel (modelName=None, functionName=None, algo-
                                     rithmName=None, numberOfTransactions=None,
                                     maxNumberOfItemsPerTA=None, avgNumberOf-
                                     ItemsPerTA=None, minimumSupport=None, min-
                                     imumConfidence=None, lengthLimit=None, num-
                                     berOfItems=None, numberOfItemsets=None,
                                     numberOfRules=None, isScorable=True, Min-
                                     ingSchema=None, Output=None, ModelStats=None,
                                     LocalTransformations=None, Item=None, Itemset=None,
                                     AssociationRule=None, ModelVerification=None, Exten-
                                     sion=None)
Bases: PMML44Super.GeneratedsSuper
add_AssociationRule (value)
add_Extension (value)
add_Item (value)
add_Itemset (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='AssociationModel', namespacedef="",
          pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='AssociationModel')

```

```
exportChildren (outfile, level, namespace_=' ', name_='AssociationModel', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='AssociationModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_AssociationRule ()
get_Extension ()
get_Item ()
get_Itemset ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_algorithmName ()
get_avgNumberOfItemsPerTA ()
get_functionName ()
get_isScorable ()
get_lengthLimit ()
get_maxNumberOfItemsPerTA ()
get_minimumConfidence ()
get_minimumSupport ()
get_modelName ()
get_numberOfItems ()
get_numberOfItemsets ()
get_numberOfRules ()
get_numberOfTransactions ()
hasContent_ ()
insert_AssociationRule_at (index, value)
insert_Extension_at (index, value)
insert_Item_at (index, value)
insert_Itemset_at (index, value)
replace_AssociationRule_at (index, value)
replace_Extension_at (index, value)
replace_Item_at (index, value)
replace_Itemset_at (index, value)
```

```

set_AssociationRule (AssociationRule)
set_Extension (Extension)
set_Item (Item)
set_Itemset (Itemset)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_algorithmName (algorithmName)
set_avgNumberOfItemsPerTA (avgNumberOfItemsPerTA)
set_functionName (functionName)
set_isScorable (isScorable)
set_lengthLimit (lengthLimit)
set_maxNumberOfItemsPerTA (maxNumberOfItemsPerTA)
set_minimumConfidence (minimumConfidence)
set_minimumSupport (minimumSupport)
set_modelName (modelName)
set_numberOfItems (numberOfItems)
set_numberOfItemsets (numberOfItemsets)
set_numberOfRules (numberOfRules)
set_numberOfTransactions (numberOfTransactions)
subclass = None
superclass = None
to_etree (parent_element=None, name_='AssociationModel', mapping_=None)
validate_INT_NUMBER (value)
validate_MINING_FUNCTION (value)
validate_PROB_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.AssociationRule (antecedent=None, consequent=None, support=None, confidence=None, lift=None, leverage=None, affinity=None, id=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='AssociationRule', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='AssociationRule')
exportChildren (outfile, level, namespace_="", name_='AssociationRule', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='AssociationRule')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_affinity()
get_antecedent()
get_confidence()
get_consequent()
get_id()
get_leverage()
get_lift()
get_support()
hasContent_()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_affinity (affinity)
set_antecedent (antecedent)
set_confidence (confidence)
set_consequent (consequent)
set_id (id)
set_leverage (leverage)
set_lift (lift)
set_support (support)
subclass = None
superclass = None
to_etree (parent_element=None, name_='AssociationRule', mapping_=None)
validate_PROB_NUMBER (value)

class PMML44Super.Attribute (reasonCode=None, partialScore=None, Extension=None, SimplePredicate=None, CompoundPredicate=None, SimpleSetPredicate=None, True_=None, False_=None, ComplexPartialScore=None)
    Bases: PMML44Super.GeneratedsSuper
```

```
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Attribute', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Attribute')
exportChildren (outfile, level, namespace_="", name_='Attribute', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Attribute')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_ComplexPartialScore ()
get_CompoundPredicate ()
get_Extension ()
get_False ()
get_SimplePredicate ()
get_SimpleSetPredicate ()
get_True ()
get_partialScore ()
get_reasonCode ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_ComplexPartialScore (ComplexPartialScore)
set_CompoundPredicate (CompoundPredicate)
set_Extension (Extension)
set_False (False_)
set_SimplePredicate (SimplePredicate)
set_SimpleSetPredicate (SimpleSetPredicate)
set_True (True_)
set_partialScore (partialScore)
set_reasonCode (reasonCode)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Attribute', mapping_=None)
validate_NUMBER (value)
```

```
class PMML44Super.BaseCumHazardTables (maxTime=None, Extension=None, BaselineStratum=None, BaselineCell=None)
    Bases: PMML44Super.GeneratedsSuper
    add_BaselineCell (value)
    add_BaselineStratum (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='BaseCumHazardTables', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='BaseCumHazardTables')
    exportChildren (outfile, level, namespace_="", name_='BaseCumHazardTables', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='BaseCumHazardTables')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_BaselineCell ()
    get_BaselineStratum ()
    get_Extension ()
    get_maxTime ()
    hasContent_ ()
    insert_BaselineCell_at (index, value)
    insert_BaselineStratum_at (index, value)
    insert_Extension_at (index, value)
    replace_BaselineCell_at (index, value)
    replace_BaselineStratum_at (index, value)
    replace_Extension_at (index, value)
    set_BaselineCell (BaselineCell)
    set_BaselineStratum (BaselineStratum)
    set_Extension (Extension)
    set_maxTime (maxTime)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='BaseCumHazardTables', mapping_=None)
    validate_REAL_NUMBER (value)
```

```

class PMML44Super.Baseline (AnyDistribution=None, GaussianDistribution=None, PoissonDistri-
                             bution=None, UniformDistribution=None, CountTable=None, Nor-
                             malizedCountTable=None, FieldRef=None)
    Bases: PMML44Super.GeneratedsSuper
    add_FieldRef (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='Baseline', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='Baseline')
    exportChildren (outfile, level, namespace="", name_='Baseline', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Baseline')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_AnyDistribution ()
    get_CountTable ()
    get_FieldRef ()
    get_GaussianDistribution ()
    get_NormalizedCountTable ()
    get_PoissonDistribution ()
    get_UniformDistribution ()
    hasContent_ ()
    insert_FieldRef_at (index, value)
    replace_FieldRef_at (index, value)
    set_AnyDistribution (AnyDistribution)
    set_CountTable (CountTable)
    set_FieldRef (FieldRef)
    set_GaussianDistribution (GaussianDistribution)
    set_NormalizedCountTable (NormalizedCountTable)
    set_PoissonDistribution (PoissonDistribution)
    set_UniformDistribution (UniformDistribution)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Baseline', mapping_=None)

class PMML44Super.BaselineCell (time=None, cumHazard=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

```

```
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='BaselineCell', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='BaselineCell')
exportChildren (outfile, level, namespace_="", name_='BaselineCell', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='BaselineCell')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_cumHazard ()
get_time ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_cumHazard (cumHazard)
set_time (time)
subclass = None
superclass = None
to_etree (parent_element=None, name_='BaselineCell', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.BaselineModel (modelName=None, functionName=None, algorithm-
                                Name=None, isScorable=True, MiningSchema=None,
                                Output=None, ModelStats=None, ModelExplanation=None,
                                Targets=None, LocalTransformations=None, TestDistribu-
                                tions=None, ModelVerification=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='BaselineModel', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='BaselineModel')
    exportChildren (outfile, level, namespace_="", name_='BaselineModel', fromsubclass_=False,
                    pretty_print=True)
```

```
exportLiteral (outfile, level, name_='BaselineModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_Targets ()
get_TestDistributions ()
get_algorithmName ()
get_functionName ()
get_isScorable ()
get_modelName ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_Targets (Targets)
set_TestDistributions (TestDistributions)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
subclass = None
superclass = None
to_etree (parent_element=None, name_='BaselineModel', mapping_=None)
```

```
    validate_MINING_FUNCTION (value)

class PMML44Super.BaselineStratum (value=None, label=None, maxTime=None, Extension=None, BaselineCell=None)
    Bases: PMML44Super.GeneratedsSuper
    add_BaselineCell (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='BaselineStratum', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='BaselineStratum')
    exportChildren (outfile, level, namespace_="", name_='BaselineStratum', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='BaselineStratum')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_BaselineCell ()
    get_Extension ()
    get_label ()
    get_maxTime ()
    get_value ()
    hasContent_ ()
    insert_BaselineCell_at (index, value)
    insert_Extension_at (index, value)
    replace_BaselineCell_at (index, value)
    replace_Extension_at (index, value)
    set_BaselineCell (BaselineCell)
    set_Extension (Extension)
    set_label (label)
    set_maxTime (maxTime)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='BaselineStratum', mapping_=None)
    validate_REAL_NUMBER (value)
```

```
class PMML44Super.BayesInput (fieldName=None, Extension=None, TargetValueStats=None, De-  
                                rivedField=None, PairCounts=None)  
    Bases: PMML44Super.GeneratedsSuper  
  
    add_Extension (value)  
  
    add_PairCounts (value)  
  
    build (node)  
  
    buildAttributes (node, attrs, already_processed)  
  
    buildChildren (child_, node, nodeName_, fromsubclass_=False)  
  
    export (outfile, level, namespace_="", name_='BayesInput', namespacedef_="", pretty_print=True)  
  
    exportAttributes (outfile, level, already_processed, namespace_="", name_='BayesInput')  
  
    exportChildren (outfile, level, namespace_="", name_='BayesInput', fromsubclass_=False,  
                    pretty_print=True)  
  
    exportLiteral (outfile, level, name_='BayesInput')  
  
    exportLiteralAttributes (outfile, level, already_processed, name_)  
  
    exportLiteralChildren (outfile, level, name_)  
  
    static factory (*args_, **kwargs_)  
  
    get_DerivedField ()  
  
    get_Extension ()  
  
    get_PairCounts ()  
  
    get_TargetValueStats ()  
  
    get_fieldName ()  
  
    hasContent_ ()  
  
    insert_Extension_at (index, value)  
  
    insert_PairCounts_at (index, value)  
  
    replace_Extension_at (index, value)  
  
    replace_PairCounts_at (index, value)  
  
    set_DerivedField (DerivedField)  
  
    set_Extension (Extension)  
  
    set_PairCounts (PairCounts)  
  
    set_TargetValueStats (TargetValueStats)  
  
    set_fieldName (fieldName)  
  
    subclass = None  
  
    superclass = None  
  
    to_etree (parent_element=None, name_='BayesInput', mapping_=None)  
  
    validate_FIELD_NAME (value)  
  
class PMML44Super.BayesInputs (Extension=None, BayesInput=None)  
    Bases: PMML44Super.GeneratedsSuper  
  
    add_BayesInput (value)
```

```
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='BayesInputs', namespacesdef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='BayesInputs')
exportChildren (outfile, level, namespace_="", name_='BayesInputs', fromsubclass_=False,
    pretty_print=True)
exportLiteral (outfile, level, name_='BayesInputs')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_BayesInput ()
get_Extension ()
hasContent_ ()
insert_BayesInput_at (index, value)
insert_Extension_at (index, value)
replace_BayesInput_at (index, value)
replace_Extension_at (index, value)
set_BayesInput (BayesInput)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='BayesInputs', mapping_=None)
class PMML44Super.BayesOutput (fieldName=None, Extension=None, TargetValueCounts=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='BayesOutput', namespacesdef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='BayesOutput')
    exportChildren (outfile, level, namespace_="", name_='BayesOutput', fromsubclass_=False,
        pretty_print=True)
    exportLiteral (outfile, level, name_='BayesOutput')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
```

```

static factory (*args_, **kwargs_)
get_Extension()
get_TargetValueCounts()
get_fieldName()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_TargetValueCounts(TargetValueCounts)
set_fieldName(fieldName)
subclass = None
superclass = None
to_etree(parent_element=None, name_='BayesOutput', mapping_=None)
validate_FIELD_NAME(value)

class PMML44Super.BayesianNetworkModel(modelName=None, functionName=None, al-
                                         gorithmName=None, modelType='General',
                                         inferenceMethod='Other', isScorable=True,
                                         MiningSchema=None, Output=None, Mod-
                                         elStats=None, ModelExplanation=None, Tar-
                                         gets=None, LocalTransformations=None, Bayesian-
                                         NetworkNodes=None, ModelVerification=None,
                                         Extension=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension(value)

build(node)

buildAttributes(node, attrs, already_processed)

buildChildren(child_, node, nodeName_, fromsubclass_=False)

export(outfile, level, namespace_="", name_='BayesianNetworkModel', namespacedef="",
        pretty_print=True)

exportAttributes(outfile, level, already_processed, namespace_="",
                 name_='BayesianNetworkModel')

exportChildren(outfile, level, namespace_="", name_='BayesianNetworkModel', fromsub-
               class_=False, pretty_print=True)

exportLiteral(outfile, level, name_='BayesianNetworkModel')

exportLiteralAttributes(outfile, level, already_processed, name_)

exportLiteralChildren(outfile, level, name_)

static factory (*args_, **kwargs_)

get_BayesianNetworkNodes()

get_Extension()

get_LocalTransformations()

```

```
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_Targets ()
get_algorithmName ()
get_functionName ()
get_inferenceMethod ()
get_isScorable ()
get_modelName ()
get_modelType ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_BayesianNetworkNodes (BayesianNetworkNodes)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_inferenceMethod (inferenceMethod)
set_isScorable (isScorable)
set_modelName (modelName)
set_modelType (modelType)
subclass = None
superclass = None
to_etree (parent_element=None, name_='BayesianNetworkModel', mapping_=None)
validate_BN_TYPE (value)
validate_INFERENCE_TYPE (value)
validate_MINING_FUNCTION (value)
```

```

class PMML44Super.BayesianNetworkNodes (Extension=None, DiscreteNode=None, Continu-
                                         ousNode=None)
    Bases: PMML44Super.GeneratedsSuper
    add_ContinuousNode (value)
    add_DiscreteNode (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='BayesianNetworkNodes', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='BayesianNetworkNodes')
    exportChildren (outfile, level, namespace_="", name_='BayesianNetworkNodes', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='BayesianNetworkNodes')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_ContinuousNode ()
    get_DiscreteNode ()
    get_Extension ()
    hasContent_ ()
    insert_ContinuousNode_at (index, value)
    insert_DiscreteNode_at (index, value)
    insert_Extension_at (index, value)
    replace_ContinuousNode_at (index, value)
    replace_DiscreteNode_at (index, value)
    replace_Extension_at (index, value)
    set_ContinuousNode (ContinuousNode)
    set_DiscreteNode (DiscreteNode)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='BayesianNetworkNodes', mapping_=None)

class PMML44Super.BlockIndicator (field=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)

```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='BlockIndicator', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='BlockIndicator')
exportChildren (outfile, level, namespace_="", name_='BlockIndicator', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='BlockIndicator')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_field ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_field (field)
subclass = None
superclass = None
to_etree (parent_element=None, name_='BlockIndicator', mapping_=None)
validate_FIELD_NAME (value)

class PMML44Super.BoundaryValueMeans (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='BoundaryValueMeans', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                       name_='BoundaryValueMeans')
    exportChildren (outfile, level, namespace_="", name_='BoundaryValueMeans', fromsub-
                     class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='BoundaryValueMeans')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
```

```

    get_Extension()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Array(Array)
    set_Extension(Extension)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='BoundaryValueMeans', mapping_=None)
class PMML44Super.BoundaryValues(Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='BoundaryValues', namespacedef_="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='BoundaryValues')
    exportChildren(outfile, level, namespace_="", name_='BoundaryValues', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral(outfile, level, name_='BoundaryValues')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    get_Extension()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Array(Array)
    set_Extension(Extension)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='BoundaryValues', mapping_=None)
class PMML44Super.COUNT_TABLE_TYPE(sample=None, Extension=None, FieldValue=None, Field-
                                   ValueCount=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)

```

```
add_FieldValue (value)
add_FieldValueCount (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='COUNT-TABLE-TYPE', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='COUNT-TABLE-
        TYPE')
exportChildren (outfile, level, namespace_="", name_='COUNT-TABLE-TYPE', fromsub-
        class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='COUNT-TABLE-TYPE')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_FieldValue ()
get_FieldValueCount ()
get_sample ()
hasContent_ ()
insert_Extension_at (index, value)
insert_FieldValueCount_at (index, value)
insert_FieldValue_at (index, value)
replace_Extension_at (index, value)
replace_FieldValueCount_at (index, value)
replace_FieldValue_at (index, value)
set_Extension (Extension)
set_FieldValue (FieldValue)
set_FieldValueCount (FieldValueCount)
set_sample (sample)
subclass = None
superclass = None
to_etree (parent_element=None, name_='COUNT-TABLE-TYPE', mapping_=None)
validate_NUMBER (value)
class PMML44Super.CategoricalPredictor (name=None, value=None, coefficient=None, Exten-
        sion=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
```

```

build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_=", name_='CategoricalPredictor', namespacedef_=",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_=",
                    name_='CategoricalPredictor')
exportChildren (outfile, level, namespace_=", name_='CategoricalPredictor', fromsub-
                  class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='CategoricalPredictor')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_coefficient ()
get_name ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_coefficient (coefficient)
set_name (name)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='CategoricalPredictor', mapping_=None)
validate_FIELD_NAME (value)
validate_REAL_NUMBER (value)

class PMML44Super.Categories (Extension=None, Category=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Category (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_=", name_='Categories', namespacedef_=", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_=", name_='Categories')

```

```
exportChildren (outfile, level, namespace_="", name_='Categories', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Categories')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Category ()
get_Extension ()
hasContent_ ()
insert_Category_at (index, value)
insert_Extension_at (index, value)
replace_Category_at (index, value)
replace_Extension_at (index, value)
set_Category (Category)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Categories', mapping_=None)
class PMML44Super.Category (value=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Category', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Category')
    exportChildren (outfile, level, namespace_="", name_='Category', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Category')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_value ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
```

```

    set_Extension (Extension)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Category', mapping_=None)
class PMML44Super.Characteristic (name=None, reasonCode=None, baselineScore=None, Ex-
    tension=None, Attribute=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Attribute (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Characteristic', namespacesdef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Characteristic')
    exportChildren (outfile, level, namespace_="", name_='Characteristic', fromsubclass_=False,
        pretty_print=True)
    exportLiteral (outfile, level, name_='Characteristic')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Attribute ()
    get_Extension ()
    get_baselineScore ()
    get_name ()
    get_reasonCode ()
    hasContent_ ()
    insert_Attribute_at (index, value)
    insert_Extension_at (index, value)
    replace_Attribute_at (index, value)
    replace_Extension_at (index, value)
    set_Attribute (Attribute)
    set_Extension (Extension)
    set_baselineScore (baselineScore)
    set_name (name)
    set_reasonCode (reasonCode)
    subclass = None

```

```
    superclass = None

    to_etree (parent_element=None, name_='Characteristic', mapping_=None)

    validate_NUMBER (value)

class PMML44Super.Characteristics (Extension=None, Characteristic=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Characteristic (value)

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Characteristics', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Characteristics')

    exportChildren (outfile, level, namespace_="", name_='Characteristics', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='Characteristics')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Characteristic ()

    get_Extension ()

    hasContent_ ()

    insert_Characteristic_at (index, value)

    insert_Extension_at (index, value)

    replace_Characteristic_at (index, value)

    replace_Extension_at (index, value)

    set_Characteristic (Characteristic)

    set_Extension (Extension)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Characteristics', mapping_=None)

class PMML44Super.ChildParent (childField=None, parentField=None, parentLevelField=None,
                                isRecursive='no', Extension=None, FieldColumnPair=None,
                                TableLocator=None, InlineTable=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_FieldColumnPair (value)

    build (node)
```

```

buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='ChildParent', namespacesdef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='ChildParent')
exportChildren (outfile, level, namespace_="", name_='ChildParent', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='ChildParent')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_FieldColumnPair()
get_InlineTable()
get_TableLocator()
get_childField()
get_isRecursive()
get_parentField()
get_parentLevelField()
hasContent_()
insert_Extension_at (index, value)
insert_FieldColumnPair_at (index, value)
replace_Extension_at (index, value)
replace_FieldColumnPair_at (index, value)
set_Extension (Extension)
set_FieldColumnPair (FieldColumnPair)
set_InlineTable (InlineTable)
set_TableLocator (TableLocator)
set_childField (childField)
set_isRecursive (isRecursive)
set_parentField (parentField)
set_parentLevelField (parentLevelField)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ChildParent', mapping_=None)
class PMML44Super.ClassLabels (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)

```

```
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='ClassLabels', namespacesdef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='ClassLabels')
exportChildren (outfile, level, namespace_="", name_='ClassLabels', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='ClassLabels')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_Extension ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Array (Array)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ClassLabels', mapping_=None)
class PMML44Super.Cluster (id=None, name=None, size=None, Extension=None, Kohonen-
                          Map=None, Array=None, Partition=None, Covariances=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Cluster', namespacesdef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Cluster')
    exportChildren (outfile, level, namespace_="", name_='Cluster', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Cluster')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Covariances ()
```

```

    get_Extension()
    get_KohonenMap()
    get_Partition()
    get_id()
    get_name()
    get_size()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Array(Array)
    set_Covariances(Covariances)
    set_Extension(Extension)
    set_KohonenMap(KohonenMap)
    set_Partition(Partition)
    set_id(id)
    set_name(name)
    set_size(size)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Cluster', mapping_=None)
class PMML44Super.ClusteringField(field=None, isCenterField='true', fieldWeight='1', similarityScale=None, compareFunction=None, Extension=None, Comparisons=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='ClusteringField', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='ClusteringField')
    exportChildren(outfile, level, namespace_="", name_='ClusteringField', fromsubclass_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='ClusteringField')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Comparisons()

```

```
get_Extension()
get_compareFunction()
get_field()
get_fieldWeight()
get_isCenterField()
get_similarityScale()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Comparisons(Comparisons)
set_Extension(Extension)
set_compareFunction(compareFunction)
set_field(field)
set_fieldWeight(fieldWeight)
set_isCenterField(isCenterField)
set_similarityScale(similarityScale)
subclass = None
superclass = None
to_etree(parent_element=None, name_='ClusteringField', mapping_=None)
validate_COMPARE_FUNCTION(value)
validate_FIELD_NAME(value)
validate_REAL_NUMBER(value)
class PMML44Super.ClusteringModel(modelName=None, functionName=None, algorithm-
                                Name=None, modelClass=None, numberOfClusters=None,
                                isScorable=True, MiningSchema=None, Output=None,
                                ModelStats=None, ModelExplanation=None, LocalTrans-
                                formations=None, ComparisonMeasure=None, Clustering-
                                Field=None, MissingValueWeights=None, Cluster=None,
                                ModelVerification=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper
add_Cluster(value)
add_ClusteringField(value)
add_Extension(value)
build(node)
buildAttributes(node, attrs, already_processed)
buildChildren(child_, node, nodeName_, fromsubclass_=False)
export(outfile, level, namespace_="", name_='ClusteringModel', namespacedef="",
        pretty_print=True)
exportAttributes(outfile, level, already_processed, namespace_="", name_='ClusteringModel')
```

```
exportChildren (outfile, level, namespace_="", name_='ClusteringModel', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='ClusteringModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Cluster ()
get_ClusteringField ()
get_ComparisonMeasure ()
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_MissingValueWeights ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_algorithmName ()
get_functionName ()
get_isScorable ()
get_modelClass ()
get_modelName ()
get_numberOfClusters ()
hasContent_ ()
insert_Cluster_at (index, value)
insert_ClusteringField_at (index, value)
insert_Extension_at (index, value)
replace_Cluster_at (index, value)
replace_ClusteringField_at (index, value)
replace_Extension_at (index, value)
set_Cluster (Cluster)
set_ClusteringField (ClusteringField)
set_ComparisonMeasure (ComparisonMeasure)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_MissingValueWeights (MissingValueWeights)
```

```
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelClass (modelClass)
set_modelName (modelName)
set_numberOfClusters (numberOfClusters)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ClusteringModel', mapping_=None)
validate_INT_NUMBER (value)
validate_MINING_FUNCTION (value)
class PMML44Super.ClusteringModelQuality (dataName=None, SSE=None, SSB=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ClusteringModelQuality', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='ClusteringModelQuality')
    exportChildren (outfile, level, namespace_="", name_='ClusteringModelQuality', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='ClusteringModelQuality')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_SSB ()
    get_SSE ()
    get_dataName ()
    hasContent_ ()
    insert_Extension_at (index, value)
```

```

    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_SSB (SSB)
    set_SSE (SSE)
    set_dataName (dataName)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='ClusteringModelQuality', mapping_=None)
    validate_NUMBER (value)

class PMML44Super.Coefficient (value='0', Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Coefficient', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Coefficient')
    exportChildren (outfile, level, namespace_="", name_='Coefficient', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Coefficient')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_value ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Coefficient', mapping_=None)
    validate_REAL_NUMBER (value)

class PMML44Super.Coefficients (numberOfCoefficients=None, absoluteValue='0', Extension=None, Coefficient=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Coefficient (value)

```

```
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Coefficients', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Coefficients')
exportChildren (outfile, level, namespace_="", name_='Coefficients', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Coefficients')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Coefficient ()
get_Extension ()
get_absoluteValue ()
get_numberOfCoefficients ()
hasContent_ ()
insert_Coefficient_at (index, value)
insert_Extension_at (index, value)
replace_Coefficient_at (index, value)
replace_Extension_at (index, value)
set_Coefficient (Coefficient)
set_Extension (Extension)
set_absoluteValue (absoluteValue)
set_numberOfCoefficients (numberOfCoefficients)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Coefficients', mapping_=None)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.ComparisonMeasure (kind=None, compareFunction='absDiff', mini-
                                     mum=None, maximum=None, Extension=None,
                                     euclidean=None, squaredEuclidean=None, cheby-
                                     chev=None, cityBlock=None, minkowski=None, sim-
                                     pleMatching=None, jaccard=None, tanimoto=None,
                                     binarySimilarity=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
```

```

buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='ComparisonMeasure', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
        name_='ComparisonMeasure')
exportChildren (outfile, level, namespace_="", name_='ComparisonMeasure', fromsub-
        class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='ComparisonMeasure')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_binarySimilarity ()
get_chebychev ()
get_cityBlock ()
get_compareFunction ()
get_euclidean ()
get_jaccard ()
get_kind ()
get_maximum ()
get_minimum ()
get_minkowski ()
get_simpleMatching ()
get_squaredEuclidean ()
get_tanimoto ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_binarySimilarity (binarySimilarity)
set_chebychev (chebychev)
set_cityBlock (cityBlock)
set_compareFunction (compareFunction)
set_euclidean (euclidean)
set_jaccard (jaccard)
set_kind (kind)

```

```
    set_maximum(maximum)
    set_minimum(minimum)
    set_minkowski(minkowski)
    set_simpleMatching(simpleMatching)
    set_squaredEuclidean(squaredEuclidean)
    set_tanimoto(tanimoto)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='ComparisonMeasure', mapping_=None)
    validate_COMPARE_FUNCTION(value)
    validate_NUMBER(value)
class PMML44Super.Comparisons(Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='Comparisons', namespacedef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='Comparisons')
    exportChildren(outfile, level, namespace_="", name_='Comparisons', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='Comparisons')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_Matrix()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_Matrix(Matrix)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Comparisons', mapping_=None)
```

```

class PMML44Super.ComplexPartialScore (Extension=None, FieldRef=None, Constant=None,
                                         NormContinuous=None, NormDiscrete=None, Dis-
                                         cretize=None, MapValues=None, TextIndex=None,
                                         Apply=None, Aggregate=None, Lag=None)

Bases: PMML44Super.GeneratedSuper

add_Extension (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile, level, namespace="", name_='ComplexPartialScore', namespacedef="",
        pretty_print=True)

exportAttributes (outfile, level, already_processed, namespace="",
                  name_='ComplexPartialScore')

exportChildren (outfile, level, namespace="", name_='ComplexPartialScore', fromsub-
                class_=False, pretty_print=True)

exportLiteral (outfile, level, name_='ComplexPartialScore')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_Aggregate ()

get_Apply ()

get_Constant ()

get_Discretize ()

get_Extension ()

get_FieldRef ()

get_Lag ()

get_MapValues ()

get_NormContinuous ()

get_NormDiscrete ()

get_TextIndex ()

hasContent_ ()

insert_Extension_at (index, value)

replace_Extension_at (index, value)

set_Aggregate (Aggregate)

set_Apply (Apply)

set_Constant (Constant)

set_Discretize (Discretize)

set_Extension (Extension)

set_FieldRef (FieldRef)

```

```
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ComplexPartialScore', mapping_=None)
class PMML44Super.CompoundPredicate (booleanOperator=None, Extension=None, SimplePred-
                                     icate=None, CompoundPredicate_member=None, Sim-
                                     pleSetPredicate=None, True_=None, False_=None)
Bases: PMML44Super.GeneratedsSuper
add_CompoundPredicate (value)
add_Extension (value)
add_False (value)
add_SimplePredicate (value)
add_SimpleSetPredicate (value)
add_True (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='CompoundPredicate', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
                  name_='CompoundPredicate')
exportChildren (outfile, level, namespace_="", name_='CompoundPredicate', fromsub-
                class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='CompoundPredicate')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_CompoundPredicate ()
get_Extension ()
get_False ()
get_SimplePredicate ()
get_SimpleSetPredicate ()
get_True ()
get_booleanOperator ()
hasContent_ ()
```

```

insert_CompoundPredicate_at (index, value)
insert_Extension_at (index, value)
insert_False_at (index, value)
insert_SimplePredicate_at (index, value)
insert_SimpleSetPredicate_at (index, value)
insert_True_at (index, value)
replace_CompoundPredicate_at (index, value)
replace_Extension_at (index, value)
replace_False_at (index, value)
replace_SimplePredicate_at (index, value)
replace_SimpleSetPredicate_at (index, value)
replace_True_at (index, value)
set_CompoundPredicate (CompoundPredicate)
set_Extension (Extension)
set_False (False_)
set_SimplePredicate (SimplePredicate)
set_SimpleSetPredicate (SimpleSetPredicate)
set_True (True_)
set_booleanOperator (booleanOperator)
subclass = None
superclass = None
to_etree (parent_element=None, name_='CompoundPredicate', mapping_=None)
class PMML44Super.CompoundRule (Extension=None, SimplePredicate=None, Com-
                                poundPredicate=None, SimpleSetPredicate=None,
                                True_=None, False_=None, SimpleRule=None, Com-
                                poundRule_member=None)
Bases: PMML44Super.GeneratedsSuper
add_CompoundRule (value)
add_Extension (value)
add_SimpleRule (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='CompoundRule', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='CompoundRule')
exportChildren (outfile, level, namespace_="", name_='CompoundRule', fromsubclass_=False,
                pretty_print=True)

```

```
exportLiteral (outfile, level, name_='CompoundRule')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_CompoundPredicate ()
get_CompoundRule ()
get_Extension ()
get_False ()
get_SimplePredicate ()
get_SimpleRule ()
get_SimpleSetPredicate ()
get_True ()
hasContent_ ()
insert_CompoundRule_at (index, value)
insert_Extension_at (index, value)
insert_SimpleRule_at (index, value)
replace_CompoundRule_at (index, value)
replace_Extension_at (index, value)
replace_SimpleRule_at (index, value)
set_CompoundPredicate (CompoundPredicate)
set_CompoundRule (CompoundRule)
set_Extension (Extension)
set_False (False_)
set_SimplePredicate (SimplePredicate)
set_SimpleRule (SimpleRule)
set_SimpleSetPredicate (SimpleSetPredicate)
set_True (True_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='CompoundRule', mapping_=None)
class PMML44Super.Con (from_=None, weight=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
```

```
export (outfile, level, namespace_="", name_='Con', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Con')
exportChildren (outfile, level, namespace_="", name_='Con', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Con')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_from()
get_weight()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_from (from_)
set_weight (weight)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Con', mapping_=None)
validate_NN_NEURON_IDREF (value)
validate_REAL_NUMBER (value)

class PMML44Super.ConfusionMatrix (Extension=None, ClassLabels=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ConfusionMatrix', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ConfusionMatrix')
    exportChildren (outfile, level, namespace_="", name_='ConfusionMatrix', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ConfusionMatrix')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_ClassLabels()
```

```
get_Extension()
get_Matrix()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_ClassLabels(ClassLabels)
set_Extension(Extension)
set_Matrix(Matrix)
subclass = None
superclass = None
to_etree(parent_element=None, name_='ConfusionMatrix', mapping_=None)
class PMML44Super.ConsequentSequence(Extension=None, SequenceReference=None,
                                     Time=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='ConsequentSequence', namespacedef="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="",
                     name_='ConsequentSequence')
    exportChildren(outfile, level, namespace_="", name_='ConsequentSequence', fromsub-
                   class_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='ConsequentSequence')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_SequenceReference()
    get_Time()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_SequenceReference(SequenceReference)
    set_Time(Time)
    subclass = None
```



```

    superclass = None

    to_etree (parent_element=None, name_='ConsequentSequence', mapping_=None)

class PMML44Super.Constant (dataType=None, valueOf_=None)
    Bases: PMML44Super.GeneratedsSuper

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Constant', namespacesdef="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Constant')

    exportChildren (outfile, level, namespace_="", name_='Constant', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='Constant')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_dataType ()

    get_valueOf_ ()

    hasContent_ ()

    set_dataType (dataType)

    set_valueOf_ (valueOf_)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Constant', mapping_=None)

    validate_DATATYPE (value)

class PMML44Super.Constraints (minimumNumberOfItems='1', maximumNumberOfItems=None,
                                minimumNumberOfAntecedentItems='1', maximumNumberOfAntecedentItems=None,
                                minimumNumberOfConsequentItems='1', maximumNumberOfConsequentItems=None,
                                minimumSupport='0', minimumConfidence='0', minimumLift='0', minimumTotalSequenceTime='0',
                                maximumTotalSequenceTime=None, minimumItemsetSeparationTime='0',
                                maximumItemsetSeparationTime=None, minimumAntConsSeparationTime='0',
                                maximumAntConsSeparationTime=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Constraints', namespacesdef="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Constraints')

```

```
exportChildren (outfile, level, namespace="", name_='Constraints', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Constraints')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_maximumAntConsSeparationTime ()
get_maximumItemsetSeparationTime ()
get_maximumNumberOfAntecedentItems ()
get_maximumNumberOfConsequentItems ()
get_maximumNumberOfItems ()
get_maximumTotalSequenceTime ()
get_minimumAntConsSeparationTime ()
get_minimumConfidence ()
get_minimumItemsetSeparationTime ()
get_minimumLift ()
get_minimumNumberOfAntecedentItems ()
get_minimumNumberOfConsequentItems ()
get_minimumNumberOfItems ()
get_minimumSupport ()
get_minimumTotalSequenceTime ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_maximumAntConsSeparationTime (maximumAntConsSeparationTime)
set_maximumItemsetSeparationTime (maximumItemsetSeparationTime)
set_maximumNumberOfAntecedentItems (maximumNumberOfAntecedentItems)
set_maximumNumberOfConsequentItems (maximumNumberOfConsequentItems)
set_maximumNumberOfItems (maximumNumberOfItems)
set_maximumTotalSequenceTime (maximumTotalSequenceTime)
set_minimumAntConsSeparationTime (minimumAntConsSeparationTime)
set_minimumConfidence (minimumConfidence)
set_minimumItemsetSeparationTime (minimumItemsetSeparationTime)
set_minimumLift (minimumLift)
set_minimumNumberOfAntecedentItems (minimumNumberOfAntecedentItems)
```

```
set_minimumNumberOfConsequentItems (minimumNumberOfConsequentItems)
set_minimumNumberOfItems (minimumNumberOfItems)
set_minimumSupport (minimumSupport)
set_minimumTotalSequenceTime (minimumTotalSequenceTime)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Constraints', mapping_=None)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.ContStats (totalValuesSum=None, totalSquaresSum=None, Extension=None,
                             Interval=None, NUM_ARRAY=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_Interval (value)
    add_NUM_ARRAY (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ContStats', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ContStats')
    exportChildren (outfile, level, namespace_="", name_='ContStats', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ContStats')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Interval ()
    get_NUM_ARRAY ()
    get_totalSquaresSum ()
    get_totalValuesSum ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_Interval_at (index, value)
    insert_NUM_ARRAY_at (index, value)
    replace_Extension_at (index, value)
    replace_Interval_at (index, value)
```

```
replace_NUM_ARRAY_at (index, value)
set_Extension (Extension)
set_Interval (Interval)
set_NUM_ARRAY (NUM_ARRAY)
set_totalSquaresSum (totalSquaresSum)
set_totalValuesSum (totalValuesSum)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ContStats', mapping_=None)
validate_NUMBER (value)
class PMML44Super.ContinuousConditionalProbability (count=None, Extension=None,
                                                    ParentValue=None, Continuous-
                                                    Distribution=None)
    Bases: PMML44Super.GeneratedsSuper
    add_ContinuousDistribution (value)
    add_Extension (value)
    add_ParentValue (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ContinuousConditionalProbability', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='ContinuousConditionalProbability')
    exportChildren (outfile, level, namespace_="", name_='ContinuousConditionalProbability', from-
                    subclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='ContinuousConditionalProbability')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_ContinuousDistribution ()
    get_Extension ()
    get_ParentValue ()
    get_count ()
    hasContent_ ()
    insert_ContinuousDistribution_at (index, value)
    insert_Extension_at (index, value)
    insert_ParentValue_at (index, value)
    replace_ContinuousDistribution_at (index, value)
```

```

replace_Extension_at (index, value)
replace_ParentValue_at (index, value)
set_ContinuousDistribution (ContinuousDistribution)
set_Extension (Extension)
set_ParentValue (ParentValue)
set_count (count)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ContinuousConditionalProbability', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.ContinuousDistribution (Extension=None, TriangularDistribution-
                                           ForBN=None, NormalDistributionForBN=None,
                                           LognormalDistributionForBN=None, Uniform-
                                           DistributionForBN=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='ContinuousDistribution', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
                   name_='ContinuousDistribution')
exportChildren (outfile, level, namespace_="", name_='ContinuousDistribution', fromsub-
                  class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='ContinuousDistribution')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_LognormalDistributionForBN ()
get_NormalDistributionForBN ()
get_TriangularDistributionForBN ()
get_UniformDistributionForBN ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_LognormalDistributionForBN (LognormalDistributionForBN)

```

```
set_NormalDistributionForBN (NormalDistributionForBN)
set_TriangularDistributionForBN (TriangularDistributionForBN)
set_UniformDistributionForBN (UniformDistributionForBN)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ContinuousDistribution', mapping_=None)
class PMML44Super.ContinuousNode (name=None, count=None, Extension=None, Derived-
                                Field=None, ContinuousConditionalProbability=None,
                                ContinuousDistribution=None)
Bases: PMML44Super.GeneratedSuper
add_ContinuousConditionalProbability (value)
add_ContinuousDistribution (value)
add_DerivedField (value)
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='ContinuousNode', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='ContinuousNode')
exportChildren (outfile, level, namespace_="", name_='ContinuousNode', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='ContinuousNode')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_ContinuousConditionalProbability ()
get_ContinuousDistribution ()
get_DerivedField ()
get_Extension ()
get_count ()
get_name ()
hasContent_ ()
insert_ContinuousConditionalProbability_at (index, value)
insert_ContinuousDistribution_at (index, value)
insert_DerivedField_at (index, value)
insert_Extension_at (index, value)
replace_ContinuousConditionalProbability_at (index, value)
```

```

replace_ContinuousDistribution_at (index, value)
replace_DerivedField_at (index, value)
replace_Extension_at (index, value)
set_ContinuousConditionalProbability (ContinuousConditionalProbability)
set_ContinuousDistribution (ContinuousDistribution)
set_DerivedField (DerivedField)
set_Extension (Extension)
set_count (count)
set_name (name)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ContinuousNode', mapping_=None)
validate_FIELD_NAME (value)
validate_REAL_NUMBER (value)

class PMML44Super.CorrelationFields (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='CorrelationFields', namespacedef_="",
        pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='CorrelationFields')
    exportChildren (outfile, level, namespace_="", name_='CorrelationFields', fromsubclass_=False,
        pretty_print=True)
    exportLiteral (outfile, level, name_='CorrelationFields')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
    subclass = None

```

```
    superclass = None

    to_etree (parent_element=None, name_='CorrelationFields', mapping_=None)

class PMML44Super.CorrelationMethods (Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='CorrelationMethods', namespacedef_="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='CorrelationMethods')

    exportChildren (outfile, level, namespace_="", name_='CorrelationMethods', fromsub-
                    class_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='CorrelationMethods')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_Matrix ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_Matrix (Matrix)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='CorrelationMethods', mapping_=None)

class PMML44Super.CorrelationValues (Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='CorrelationValues', namespacedef_="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='CorrelationValues')

    exportChildren (outfile, level, namespace_="", name_='CorrelationValues', fromsubclass_=False,
                    pretty_print=True)
```



```

exportLiteral (outfile, level, name_='CorrelationValues')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_Matrix()
hasContent_()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_Matrix (Matrix)
subclass = None
superclass = None
to_etree (parent_element=None, name_='CorrelationValues', mapping_=None)
class PMML44Super.Correlations (Extension=None, CorrelationFields=None, CorrelationVal-
                                ues=None, CorrelationMethods=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Correlations', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Correlations')
    exportChildren (outfile, level, namespace_="", name_='Correlations', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Correlations')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_CorrelationFields()
    get_CorrelationMethods()
    get_CorrelationValues()
    get_Extension()
    hasContent_()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_CorrelationFields (CorrelationFields)

```

```
    set_CorrelationMethods (CorrelationMethods)
    set_CorrelationValues (CorrelationValues)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Correlations', mapping_=None)
class PMML44Super.Counts (totalFreq=None, missingFreq=None, invalidFreq=None, cardinal-
                           ity=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Counts', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Counts')
    exportChildren (outfile, level, namespace_="", name_='Counts', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Counts')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_cardinality ()
    get_invalidFreq ()
    get_missingFreq ()
    get_totalFreq ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_cardinality (cardinality)
    set_invalidFreq (invalidFreq)
    set_missingFreq (missingFreq)
    set_totalFreq (totalFreq)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Counts', mapping_=None)
```

```

    validate_NUMBER (value)

class PMML44Super.Covariances (Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Covariances', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Covariances')
    exportChildren (outfile, level, namespace_="", name_='Covariances', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Covariances')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Matrix ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_Matrix (Matrix)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Covariances', mapping_=None)

class PMML44Super.CovariateList (Extension=None, Predictor=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_Predictor (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='CovariateList', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='CovariateList')
    exportChildren (outfile, level, namespace_="", name_='CovariateList', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='CovariateList')
    exportLiteralAttributes (outfile, level, already_processed, name_)

```

```
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Predictor ()
hasContent_ ()
insert_Extension_at (index, value)
insert_Predictor_at (index, value)
replace_Extension_at (index, value)
replace_Predictor_at (index, value)
set_Extension (Extension)
set_Predictor (Predictor)
subclass = None
superclass = None
to_etree (parent_element=None, name_='CovariateList', mapping_=None)
class PMML44Super.DataDictionary (numberOfFields=None, Extension=None, DataField=None,
                                   Taxonomy=None)
    Bases: PMML44Super.GeneratedSuper
    add_DataField (value)
    add_Extension (value)
    add_Taxonomy (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='DataDictionary', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='DataDictionary')
    exportChildren (outfile, level, namespace_="", name_='DataDictionary', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='DataDictionary')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_DataField ()
    get_Extension ()
    get_Taxonomy ()
    get_numberOfFields ()
    hasContent_ ()
    insert_DataField_at (index, value)
```

```
insert_Extension_at (index, value)
insert_Taxonomy_at (index, value)
replace_DataField_at (index, value)
replace_Extension_at (index, value)
replace_Taxonomy_at (index, value)
set_DataField (DataField)
set_Extension (Extension)
set_Taxonomy (Taxonomy)
set_numberOfFields (numberOfFields)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DataDictionary', mapping_=None)
class PMML44Super.DataField (name=None, displayName=None, optype=None, dataType=None,
                             mimeType=None, taxonomy=None, isCyclic='0', Extension=None,
                             Interval=None, Value=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
add_Interval (value)
add_Value (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='DataField', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='DataField')
exportChildren (outfile, level, namespace_="", name_='DataField', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='DataField')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Interval ()
get_Value ()
get_dataType ()
get_displayName ()
get_isCyclic ()
get_mimeType ()
get_name ()
```

```
get_optype ()
get_taxonomy ()
hasContent_ ()
insert_Extension_at (index, value)
insert_Interval_at (index, value)
insert_Value_at (index, value)
replace_Extension_at (index, value)
replace_Interval_at (index, value)
replace_Value_at (index, value)
set_Extension (Extension)
set_Interval (Interval)
set_Value (Value)
set_dataType (dataType)
set_displayName (displayName)
set_isCyclic (isCyclic)
set_mimeType (mimeType)
set_name (name)
set_optype (optype)
set_taxonomy (taxonomy)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DataField', mapping_=None)
validate_DATATYPE (value)
validate_FIELD_NAME (value)
validate_OPTYPE (value)
class PMML44Super.Decision (value=None, displayValue=None, description=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Decision', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Decision')
    exportChildren (outfile, level, namespace_="", name_='Decision', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='Decision')
```

```

exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_description()
get_displayValue()
get_value()
hasContent_()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_description (description)
set_displayValue (displayValue)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Decision', mapping_=None)

class PMML44Super.DecisionTree (modelName=None,      functionName=None,      algorithm-
                                Name=None,      missingValueStrategy='none',      missingVal-
                                uePenalty='1.0',      noTrueChildStrategy='returnNullPrediction',
                                splitCharacteristic='multiSplit',      Extension=None,      Out-
                                put=None,      ModelStats=None,      Targets=None,      LocalTrans-
                                formations=None,      ResultField=None,      Node=None)

Bases: PMML44Super.GeneratedSuper

add_Extension (value)
add_ResultField (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace="", name_='DecisionTree', namespacedef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="", name_='DecisionTree')
exportChildren (outfile, level, namespace="", name_='DecisionTree', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='DecisionTree')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_LocalTransformations()

```

```
get_ModelStats ()
get_Node ()
get_Output ()
get_ResultField ()
get_Targets ()
get_algorithmName ()
get_functionName ()
get_missingValuePenalty ()
get_missingValueStrategy ()
get_modelName ()
get_noTrueChildStrategy ()
get_splitCharacteristic ()
hasContent_ ()
insert_Extension_at (index, value)
insert_ResultField_at (index, value)
replace_Extension_at (index, value)
replace_ResultField_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_ModelStats (ModelStats)
set_Node (Node)
set_Output (Output)
set_ResultField (ResultField)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_missingValuePenalty (missingValuePenalty)
set_missingValueStrategy (missingValueStrategy)
set_modelName (modelName)
set_noTrueChildStrategy (noTrueChildStrategy)
set_splitCharacteristic (splitCharacteristic)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DecisionTree', mapping_=None)
validate_MINING_FUNCTION (value)
validate_MISSING_VALUE_STRATEGY (value)
```



```
validate_NO_TRUE_CHILD_STRATEGY (value)
validate_PROB_NUMBER (value)
class PMML44Super.Decisions (businessProblem=None, description=None, Extension=None, Deci-
                             sion=None)
    Bases: PMML44Super.GeneratedSuper
    add_Decision (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Decisions', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Decisions')
    exportChildren (outfile, level, namespace_="", name_='Decisions', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Decisions')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Decision ()
    get_Extension ()
    get_businessProblem ()
    get_description ()
    hasContent_ ()
    insert_Decision_at (index, value)
    insert_Extension_at (index, value)
    replace_Decision_at (index, value)
    replace_Extension_at (index, value)
    set_Decision (Decision)
    set_Extension (Extension)
    set_businessProblem (businessProblem)
    set_description (description)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Decisions', mapping_=None)
```

```
class PMML44Super.DeepNetwork(modelName=None, functionName=None, algorithmName=None,  
                               normalizationMethod='none', numberOfLayers=None, isS-  
                               corable=True, MiningSchema=None, Output=None, Model-  
                               Stats=None, ModelExplanation=None, Targets=None, Local-  
                               Transformations=None, TrainingParameters=None, Network-  
                               Layer=None, NeuralOutputs=None, ModelVerification=None,  
                               Extension=None)  
Bases: PMML44Super.GeneratedsSuper  
  
    add_Extension(value)  
    add_NetworkLayer(value)  
    build(node)  
    buildAttributes(node, attrs, already_processed)  
    buildChildren(child_, node, nodeName_, fromsubclass_=False)  
    export(outfile, level, namespace_="", name_='DeepNetwork', namespacedef_="", pretty_print=True)  
    exportAttributes(outfile, level, already_processed, namespace_="", name_='DeepNetwork')  
    exportChildren(outfile, level, namespace_="", name_='DeepNetwork', fromsubclass_=False,  
                   pretty_print=True)  
    exportLiteral(outfile, level, name_='DeepNetwork')  
    exportLiteralAttributes(outfile, level, already_processed, name_)  
    exportLiteralChildren(outfile, level, name_)  
    static factory(*args_, **kwargs_)  
    get_Extension()  
    get_LocalTransformations()  
    get_MiningSchema()  
    get_ModelExplanation()  
    get_ModelStats()  
    get_ModelVerification()  
    get_NetworkLayer()  
    get_NeuralOutputs()  
    get_Output()  
    get_Targets()  
    get_TrainingParameters()  
    get_algorithmName()  
    get_functionName()  
    get_isScorable()  
    get_modelName()  
    get_normalizationMethod()  
    get_numberOfLayers()  
    hasContent_()
```

```

insert_Extension_at (index, value)
insert_NetworkLayer_at (index, value)
replace_Extension_at (index, value)
replace_NetworkLayer_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_NetworkLayer (NetworkLayer)
set_NeuralOutputs (NeuralOutputs)
set_Output (Output)
set_Targets (Targets)
set_TrainingParameters (TrainingParameters)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
set_normalizationMethod (normalizationMethod)
set_numberOfLayers (numberOfLayers)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DeepNetwork', mapping_=None)
validate_MINING_FUNCTION (value)
validate_NN_NORMALIZATION_METHOD (value)
class PMML44Super.DefineFunction (name=None, otype=None, dataType=None, Extension=None, ParameterField=None, FieldRef=None, Constant=None, NormContinuous=None, NormDiscrete=None, Discretize=None, MapValues=None, TextIndex=None, Apply=None, Aggregate=None, Lag=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
add_ParameterField (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='DefineFunction', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='DefineFunction')
exportChildren (outfile, level, namespace_="", name_='DefineFunction', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='DefineFunction')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Aggregate()
get_Apply()
get_Constant()
get_Discretize()
get_Extension()
get_FieldRef()
get_Lag()
get_MapValues()
get_NormContinuous()
get_NormDiscrete()
get_ParameterField()
get_TextIndex()
get_dataType()
get_name()
get_optype()
hasContent_()
insert_Extension_at (index, value)
insert_ParameterField_at (index, value)
replace_Extension_at (index, value)
replace_ParameterField_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
```

```
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_ParameterField (ParameterField)
set_TextIndex (TextIndex)
set_dataType (dataType)
set_name (name)
set_optype (optype)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DefineFunction', mapping_=None)
validate_DATATYPE (value)
validate_OPTYPE (value)
class PMML44Super.Delimiter (delimiter=None, gap=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Delimiter', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Delimiter')
    exportChildren (outfile, level, namespace_="", name_='Delimiter', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Delimiter')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_delimiter ()
    get_gap ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_delimiter (delimiter)
    set_gap (gap)
    subclass = None
    superclass = None
```

```
to_etree (parent_element=None, name_='Delimiter', mapping_=None)

validate_DELIMITER (value)

validate_GAP (value)

class PMML44Super.Denominator (Extension=None, NonseasonalFactor=None, SeasonalFactor=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Denominator', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Denominator')
    exportChildren (outfile, level, namespace_="", name_='Denominator', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='Denominator')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_NonseasonalFactor ()
    get_SeasonalFactor ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_NonseasonalFactor (NonseasonalFactor)
    set_SeasonalFactor (SeasonalFactor)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Denominator', mapping_=None)

class PMML44Super.DerivedField (name=None, displayName=None, otype=None,
    dataType=None, datasetName=None, trainingBackend=None,
    architectureName=None, Extension=None, FieldRef=None,
    Constant=None, NormContinuous=None, NormDiscrete=None,
    Discretize=None, MapValues=None, TextIndex=None, Apply=None,
    Aggregate=None, Lag=None, Value=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_Value (value)
    build (node)
```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='DerivedField', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='DerivedField')
exportChildren (outfile, level, namespace_="", name_='DerivedField', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='DerivedField')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Aggregate ()
get_Apply ()
get_Constant ()
get_Discretize ()
get_Extension ()
get_FieldRef ()
get_Lag ()
get_MapValues ()
get_NormContinuous ()
get_NormDiscrete ()
get_TextIndex ()
get_Value ()
get_architectureName ()
get_dataType ()
get_datasetName ()
get_displayName ()
get_name ()
get_optype ()
get_trainingBackend ()
hasContent_ ()
insert_Extension_at (index, value)
insert_Value_at (index, value)
replace_Extension_at (index, value)
replace_Value_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
```

```
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
set_Value (Value)
set_architectureName (architectureName)
set_dataType (dataType)
set_datasetName (datasetName)
set_displayName (displayName)
set_name (name)
set_optype (optype)
set_trainingBackend (trainingBackend)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DerivedField', mapping_=None)
validate_ARCHITECTURENAME (value)
validate_BACKEND (value)
validate_DATATYPE (value)
validate_FIELD_NAME (value)
validate_OPTYPE (value)
class PMML44Super.DiscrStats (modalValue=None, Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Array (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='DiscrStats', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='DiscrStats')
    exportChildren (outfile, level, namespace_="", name_='DiscrStats', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='DiscrStats')
    exportLiteralAttributes (outfile, level, already_processed, name_)
```



```

exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_Extension ()
get_modalValue ()
hasContent_ ()
insert_Array_at (index, value)
insert_Extension_at (index, value)
replace_Array_at (index, value)
replace_Extension_at (index, value)
set_Array (Array)
set_Extension (Extension)
set_modalValue (modalValue)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DiscrStats', mapping_=None)
class PMML44Super.DiscreteConditionalProbability (count=None,      Extension=None,
                                                    ParentValue=None,  ValueProbabil-
                                                    ity=None)

Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
add_ParentValue (value)
add_ValueProbability (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='DiscreteConditionalProbability', namespacedef="",
        pretty_print=True)
exportAttributes (outfile,      level,      already_processed,      namespace_="",
                  name_='DiscreteConditionalProbability')
exportChildren (outfile, level, namespace_="", name_='DiscreteConditionalProbability', fromsub-
                  class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='DiscreteConditionalProbability')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_ParentValue ()
get_ValueProbability ()

```

```
get_count ()
hasContent_ ()
insert_Extension_at (index, value)
insert_ParentValue_at (index, value)
insert_ValueProbability_at (index, value)
replace_Extension_at (index, value)
replace_ParentValue_at (index, value)
replace_ValueProbability_at (index, value)
set_Extension (Extension)
set_ParentValue (ParentValue)
set_ValueProbability (ValueProbability)
set_count (count)
subclass = None
superclass = None
to_etree (parent_element=None, name_='DiscreteConditionalProbability', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.DiscreteNode (name=None, count=None, Extension=None, Derived-
                                Field=None, DiscreteConditionalProbability=None, Val-
                                ueProbability=None)
Bases: PMML44Super.GeneratedsSuper
add_DerivedField (value)
add_DiscreteConditionalProbability (value)
add_Extension (value)
add_ValueProbability (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='DiscreteNode', namespacedef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='DiscreteNode')
exportChildren (outfile, level, namespace_="", name_='DiscreteNode', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='DiscreteNode')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_DerivedField ()
get_DiscreteConditionalProbability ()
get_Extension ()
```

```

get_ValueProbability()
get_count()
get_name()
hasContent_()
insert_DerivedField_at(index, value)
insert_DiscreteConditionalProbability_at(index, value)
insert_Extension_at(index, value)
insert_ValueProbability_at(index, value)
replace_DerivedField_at(index, value)
replace_DiscreteConditionalProbability_at(index, value)
replace_Extension_at(index, value)
replace_ValueProbability_at(index, value)
set_DerivedField(DerivedField)
set_DiscreteConditionalProbability(DiscreteConditionalProbability)
set_Extension(Extension)
set_ValueProbability(ValueProbability)
set_count(count)
set_name(name)
subclass = None
superclass = None
to_etree(parent_element=None, name_='DiscreteNode', mapping_=None)
validate_FIELD_NAME(value)
validate_REAL_NUMBER(value)
class PMML44Super.Discretize(field=None, mapMissingTo=None, defaultValue=None,
                             dataType=None, Extension=None, DiscretizeBin=None)
    Bases: PMML44Super.GeneratedSuper
    add_DiscretizeBin(value)
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace="", name_='Discretize', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace="", name_='Discretize')
    exportChildren(outfile, level, namespace="", name_='Discretize', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='Discretize')
    exportLiteralAttributes(outfile, level, already_processed, name_)

```

```
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_DiscretizeBin ()
get_Extension ()
get_dataType ()
get_defaultValue ()
get_field ()
get_mapMissingTo ()
hasContent_ ()
insert_DiscretizeBin_at (index, value)
insert_Extension_at (index, value)
replace_DiscretizeBin_at (index, value)
replace_Extension_at (index, value)
set_DiscretizeBin (DiscretizeBin)
set_Extension (Extension)
set_dataType (dataType)
set_defaultValue (defaultValue)
set_field (field)
set_mapMissingTo (mapMissingTo)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Discretize', mapping_=None)
validate_DATATYPE (value)
validate_FIELD_NAME (value)
class PMML44Super.DiscretizeBin (binValue=None, Extension=None, Interval=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_=' ', name_='DiscretizeBin', namespacedef_=' ', pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_=' ', name_='DiscretizeBin')
    exportChildren (outfile, level, namespace_=' ', name_='DiscretizeBin', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='DiscretizeBin')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
```

```

    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Interval ()
    get_binValue ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_Interval (Interval)
    set_binValue (binValue)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='DiscretizeBin', mapping_=None)
class PMML44Super.DocumentTermMatrix (Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='DocumentTermMatrix', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='DocumentTermMatrix')
    exportChildren (outfile, level, namespace_="", name_='DocumentTermMatrix', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='DocumentTermMatrix')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Matrix ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_Matrix (Matrix)
    subclass = None
    superclass = None

```

```
    to_etree (parent_element=None, name_='DocumentTermMatrix', mapping_=None)
class PMML44Super.DynamicRegressor (field=None, transformation='none', delay='0', fu-
                                     tureValuesMethod='constant', targetField=None, Ex-
                                     tension=None, Numerator=None, Denominator=None,
                                     RegressorValues=None)
Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='DynamicRegressor', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='DynamicRegressor')
    exportChildren (outfile, level, namespace_="", name_='DynamicRegressor', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='DynamicRegressor')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Denominator ()
    get_Extension ()
    get_Numerator ()
    get_RegressorValues ()
    get_delay ()
    get_field ()
    get_futureValuesMethod ()
    get_targetField ()
    get_transformation ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Denominator (Denominator)
    set_Extension (Extension)
    set_Numerator (Numerator)
    set_RegressorValues (RegressorValues)
    set_delay (delay)
    set_field (field)
    set_futureValuesMethod (futureValuesMethod)
```

```

    set_targetField(targetField)
    set_transformation(transformation)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='DynamicRegressor', mapping_=None)
    validate_FIELD_NAME(value)
    validate_INT_NUMBER(value)
class PMML44Super.EventValues(Extension=None, Value=None, Interval=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    add_Interval(value)
    add_Value(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='EventValues', namespacesdef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='EventValues')
    exportChildren(outfile, level, namespace_="", name_='EventValues', fromsubclass_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='EventValues')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_Interval()
    get_Value()
    hasContent__()
    insert_Extension_at(index, value)
    insert_Interval_at(index, value)
    insert_Value_at(index, value)
    replace_Extension_at(index, value)
    replace_Interval_at(index, value)
    replace_Value_at(index, value)
    set_Extension(Extension)
    set_Interval(Interval)
    set_Value(Value)
    subclass = None

```

```
    superclass = None

    to_etree (parent_element=None, name_='EventValues', mapping_=None)

class PMML44Super.ExponentialSmoothing (RMSE=None, transformation='none',
                                         Level=None, Trend_ExpoSmooth=None, Seasonality_ExpoSmooth=None, TimeValue=None)

    Bases: PMML44Super.GeneratedSuper

    add_TimeValue (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace="", name_='ExponentialSmoothing', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace="",
                     name_='ExponentialSmoothing')

    exportChildren (outfile, level, namespace="", name_='ExponentialSmoothing', fromsubclass_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='ExponentialSmoothing')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Level ()

    get_RMSE ()

    get_Seasonality_ExpoSmooth ()

    get_TimeValue ()

    get_Trend_ExpoSmooth ()

    get_transformation ()

    hasContent_ ()

    insert_TimeValue_at (index, value)

    replace_TimeValue_at (index, value)

    set_Level (Level)

    set_RMSE (RMSE)

    set_Seasonality_ExpoSmooth (Seasonality_ExpoSmooth)

    set_TimeValue (TimeValue)

    set_Trend_ExpoSmooth (Trend_ExpoSmooth)

    set_transformation (transformation)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='ExponentialSmoothing', mapping_=None)

    validate_REAL_NUMBER (value)
```

```

class PMML44Super.Extension (extender=None, name=None, value=None, anytypeobjs_=None)
    Bases: PMML44Super.GeneratedsSuper

    add_anytypeobjs_ (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Extension', namespacesdef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Extension')

    exportChildren (outfile, level, namespace_="", name_='Extension', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='Extension')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_anytypeobjs_ ()

    get_extender ()

    get_name ()

    get_value ()

    hasContent_ ()

    insert_anytypeobjs_ (index, value)

    set_anytypeobjs_ (anytypeobjs_)

    set_extender (extender)

    set_name (name)

    set_value (value)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Extension', mapping_=None)

class PMML44Super.FactorList (Extension=None, Predictor=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_Predictor (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='FactorList', namespacesdef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='FactorList')

    exportChildren (outfile, level, namespace_="", name_='FactorList', fromsubclass_=False,
                    pretty_print=True)

```

```
exportLiteral (outfile, level, name_='FactorList')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Predictor ()
hasContent_ ()
insert_Extension_at (index, value)
insert_Predictor_at (index, value)
replace_Extension_at (index, value)
replace_Predictor_at (index, value)
set_Extension (Extension)
set_Predictor (Predictor)
subclass = None
superclass = None
to_etree (parent_element=None, name_='FactorList', mapping_=None)
class PMML44Super.False_ (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='False', namespacesdef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='False')
    exportChildren (outfile, level, namespace_="", name_='False', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='False')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
```

```

    to_etree (parent_element=None, name_='False', mapping_=None)

class PMML44Super.FieldColumnPair (field=None, column=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FieldColumnPair', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FieldColumnPair')
    exportChildren (outfile, level, namespace_="", name_='FieldColumnPair', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='FieldColumnPair')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_column ()
    get_field ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_column (column)
    set_field (field)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='FieldColumnPair', mapping_=None)
    validate_FIELD_NAME (value)

class PMML44Super.FieldRef (field=None, mapMissingTo=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FieldRef', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FieldRef')

```

```
exportChildren (outfile, level, namespace_="", name_='FieldRef', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='FieldRef')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_field ()
get_mapMissingTo ()
hasContent _ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_field (field)
set_mapMissingTo (mapMissingTo)
subclass = None
superclass = None
to_etree (parent_element=None, name_='FieldRef', mapping_=None)
validate_FIELD_NAME (value)
class PMML44Super.FieldValue (field=None, value=None, Extension=None, Field-
                             Value_member=None, FieldValueCount=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_FieldValue (value)
    add_FieldValueCount (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FieldValue', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FieldValue')
    exportChildren (outfile, level, namespace_="", name_='FieldValue', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='FieldValue')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_FieldValue ()
```

```

get_FieldValueCount ()
get_field ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
insert_FieldValueCount_at (index, value)
insert_FieldValue_at (index, value)
replace_Extension_at (index, value)
replace_FieldValueCount_at (index, value)
replace_FieldValue_at (index, value)
set_Extension (Extension)
set_FieldValue (FieldValue)
set_FieldValueCount (FieldValueCount)
set_field (field)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='FieldValue', mapping_=None)
validate_FIELD_NAME (value)
class PMML44Super.FieldValueCount (field=None, value=None, count=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FieldValueCount', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FieldValueCount')
    exportChildren (outfile, level, namespace_="", name_='FieldValueCount', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='FieldValueCount')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_count ()
    get_field ()

```

```
get_value()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_count(count)
set_field(field)
set_value(value)
subclass = None
superclass = None
to_etree(parent_element=None, name_='FieldValueCount', mapping_=None)
validate_FIELD_NAME(value)
validate_NUMBER(value)

class PMML44Super.FinalNoise(Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='FinalNoise', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='FinalNoise')
    exportChildren(outfile, level, namespace_="", name_='FinalNoise', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='FinalNoise')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    hasContent_()
    set_Array(Array)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='FinalNoise', mapping_=None)

class PMML44Super.FinalNu(Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
```

```

export (outfile, level, namespace_="", name_='FinalNu', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='FinalNu')
exportChildren (outfile, level, namespace_="", name_='FinalNu', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='FinalNu')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
hasContent_ ()
set_Array (Array)
subclass = None
superclass = None
to_etree (parent_element=None, name_='FinalNu', mapping_=None)
class PMML44Super.FinalOmega (Matrix=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FinalOmega', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FinalOmega')
    exportChildren (outfile, level, namespace_="", name_='FinalOmega', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='FinalOmega')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Matrix ()
    hasContent_ ()
    set_Matrix (Matrix)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='FinalOmega', mapping_=None)
class PMML44Super.FinalPredictedNoise (Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='FinalPredictedNoise', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
                  name_='FinalPredictedNoise')
exportChildren (outfile, level, namespace_="", name_='FinalPredictedNoise', fromsub-
                  class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='FinalPredictedNoise')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
hasContent_ ()
set_Array (Array)
subclass = None
superclass = None
to_etree (parent_element=None, name_='FinalPredictedNoise', mapping_=None)

class PMML44Super.FinalStateVector (Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FinalStateVector', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FinalStateVector')
    exportChildren (outfile, level, namespace_="", name_='FinalStateVector', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='FinalStateVector')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    hasContent_ ()
    set_Array (Array)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='FinalStateVector', mapping_=None)

class PMML44Super.FinalTheta (Theta=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Theta (value)
```



```

    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='FinalTheta', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='FinalTheta')
    exportChildren (outfile, level, namespace_="", name_='FinalTheta', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='FinalTheta')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Theta ()
    hasContent_ ()
    insert_Theta_at (index, value)
    replace_Theta_at (index, value)
    set_Theta (Theta)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='FinalTheta', mapping_=None)
class PMML44Super.GARCH (Extension=None, ARMAPart=None, GARCHPart=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='GARCH', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='GARCH')
    exportChildren (outfile, level, namespace_="", name_='GARCH', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='GARCH')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_ARMAPart ()
    get_Extension ()
    get_GARCHPart ()
    hasContent_ ()

```

```
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_ARMAPart (ARMAPart)
set_Extension (Extension)
set_GARCHPart (GARCHPart)
subclass = None
superclass = None
to_etree (parent_element=None, name_='GARCH', mapping_=None)
class PMML44Super.GARCHPart (constant='0', gp=None, gq=None, Extension=None, Residual-
    SquareCoefficients=None, VarianceCoefficients=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='GARCHPart', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='GARCHPart')
    exportChildren (outfile, level, namespace_="", name_='GARCHPart', fromsubclass_=False,
        pretty_print=True)
    exportLiteral (outfile, level, name_='GARCHPart')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_ResidualSquareCoefficients ()
    get_VarianceCoefficients ()
    get_constant ()
    get_gp ()
    get_gq ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_ResidualSquareCoefficients (ResidualSquareCoefficients)
    set_VarianceCoefficients (VarianceCoefficients)
    set_constant (constant)
    set_gp (gp)
```

```

    set_gq(gq)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='GARCHPart', mapping_=None)
    validate_INT_NUMBER(value)
    validate_REAL_NUMBER(value)
class PMML44Super.GaussianDistribution(mean=None, variance=None, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace="", name_='GaussianDistribution', namespacedef="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace="",
                     name_='GaussianDistribution')
    exportChildren(outfile, level, namespace="", name_='GaussianDistribution', fromsub-
                   class_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='GaussianDistribution')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_mean()
    get_variance()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_mean(mean)
    set_variance(variance)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='GaussianDistribution', mapping_=None)
    validate_REAL_NUMBER(value)

```

```
class PMML44Super.GaussianProcessModel (modelName=None,  functionName=None,  al-
                                         gorithmName=None,  optimizer=None,  isS-
                                         corable=True, MiningSchema=None, Output=None,
                                         ModelStats=None, ModelExplanation=None, Tar-
                                         gets=None, LocalTransformations=None, Radi-
                                         alBasisKernel=None, ARDSquaredExponentialK-
                                         ernel=None, AbsoluteExponentialKernel=None,
                                         GeneralizedExponentialKernel=None, Train-
                                         ingInstances=None, ModelVerification=None,
                                         Extension=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile,  level,  namespace_="",  name_='GaussianProcessModel',  namespacedef="",
        pretty_print=True)

exportAttributes (outfile,          level,          already_processed,          namespace_="",
                  name_='GaussianProcessModel')

exportChildren (outfile,  level,  namespace_="",  name_='GaussianProcessModel',  fromsub-
                  class_=False, pretty_print=True)

exportLiteral (outfile, level, name_='GaussianProcessModel')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_ARDSquaredExponentialKernel ()

get_AbsoluteExponentialKernel ()

get_Extension ()

get_GeneralizedExponentialKernel ()

get_LocalTransformations ()

get_MiningSchema ()

get_ModelExplanation ()

get_ModelStats ()

get_ModelVerification ()

get_Output ()

get_RadialBasisKernel ()

get_Targets ()

get_TrainingInstances ()

get_algorithmName ()

get_functionName ()

get_isScorable ()
```

```
get_modelName()
get_optimizer()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_ARDSquaredExponentialKernel(ARDSquaredExponentialKernel)
set_AbsoluteExponentialKernel(AbsoluteExponentialKernel)
set_Extension(Extension)
set_GeneralizedExponentialKernel(GeneralizedExponentialKernel)
set_LocalTransformations(LocalTransformations)
set_MiningSchema(MiningSchema)
set_ModelExplanation(ModelExplanation)
set_ModelStats(ModelStats)
set_ModelVerification(ModelVerification)
set_Output(Output)
set_RadialBasisKernel(RadialBasisKernel)
set_Targets(Targets)
set_TrainingInstances(TrainingInstances)
set_algorithmName(algorithmName)
set_functionName(functionName)
set_isScorable(isScorable)
set_modelName(modelName)
set_optimizer(optimizer)
subclass = None
superclass = None
to_etree(parent_element=None, name_='GaussianProcessModel', mapping_=None)
validate_MINING_FUNCTION(value)
```

```
class PMML44Super.GeneralRegressionModel (targetVariableName=None,  modelType=None,
                                           modelName=None,          functionName=None,
                                           algorithmName=None,       targetReferenceCat-
                                           egory=None,               cumulativeLink=None,   link-
                                           Function=None,            linkParameter=None,   tri-
                                           alsVariable=None,         trialsValue=None,     distri-
                                           bution=None,             distParameter=None,   offsetVari-
                                           able=None,               offsetValue=None,     modelDF=None,
                                           endTimeVariable=None,      startTimeVari-
                                           able=None,               subjectIDVariable=None, status-
                                           Variable=None,           baselineStrataVariable=None,
                                           isScorable=True,         MiningSchema=None,    Out-
                                           put=None,               ModelStats=None,      ModelExpla-
                                           nation=None,             Targets=None,         LocalTrans-
                                           formations=None,         ParameterList=None,   FactorList=None,     CovariateList=None, PP-
                                           Matrix=None,             PCovMatrix=None,      ParamMa-
                                           trix=None,               EventValues=None,     BaseCumHaz-
                                           ardTables=None,         ModelVerification=None,
                                           Extension=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile,  level,  namespace_="",  name_='GeneralRegressionModel',  namespacedef="",
        pretty_print=True)

exportAttributes (outfile,          level,          already_processed,          namespace_="",
                  name_='GeneralRegressionModel')

exportChildren (outfile,  level,  namespace_="",  name_='GeneralRegressionModel',  fromsub-
                  class_=False, pretty_print=True)

exportLiteral (outfile, level, name_='GeneralRegressionModel')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_BaseCumHazardTables ()

get_CovariateList ()

get_EventValues ()

get_Extension ()

get_FactorList ()

get_LocalTransformations ()

get_MiningSchema ()

get_ModelExplanation ()

get_ModelStats ()
```

```
get_ModelVerification()  
get_Output()  
get_PCovMatrix()  
get_PPMatrix()  
get_ParamMatrix()  
get_ParameterList()  
get_Targets()  
get_algorithmName()  
get_baselineStrataVariable()  
get_cumulativeLink()  
get_distParameter()  
get_distribution()  
get_endTimeVariable()  
get_functionName()  
get_isScorable()  
get_linkFunction()  
get_linkParameter()  
get_modelDF()  
get_modelName()  
get_modelType()  
get_offsetValue()  
get_offsetVariable()  
get_startTimeVariable()  
get_statusVariable()  
get_subjectIDVariable()  
get_targetReferenceCategory()  
get_targetVariableName()  
get_trialsValue()  
get_trialsVariable()  
hasContent_  
insert_Extension_at(index, value)  
replace_Extension_at(index, value)  
set_BaseCumHazardTables(BaseCumHazardTables)  
set_CovariateList(CovariateList)  
set_EventValues(EventValues)  
set_Extension(Extension)
```

```
set_FactorList (FactorList)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_PCovMatrix (PCovMatrix)
set_PPMatrix (PPMatrix)
set_ParamMatrix (ParamMatrix)
set_ParameterList (ParameterList)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_baselineStrataVariable (baselineStrataVariable)
set_cumulativeLink (cumulativeLink)
set_distParameter (distParameter)
set_distribution (distribution)
set_endTimeVariable (endTimeVariable)
set_functionName (functionName)
set_isScorable (isScorable)
set_linkFunction (linkFunction)
set_linkParameter (linkParameter)
set_modelIDF (modelIDF)
set_modelName (modelName)
set_modelType (modelType)
set_offsetValue (offsetValue)
set_offsetVariable (offsetVariable)
set_startTimeVariable (startTimeVariable)
set_statusVariable (statusVariable)
set_subjectIDVariable (subjectIDVariable)
set_targetReferenceCategory (targetReferenceCategory)
set_targetVariableName (targetVariableName)
set_trialsValue (trialsValue)
set_trialsVariable (trialsVariable)
subclass = None
superclass = None
```



```

to_etree (parent_element=None, name_='GeneralRegressionModel', mapping_=None)

validate_CUMULATIVE_LINK_FUNCTION (value)

validate_FIELD_NAME (value)

validate_INT_NUMBER (value)

validate_LINK_FUNCTION (value)

validate_MINING_FUNCTION (value)

validate_REAL_NUMBER (value)

class PMML44Super.GeneralizedExponentialKernel (description=None, gamma='1', noi-
                                                seVariance='1', degree='1', Exten-
                                                sion=None, Lambda=None)

    Bases: PMML44Super.GeneratedSuper

    add_Extension (value)

    add_Lambda (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace="", name_='GeneralizedExponentialKernel', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace="",
                      name_='GeneralizedExponentialKernel')

    exportChildren (outfile, level, namespace="", name_='GeneralizedExponentialKernel', fromsub-
                    class_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='GeneralizedExponentialKernel')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_Lambda ()

    get_degree ()

    get_description ()

    get_gamma ()

    get_noiseVariance ()

    hasContent_ ()

    insert_Extension_at (index, value)

    insert_Lambda_at (index, value)

    replace_Extension_at (index, value)

    replace_Lambda_at (index, value)

    set_Extension (Extension)

    set_Lambda (Lambda)

```

```
    set_degree (degree)
    set_description (description)
    set_gamma (gamma)
    set_noiseVariance (noiseVariance)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='GeneralizedExponentialKernel', mapping_=None)
    validate_REAL_NUMBER (value)
class PMML44Super.HVector (Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='HVector', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='HVector')
    exportChildren (outfile, level, namespace_="", name_='HVector', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='HVector')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    hasContent_ ()
    set_Array (Array)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='HVector', mapping_=None)
class PMML44Super.Header (copyright=None, description=None, modelVersion=None, Extension=None, Application=None, Annotation=None, Timestamp=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Annotation (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Header', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Header')
```

```

exportChildren (outfile, level, namespace_="", name_='Header', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Header')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Annotation ()
get_Application ()
get_Extension ()
get_Timestamp ()
get_copyright ()
get_description ()
get_modelVersion ()
hasContent_ ()
insert_Annotation_at (index, value)
insert_Extension_at (index, value)
replace_Annotation_at (index, value)
replace_Extension_at (index, value)
set_Annotation (Annotation)
set_Application (Application)
set_Extension (Extension)
set_Timestamp (Timestamp)
set_copyright (copyright)
set_description (description)
set_modelVersion (modelVersion)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Header', mapping_=None)
class PMML44Super.INT_Entries
    Bases: PMML44Super.GeneratedSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='INT-Entries', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='INT-Entries')
    exportChildren (outfile, level, namespace_="", name_='INT-Entries', fromsubclass_=False,
                    pretty_print=True)

```

```
    exportLiteral (outfile, level, name_='INT-Entries')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    hasContent_ ()
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='INT-Entries', mapping_=None)
class PMML44Super.INT_SparseArray (n=None,          defaultValue='0',          Indices=None,
                                   INT_Entries=None)
    Bases: PMML44Super.GeneratedSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='INT-SparseArray', namespacedef="",
           pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='INT-SparseArray')
    exportChildren (outfile, level, namespace="", name_='INT-SparseArray', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral (outfile, level, name_='INT-SparseArray')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_INT_Entries ()
    get_Indices ()
    get_defaultValue ()
    get_n ()
    hasContent_ ()
    set_INT_Entries (INT_Entries)
    set_Indices (Indices)
    set_defaultValue (defaultValue)
    set_n (n)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='INT-SparseArray', mapping_=None)
    validate_INT_Entries (value)
    validate_INT_NUMBER (value)
    validate_Indices (value)
```

```

class PMML44Super.Indices
    Bases: PMML44Super.GeneratedsSuper

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Indices', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Indices')

    exportChildren (outfile, level, namespace_="", name_='Indices', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='Indices')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    hasContent_ ()

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Indices', mapping_=None)

class PMML44Super.InlineTable (Extension=None, row=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_row (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='InlineTable', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='InlineTable')

    exportChildren (outfile, level, namespace_="", name_='InlineTable', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='InlineTable')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_row ()

    hasContent_ ()

    insert_Extension_at (index, value)

    insert_row_at (index, value)

    replace_Extension_at (index, value)

```

```
replace_row_at (index, value)
set_Extension (Extension)
set_row (row)
subclass = None
superclass = None
to_etree (parent_element=None, name_='InlineTable', mapping_=None)
class PMML44Super.InstanceField (field=None, column=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='InstanceField', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='InstanceField')
    exportChildren (outfile, level, namespace_="", name_='InstanceField', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='InstanceField')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_column ()
    get_field ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_column (column)
    set_field (field)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='InstanceField', mapping_=None)
    validate_FIELD_NAME (value)
class PMML44Super.InstanceFields (Extension=None, InstanceField=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_InstanceField (value)
```

```

build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='InstanceFields', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='InstanceFields')
exportChildren (outfile, level, namespace_="", name_='InstanceFields', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='InstanceFields')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_InstanceField ()
hasContent_ ()
insert_Extension_at (index, value)
insert_InstanceField_at (index, value)
replace_Extension_at (index, value)
replace_InstanceField_at (index, value)
set_Extension (Extension)
set_InstanceField (InstanceField)
subclass = None
superclass = None
to_etree (parent_element=None, name_='InstanceFields', mapping_=None)
class PMML44Super.Interval (closure=None, leftMargin=None, rightMargin=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Interval', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Interval')
    exportChildren (outfile, level, namespace_="", name_='Interval', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Interval')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)

```

```
get_Extension()
get_closure()
get_leftMargin()
get_rightMargin()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_closure(closure)
set_leftMargin(leftMargin)
set_rightMargin(rightMargin)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Interval', mapping_=None)
validate_NUMBER(value)

class PMML44Super.Item(id=None, value=None, field=None, category=None, mappedValue=None,
                       weight=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='Item', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='Item')
    exportChildren(outfile, level, namespace_="", name_='Item', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral(outfile, level, name_='Item')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_category()
    get_field()
    get_id()
    get_mappedValue()
    get_value()
    get_weight()
```



```

hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_category(category)
set_field(field)
set_id(id)
set_mappedValue(mappedValue)
set_value(value)
set_weight(weight)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Item', mapping_=None)
validate_FIELD_NAME(value)
validate_REAL_NUMBER(value)
class PMML44Super.ItemRef(itemRef=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='ItemRef', namespacedef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='ItemRef')
    exportChildren(outfile, level, namespace_="", name_='ItemRef', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='ItemRef')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_itemRef()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_itemRef(itemRef)
    subclass = None

```

```
    superclass = None

    to_etree (parent_element=None, name_='ItemRef', mapping_=None)

class PMML44Super.Itemset (id=None, support=None, numberOfItems=None, Extension=None,
                           ItemRef=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_ItemRef (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Itemset', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Itemset')

    exportChildren (outfile, level, namespace_="", name_='Itemset', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='Itemset')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_ItemRef ()

    get_id ()

    get_numberOfItems ()

    get_support ()

    hasContent_ ()

    insert_Extension_at (index, value)

    insert_ItemRef_at (index, value)

    replace_Extension_at (index, value)

    replace_ItemRef_at (index, value)

    set_Extension (Extension)

    set_ItemRef (ItemRef)

    set_id (id)

    set_numberOfItems (numberOfItems)

    set_support (support)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Itemset', mapping_=None)

    validate_PROB_NUMBER (value)
```

```

class PMML44Super.KNNInput (field=None, fieldWeight='1', compareFunction=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='KNNInput', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='KNNInput')
    exportChildren (outfile, level, namespace="", name_='KNNInput', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='KNNInput')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_compareFunction ()
    get_field ()
    get_fieldWeight ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_compareFunction (compareFunction)
    set_field (field)
    set_fieldWeight (fieldWeight)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='KNNInput', mapping_=None)
    validate_COMPARE_FUNCTION (value)
    validate_FIELD_NAME (value)
    validate_REAL_NUMBER (value)

class PMML44Super.KNNInputs (Extension=None, KNNInput=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_KNNInput (value)
    build (node)
    buildAttributes (node, attrs, already_processed)

```

```
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='KNNInputs', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='KNNInputs')
exportChildren (outfile, level, namespace_="", name_='KNNInputs', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='KNNInputs')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_KNNInput ()
hasContent_ ()
insert_Extension_at (index, value)
insert_KNNInput_at (index, value)
replace_Extension_at (index, value)
replace_KNNInput_at (index, value)
set_Extension (Extension)
set_KNNInput (KNNInput)
subclass = None
superclass = None
to_etree (parent_element=None, name_='KNNInputs', mapping_=None)
class PMML44Super.KalmanState (FinalOmega=None, FinalStateVector=None, HVector=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='KalmanState', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='KalmanState')
    exportChildren (outfile, level, namespace_="", name_='KalmanState', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='KalmanState')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_FinalOmega ()
    get_FinalStateVector ()
    get_HVector ()
```

```

hasContent_()
set_FinalOmega (FinalOmega)
set_FinalStateVector (FinalStateVector)
set_HVector (HVector)
subclass = None
superclass = None
to_etree (parent_element=None, name_='KalmanState', mapping_=None)
class PMML44Super.KohonenMap (coord1=None, coord2=None, coord3=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='KohonenMap', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='KohonenMap')
    exportChildren (outfile, level, namespace_="", name_='KohonenMap', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='KohonenMap')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_coord1 ()
    get_coord2 ()
    get_coord3 ()
    hasContent_()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_coord1 (coord1)
    set_coord2 (coord2)
    set_coord3 (coord3)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='KohonenMap', mapping_=None)
class PMML44Super.Lag (field=None, n=1, aggregate='none', Extension=None, BlockIndica-
                      tor=None)
    Bases: PMML44Super.GeneratedsSuper

```

```
add_BlockIndicator (value)
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Lag', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Lag')
exportChildren (outfile, level, namespace_="", name_='Lag', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Lag')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_BlockIndicator ()
get_Extension ()
get_aggregate ()
get_field ()
get_n ()
hasContent_ ()
insert_BlockIndicator_at (index, value)
insert_Extension_at (index, value)
replace_BlockIndicator_at (index, value)
replace_Extension_at (index, value)
set_BlockIndicator (BlockIndicator)
set_Extension (Extension)
set_aggregate (aggregate)
set_field (field)
set_n (n)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Lag', mapping_=None)
validate_FIELD_NAME (value)
class PMML44Super.Lambda (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
```

```

buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Lambda', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Lambda')
exportChildren (outfile, level, namespace_="", name_='Lambda', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Lambda')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_Extension ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Array (Array)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Lambda', mapping_=None)
class PMML44Super.LayerBias (biasShape=None, biasFlattenAxis=None, Extension=None, valueOf_=None, mixedclass_=None, content_=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='LayerBias', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='LayerBias')
    exportChildren (outfile, level, namespace_="", name_='LayerBias', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='LayerBias')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_biasFlattenAxis ()
    get_biasShape ()
    get_valueOf_ ()

```

```
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_biasFlattenAxis(biasFlattenAxis)
set_biasShape(biasShape)
set_valueOf_(valueOf_)
subclass = None
superclass = None
to_etree(parent_element=None, name_='LayerBias', mapping_=None)

class PMML44Super.LayerParameters(activationFunction=None, inputDimension=None, outputDimension=None, featureMaps=None, kernel=None, pad=None, stride=None, dilationRate=None, poolSize=None, depthMultiplier=None, paddingDims=None, croppingDims=None, upsamplingSize=None, batchNormalizationEpsilon=None, flattenAxis=None, batchNormalizationAxis=None, batchNormalizationMomentum=None, batchNormalizationCenter=None, batchNormalizationScale=None, gaussianNoiseStdev=None, gaussianDropoutRate=None, alphaDropoutRate=None, alphaDropoutSeed=None, betaInitializer=None, gammaInitializer=None, movingMeanInitializer=None, movingVarianceInitializer=None, betaRegularizer=None, gammaRegularizer=None, betaConstraint=None, gammaConstraint=None, kernelInitializer=None, biasInitializer=None, kernelRegularizer=None, biasRegularizer=None, kernelConstraint=None, biasConstraint=None, depthwiseConstraint=None, pointwiseConstraint=None, batchSize=None, dropoutRate=None, dropoutNoiseShape=None, dropoutSeed=None, generalLUAlpha=None, reshapeTarget=None, permuteDims=None, repeatVectorTimes=None, activityRegularizerL1=None, activityRegularizerL2=None, maskValue=None, mergeLayerOp=None, mergeLayerDotOperationAxis=None, mergeLayerDotNormalize=None, mergeLayerConcatOperationAxes=None, slicingAxis=None, Extension=None)

Bases: PMML44Super.GeneratedSuper

add_Extension(value)

build(node)

buildAttributes(node, attrs, already_processed)

buildChildren(child_, node, nodeName_, fromsubclass_=False)

export(outfile, level, namespace="", name_='LayerParameters', namespacedef="", pretty_print=True)

exportAttributes(outfile, level, already_processed, namespace="", name_='LayerParameters')

exportChildren(outfile, level, namespace="", name_='LayerParameters', fromsubclass_=False, pretty_print=True)
```



```
exportLiteral (outfile, level, name_='LayerParameters')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_activationFunction ()
get_activityRegularizerL1 ()
get_activityRegularizerL2 ()
get_alphaDropoutRate ()
get_alphaDropoutSeed ()
get_batchNormalizationAxis ()
get_batchNormalizationCenter ()
get_batchNormalizationEpsilon ()
get_batchNormalizationMomentum ()
get_batchNormalizationScale ()
get_batchSize ()
get_betaConstraint ()
get_betaInitializer ()
get_betaRegularizer ()
get_biasConstraint ()
get_biasInitializer ()
get_biasRegularizer ()
get_croppingDims ()
get_depthMultiplier ()
get_depthwiseConstraint ()
get_dilationRate ()
get_dropoutNoiseShape ()
get_dropoutRate ()
get_dropoutSeed ()
get_featureMaps ()
get_flattenAxis ()
get_gammaConstraint ()
get_gammaInitializer ()
get_gammaRegularizer ()
get_gaussianDropoutRate ()
get_gaussianNoiseStdev ()
```

```
get_generalLUAlpha ()
get_inputDimension ()
get_kernel ()
get_kernelConstraint ()
get_kernelInitializer ()
get_kernelRegularizer ()
get_maskValue ()
get_mergeLayerConcatOperationAxes ()
get_mergeLayerDotNormalize ()
get_mergeLayerDotOperationAxis ()
get_mergeLayerOp ()
get_movingMeanInitializer ()
get_movingVarianceInitializer ()
get_outputDimension ()
get_pad ()
get_paddingDims ()
get_permuteDims ()
get_pointwiseConstraint ()
get_poolSize ()
get_repeatVectorTimes ()
get_reshapeTarget ()
get_slicingAxis ()
get_stride ()
get_upsamplingSize ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_activationFunction (activationFunction)
set_activityRegularizerL1 (activityRegularizerL1)
set_activityRegularizerL2 (activityRegularizerL2)
set_alphaDropoutRate (alphaDropoutRate)
set_alphaDropoutSeed (alphaDropoutSeed)
set_batchNormalizationAxis (batchNormalizationAxis)
set_batchNormalizationCenter (batchNormalizationCenter)
set_batchNormalizationEpsilon (batchNormalizationEpsilon)
```

set_batchNormalizationMomentum (*batchNormalizationMomentum*)

set_batchNormalizationScale (*batchNormalizationScale*)

set_batchSize (*batchSize*)

set_betaConstraint (*betaConstraint*)

set_betaInitializer (*betaInitializer*)

set_betaRegularizer (*betaRegularizer*)

set_biasConstraint (*biasConstraint*)

set_biasInitializer (*biasInitializer*)

set_biasRegularizer (*biasRegularizer*)

set_croppingDims (*croppingDims*)

set_depthMultiplier (*depthMultiplier*)

set_depthwiseConstraint (*depthwiseConstraint*)

set_dilationRate (*dilationRate*)

set_dropoutNoiseShape (*dropoutNoiseShape*)

set_dropoutRate (*dropoutRate*)

set_dropoutSeed (*dropoutSeed*)

set_featureMaps (*featureMaps*)

set_flattenAxis (*flattenAxis*)

set_gammaConstraint (*gammaConstraint*)

set_gammaInitializer (*gammaInitializer*)

set_gammaRegularizer (*gammaRegularizer*)

set_gaussianDropoutRate (*gaussianDropoutRate*)

set_gaussianNoiseStddev (*gaussianNoiseStddev*)

set_generalLUAlpha (*generalLUAlpha*)

set_inputDimension (*inputDimension*)

set_kernel (*kernel*)

set_kernelConstraint (*kernelConstraint*)

set_kernelInitializer (*kernelInitializer*)

set_kernelRegularizer (*kernelRegularizer*)

set_maskValue (*maskValue*)

set_mergeLayerConcatOperationAxes (*mergeLayerConcatOperationAxes*)

set_mergeLayerDotNormalize (*mergeLayerDotNormalize*)

set_mergeLayerDotOperationAxis (*mergeLayerDotOperationAxis*)

set_mergeLayerOp (*mergeLayerOp*)

set_movingMeanInitializer (*movingMeanInitializer*)

set_movingVarianceInitializer (*movingVarianceInitializer*)

```
set_outputDimension (outputDimension)
set_pad (pad)
set_paddingDims (paddingDims)
set_permuteDims (permuteDims)
set_pointwiseConstraint (pointwiseConstraint)
set_poolSize (poolSize)
set_repeatVectorTimes (repeatVectorTimes)
set_reshapeTarget (reshapeTarget)
set_slicingAxis (slicingAxis)
set_stride (stride)
set_upsamplingSize (upsamplingSize)
subclass = None
superclass = None
to_etree (parent_element=None, name_='LayerParameters', mapping_=None)
validate_DNN_ACTIVATION_FUNCTION (value)
validate_INT_NUMBER (value)
validate_LAYEROP_TYPE (value)
validate_REAL_NUMBER (value)
class PMML44Super.LayerWeights (weightsShape=None, weightsFlattenAxis=None, Extension=None, valueOf_=None, mixedclass_=None, content_=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='LayerWeights', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='LayerWeights')
exportChildren (outfile, level, namespace_="", name_='LayerWeights', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='LayerWeights')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_valueOf_ ()
get_weightsFlattenAxis ()
get_weightsShape ()
```

```

    hasContent_()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_valueOf_ (valueOf_)
    set_weightsFlattenAxis (weightsFlattenAxis)
    set_weightsShape (weightsShape)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='LayerWeights', mapping_=None)

class PMML44Super.Level (alpha=None, smoothedValue=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Level', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Level')
    exportChildren (outfile, level, namespace_="", name_='Level', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Level')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_alpha ()
    get_smoothedValue ()
    hasContent_()
    set_alpha (alpha)
    set_smoothedValue (smoothedValue)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Level', mapping_=None)
    validate_REAL_NUMBER (value)

class PMML44Super.LiftData (targetFieldValue=None, targetFieldDisplayValue=None, rank-
                           ingQuality=None, Extension=None, ModelLiftGraph=None, Opti-
                           mumLiftGraph=None, RandomLiftGraph=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)

```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='LiftData', namespacesdef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='LiftData')
exportChildren (outfile, level, namespace_="", name_='LiftData', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='LiftData')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_ModelLiftGraph ()
get_OptimumLiftGraph ()
get_RandomLiftGraph ()
get_rankingQuality ()
get_targetFieldDisplayValue ()
get_targetFieldValue ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_ModelLiftGraph (ModelLiftGraph)
set_OptimumLiftGraph (OptimumLiftGraph)
set_RandomLiftGraph (RandomLiftGraph)
set_rankingQuality (rankingQuality)
set_targetFieldDisplayValue (targetFieldDisplayValue)
set_targetFieldValue (targetFieldValue)
subclass = None
superclass = None
to_etree (parent_element=None, name_='LiftData', mapping_=None)
validate_NUMBER (value)

class PMML44Super.LiftGraph (Extension=None, XCoordinates=None, YCoordinates=None, Bound-
                             aryValues=None, BoundaryValueMeans=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
```

```

buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='LiftGraph', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='LiftGraph')
exportChildren (outfile, level, namespace_="", name_='LiftGraph', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='LiftGraph')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_BoundaryValueMeans ()
get_BoundaryValues ()
get_Extension ()
get_XCoordinates ()
get_YCoordinates ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_BoundaryValueMeans (BoundaryValueMeans)
set_BoundaryValues (BoundaryValues)
set_Extension (Extension)
set_XCoordinates (XCoordinates)
set_YCoordinates (YCoordinates)
subclass = None
superclass = None
to_etree (parent_element=None, name_='LiftGraph', mapping_=None)
class PMML44Super.LinearKernelType (description=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='LinearKernelType', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='LinearKernelType')
    exportChildren (outfile, level, namespace_="", name_='LinearKernelType', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='LinearKernelType')
    exportLiteralAttributes (outfile, level, already_processed, name_)

```

```
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_description ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_description (description)
subclass = None
superclass = None
to_etree (parent_element=None, name_='LinearKernelType', mapping_=None)
class PMML44Super.LinearNorm (orig=None, norm=None, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='LinearNorm', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='LinearNorm')
    exportChildren (outfile, level, namespace_="", name_='LinearNorm', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='LinearNorm')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_norm ()
    get_orig ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_norm (norm)
    set_orig (orig)
    subclass = None
    superclass = None
```



```

    to_etree (parent_element=None, name_='LinearNorm', mapping_=None)

    validate_NUMBER (value)

class PMML44Super.LocalTransformations (Extension=None, DerivedField=None)
    Bases: PMML44Super.GeneratedsSuper

    add_DerivedField (value)

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='LocalTransformations', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='LocalTransformations')

    exportChildren (outfile, level, namespace_="", name_='LocalTransformations', fromsub-
                    class_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='LocalTransformations')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_DerivedField ()

    get_Extension ()

    hasContent_ ()

    insert_DerivedField_at (index, value)

    insert_Extension_at (index, value)

    replace_DerivedField_at (index, value)

    replace_Extension_at (index, value)

    set_DerivedField (DerivedField)

    set_Extension (Extension)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='LocalTransformations', mapping_=None)

class PMML44Super.LognormalDistributionForBN (Extension=None, Mean=None, Vari-
                                              ance=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='LognormalDistributionForBN', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
                   name_='LognormalDistributionForBN')
exportChildren (outfile, level, namespace_="", name_='LognormalDistributionForBN', fromsub-
                  class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='LognormalDistributionForBN')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Mean ()
get_Variance ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_Mean (Mean)
set_Variance (Variance)
subclass = None
superclass = None
to_etree (parent_element=None, name_='LognormalDistributionForBN', mapping_=None)
class PMML44Super.Losses (loss=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Losses', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Losses')
    exportChildren (outfile, level, namespace_="", name_='Losses', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Losses')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_loss ()
```

```

hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_loss(loss)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Losses', mapping_=None)
validate_LOSS_TYPE(value)

class PMML44Super.Lower(Extension=None, FieldRef=None, Constant=None, NormContinu-
                        ous=None, NormDiscrete=None, Discretize=None, MapValues=None,
                        TextIndex=None, Apply=None, Aggregate=None, Lag=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='Lower', namespacedef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='Lower')
    exportChildren(outfile, level, namespace_="", name_='Lower', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='Lower')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Aggregate()
    get_Apply()
    get_Constant()
    get_Discretize()
    get_Extension()
    get_FieldRef()
    get_Lag()
    get_MapValues()
    get_NormContinuous()
    get_NormDiscrete()
    get_TextIndex()
    hasContent_()
    insert_Extension_at(index, value)

```

```
replace_Extension_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Lower', mapping_=None)
class PMML44Super.MA (Extension=None, MACoefficients=None, Residuals=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='MA', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='MA')
    exportChildren (outfile, level, namespace_="", name_='MA', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='MA')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_MACoefficients ()
    get_Residuals ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_MACoefficients (MACoefficients)
```

```

    set_Residuals (Residuals)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='MA', mapping_=None)

class PMML44Super.MACoefficients (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='MACoefficients', namespacedef_="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='MACoefficients')

    exportChildren (outfile, level, namespace_="", name_='MACoefficients', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='MACoefficients')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Array ()

    get_Extension ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Array (Array)

    set_Extension (Extension)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='MACoefficients', mapping_=None)

class PMML44Super.MapValues (mapMissingTo=None, defaultValue=None, outputColumn=None,
                             dataType=None, Extension=None, FieldColumnPair=None, Table-
                             Locator=None, InlineTable=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_FieldColumnPair (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='MapValues', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='MapValues')
exportChildren (outfile, level, namespace_="", name_='MapValues', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='MapValues')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_FieldColumnPair()
get_InlineTable()
get_TableLocator()
get_dataType()
get_defaultValue()
get_mapMissingTo()
get_outputColumn()
hasContent()
insert_Extension_at (index, value)
insert_FieldColumnPair_at (index, value)
replace_Extension_at (index, value)
replace_FieldColumnPair_at (index, value)
set_Extension (Extension)
set_FieldColumnPair (FieldColumnPair)
set_InlineTable (InlineTable)
set_TableLocator (TableLocator)
set_dataType (dataType)
set_defaultValue (defaultValue)
set_mapMissingTo (mapMissingTo)
set_outputColumn (outputColumn)
subclass = None
superclass = None
to_etree (parent_element=None, name_='MapValues', mapping_=None)
validate_DATATYPE (value)
class PMML44Super.MatCell (row=None, col=None, valueOf_=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
```

```

buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='MatCell', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='MatCell')
exportChildren (outfile, level, namespace_="", name_='MatCell', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='MatCell')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_col ()
get_row ()
get_valueOf_ ()
hasContent_ ()
set_col (col)
set_row (row)
set_valueOf_ (valueOf_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='MatCell', mapping_=None)
validate_INT_NUMBER (value)

class PMML44Super.Matrix (kind='any', nbRows=None, nbCols=None, diagDefault=None, offDi-
                        agDefault=None, Array=None, MatCell=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Array (value)
    add_MatCell (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Matrix', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Matrix')
    exportChildren (outfile, level, namespace_="", name_='Matrix', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Matrix')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_MatCell ()

```

```
get_diagDefault ()
get_kind ()
get_nbCols ()
get_nRows ()
get_offDiagDefault ()
hasContent_ ()
insert_Array_at (index, value)
insert_MatCell_at (index, value)
replace_Array_at (index, value)
replace_MatCell_at (index, value)
set_Array (Array)
set_MatCell (MatCell)
set_diagDefault (diagDefault)
set_kind (kind)
set_nbCols (nbCols)
set_nRows (nbRows)
set_offDiagDefault (offDiagDefault)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Matrix', mapping_=None)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)

class PMML44Super.MaximumLikelihoodStat (method=None, periodDeficit='0',
                                         KalmanState=None, ThetaRecursionState=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='MaximumLikelihoodStat', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="",
                     name_='MaximumLikelihoodStat')
    exportChildren (outfile, level, namespace="", name_='MaximumLikelihoodStat', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='MaximumLikelihoodStat')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
```



```

get_KalmanState ()
get_ThetaRecursionState ()
get_method ()
get_periodDeficit ()
hasContent_ ()
set_KalmanState (KalmanState)
set_ThetaRecursionState (ThetaRecursionState)
set_method (method)
set_periodDeficit (periodDeficit)
subclass = None
superclass = None
to_etree (parent_element=None, name_='MaximumLikelihoodStat', mapping_=None)
validate_INT_NUMBER (value)
class PMML44Super.Mean (Extension=None, FieldRef=None, Constant=None, NormContinuous=None,
                        NormDiscrete=None, Discretize=None, MapValues=None, TextIndex=None,
                        Apply=None, Aggregate=None, Lag=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace="", name_='Mean', namespacedef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="", name_='Mean')
exportChildren (outfile, level, namespace="", name_='Mean', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Mean')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Aggregate ()
get_Apply ()
get_Constant ()
get_Discretize ()
get_Extension ()
get_FieldRef ()
get_Lag ()
get_MapValues ()
get_NormContinuous ()

```

```
get_NormDiscrete ()
get_TextIndex ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Mean', mapping_=None)
class PMML44Super.MeasurementMatrix (Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='MeasurementMatrix', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='MeasurementMatrix')
    exportChildren (outfile, level, namespace_="", name_='MeasurementMatrix', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='MeasurementMatrix')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Matrix ()
    hasContent_ ()
```

```

    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_Matrix (Matrix)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='MeasurementMatrix', mapping_=None)
class PMML44Super.Metrics (top_k_categories_for_accuracy=None,      metric=None,      Exten-
                           sion=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Metrics', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Metrics')
    exportChildren (outfile, level, namespace_="", name_='Metrics', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Metrics')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_metric ()
    get_top_k_categories_for_accuracy ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_metric (metric)
    set_top_k_categories_for_accuracy (top_k_categories_for_accuracy)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Metrics', mapping_=None)
    validate_INT_NUMBER (value)
    validate_METRIC_TYPE (value)
class PMML44Super.MiningBuildTask (Extension=None)
    Bases: PMML44Super.GeneratedsSuper

```

```
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='MiningBuildTask', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='MiningBuildTask')
exportChildren (outfile, level, namespace_="", name_='MiningBuildTask', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='MiningBuildTask')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
hasContent _ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='MiningBuildTask', mapping_=None)
class PMML44Super.MiningField (name=None, usageType='active', optype=None, importance=None,
                                outliers='asIs', lowValue=None, highValue=None,
                                missingValueReplacement=None, missingValueTreatment=None,
                                invalidValueTreatment='returnInvalid', invalidValueReplacement=None,
                                Extension=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='MiningField', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='MiningField')
exportChildren (outfile, level, namespace_="", name_='MiningField', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='MiningField')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
```

```
get_Extension()  
get_highValue()  
get_importance()  
get_invalidValueReplacement()  
get_invalidValueTreatment()  
get_lowValue()  
get_missingValueReplacement()  
get_missingValueTreatment()  
get_name()  
get_optype()  
get_outliers()  
get_usageType()  
hasContent_  
insert_Extension_at(index, value)  
replace_Extension_at(index, value)  
set_Extension(Extension)  
set_highValue(highValue)  
set_importance(importance)  
set_invalidValueReplacement(invalidValueReplacement)  
set_invalidValueTreatment(invalidValueTreatment)  
set_lowValue(lowValue)  
set_missingValueReplacement(missingValueReplacement)  
set_missingValueTreatment(missingValueTreatment)  
set_name(name)  
set_optype(optype)  
set_outliers(outliers)  
set_usageType(usageType)  
subclass = None  
superclass = None  
to_etree(parent_element=None, name_='MiningField', mapping_=None)  
validate_FIELD_NAME(value)  
validate_FIELD_USAGE_TYPE(value)  
validate_INVALID_VALUE_TREATMENT_METHOD(value)  
validate_MISSING_VALUE_TREATMENT_METHOD(value)  
validate_NUMBER(value)  
validate_OPTYPE(value)
```

```
validate_OUTLIER_TREATMENT_METHOD (value)

validate_PROB_NUMBER (value)

class PMML44Super.MiningModel (modelName=None, functionName=None, algorithmName=None,
                                isScorable=True, MiningSchema=None, Output=None, Model-
                                Stats=None, ModelExplanation=None, Targets=None, Local-
                                Transformations=None, Regression=None, DecisionTree=None,
                                Segmentation=None, ModelVerification=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper

add_DecisionTree (value)

add_Extension (value)

add_Regression (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile, level, namespace="", name_='MiningModel', namespacedef="", pretty_print=True)

exportAttributes (outfile, level, already_processed, namespace="", name_='MiningModel')

exportChildren (outfile, level, namespace="", name_='MiningModel', fromsubclass_=False,
                pretty_print=True)

exportLiteral (outfile, level, name_='MiningModel')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_DecisionTree ()

get_Extension ()

get_LocalTransformations ()

get_MiningSchema ()

get_ModelExplanation ()

get_ModelStats ()

get_ModelVerification ()

get_Output ()

get_Regression ()

get_Segmentation ()

get_Targets ()

get_algorithmName ()

get_functionName ()

get_isScorable ()

get_modelName ()

hasContent_ ()
```

```

insert_DecisionTree_at (index, value)
insert_Extension_at (index, value)
insert_Regression_at (index, value)
replace_DecisionTree_at (index, value)
replace_Extension_at (index, value)
replace_Regression_at (index, value)
set_DecisionTree (DecisionTree)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_Regression (Regression)
set_Segmentation (Segmentation)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
subclass = None
superclass = None
to_etree (parent_element=None, name_='MiningModel', mapping_=None)
validate_MINING_FUNCTION (value)
class PMML44Super.MiningSchema (Extension=None, MiningField=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    add_MiningField (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='MiningSchema', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='MiningSchema')
    exportChildren (outfile, level, namespace_="", name_='MiningSchema', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='MiningSchema')

```

```
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_MiningField ()
hasContent_ ()
insert_Extension_at (index, value)
insert_MiningField_at (index, value)
replace_Extension_at (index, value)
replace_MiningField_at (index, value)
set_Extension (Extension)
set_MiningField (MiningField)
subclass = None
superclass = None
to_etree (parent_element=None, name_='MiningSchema', mapping_=None)
class PMML44Super.MissingValueWeights (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='MissingValueWeights', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                       name_='MissingValueWeights')
    exportChildren (outfile, level, namespace_="", name_='MissingValueWeights', fromsub-
                     class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='MissingValueWeights')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
```



```

    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='MissingValueWeights', mapping_=None)
class PMML44Super.ModelExplanation (Extension=None, PredictiveModelQuality=None, Cluster-
                                     ingModelQuality=None, Correlations=None)
    Bases: PMML44Super.GeneratedsSuper
    add_ClusteringModelQuality (value)
    add_Extension (value)
    add_PredictiveModelQuality (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ModelExplanation', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ModelExplanation')
    exportChildren (outfile, level, namespace_="", name_='ModelExplanation', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ModelExplanation')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_ClusteringModelQuality ()
    get_Correlations ()
    get_Extension ()
    get_PredictiveModelQuality ()
    hasContent_ ()
    insert_ClusteringModelQuality_at (index, value)
    insert_Extension_at (index, value)
    insert_PredictiveModelQuality_at (index, value)
    replace_ClusteringModelQuality_at (index, value)
    replace_Extension_at (index, value)
    replace_PredictiveModelQuality_at (index, value)
    set_ClusteringModelQuality (ClusteringModelQuality)
    set_Correlations (Correlations)
    set_Extension (Extension)
    set_PredictiveModelQuality (PredictiveModelQuality)
    subclass = None
    superclass = None

```

```
    to_etree (parent_element=None, name_='ModelExplanation', mapping_=None)

class PMML44Super.ModelLiftGraph (Extension=None, LiftGraph=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='ModelLiftGraph', namespacedef_="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='ModelLiftGraph')

    exportChildren (outfile, level, namespace_="", name_='ModelLiftGraph', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='ModelLiftGraph')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_LiftGraph ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_LiftGraph (LiftGraph)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='ModelLiftGraph', mapping_=None)

class PMML44Super.ModelStats (Extension=None, UnivariateStats=None, MultivariateStats=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_MultivariateStats (value)

    add_UnivariateStats (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='ModelStats', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='ModelStats')

    exportChildren (outfile, level, namespace_="", name_='ModelStats', fromsubclass_=False,
                    pretty_print=True)
```

```

exportLiteral (outfile, level, name_='ModelStats')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_MultivariateStats()
get_UnivariateStats()
hasContent_()
insert_Extension_at (index, value)
insert_MultivariateStats_at (index, value)
insert_UnivariateStats_at (index, value)
replace_Extension_at (index, value)
replace_MultivariateStats_at (index, value)
replace_UnivariateStats_at (index, value)
set_Extension (Extension)
set_MultivariateStats (MultivariateStats)
set_UnivariateStats (UnivariateStats)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ModelStats', mapping_=None)
class PMML44Super.ModelVerification (recordCount=None, fieldCount=None, Extension=None,
                                         VerificationFields=None, InlineTable=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ModelVerification', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ModelVerification')
    exportChildren (outfile, level, namespace_="", name_='ModelVerification', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ModelVerification')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension()
    get_InlineTable()

```

```
get_VerificationFields ()
get_fieldCount ()
get_recordCount ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_InlineTable (InlineTable)
set_VerificationFields (VerificationFields)
set_fieldCount (fieldCount)
set_recordCount (recordCount)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ModelVerification', mapping_=None)
validate_INT_NUMBER (value)

class PMML44Super.MultivariateStat (name=None, category=None, exponent='1', isIntercept=False, importance=None, stdError=None, tValue=None, chiSquareValue=None, fStatistic=None, dF=None, pValueAlpha=None, pValueInitial=None, pValueFinal=None, confidenceLevel='0.95', confidenceLowerBound=None, confidenceUpperBound=None, Extension=None)

Bases: PMML44Super.GeneratedSuper

add_Extension (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile, level, namespace_="", name_='MultivariateStat', namespacedef="", pretty_print=True)

exportAttributes (outfile, level, already_processed, namespace_="", name_='MultivariateStat')

exportChildren (outfile, level, namespace_="", name_='MultivariateStat', fromsubclass_=False, pretty_print=True)

exportLiteral (outfile, level, name_='MultivariateStat')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_Extension ()

get_category ()

get_chiSquareValue ()

get_confidenceLevel ()
```

```
get_confidenceLowerBound()  
get_confidenceUpperBound()  
get_dF()  
get_exponent()  
get_fStatistic()  
get_importance()  
get_isIntercept()  
get_name()  
get_pValueAlpha()  
get_pValueFinal()  
get_pValueInitial()  
get_stdError()  
get_tValue()  
hasContent_  
insert_Extension_at(index, value)  
replace_Extension_at(index, value)  
set_Extension(Extension)  
set_category(category)  
set_chiSquareValue(chiSquareValue)  
set_confidenceLevel(confidenceLevel)  
set_confidenceLowerBound(confidenceLowerBound)  
set_confidenceUpperBound(confidenceUpperBound)  
set_dF(dF)  
set_exponent(exponent)  
set_fStatistic(fStatistic)  
set_importance(importance)  
set_isIntercept(isIntercept)  
set_name(name)  
set_pValueAlpha(pValueAlpha)  
set_pValueFinal(pValueFinal)  
set_pValueInitial(pValueInitial)  
set_stdError(stdError)  
set_tValue(tValue)  
subclass = None  
superclass = None  
to_etree(parent_element=None, name_='MultivariateStat', mapping_=None)
```

```
    validate_INT_NUMBER (value)
    validate_NUMBER (value)
    validate_PROB_NUMBER (value)
class PMML44Super.MultivariateStats (targetCategory=None, Extension=None, MultivariateStat=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_MultivariateStat (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='MultivariateStats', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='MultivariateStats')
    exportChildren (outfile, level, namespace="", name_='MultivariateStats', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='MultivariateStats')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_MultivariateStat ()
    get_targetCategory ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_MultivariateStat_at (index, value)
    replace_Extension_at (index, value)
    replace_MultivariateStat_at (index, value)
    set_Extension (Extension)
    set_MultivariateStat (MultivariateStat)
    set_targetCategory (targetCategory)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='MultivariateStats', mapping_=None)
class PMML44Super.Nadam (learningRate=None, beta_1=None, beta_2=None, schedule_decay=None,
                        epsilon=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
```

```

buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Nadam', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Nadam')
exportChildren (outfile, level, namespace_="", name_='Nadam', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Nadam')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_beta_1()
get_beta_2()
get_epsilon()
get_learningRate()
get_schedule_decay()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_beta_1 (beta_1)
set_beta_2 (beta_2)
set_epsilon (epsilon)
set_learningRate (learningRate)
set_schedule_decay (schedule_decay)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Nadam', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.NaiveBayesModel (modelName=None, threshold=None, functionName=None,
                                     algorithmName=None, isScorable=True, MiningSchema=None,
                                     Output=None, ModelStats=None, ModelExplanation=None,
                                     Targets=None, LocalTransformations=None, BayesInputs=None,
                                     BayesOutput=None, ModelVerification=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)

```

```
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_=", name_='NaiveBayesModel', namespacedef_=",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_=", name_='NaiveBayesModel')
exportChildren (outfile, level, namespace_=", name_='NaiveBayesModel', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='NaiveBayesModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_BayesInputs ()
get_BayesOutput ()
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_Targets ()
get_algorithmName ()
get_functionName ()
get_isScorable ()
get_modelName ()
get_threshold ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_BayesInputs (BayesInputs)
set_BayesOutput (BayesOutput)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
```



```

set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
set_threshold (threshold)
subclass = None
superclass = None
to_etree (parent_element=None, name_='NaiveBayesModel', mapping_=None)
validate_MINING_FUNCTION (value)
validate_REAL_NUMBER (value)

class PMML44Super.NearestNeighborModel (modelName=None, functionName=None, algo-
algorithmName=None, numberOfNeighbors=None,
continuousScoringMethod='average', categori-
calScoringMethod='majorityVote', instanceIdVari-
able=None, threshold='0.001', isScorable=True,
MiningSchema=None, Output=None, Model-
Stats=None, ModelExplanation=None, Tar-
gets=None, LocalTransformations=None, Train-
ingInstances=None, ComparisonMeasure=None,
KNNInputs=None, ModelVerification=None, Exten-
sion=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile, level, namespace_="", name_='NearestNeighborModel', namespacedef_="",
pretty_print=True)

exportAttributes (outfile, level, already_processed, namespace_="",
name_='NearestNeighborModel')

exportChildren (outfile, level, namespace_="", name_='NearestNeighborModel', fromsub-
class_=False, pretty_print=True)

exportLiteral (outfile, level, name_='NearestNeighborModel')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_ComparisonMeasure ()

get_Extension ()

get_KNNInputs ()

get_LocalTransformations ()

```

```
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_Targets ()
get_TrainingInstances ()
get_algorithmName ()
get_categoricalScoringMethod ()
get_continuousScoringMethod ()
get_functionName ()
get_instanceIdVariable ()
get_isScorable ()
get_modelName ()
get_numberOfNeighbors ()
get_threshold ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_ComparisonMeasure (ComparisonMeasure)
set_Extension (Extension)
set_KNNInputs (KNNInputs)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_Targets (Targets)
set_TrainingInstances (TrainingInstances)
set_algorithmName (algorithmName)
set_categoricalScoringMethod (categoricalScoringMethod)
set_continuousScoringMethod (continuousScoringMethod)
set_functionName (functionName)
set_instanceIdVariable (instanceIdVariable)
set_isScorable (isScorable)
```

```
set_modelName(modelName)
set_numberOfNeighbors(numberOfNeighbors)
set_threshold(threshold)
subclass = None
superclass = None
to_etree(parent_element=None, name_='NearestNeighborModel', mapping_=None)
validate_CAT_SCORING_METHOD(value)
validate_CONT_SCORING_METHOD(value)
validate_FIELD_NAME(value)
validate_INT_NUMBER(value)
validate_MINING_FUNCTION(value)
validate_REAL_NUMBER(value)
class PMML44Super.NetworkLayer(normalizationMethod='none', layerType=None, layerId=None,
                                connectionLayerId=None, inputFieldName=None, Extension=None,
                                LayerParameters=None, LayerWeights=None, LayerBias=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='NetworkLayer', namespacedef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='NetworkLayer')
    exportChildren(outfile, level, namespace_="", name_='NetworkLayer', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='NetworkLayer')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_LayerBias()
    get_LayerParameters()
    get_LayerWeights()
    get_connectionLayerId()
    get_inputFieldName()
    get_layerId()
    get_layerType()
    get_normalizationMethod()
```

```
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_LayerBias(LayerBias)
set_LayerParameters(LayerParameters)
set_LayerWeights(LayerWeights)
set_connectionLayerId(connectionLayerId)
set_inputFieldName(inputFieldName)
set_layerId(layerId)
set_layerType(layerType)
set_normalizationMethod(normalizationMethod)
subclass = None
superclass = None
to_etree(parent_element=None, name_='NetworkLayer', mapping_=None)
validate_FIELD_NAME(value)
validate_LAYER_TYPE(value)
validate_NN_NORMALIZATION_METHOD(value)
class PMML44Super.NeuralInput(id=None, Extension=None, DerivedField=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='NeuralInput', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='NeuralInput')
    exportChildren(outfile, level, namespace_="", name_='NeuralInput', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='NeuralInput')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_DerivedField()
    get_Extension()
    get_id()
    hasContent_()
    insert_Extension_at(index, value)
```

```

    replace_Extension_at (index, value)
    set_DerivedField (DerivedField)
    set_Extension (Extension)
    set_id (id)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='NeuralInput', mapping_=None)
    validate_NN_NEURON_ID (value)
class PMML44Super.NeuralInputs (numberOfInputs=None, Extension=None, NeuralInput=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_NeuralInput (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_=' ', name_='NeuralInputs', namespacedef_=' ', pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_=' ', name_='NeuralInputs')
    exportChildren (outfile, level, namespace_=' ', name_='NeuralInputs', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='NeuralInputs')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_NeuralInput ()
    get_numberOfInputs ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_NeuralInput_at (index, value)
    replace_Extension_at (index, value)
    replace_NeuralInput_at (index, value)
    set_Extension (Extension)
    set_NeuralInput (NeuralInput)
    set_numberOfInputs (numberOfInputs)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='NeuralInputs', mapping_=None)

```

```
class PMML44Super.NeuralLayer (numberOfNeurons=None, activationFunction=None, thresh-  
                                old=None, width=None, altitude=None, normalization-  
                                Method=None, Extension=None, Neuron=None)  
    Bases: PMML44Super.GeneratedSuper  
  
    add_Extension (value)  
  
    add_Neuron (value)  
  
    build (node)  
  
    buildAttributes (node, attrs, already_processed)  
  
    buildChildren (child_, node, nodeName_, fromsubclass_=False)  
  
    export (outfile, level, namespace_="", name_='NeuralLayer', namespacedef_="", pretty_print=True)  
  
    exportAttributes (outfile, level, already_processed, namespace_="", name_='NeuralLayer')  
  
    exportChildren (outfile, level, namespace_="", name_='NeuralLayer', fromsubclass_=False,  
                    pretty_print=True)  
  
    exportLiteral (outfile, level, name_='NeuralLayer')  
  
    exportLiteralAttributes (outfile, level, already_processed, name_)  
  
    exportLiteralChildren (outfile, level, name_)  
  
    static factory (*args_, **kwargs_)  
  
    get_Extension ()  
  
    get_Neuron ()  
  
    get_activationFunction ()  
  
    get_altitude ()  
  
    get_normalizationMethod ()  
  
    get_numberOfNeurons ()  
  
    get_threshold ()  
  
    get_width ()  
  
    hasContent_ ()  
  
    insert_Extension_at (index, value)  
  
    insert_Neuron_at (index, value)  
  
    replace_Extension_at (index, value)  
  
    replace_Neuron_at (index, value)  
  
    set_Extension (Extension)  
  
    set_Neuron (Neuron)  
  
    set_activationFunction (activationFunction)  
  
    set_altitude (altitude)  
  
    set_normalizationMethod (normalizationMethod)  
  
    set_numberOfNeurons (numberOfNeurons)  
  
    set_threshold (threshold)  
  
    set_width (width)
```

```

subclass = None
superclass = None
to_etree (parent_element=None, name_='NeuralLayer', mapping_=None)
validate_ACTIVATION_FUNCTION (value)
validate_NN_NORMALIZATION_METHOD (value)
validate_REAL_NUMBER (value)
class PMML44Super.NeuralNetwork (modelName=None,      functionName=None,      algorithm-
                                Name=None,      activationFunction=None,      normaliza-
                                tionMethod='none',      threshold='0',      width=None,      alti-
                                tude='1.0',      numberOfLayers=None,      isScorable=True,
                                MiningSchema=None,      Output=None,      ModelStats=None,
                                ModelExplanation=None,      Targets=None,      LocalTransfor-
                                mations=None,      NeuralInputs=None,      NeuralLayer=None,
                                NeuralOutputs=None,      ModelVerification=None,      Exten-
                                sion=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
add_NeuralLayer (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile,      level,      namespace_="",      name_='NeuralNetwork',      namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='NeuralNetwork')
exportChildren (outfile, level, namespace_="", name_='NeuralNetwork', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='NeuralNetwork')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_NeuralInputs ()
get_NeuralLayer ()
get_NeuralOutputs ()
get_Output ()

```

```
get_Targets ()
get_activationFunction ()
get_algorithmName ()
get_altitude ()
get_functionName ()
get_isScorable ()
get_modelName ()
get_normalizationMethod ()
get_numberOfLayers ()
get_threshold ()
get_width ()
hasContent_ ()
insert_Extension_at (index, value)
insert_NeuralLayer_at (index, value)
replace_Extension_at (index, value)
replace_NeuralLayer_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_NeuralInputs (NeuralInputs)
set_NeuralLayer (NeuralLayer)
set_NeuralOutputs (NeuralOutputs)
set_Output (Output)
set_Targets (Targets)
set_activationFunction (activationFunction)
set_algorithmName (algorithmName)
set_altitude (altitude)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
set_normalizationMethod (normalizationMethod)
set_numberOfLayers (numberOfLayers)
set_threshold (threshold)
```



```

    set_width(width)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='NeuralNetwork', mapping_=None)
    validate_ACTIVATION_FUNCTION(value)
    validate_MINING_FUNCTION(value)
    validate_NN_NORMALIZATION_METHOD(value)
    validate_REAL_NUMBER(value)
class PMML44Super.NeuralOutput(outputNeuron=None, Extension=None, DerivedField=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='NeuralOutput', namespacedef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='NeuralOutput')
    exportChildren(outfile, level, namespace_="", name_='NeuralOutput', fromsubclass_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='NeuralOutput')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_DerivedField()
    get_Extension()
    get_outputNeuron()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_DerivedField(DerivedField)
    set_Extension(Extension)
    set_outputNeuron(outputNeuron)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='NeuralOutput', mapping_=None)
    validate_NN_NEURON_IDREF(value)
class PMML44Super.NeuralOutputs(numberOfOutputs=None, Extension=None, NeuralOutput=None)
    Bases: PMML44Super.GeneratedsSuper

```

```
add_Extension (value)
add_NeuralOutput (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='NeuralOutputs', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='NeuralOutputs')
exportChildren (outfile, level, namespace_="", name_='NeuralOutputs', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='NeuralOutputs')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_NeuralOutput ()
get_numberOfOutputs ()
hasContent_ ()
insert_Extension_at (index, value)
insert_NeuralOutput_at (index, value)
replace_Extension_at (index, value)
replace_NeuralOutput_at (index, value)
set_Extension (Extension)
set_NeuralOutput (NeuralOutput)
set_numberOfOutputs (numberOfOutputs)
subclass = None
superclass = None
to_etree (parent_element=None, name_='NeuralOutputs', mapping_=None)
class PMML44Super.Neuron (id=None, bias=None, width=None, altitude=None, Extension=None,
                        Con=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Con (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Neuron', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Neuron')
```

```

exportChildren (outfile, level, namespace_="", name_='Neuron', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Neuron')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Con ()
get_Extension ()
get_altitude ()
get_bias ()
get_id ()
get_width ()
hasContent_ ()
insert_Con_at (index, value)
insert_Extension_at (index, value)
replace_Con_at (index, value)
replace_Extension_at (index, value)
set_Con (Con)
set_Extension (Extension)
set_altitude (altitude)
set_bias (bias)
set_id (id)
set_width (width)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Neuron', mapping_=None)
validate_NN_NEURON_ID (value)
validate_REAL_NUMBER (value)
class PMML44Super.Node (id=None, score=None, recordCount=None, defaultChild=None, SimplePredicate=None, CompoundPredicate=None, SimpleSetPredicate=None, True_=None, False_=None, Partition=None, ScoreDistribution=None, Node_member=None, Extension=None, Regression=None, DecisionTree=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
add_Node (value)
add_ScoreDistribution (value)
build (node)

```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Node', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Node')
exportChildren (outfile, level, namespace_="", name_='Node', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Node')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_CompoundPredicate ()
get_DecisionTree ()
get_Extension ()
get_False ()
get_Node ()
get_Partition ()
get_Regression ()
get_ScoreDistribution ()
get_SimplePredicate ()
get_SimpleSetPredicate ()
get_True ()
get_defaultChild ()
get_id ()
get_recordCount ()
get_score ()
hasContent_ ()
insert_Extension_at (index, value)
insert_Node_at (index, value)
insert_ScoreDistribution_at (index, value)
replace_Extension_at (index, value)
replace_Node_at (index, value)
replace_ScoreDistribution_at (index, value)
set_CompoundPredicate (CompoundPredicate)
set_DecisionTree (DecisionTree)
set_Extension (Extension)
set_False (False_)
set_Node (Node)
```

```

    set_Partition (Partition)
    set_Regression (Regression)
    set_ScoreDistribution (ScoreDistribution)
    set_SimplePredicate (SimplePredicate)
    set_SimpleSetPredicate (SimpleSetPredicate)
    set_True (True_)
    set_defaultChild (defaultChild)
    set_id (id)
    set_recordCount (recordCount)
    set_score (score)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Node', mapping_=None)
    validate_NUMBER (value)
class PMML44Super.NonseasonalComponent (p=None, d=None, q=None, Extension=None,
                                         AR=None, MA=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='NonseasonalComponent', namespacedef="",
           pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='NonseasonalComponent')
    exportChildren (outfile, level, namespace_="", name_='NonseasonalComponent', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='NonseasonalComponent')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_AR ()
    get_Extension ()
    get_MA ()
    get_d ()
    get_p ()
    get_q ()
    hasContent_ ()

```

```
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_AR (AR)
set_Extension (Extension)
set_MA (MA)
set_d (d)
set_p (p)
set_q (q)
subclass = None
superclass = None
to_etree (parent_element=None, name_='NonseasonalComponent', mapping_=None)
validate_INT_NUMBER (value)
class PMML44Super.NonseasonalFactor (difference='0', maximumOrder=None, Extension=None,
                                     Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_=' ', name_='NonseasonalFactor', namespacedef_=' ',
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_=' ',
                     name_='NonseasonalFactor')
    exportChildren (outfile, level, namespace_=' ', name_='NonseasonalFactor', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='NonseasonalFactor')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    get_difference ()
    get_maximumOrder ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
```

```

    set_difference (difference)
    set_maximumOrder (maximumOrder)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='NonseasonalFactor', mapping_=None)
    validate_INT_NUMBER (value)
class PMML44Super.NormContinuous (mapMissingTo=None, field=None, outliers='asIs', Extension=None, LinearNorm=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    add_LinearNorm (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='NormContinuous', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='NormContinuous')
    exportChildren (outfile, level, namespace="", name_='NormContinuous', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='NormContinuous')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_LinearNorm ()
    get_field ()
    get_mapMissingTo ()
    get_outliers ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_LinearNorm_at (index, value)
    replace_Extension_at (index, value)
    replace_LinearNorm_at (index, value)
    set_Extension (Extension)
    set_LinearNorm (LinearNorm)
    set_field (field)
    set_mapMissingTo (mapMissingTo)
    set_outliers (outliers)

```

```
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='NormContinuous', mapping_=None)
    validate_FIELD_NAME (value)
    validate_NUMBER (value)
    validate_OUTLIER_TREATMENT_METHOD (value)
class PMML44Super.NormDiscrete (field=None, value=None, mapMissingTo=None, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='NormDiscrete', namespacesdef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='NormDiscrete')
    exportChildren (outfile, level, namespace_="", name_='NormDiscrete', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='NormDiscrete')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_field ()
    get_mapMissingTo ()
    get_value ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_field (field)
    set_mapMissingTo (mapMissingTo)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='NormDiscrete', mapping_=None)
    validate_FIELD_NAME (value)
    validate_NUMBER (value)
```

```

class PMML44Super.NormalDistributionForBN (Extension=None,      Mean=None,      Vari-
                                         ance=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='NormalDistributionForBN', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='NormalDistributionForBN')
    exportChildren (outfile, level, namespace_="", name_='NormalDistributionForBN', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='NormalDistributionForBN')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Mean ()
    get_Variance ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_Mean (Mean)
    set_Variance (Variance)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='NormalDistributionForBN', mapping_=None)

class PMML44Super.Numerator (Extension=None,      NonseasonalFactor=None,      SeasonalFac-
                              tor=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Numerator', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Numerator')
    exportChildren (outfile, level, namespace_="", name_='Numerator', fromsubclass_=False,
                    pretty_print=True)

```

```
exportLiteral (outfile, level, name_='Numerator')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_NonseasonalFactor ()
get_SeasonalFactor ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_NonseasonalFactor (NonseasonalFactor)
set_SeasonalFactor (SeasonalFactor)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Numerator', mapping_=None)
class PMML44Super.NumericInfo (minimum=None, maximum=None, mean=None, standardDeviation=None, median=None, interQuartileRange=None, Extension=None, Quantile=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
add_Quantile (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='NumericInfo', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='NumericInfo')
exportChildren (outfile, level, namespace_="", name_='NumericInfo', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='NumericInfo')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Quantile ()
get_interQuartileRange ()
get_maximum ()
get_mean ()
```

```

get_median()
get_minimum()
get_standardDeviation()
hasContent_()
insert_Extension_at(index, value)
insert_Quantile_at(index, value)
replace_Extension_at(index, value)
replace_Quantile_at(index, value)
set_Extension(Extension)
set_Quantile(Quantile)
set_interQuartileRange(interQuartileRange)
set_maximum(maximum)
set_mean(mean)
set_median(median)
set_minimum(minimum)
set_standardDeviation(standardDeviation)
subclass = None
superclass = None
to_etree(parent_element=None, name_='NumericInfo', mapping_=None)
validate_NUMBER(value)

class PMML44Super.NumericPredictor(name=None, exponent='1', coefficient=None, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='NumericPredictor', namespacedef="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='NumericPredictor')
    exportChildren(outfile, level, namespace_="", name_='NumericPredictor', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral(outfile, level, name_='NumericPredictor')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_coefficient()

```

```
get_exponent ()
get_name ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_coefficient (coefficient)
set_exponent (exponent)
set_name (name)
subclass = None
superclass = None
to_etree (parent_element=None, name_='NumericPredictor', mapping_=None)
validate_FIELD_NAME (value)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.Optimizers (clipnorm=None, clipvalue=None, Extension=None, SGD=None,
                             RMSprop=None, Adagrad=None, Adadelat=None, Adam=None,
                             Adamax=None, Nadam=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Adadelat (value)
    add_Adagrad (value)
    add_Adam (value)
    add_Adamax (value)
    add_Extension (value)
    add_Nadam (value)
    add_RMSprop (value)
    add_SGD (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Optimizers', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Optimizers')
    exportChildren (outfile, level, namespace_="", name_='Optimizers', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Optimizers')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
```

```
get_Adadelta()  
get_Adagrad()  
get_Adam()  
get_Adamax()  
get_Extension()  
get_Nadam()  
get_RMSprop()  
get_SGD()  
get_clipnorm()  
get_clipvalue()  
hasContent_  
insert_Adadelta_at(index, value)  
insert_Adagrad_at(index, value)  
insert_Adam_at(index, value)  
insert_Adamax_at(index, value)  
insert_Extension_at(index, value)  
insert_Nadam_at(index, value)  
insert_RMSprop_at(index, value)  
insert_SGD_at(index, value)  
replace_Adadelta_at(index, value)  
replace_Adagrad_at(index, value)  
replace_Adam_at(index, value)  
replace_Adamax_at(index, value)  
replace_Extension_at(index, value)  
replace_Nadam_at(index, value)  
replace_RMSprop_at(index, value)  
replace_SGD_at(index, value)  
set_Adadelta(Adadelta)  
set_Adagrad(Adagrad)  
set_Adam(Adam)  
set_Adamax(Adamax)  
set_Extension(Extension)  
set_Nadam(Nadam)  
set_RMSprop(RMSprop)  
set_SGD(SGD)  
set_clipnorm(clipnorm)
```

```
    set_clipvalue (clipvalue)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Optimizers', mapping_=None)

    validate_REAL_NUMBER (value)

class PMML44Super.OptimumLiftGraph (Extension=None, LiftGraph=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='OptimumLiftGraph', namespacedef_="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='OptimumLiftGraph')

    exportChildren (outfile, level, namespace_="", name_='OptimumLiftGraph', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='OptimumLiftGraph')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_LiftGraph ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_LiftGraph (LiftGraph)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='OptimumLiftGraph', mapping_=None)

class PMML44Super.OutlierEffect (type_=None, startTime=None, magnitude=None, dampingCo-
                                efficient=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='OutlierEffect', namespacedef_="", pretty_print=True)
```

```

exportAttributes (outfile, level, already_processed, namespace_="", name_='OutlierEffect')
exportChildren (outfile, level, namespace_="", name_='OutlierEffect', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='OutlierEffect')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_dampingCoefficient()
get_magnitude()
get_startTime()
get_type()
hasContent_()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_dampingCoefficient (dampingCoefficient)
set_magnitude (magnitude)
set_startTime (startTime)
set_type (type_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='OutlierEffect', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.Output (Extension=None, OutputField=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_OutputField (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Output', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Output')
    exportChildren (outfile, level, namespace_="", name_='Output', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Output')
    exportLiteralAttributes (outfile, level, already_processed, name_)

```

```
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_OutputField ()
hasContent_ ()
insert_Extension_at (index, value)
insert_OutputField_at (index, value)
replace_Extension_at (index, value)
replace_OutputField_at (index, value)
set_Extension (Extension)
set_OutputField (OutputField)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Output', mapping_=None)
class PMML44Super.OutputField (name=None, displayName=None, optype=None, dataType=None,
                                targetField=None, feature='predictedValue', value=None,
                                numTopCategories=None, threshold=None, ruleFea-
                                ture='consequent', algorithm='exclusiveRecommendation',
                                rank='1', rankBasis='confidence', rankOrder='descending',
                                isMultiValued='0', segmentId=None, isFinalResult=True, Exten-
                                sion=None, Decisions=None, FieldRef=None, Constant=None,
                                NormContinuous=None, NormDiscrete=None, Discretize=None,
                                MapValues=None, TextIndex=None, Apply=None, Aggre-
                                gate=None, Lag=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='OutputField', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='OutputField')
exportChildren (outfile, level, namespace_="", name_='OutputField', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='OutputField')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Aggregate ()
get_Apply ()
get_Constant ()
```



```
get_Decisions()
get_Discretize()
get_Extension()
get_FieldRef()
get_Lag()
get_MapValues()
get_NormContinuous()
get_NormDiscrete()
get_TextIndex()
get_algorithm()
get_dataType()
get_displayName()
get_feature()
get_isFinalResult()
get_isMultiValued()
get_name()
get_numTopCategories()
get_optype()
get_rank()
get_rankBasis()
get_rankOrder()
get_ruleFeature()
get_segmentId()
get_targetField()
get_threshold()
get_value()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Aggregate(Aggregate)
set_Apply(Apply)
set_Constant(Constant)
set_Decisions(Decisions)
set_Discretize(Discretize)
set_Extension(Extension)
set_FieldRef(FieldRef)
```

```
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
set_algorithm (algorithm)
set_dataType (dataType)
set_displayName (displayName)
set_feature (feature)
set_isFinalResult (isFinalResult)
set_isMultiValued (isMultiValued)
set_name (name)
set_numTopCategories (numTopCategories)
set_optype (optype)
set_rank (rank)
set_rankBasis (rankBasis)
set_rankOrder (rankOrder)
set_ruleFeature (ruleFeature)
set_segmentId (segmentId)
set_targetField (targetField)
set_threshold (threshold)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='OutputField', mapping_=None)
validate_DATATYPE (value)
validate_FIELD_NAME (value)
validate_INT_NUMBER (value)
validate_OPTYPE (value)
validate_REAL_NUMBER (value)
validate_RESULT_FEATURE (value)
validate_RULE_FEATURE (value)
class PMML44Super.PCell (targetCategory=None, parameterName=None, beta=None, df=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='PCell', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='PCell')
exportChildren (outfile, level, namespace_="", name_='PCell', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='PCell')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_beta()
get_df()
get_parameterName()
get_targetCategory()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_beta (beta)
set_df (df)
set_parameterName (parameterName)
set_targetCategory (targetCategory)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PCell', mapping_=None)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.PCovCell (pRow=None, pCol=None, tRow=None, tCol=None, value=None, tar-
                           getCategory=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PCovCell', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PCovCell')
```

```
exportChildren (outfile, level, namespace_="", name_='PCovCell', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='PCovCell')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_pCol ()
get_pRow ()
get_tCol ()
get_tRow ()
get_targetCategory ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_pCol (pCol)
set_pRow (pRow)
set_tCol (tCol)
set_tRow (tRow)
set_targetCategory (targetCategory)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PCovCell', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.PCovMatrix (type_=None, Extension=None, PCovCell=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_PCovCell (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PCovMatrix', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PCovMatrix')
```

```

exportChildren (outfile, level, namespace_=' ', name_='PCovMatrix', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='PCovMatrix')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_PCovCell ()
get_type ()
hasContent _ ()
insert_Extension_at (index, value)
insert_PCovCell_at (index, value)
replace_Extension_at (index, value)
replace_PCovCell_at (index, value)
set_Extension (Extension)
set_PCovCell (PCovCell)
set_type (type_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PCovMatrix', mapping_=None)

class PMML44Super.PMML (version=None, Header=None, MiningBuildTask=None, DataDic-
                        tionary=None, TransformationDictionary=None, AssociationModel=None,
                        BayesianNetworkModel=None, BaselineModel=None, Clustering-
                        Model=None, DeepNetwork=None, AnomalyDetectionModel=None,
                        GaussianProcessModel=None, GeneralRegressionModel=None,
                        MiningModel=None, NaiveBayesModel=None, NearestNeighbor-
                        Model=None, NeuralNetwork=None, RegressionModel=None, RuleSet-
                        Model=None, SequenceModel=None, Scorecard=None, SupportVec-
                        torMachineModel=None, TextModel=None, TimeSeriesModel=None,
                        TreeModel=None, Extension=None)
Bases: PMML44Super.GeneratedSuper
add_AnomalyDetectionModel (value)
add_AssociationModel (value)
add_BaselineModel (value)
add_BayesianNetworkModel (value)
add_ClusteringModel (value)
add_DeepNetwork (value)
add_Extension (value)
add_GaussianProcessModel (value)
add_GeneralRegressionModel (value)

```

```
add_MiningModel (value)
add_NaiveBayesModel (value)
add_NearestNeighborModel (value)
add_NeuralNetwork (value)
add_RegressionModel (value)
add_RuleSetModel (value)
add_Scorecard (value)
add_SequenceModel (value)
add_SupportVectorMachineModel (value)
add_TextModel (value)
add_TimeSeriesModel (value)
add_TreeModel (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='PMML', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='PMML')
exportChildren (outfile, level, namespace_="", name_='PMML', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='PMML')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_AnomalyDetectionModel ()
get_AssociationModel ()
get_BaselineModel ()
get_BayesianNetworkModel ()
get_ClusteringModel ()
get_DataDictionary ()
get_DeepNetwork ()
get_Extension ()
get_GaussianProcessModel ()
get_GeneralRegressionModel ()
get_Header ()
get_MiningBuildTask ()
get_MiningModel ()
get_NaiveBayesModel ()
```

```
get_NearestNeighborModel()  
get_NeuralNetwork()  
get_RegressionModel()  
get_RuleSetModel()  
get_Scorecard()  
get_SequenceModel()  
get_SupportVectorMachineModel()  
get_TextModel()  
get_TimeSeriesModel()  
get_TransformationDictionary()  
get_TreeModel()  
get_version()  
hasContent_  
insert_AnomalyDetectionModel_at(index, value)  
insert_AssociationModel_at(index, value)  
insert_BaselineModel_at(index, value)  
insert_BayesianNetworkModel_at(index, value)  
insert_ClusteringModel_at(index, value)  
insert_DeepNetwork_at(index, value)  
insert_Extension_at(index, value)  
insert_GaussianProcessModel_at(index, value)  
insert_GeneralRegressionModel_at(index, value)  
insert_MiningModel_at(index, value)  
insert_NaiveBayesModel_at(index, value)  
insert_NearestNeighborModel_at(index, value)  
insert_NeuralNetwork_at(index, value)  
insert_RegressionModel_at(index, value)  
insert_RuleSetModel_at(index, value)  
insert_Scorecard_at(index, value)  
insert_SequenceModel_at(index, value)  
insert_SupportVectorMachineModel_at(index, value)  
insert_TextModel_at(index, value)  
insert_TimeSeriesModel_at(index, value)  
insert_TreeModel_at(index, value)  
replace_AnomalyDetectionModel_at(index, value)  
replace_AssociationModel_at(index, value)
```

```
replace_BaselineModel_at (index, value)
replace_BayesianNetworkModel_at (index, value)
replace_ClusteringModel_at (index, value)
replace_DeepNetwork_at (index, value)
replace_Extension_at (index, value)
replace_GaussianProcessModel_at (index, value)
replace_GeneralRegressionModel_at (index, value)
replace_MiningModel_at (index, value)
replace_NaiveBayesModel_at (index, value)
replace_NearestNeighborModel_at (index, value)
replace_NeuralNetwork_at (index, value)
replace_RegressionModel_at (index, value)
replace_RuleSetModel_at (index, value)
replace_Scorecard_at (index, value)
replace_SequenceModel_at (index, value)
replace_SupportVectorMachineModel_at (index, value)
replace_TextModel_at (index, value)
replace_TimeSeriesModel_at (index, value)
replace_TreeModel_at (index, value)
set_AnomalyDetectionModel (AnomalyDetectionModel)
set_AssociationModel (AssociationModel)
set_BaselineModel (BaselineModel)
set_BayesianNetworkModel (BayesianNetworkModel)
set_ClusteringModel (ClusteringModel)
set_DataDictionary (DataDictionary)
set_DeepNetwork (DeepNetwork)
set_Extension (Extension)
set_GaussianProcessModel (GaussianProcessModel)
set_GeneralRegressionModel (GeneralRegressionModel)
set_Header (Header)
set_MiningBuildTask (MiningBuildTask)
set_MiningModel (MiningModel)
set_NaiveBayesModel (NaiveBayesModel)
set_NearestNeighborModel (NearestNeighborModel)
set_NeuralNetwork (NeuralNetwork)
set_RegressionModel (RegressionModel)
```



```

    set_RuleSetModel (RuleSetModel)
    set_Scorecard (Scorecard)
    set_SequenceModel (SequenceModel)
    set_SupportVectorMachineModel (SupportVectorMachineModel)
    set_TextModel (TextModel)
    set_TimeSeriesModel (TimeSeriesModel)
    set_TransformationDictionary (TransformationDictionary)
    set_TreeModel (TreeModel)
    set_version (version)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='PMML', mapping_=None)
class PMML44Super.PPCell (value=None, predictorName=None, parameterName=None, targetCategory=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PPCell', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PPCell')
    exportChildren (outfile, level, namespace_="", name_='PPCell', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='PPCell')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_parameterName ()
    get_predictorName ()
    get_targetCategory ()
    get_value ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_parameterName (parameterName)

```

```
    set_predictorName (predictorName)
    set_targetCategory (targetCategory)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='PPCell', mapping_=None)
    validate_FIELD_NAME (value)
class PMML44Super.PPMatrix (Extension=None, PPCell=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_PPCell (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PPMatrix', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PPMatrix')
    exportChildren (outfile, level, namespace_="", name_='PPMatrix', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='PPMatrix')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_PPCell ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_PPCell_at (index, value)
    replace_Extension_at (index, value)
    replace_PPCell_at (index, value)
    set_Extension (Extension)
    set_PPCell (PPCell)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='PPMatrix', mapping_=None)
class PMML44Super.PairCounts (value=None, Extension=None, TargetValueCounts=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
```

```

build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='PairCounts', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='PairCounts')
exportChildren (outfile, level, namespace_="", name_='PairCounts', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='PairCounts')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_TargetValueCounts ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_TargetValueCounts (TargetValueCounts)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PairCounts', mapping_=None)
class PMML44Super.ParamMatrix (Extension=None, PCell=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_PCell (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ParamMatrix', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ParamMatrix')
    exportChildren (outfile, level, namespace_="", name_='ParamMatrix', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ParamMatrix')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)

```

```
static factory (*args_, **kwargs_)
get_Extension()
get_PCell()
hasContent_()
insert_Extension_at (index, value)
insert_PCell_at (index, value)
replace_Extension_at (index, value)
replace_PCell_at (index, value)
set_Extension (Extension)
set_PCell (PCell)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ParamMatrix', mapping_=None)
class PMML44Super.Parameter (name=None, value=None, label=None, referencePoint='0', Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Parameter', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Parameter')
    exportChildren (outfile, level, namespace_="", name_='Parameter', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='Parameter')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension()
    get_label()
    get_name()
    get_referencePoint()
    get_value()
    hasContent_()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
```

```

    set_label(label)
    set_name(name)
    set_referencePoint(referencePoint)
    set_value(value)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Parameter', mapping_=None)
    validate_REAL_NUMBER(value)
class PMML44Super.ParameterField(name=None, otype=None, dataType=None, display-
                                Name=None)
    Bases: PMML44Super.GeneratedSuper
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='ParameterField', namespacedef_="",
          pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='ParameterField')
    exportChildren(outfile, level, namespace_="", name_='ParameterField', fromsubclass_=False,
                  pretty_print=True)
    exportLiteral(outfile, level, name_='ParameterField')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_dataType()
    get_displayName()
    get_name()
    get_otype()
    hasContent_()
    set_dataType(dataType)
    set_displayName(displayName)
    set_name(name)
    set_otype(otype)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='ParameterField', mapping_=None)
    validate_DATATYPE(value)
    validate_FIELD_NAME(value)
    validate_OPTYPE(value)

```

```
class PMML44Super.ParameterList (Extension=None, Parameter=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_Parameter (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='ParameterList', namespacesdef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='ParameterList')

    exportChildren (outfile, level, namespace_="", name_='ParameterList', fromsubclass_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='ParameterList')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_Parameter ()

    hasContent_ ()

    insert_Extension_at (index, value)

    insert_Parameter_at (index, value)

    replace_Extension_at (index, value)

    replace_Parameter_at (index, value)

    set_Extension (Extension)

    set_Parameter (Parameter)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='ParameterList', mapping_=None)

class PMML44Super.ParentValue (parent=None, value=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='ParentValue', namespacesdef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='ParentValue')

    exportChildren (outfile, level, namespace_="", name_='ParentValue', fromsubclass_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='ParentValue')
```

```

exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_parent ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_parent (parent)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ParentValue', mapping_=None)
validate_FIELD_NAME (value)

class PMML44Super.Partition (name=None, size=None, Extension=None, PartitionField-
                               Stats=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    add_PartitionFieldStats (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='Partition', namespacedef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='Partition')
    exportChildren (outfile, level, namespace="", name_='Partition', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Partition')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_PartitionFieldStats ()
    get_name ()
    get_size ()
    hasContent_ ()

```

```
insert_Extension_at (index, value)
insert_PartitionFieldStats_at (index, value)
replace_Extension_at (index, value)
replace_PartitionFieldStats_at (index, value)
set_Extension (Extension)
set_PartitionFieldStats (PartitionFieldStats)
set_name (name)
set_size (size)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Partition', mapping_=None)
validate_NUMBER (value)
class PMML44Super.PartitionFieldStats (field=None, weighted='0', Extension=None,
                                         Counts=None, NumericInfo=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Array (value)
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PartitionFieldStats', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PartitionFieldStats')
    exportChildren (outfile, level, namespace_="", name_='PartitionFieldStats', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='PartitionFieldStats')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Counts ()
    get_Extension ()
    get_NumericInfo ()
    get_field ()
    get_weighted ()
    hasContent_ ()
    insert_Array_at (index, value)
    insert_Extension_at (index, value)
```



```
replace_Array_at (index, value)
replace_Extension_at (index, value)
set_Array (Array)
set_Counts (Counts)
set_Extension (Extension)
set_NumericInfo (NumericInfo)
set_field (field)
set_weighted (weighted)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PartitionFieldStats', mapping_=None)
validate_FIELD_NAME (value)
class PMML44Super.PastVariances (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PastVariances', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PastVariances')
    exportChildren (outfile, level, namespace_="", name_='PastVariances', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='PastVariances')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='PastVariances', mapping_=None)
```

```
class PMML44Super.PoissonDistribution (mean=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='PoissonDistribution', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='PoissonDistribution')

    exportChildren (outfile, level, namespace_="", name_='PoissonDistribution', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='PoissonDistribution')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_mean ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_mean (mean)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='PoissonDistribution', mapping_=None)

    validate_REAL_NUMBER (value)

class PMML44Super.PolynomialKernelType (description=None, gamma='1', coef0='1', de-
                                         gree='1', Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='PolynomialKernelType', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='PolynomialKernelType')

    exportChildren (outfile, level, namespace_="", name_='PolynomialKernelType', fromsub-
                    class_=False, pretty_print=True)
```

```

exportLiteral (outfile, level, name_='PolynomialKernelType')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_coef0 ()
get_degree ()
get_description ()
get_gamma ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_coef0 (coef0)
set_degree (degree)
set_description (description)
set_gamma (gamma)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PolynomialKernelType', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.PredictiveModelQuality (targetField=None,          dataName=None,
                                         dataUsage='training',      meanError=None,
                                         meanAbsoluteError=None,     meanSquared-
                                         Error=None,                rootMeanSquaredError=None,
                                         r_squared=None, adj_r_squared=None, sum-
                                         SquaredError=None,         sumSquaredRegres-
                                         sion=None,                 numOfRecords=None,      nu-
                                         mOfRecordsWeighted=None,   numOfPredic-
                                         tors=None, degreesOfFreedom=None, fStatis-
                                         tic=None, AIC=None, BIC=None, AICc=None,
                                         Extension=None,            ConfusionMatrix=None,
                                         LiftData=None, ROC=None)

Bases: PMML44Super.GeneratedSuper
add_Extension (value)
add_LiftData (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='PredictiveModelQuality', namespacedef="",
         pretty_print=True)

```

```
exportAttributes (outfile, level, already_processed, namespace_=",  
                  name_='PredictiveModelQuality')  
exportChildren (outfile, level, namespace_=", name_='PredictiveModelQuality', fromsub-  
                  class_=False, pretty_print=True)  
exportLiteral (outfile, level, name_='PredictiveModelQuality')  
exportLiteralAttributes (outfile, level, already_processed, name_)  
exportLiteralChildren (outfile, level, name_)  
static factory (*args_, **kwargs_)  
get_AIC()  
get_AICc()  
get_BIC()  
get_ConfusionMatrix()  
get_Extension()  
get_LiftData()  
get_ROC()  
get_adj_r_squared()  
get_dataName()  
get_dataUsage()  
get_degreesOfFreedom()  
get_fStatistic()  
get_meanAbsoluteError()  
get_meanError()  
get_meanSquaredError()  
get_numOfPredictors()  
get_numOfRecords()  
get_numOfRecordsWeighted()  
get_r_squared()  
get_rootMeanSquaredError()  
get_sumSquaredError()  
get_sumSquaredRegression()  
get_targetField()  
hasContent_()  
insert_Extension_at (index, value)  
insert_LiftData_at (index, value)  
replace_Extension_at (index, value)  
replace_LiftData_at (index, value)  
set_AIC (AIC)
```

```

set_AICc (AICc)
set_BIC (BIC)
set_ConfusionMatrix (ConfusionMatrix)
set_Extension (Extension)
set_LiftData (LiftData)
set_ROC (ROC)
set_adj_r_squared (adj_r_squared)
set_dataName (dataName)
set_dataUsage (dataUsage)
set_degreesOfFreedom (degreesOfFreedom)
set_fStatistic (fStatistic)
set_meanAbsoluteError (meanAbsoluteError)
set_meanError (meanError)
set_meanSquaredError (meanSquaredError)
set_numOfPredictors (numOfPredictors)
set_numOfRecords (numOfRecords)
set_numOfRecordsWeighted (numOfRecordsWeighted)
set_r_squared (r_squared)
set_rootMeanSquaredError (rootMeanSquaredError)
set_sumSquaredError (sumSquaredError)
set_sumSquaredRegression (sumSquaredRegression)
set_targetField (targetField)
subclass = None
superclass = None
to_etree (parent_element=None, name_='PredictiveModelQuality', mapping_=None)
validate_FIELD_NAME (value)
validate_NUMBER (value)

class PMML44Super.Predictor (name=None, contrastMatrixType=None, Extension=None, Categories=None, Matrix=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Predictor', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Predictor')

```

```
exportChildren (outfile, level, namespace_="", name_='Predictor', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='Predictor')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Categories ()
get_Extension ()
get_Matrix ()
get_contrastMatrixType ()
get_name ()
hasContent _ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Categories (Categories)
set_Extension (Extension)
set_Matrix (Matrix)
set_contrastMatrixType (contrastMatrixType)
set_name (name)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Predictor', mapping_=None)
validate_FIELD_NAME (value)

class PMML44Super.PredictorTerm (name=None, coefficient=None, Extension=None, Field-
                                Ref=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_FieldRef (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='PredictorTerm', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='PredictorTerm')
    exportChildren (outfile, level, namespace_="", name_='PredictorTerm', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='PredictorTerm')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
```

```

    static factory(*args_,**kwargs_)
    get_Extension()
    get_FieldRef()
    get_coefficient()
    get_name()
    hasContent_()
    insert_Extension_at(index, value)
    insert_FieldRef_at(index, value)
    replace_Extension_at(index, value)
    replace_FieldRef_at(index, value)
    set_Extension(Extension)
    set_FieldRef(FieldRef)
    set_coefficient(coeffcient)
    set_name(name)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='PredictorTerm', mapping_=None)
    validate_FIELD_NAME(value)
    validate_REAL_NUMBER(value)
class PMML44Super.PsiVector(targetField=None, variance=None, Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_=' ', name_='PsiVector', namespacedef_=' ', pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_=' ', name_='PsiVector')
    exportChildren(outfile, level, namespace_=' ', name_='PsiVector', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='PsiVector')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_,**kwargs_)
    get_Array()
    get_Extension()
    get_targetField()
    get_variance()

```

```
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Array(Array)
set_Extension(Extension)
set_targetField(targetField)
set_variance(variance)
subclass = None
superclass = None
to_etree(parent_element=None, name_='PsiVector', mapping_=None)
class PMML44Super.Quantile(quantileLimit=None, quantileValue=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='Quantile', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='Quantile')
    exportChildren(outfile, level, namespace_="", name_='Quantile', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='Quantile')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_quantileLimit()
    get_quantileValue()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_quantileLimit(quantileLimit)
    set_quantileValue(quantileValue)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='Quantile', mapping_=None)
    validate_NUMBER(value)
```



```

    validate_PERCENTAGE_NUMBER (value)

class PMML44Super.REAL_Entries
    Bases: PMML44Super.GeneratedsSuper

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='REAL-Entries', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='REAL-Entries')

    exportChildren (outfile, level, namespace_="", name_='REAL-Entries', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='REAL-Entries')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    hasContent_ ()

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='REAL-Entries', mapping_=None)

class PMML44Super.REAL_SparseArray (n=None, default_value='0', Indices=None,
                                     REAL_Entries=None)
    Bases: PMML44Super.GeneratedsSuper

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='REAL-SparseArray', namespacedef_="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='REAL-
                     SparseArray')

    exportChildren (outfile, level, namespace_="", name_='REAL-SparseArray', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='REAL-SparseArray')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Indices ()

    get_REAL_Entries ()

    get_defaultValue ()

    get_n ()

    hasContent_ ()

```

```
set_Indices (Indices)
set_REAL_Entries (REAL_Entries)
set_defaultValue (defaultValue)
set_n (n)
subclass = None
superclass = None
to_etree (parent_element=None, name_='REAL-SparseArray', mapping_=None)
validate_INT_NUMBER (value)
validate_Indices (value)
validate_REAL_Entries (value)
validate_REAL_NUMBER (value)
class PMML44Super.RMSprop (learningRate=None, rho=None, decayRate=None, epsilon=None, Ex-
                           tension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='RMSprop', namespacesdef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='RMSprop')
    exportChildren (outfile, level, namespace_="", name_='RMSprop', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='RMSprop')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_decayRate ()
    get_epsilon ()
    get_learningRate ()
    get_rho ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_decayRate (decayRate)
    set_epsilon (epsilon)
```

```

    set_learningRate (learningRate)

    set_rho (rho)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='RMSprop', mapping_=None)

    validate_REAL_NUMBER (value)

class PMML44Super.ROC (positiveTargetFieldValue=None, positiveTargetFieldDisplayValue=None, neg-
                        ativeTargetFieldValue=None, negativeTargetFieldDisplayValue=None, Exten-
                        sion=None, ROCGraph=None)
Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace="", name_='ROC', namespacedef="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace="", name_='ROC')

    exportChildren (outfile, level, namespace="", name_='ROC', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='ROC')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_ROCGraph ()

    get_negativeTargetFieldDisplayValue ()

    get_negativeTargetFieldValue ()

    get_positiveTargetFieldDisplayValue ()

    get_positiveTargetFieldValue ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_ROCGraph (ROCGraph)

    set_negativeTargetFieldDisplayValue (negativeTargetFieldDisplayValue)

    set_negativeTargetFieldValue (negativeTargetFieldValue)

    set_positiveTargetFieldDisplayValue (positiveTargetFieldDisplayValue)

    set_positiveTargetFieldValue (positiveTargetFieldValue)

    subclass = None

```

```
    superclass = None
    to_etree (parent_element=None, name_='ROC', mapping_=None)
class PMML44Super.ROCGraph (Extension=None, XCoordinates=None, YCoordinates=None, BoundaryValues=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ROCGraph', namespacesdef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ROCGraph')
    exportChildren (outfile, level, namespace_="", name_='ROCGraph', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='ROCGraph')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_BoundaryValues ()
    get_Extension ()
    get_XCoordinates ()
    get_YCoordinates ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_BoundaryValues (BoundaryValues)
    set_Extension (Extension)
    set_XCoordinates (XCoordinates)
    set_YCoordinates (YCoordinates)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='ROCGraph', mapping_=None)
class PMML44Super.RadialBasisKernel (description=None, gamma='I', noiseVariance='I', lambda_='I', Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
```

```

export (outfile, level, namespace_="", name_='RadialBasisKernel', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='RadialBasisKernel')
exportChildren (outfile, level, namespace_="", name_='RadialBasisKernel', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='RadialBasisKernel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_description()
get_gamma()
get_lambda()
get_noiseVariance()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_description (description)
set_gamma (gamma)
set_lambda (lambda_)
set_noiseVariance (noiseVariance)
subclass = None
superclass = None
to_etree (parent_element=None, name_='RadialBasisKernel', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.RadialBasisKernelType (description=None, gamma='1', Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='RadialBasisKernelType', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
            name_='RadialBasisKernelType')
    exportChildren (outfile, level, namespace_="", name_='RadialBasisKernelType', fromsub-
            class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='RadialBasisKernelType')

```

```
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_description ()
get_gamma ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_description (description)
set_gamma (gamma)
subclass = None
superclass = None
to_etree (parent_element=None, name_='RadialBasisKernelType', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.RandomLiftGraph (Extension=None, LiftGraph=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='RandomLiftGraph', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='RandomLiftGraph')
    exportChildren (outfile, level, namespace_="", name_='RandomLiftGraph', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='RandomLiftGraph')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_LiftGraph ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
```

```

    set_LiftGraph (LiftGraph)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='RandomLiftGraph', mapping_=None)

class PMML44Super.Reggression (modelName=None, functionName=None, algorithmName=None,
                                normalizationMethod='none', Extension=None, Output=None,
                                ModelStats=None, Targets=None, LocalTransformations=None,
                                ResultField=None, RegressionTable=None)

Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_RegressionTable (value)

    add_ResultField (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Regression', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Regression')

    exportChildren (outfile, level, namespace_="", name_='Regression', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='Regression')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_LocalTransformations ()

    get_ModelStats ()

    get_Output ()

    get_RegressionTable ()

    get_ResultField ()

    get_Targets ()

    get_algorithmName ()

    get_functionName ()

    get_modelName ()

    get_normalizationMethod ()

    hasContent_ ()

    insert_Extension_at (index, value)

    insert_RegressionTable_at (index, value)

    insert_ResultField_at (index, value)

```

```
replace_Extension_at (index, value)
replace_RegressionTable_at (index, value)
replace_ResultField_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_ModelStats (ModelStats)
set_Output (Output)
set_RegressionTable (RegressionTable)
set_ResultField (ResultField)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_modelName (modelName)
set_normalizationMethod (normalizationMethod)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Regression', mapping_=None)
validate_MINING_FUNCTION (value)
validate_REGRESSIONNORMALIZATIONMETHOD (value)
class PMML44Super.RegressionModel (modelName=None, functionName=None, algorithm-
    Name=None, modelType=None, targetFieldName=None,
    normalizationMethod='none', isScorable=True, Min-
    ingSchema=None, Output=None, ModelStats=None,
    ModelExplanation=None, Targets=None, LocalTransfor-
    mations=None, RegressionTable=None, ModelVerifica-
    tion=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
add_RegressionTable (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace="", name_='RegressionModel', namespacedef="",
    pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="", name_='RegressionModel')
exportChildren (outfile, level, namespace="", name_='RegressionModel', fromsubclass_=False,
    pretty_print=True)
exportLiteral (outfile, level, name_='RegressionModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
```



```
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_RegressionTable ()
get_Targets ()
get_algorithmName ()
get_functionName ()
get_isScorable ()
get_modelName ()
get_modelType ()
get_normalizationMethod ()
get_targetFieldName ()
hasContent_ ()
insert_Extension_at (index, value)
insert_RegressionTable_at (index, value)
replace_Extension_at (index, value)
replace_RegressionTable_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_RegressionTable (RegressionTable)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
```

```
set_modelType (modelType)
set_normalizationMethod (normalizationMethod)
set_targetFieldName (targetFieldName)
subclass = None
superclass = None
to_etree (parent_element=None, name_='RegressionModel', mapping_=None)
validate_FIELD_NAME (value)
validate_MINING_FUNCTION (value)
validate_REGRESSIONNORMALIZATIONMETHOD (value)
class PMML44Super.RegressionTable (intercept=None, targetCategory=None, Extension=None,
                                   NumericPredictor=None, CategoricalPredictor=None, PredictorTerm=None)
    Bases: PMML44Super.GeneratedSuper
    add_CategoricalPredictor (value)
    add_Extension (value)
    add_NumericPredictor (value)
    add_PredictorTerm (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='RegressionTable', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='RegressionTable')
    exportChildren (outfile, level, namespace_="", name_='RegressionTable', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='RegressionTable')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_CategoricalPredictor ()
    get_Extension ()
    get_NumericPredictor ()
    get_PredictorTerm ()
    get_intercept ()
    get_targetCategory ()
    hasContent_ ()
    insert_CategoricalPredictor_at (index, value)
    insert_Extension_at (index, value)
```

```

insert_NumericPredictor_at (index, value)
insert_PredictorTerm_at (index, value)
replace_CategoricalPredictor_at (index, value)
replace_Extension_at (index, value)
replace_NumericPredictor_at (index, value)
replace_PredictorTerm_at (index, value)
set_CategoricalPredictor (CategoricalPredictor)
set_Extension (Extension)
set_NumericPredictor (NumericPredictor)
set_PredictorTerm (PredictorTerm)
set_intercept (intercept)
set_targetCategory (targetCategory)
subclass = None
superclass = None
to_etree (parent_element=None, name_='RegressionTable', mapping_=None)
validate_REAL_NUMBER (value)
class PMML44Super.RegressorValues (Extension=None, TimeSeries=None, TrendCoeffi-
                                cients=None, TransferFunctionValues=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='RegressorValues', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='RegressorValues')
    exportChildren (outfile, level, namespace="", name_='RegressorValues', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='RegressorValues')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_TimeSeries ()
    get_TransferFunctionValues ()
    get_TrendCoefficients ()
    hasContent_ ()
    insert_Extension_at (index, value)

```

```
replace_Extension_at (index, value)
set_Extension (Extension)
set_TimeSeries (TimeSeries)
set_TransferFunctionValues (TransferFunctionValues)
set_TrendCoefficients (TrendCoefficients)
subclass = None
superclass = None
to_etree (parent_element=None, name_='RegressorValues', mapping_=None)
class PMML44Super.ResidualSquareCoefficients (Extension=None, Residuals=None, MACo-
                                              efficients=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='ResidualSquareCoefficients', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="",
                     name_='ResidualSquareCoefficients')
    exportChildren (outfile, level, namespace="", name_='ResidualSquareCoefficients', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='ResidualSquareCoefficients')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_MACoefficients ()
    get_Residuals ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_MACoefficients (MACoefficients)
    set_Residuals (Residuals)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='ResidualSquareCoefficients', mapping_=None)
class PMML44Super.Residuals (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
```

```

    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Residuals', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Residuals')
    exportChildren (outfile, level, namespace_="", name_='Residuals', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='Residuals')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Residuals', mapping_=None)
class PMML44Super.ResultField (name=None, displayName=None, optype=None, dataType=None,
                                feature=None, value=None, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ResultField', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ResultField')
    exportChildren (outfile, level, namespace_="", name_='ResultField', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ResultField')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()

```

```
get_dataType ()
get_displayName ()
get_feature ()
get_name ()
get_optype ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_dataType (dataType)
set_displayName (displayName)
set_feature (feature)
set_name (name)
set_optype (optype)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ResultField', mapping_=None)
validate_DATATYPE (value)
validate_FIELD_NAME (value)
validate_OPTYPE (value)
validate_RESULT_FEATURE (value)
class PMML44Super.RuleSelectionMethod (criterion=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='RuleSelectionMethod', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='RuleSelectionMethod')
    exportChildren (outfile, level, namespace_="", name_='RuleSelectionMethod', fromsub-
                    class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='RuleSelectionMethod')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
```

```

    static factory(*args_,**kwargs_)
    get_Extension()
    get_criterion()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_Extension(Extension)
    set_criterion(criterion)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='RuleSelectionMethod', mapping_=None)
class PMML44Super.RuleSet(recordCount=None, nbCorrect=None, defaultScore=None, defaultCon-
                        fidence=None, Extension=None, RuleSelectionMethod=None, Score-
                        Distribution=None, SimpleRule=None, CompoundRule=None)
    Bases: PMML44Super.GeneratedsSuper
    add_CompoundRule(value)
    add_Extension(value)
    add_RuleSelectionMethod(value)
    add_ScoreDistribution(value)
    add_SimpleRule(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='RuleSet', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='RuleSet')
    exportChildren(outfile, level, namespace_="", name_='RuleSet', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='RuleSet')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_,**kwargs_)
    get_CompoundRule()
    get_Extension()
    get_RuleSelectionMethod()
    get_ScoreDistribution()
    get_SimpleRule()
    get_defaultConfidence()
    get_defaultScore()

```

```
get_nbCorrect ()
get_recordCount ()
hasContent_ ()
insert_CompoundRule_at (index, value)
insert_Extension_at (index, value)
insert_RuleSelectionMethod_at (index, value)
insert_ScoreDistribution_at (index, value)
insert_SimpleRule_at (index, value)
replace_CompoundRule_at (index, value)
replace_Extension_at (index, value)
replace_RuleSelectionMethod_at (index, value)
replace_ScoreDistribution_at (index, value)
replace_SimpleRule_at (index, value)
set_CompoundRule (CompoundRule)
set_Extension (Extension)
set_RuleSelectionMethod (RuleSelectionMethod)
set_ScoreDistribution (ScoreDistribution)
set_SimpleRule (SimpleRule)
set_defaultConfidence (defaultConfidence)
set_defaultScore (defaultScore)
set_nbCorrect (nbCorrect)
set_recordCount (recordCount)
subclass = None
superclass = None
to_etree (parent_element=None, name_='RuleSet', mapping_=None)
validate_NUMBER (value)

class PMML44Super.RuleSetModel (modelName=None,      functionName=None,      algorithm-
                                Name=None, isScorable=True, MiningSchema=None, Out-
                                put=None, ModelStats=None, ModelExplanation=None,
                                Targets=None, LocalTransformations=None, RuleSet=None,
                                ModelVerification=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='RuleSetModel', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='RuleSetModel')
```



```
exportChildren (outfile, level, namespace_="", name_='RuleSetModel', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='RuleSetModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_RuleSet ()
get_Targets ()
get_algorithmName ()
get_functionName ()
get_isScorable ()
get_modelName ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_RuleSet (RuleSet)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
subclass = None
superclass = None
```

```
to_etree (parent_element=None, name_='RuleSetModel', mapping_=None)

validate_MINING_FUNCTION (value)

class PMML44Super.SGD (learningRate=None, momentum=None, decayRate=None, nesterov=None,
                        Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='SGD', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='SGD')

    exportChildren (outfile, level, namespace_="", name_='SGD', fromsubclass_=False,
                    pretty_print=True)

    exportLiteral (outfile, level, name_='SGD')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_decayRate ()

    get_learningRate ()

    get_momentum ()

    get_nesterov ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_decayRate (decayRate)

    set_learningRate (learningRate)

    set_momentum (momentum)

    set_nesterov (nesterov)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='SGD', mapping_=None)

    validate_REAL_NUMBER (value)

class PMML44Super.ScoreDistribution (value=None, recordCount=None, confidence=None,
                                     probability=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)
```

```

build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_=", name_='ScoreDistribution', namespacedef_=",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_=", name_='ScoreDistribution')
exportChildren (outfile, level, namespace_=", name_='ScoreDistribution', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='ScoreDistribution')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_confidence ()
get_probability ()
get_recordCount ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_confidence (confidence)
set_probability (probability)
set_recordCount (recordCount)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='ScoreDistribution', mapping_=None)
validate_NUMBER (value)
validate_PROB_NUMBER (value)

class PMML44Super.Scorecard (modelName=None, functionName=None, algorithm-
                             Name=None, initialScore='0', useReasonCodes=True, rea-
                             sonCodeAlgorithm='pointsBelow', baselineScore=None, base-
                             lineMethod='other', isScorable=True, MiningSchema=None,
                             Output=None, ModelStats=None, ModelExplanation=None, Tar-
                             gets=None, LocalTransformations=None, Characteristics=None,
                             ModelVerification=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)

```

```
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Scorecard', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Scorecard')
exportChildren (outfile, level, namespace_="", name_='Scorecard', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Scorecard')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Characteristics ()
get_Extension ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_Targets ()
get_algorithmName ()
get_baselineMethod ()
get_baselineScore ()
get_functionName ()
get_initialScore ()
get_isScorable ()
get_modelName ()
get_reasonCodeAlgorithm ()
get_useReasonCodes ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Characteristics (Characteristics)
set_Extension (Extension)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
```

```

set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_Targets (Targets)
set_algorithmName (algorithmName)
set_baselineMethod (baselineMethod)
set_baselineScore (baselineScore)
set_functionName (functionName)
set_initialScore (initialScore)
set_isScorable (isScorable)
set_modelName (modelName)
set_reasonCodeAlgorithm (reasonCodeAlgorithm)
set_useReasonCodes (useReasonCodes)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Scorecard', mapping_=None)
validate_MINING_FUNCTION (value)
validate_NUMBER (value)
class PMML44Super.SeasonalComponent (P=None, D=None, Q=None, period=None, Extension=None, AR=None, MA=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SeasonalComponent', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='SeasonalComponent')
    exportChildren (outfile, level, namespace_="", name_='SeasonalComponent', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='SeasonalComponent')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_AR ()
    get_D ()
    get_Extension ()

```

```
get_MA()
get_P()
get_Q()
get_period()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_AR(AR)
set_D(D)
set_Extension(Extension)
set_MA(MA)
set_P(P)
set_Q(Q)
set_period(period)
subclass = None
superclass = None
to_etree(parent_element=None, name_='SeasonalComponent', mapping_=None)
validate_INT_NUMBER(value)
class PMML44Super.SeasonalFactor(difference='0', maximumOrder=None, Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_=' ', name_='SeasonalFactor', namespacedef_=' ',
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_=' ', name_='SeasonalFactor')
    exportChildren(outfile, level, namespace_=' ', name_='SeasonalFactor', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral(outfile, level, name_='SeasonalFactor')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    get_Extension()
    get_difference()
    get_maximumOrder()
```

```

hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Array(Array)
set_Extension(Extension)
set_difference(difference)
set_maximumOrder(maximumOrder)
subclass = None
superclass = None
to_etree(parent_element=None, name_='SeasonalFactor', mapping_=None)
validate_INT_NUMBER(value)

class PMML44Super.SeasonalTrendDecomposition
    Bases: PMML44Super.GeneratedsSuper

    build(node)

    buildAttributes(node, attrs, already_processed)

    buildChildren(child_, node, nodeName_, fromsubclass_=False)

    export(outfile, level, namespace="", name_='SeasonalTrendDecomposition', namespacedef="",
           pretty_print=True)

    exportAttributes(outfile, level, already_processed, namespace="",
                     name_='SeasonalTrendDecomposition')

    exportChildren(outfile, level, namespace="", name_='SeasonalTrendDecomposition', fromsub-
                   class_=False, pretty_print=True)

    exportLiteral(outfile, level, name_='SeasonalTrendDecomposition')

    exportLiteralAttributes(outfile, level, already_processed, name_)

    exportLiteralChildren(outfile, level, name_)

    static factory(*args_, **kwargs_)

    hasContent_()

    subclass = None

    superclass = None

    to_etree(parent_element=None, name_='SeasonalTrendDecomposition', mapping_=None)

class PMML44Super.Seasonality_ExpoSmooth(type_=None, period=None, unit=None,
                                           phase=None, delta=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper

    build(node)

    buildAttributes(node, attrs, already_processed)

    buildChildren(child_, node, nodeName_, fromsubclass_=False)

    export(outfile, level, namespace="", name_='Seasonality_ExpoSmooth', namespacedef="",
           pretty_print=True)

    exportAttributes(outfile, level, already_processed, namespace="",
                     name_='Seasonality_ExpoSmooth')

```

```
exportChildren (outfile, level, namespace_=' ', name_='Seasonality_ExpoSmooth', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='Seasonality_ExpoSmooth')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_delta ()
get_period ()
get_phase ()
get_type ()
get_unit ()
hasContent_ ()
set_Array (Array)
set_delta (delta)
set_period (period)
set_phase (phase)
set_type (type_)
set_unit (unit)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Seasonality_ExpoSmooth', mapping_=None)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)

class PMML44Super.Segment (id=None, weight='1', Extension=None, SimplePredicate=None, CompoundPredicate=None, SimpleSetPredicate=None, True_=None, False_=None, AssociationModel=None, BayesianNetworkModel=None, BaselineModel=None, ClusteringModel=None, DeepNetwork=None, AnomalyDetectionModel=None, GaussianProcessModel=None, GeneralRegressionModel=None, MiningModel=None, NaiveBayesModel=None, NearestNeighborModel=None, NeuralNetwork=None, RegressionModel=None, RuleSetModel=None, SequenceModel=None, Scorecard=None, SupportVectorMachineModel=None, TextModel=None, TimeSeriesModel=None, TreeModel=None, VariableWeight=None)
    Bases: PMML44Super.GeneratedSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
```



```
export (outfile, level, namespace_="", name_='Segment', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Segment')
exportChildren (outfile, level, namespace_="", name_='Segment', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Segment')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_AnomalyDetectionModel()
get_AssociationModel()
get_BaselineModel()
get_BayesianNetworkModel()
get_ClusteringModel()
get_CompoundPredicate()
get_DeepNetwork()
get_Extension()
get_False()
get_GaussianProcessModel()
get_GeneralRegressionModel()
get_MiningModel()
get_NaiveBayesModel()
get_NearestNeighborModel()
get_NeuralNetwork()
get_RegressionModel()
get_RuleSetModel()
get_Scorecard()
get_SequenceModel()
get_SimplePredicate()
get_SimpleSetPredicate()
get_SupportVectorMachineModel()
get_TextModel()
get_TimeSeriesModel()
get_TreeModel()
get_True()
get_VariableWeight()
get_id()
get_weight()
```

```
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_AnomalyDetectionModel(AnomalyDetectionModel)
set_AssociationModel(AssociationModel)
set_BaselineModel(BaselineModel)
set_BayesianNetworkModel(BayesianNetworkModel)
set_ClusteringModel(ClusteringModel)
set_CompoundPredicate(CompoundPredicate)
set_DeepNetwork(DeepNetwork)
set_Extension(Extension)
set_False(False_)
set_GaussianProcessModel(GaussianProcessModel)
set_GeneralRegressionModel(GeneralRegressionModel)
set_MiningModel(MiningModel)
set_NaiveBayesModel(NaiveBayesModel)
set_NearestNeighborModel(NearestNeighborModel)
set_NeuralNetwork(NeuralNetwork)
set_RegressionModel(RegressionModel)
set_RuleSetModel(RuleSetModel)
set_Scorecard(Scorecard)
set_SequenceModel(SequenceModel)
set_SimplePredicate(SimplePredicate)
set_SimpleSetPredicate(SimpleSetPredicate)
set_SupportVectorMachineModel(SupportVectorMachineModel)
set_TextModel(TextModel)
set_TimeSeriesModel(TimeSeriesModel)
set_TreeModel(TreeModel)
set_True(True_)
set_VariableWeight(VariableWeight)
set_id(id)
set_weight(weight)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Segment', mapping_=None)
validate_NUMBER(value)
```

```

class PMML44Super.Segmentation (multipleModelMethod=None, missingThreshold='1', Extension=None, Segment=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_Segment (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Segmentation', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Segmentation')
    exportChildren (outfile, level, namespace_="", name_='Segmentation', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='Segmentation')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_Segment ()
    get_missingThreshold ()
    get_multipleModelMethod ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_Segment_at (index, value)
    replace_Extension_at (index, value)
    replace_Segment_at (index, value)
    set_Extension (Extension)
    set_Segment (Segment)
    set_missingThreshold (missingThreshold)
    set_multipleModelMethod (multipleModelMethod)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Segmentation', mapping_=None)
    validate_MULTIPLE_MODEL_METHOD (value)
    validate_PROB_NUMBER (value)

class PMML44Super.Sequence (id=None, numberOfSets=None, occurrence=None, support=None, Extension=None, Delimiter=None, SetReference=None, Time=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Delimiter (value)

```

```
add_Extension (value)
add_SetReference (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Sequence', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Sequence')
exportChildren (outfile, level, namespace_="", name_='Sequence', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='Sequence')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Delimiter ()
get_Extension ()
get_SetReference ()
get_Time ()
get_id ()
get_numberOfSets ()
get_occurrence ()
get_support ()
hasContent_ ()
insert_Delimiter_at (index, value)
insert_Extension_at (index, value)
insert_SetReference_at (index, value)
replace_Delimiter_at (index, value)
replace_Extension_at (index, value)
replace_SetReference_at (index, value)
set_Delimiter (Delimiter)
set_Extension (Extension)
set_SetReference (SetReference)
set_Time (Time)
set_id (id)
set_numberOfSets (numberOfSets)
set_occurrence (occurrence)
set_support (support)
subclass = None
```

```

superclass = None
to_etree (parent_element=None, name_='Sequence', mapping_=None)
validate_ELEMENT_ID (value)
validate_INT_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.SequenceModel (modelName=None, functionName=None, algorithm-
Name=None, numberOfTransactions=None, maxNumberOf-
ItemsPerTransaction=None, avgNumberOfItemsPerTransac-
tion=None, numberOfTransactionGroups=None, maxNum-
berOfTAsPerTAGroup=None, avgNumberOfTAsPerTA-
Group=None, isScorable=True, MiningSchema=None, Model-
Stats=None, LocalTransformations=None, Constraints=None,
Item=None, Itemset=None, SetPredicate=None, Se-
quence=None, SequenceRule=None, Extension=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
add_Item (value)
add_Itemset (value)
add_Sequence (value)
add_SequenceRule (value)
add_SetPredicate (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='SequenceModel', namespacedef_="",
pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='SequenceModel')
exportChildren (outfile, level, namespace_="", name_='SequenceModel', fromsubclass_=False,
pretty_print=True)
exportLiteral (outfile, level, name_='SequenceModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Constraints ()
get_Extension ()
get_Item ()
get_Itemset ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelStats ()

```

```
get_Sequence ()
get_SequenceRule ()
get_SetPredicate ()
get_algorithmName ()
get_avgNumberOfItemsPerTransaction ()
get_avgNumberOfTAsPerTAGroup ()
get_functionName ()
get_isScorable ()
get_maxNumberOfItemsPerTransaction ()
get_maxNumberOfTAsPerTAGroup ()
get_modelName ()
get_numberOfTransactionGroups ()
get_numberOfTransactions ()
hasContent_ ()
insert_Extension_at (index, value)
insert_Item_at (index, value)
insert_Itemset_at (index, value)
insert_SequenceRule_at (index, value)
insert_Sequence_at (index, value)
insert_SetPredicate_at (index, value)
replace_Extension_at (index, value)
replace_Item_at (index, value)
replace_Itemset_at (index, value)
replace_SequenceRule_at (index, value)
replace_Sequence_at (index, value)
replace_SetPredicate_at (index, value)
set_Constraints (Constraints)
set_Extension (Extension)
set_Item (Item)
set_Itemset (Itemset)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelStats (ModelStats)
set_Sequence (Sequence)
set_SequenceRule (SequenceRule)
set_SetPredicate (SetPredicate)
```

```

    set_algorithmName (algorithmName)
    set_avgNumberOfItemsPerTransaction (avgNumberOfItemsPerTransaction)
    set_avgNumberOfTAsPerTAGroup (avgNumberOfTAsPerTAGroup)
    set_functionName (functionName)
    set_isScorable (isScorable)
    set_maxNumberOfItemsPerTransaction (maxNumberOfItemsPerTransaction)
    set_maxNumberOfTAsPerTAGroup (maxNumberOfTAsPerTAGroup)
    set_modelName (modelName)
    set_numberOfTransactionGroups (numberOfTransactionGroups)
    set_numberOfTransactions (numberOfTransactions)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='SequenceModel', mapping_=None)
    validate_INT_NUMBER (value)
    validate_MINING_FUNCTION (value)
    validate_REAL_NUMBER (value)
class PMML44Super.SequenceReference (seqId=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SequenceReference', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='SequenceReference')
    exportChildren (outfile, level, namespace_="", name_='SequenceReference', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral (outfile, level, name_='SequenceReference')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_seqId ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)

```

```
    set_seqId(seqId)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='SequenceReference', mapping_=None)
    validate_ELEMENT_ID(value)
class PMML44Super.SequenceRule(id=None, numberOfSets=None, occurrence=None, sup-
                                port=None, confidence=None, lift=None, Extension=None,
                                AntecedentSequence=None, Delimiter=None, ConsequentSe-
                                quence=None, Time=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='SequenceRule', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='SequenceRule')
    exportChildren(outfile, level, namespace_="", name_='SequenceRule', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='SequenceRule')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_AntecedentSequence()
    get_ConsequentSequence()
    get_Delimiter()
    get_Extension()
    get_Time()
    get_confidence()
    get_id()
    get_lift()
    get_numberOfSets()
    get_occurrence()
    get_support()
    hasContent_()
    insert_Extension_at(index, value)
    replace_Extension_at(index, value)
    set_AntecedentSequence(AntecedentSequence)
    set_ConsequentSequence(ConsequentSequence)
```



```

    set_Delimiter(Delimiter)
    set_Extension(Extension)
    set_Time(Time)
    set_confidence(confidence)
    set_id(id)
    set_lift(lift)
    set_numberOfSets(numberOfSets)
    set_occurrence(occurrence)
    set_support(support)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='SequenceRule, mapping_=None)
    validate_ELEMENT_ID(value)
    validate_INT_NUMBER(value)
    validate_REAL_NUMBER(value)
class PMML44Super.SetPredicate(id=None, field=None, operator=None, Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='SetPredicate', namespacedef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='SetPredicate')
    exportChildren(outfile, level, namespace_="", name_='SetPredicate', fromsubclass_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='SetPredicate')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    get_Extension()
    get_field()
    get_id()
    get_operator()
    hasContent_()
    insert_Extension_at(index, value)

```

```
replace_Extension_at (index, value)
set_Array (Array)
set_Extension (Extension)
set_field (field)
set_id (id)
set_operator (operator)
subclass = None
superclass = None
to_etree (parent_element=None, name_='SetPredicate', mapping_=None)
validate_ELEMENT_ID (value)
validate_FIELD_NAME (value)
class PMML44Super.SetReference (setId=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SetReference', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='SetReference')
    exportChildren (outfile, level, namespace_="", name_='SetReference', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='SetReference')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_setId ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_setId (setId)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='SetReference', mapping_=None)
    validate_ELEMENT_ID (value)
```

```

class PMML44Super.SigmoidKernelType (description=None, gamma='1', coef0='1', Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SigmoidKernelType', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='SigmoidKernelType')
    exportChildren (outfile, level, namespace_="", name_='SigmoidKernelType', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='SigmoidKernelType')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_coef0 ()
    get_description ()
    get_gamma ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_coef0 (coef0)
    set_description (description)
    set_gamma (gamma)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='SigmoidKernelType', mapping_=None)
    validate_REAL_NUMBER (value)

class PMML44Super.SimplePredicate (field=None, operator=None, value=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='SimplePredicate', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='SimplePredicate')
exportChildren (outfile, level, namespace_="", name_='SimplePredicate', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='SimplePredicate')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_field()
get_operator()
get_value()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_field (field)
set_operator (operator)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='SimplePredicate', mapping_=None)
validate_FIELD_NAME (value)

class PMML44Super.SimpleRule (id=None, score=None, recordCount=None, nbCorrect=None,
                             confidence='1', weight='1', Extension=None, SimplePred-
                             icate=None, CompoundPredicate=None, SimpleSetPredi-
                             cate=None, True_=None, False_=None, ScoreDistribution=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_ScoreDistribution (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SimpleRule', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='SimpleRule')
    exportChildren (outfile, level, namespace_="", name_='SimpleRule', fromsubclass_=False,
        pretty_print=True)
    exportLiteral (outfile, level, name_='SimpleRule')
```

```
exportLiteralAttributes (outfile, level, already_processed, name_)  
exportLiteralChildren (outfile, level, name_)  
static factory (*args_, **kwargs_)  
get_CompoundPredicate ()  
get_Extension ()  
get_False ()  
get_ScoreDistribution ()  
get_SimplePredicate ()  
get_SimpleSetPredicate ()  
get_True ()  
get_confidence ()  
get_id ()  
get_nbCorrect ()  
get_recordCount ()  
get_score ()  
get_weight ()  
hasContent_ ()  
insert_Extension_at (index, value)  
insert_ScoreDistribution_at (index, value)  
replace_Extension_at (index, value)  
replace_ScoreDistribution_at (index, value)  
set_CompoundPredicate (CompoundPredicate)  
set_Extension (Extension)  
set_False (False_)  
set_ScoreDistribution (ScoreDistribution)  
set_SimplePredicate (SimplePredicate)  
set_SimpleSetPredicate (SimpleSetPredicate)  
set_True (True_)  
set_confidence (confidence)  
set_id (id)  
set_nbCorrect (nbCorrect)  
set_recordCount (recordCount)  
set_score (score)  
set_weight (weight)  
subclass = None  
superclass = None
```

```
    to_etree (parent_element=None, name_='SimpleRule', mapping_=None)
    validate_NUMBER (value)
class PMML44Super.SimpleSetPredicate (field=None, booleanOperator=None, Extension=None,
                                       Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SimpleSetPredicate', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='SimpleSetPredicate')
    exportChildren (outfile, level, namespace_="", name_='SimpleSetPredicate', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='SimpleSetPredicate')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    get_booleanOperator ()
    get_field ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
    set_booleanOperator (booleanOperator)
    set_field (field)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='SimpleSetPredicate', mapping_=None)
    validate_FIELD_NAME (value)
class PMML44Super.SpectralAnalysis
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
```

```

export (outfile, level, namespace_="", name_='SpectralAnalysis', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='SpectralAnalysis')
exportChildren (outfile, level, namespace_="", name_='SpectralAnalysis', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='SpectralAnalysis')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
hasContent_ ()
subclass = None
superclass = None
to_etree (parent_element=None, name_='SpectralAnalysis', mapping_=None)
class PMML44Super.StateSpaceModel (variance=None, period='none', intercept='0', Extension=None,
        StateVector=None, TransitionMatrix=None, MeasurementMatrix=None, PsiVector=None, DynamicRe-
        gressor=None)
Bases: PMML44Super.GeneratedsSuper
add_DynamicRegressor (value)
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='StateSpaceModel', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='StateSpaceModel')
exportChildren (outfile, level, namespace_="", name_='StateSpaceModel', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='StateSpaceModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_DynamicRegressor ()
get_Extension ()
get_MeasurementMatrix ()
get_PsiVector ()
get_StateVector ()
get_TransitionMatrix ()
get_intercept ()

```

```
get_period()
get_variance()
hasContent_()
insert_DynamicRegressor_at(index, value)
insert_Extension_at(index, value)
replace_DynamicRegressor_at(index, value)
replace_Extension_at(index, value)
set_DynamicRegressor(DynamicRegressor)
set_Extension(Extension)
set_MeasurementMatrix(MeasurementMatrix)
set_PsiVector(PsiVector)
set_StateVector(StateVector)
set_TransitionMatrix(TransitionMatrix)
set_intercept(intercept)
set_period(period)
set_variance(variance)
subclass = None
superclass = None
to_etree(parent_element=None, name_='StateSpaceModel', mapping_=None)
validate_REAL_NUMBER(value)

class PMML44Super.StateVector(Extension=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace="", name_='StateVector', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace="", name_='StateVector')
    exportChildren(outfile, level, namespace="", name_='StateVector', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='StateVector')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    get_Extension()
    hasContent_()
```



```

    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='StateVector', mapping_=None)
class PMML44Super.SupportVector (vectorId=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='SupportVector', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='SupportVector')
    exportChildren (outfile, level, namespace_="", name_='SupportVector', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='SupportVector')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_vectorId ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_vectorId (vectorId)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='SupportVector', mapping_=None)
    validate_VECTOR_ID (value)
class PMML44Super.SupportVectorMachine (targetCategory=None, alternateTargetCategory=None, threshold=None, Extension=None,
                                         SupportVectors=None, Coefficients=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)

```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='SupportVectorMachine', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
                    name_='SupportVectorMachine')
exportChildren (outfile, level, namespace_="", name_='SupportVectorMachine', fromsub-
                    class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='SupportVectorMachine')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Coefficients ()
get_Extension ()
get_SupportVectors ()
get_alternateTargetCategory ()
get_targetCategory ()
get_threshold ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Coefficients (Coefficients)
set_Extension (Extension)
set_SupportVectors (SupportVectors)
set_alternateTargetCategory (alternateTargetCategory)
set_targetCategory (targetCategory)
set_threshold (threshold)
subclass = None
superclass = None
to_etree (parent_element=None, name_='SupportVectorMachine', mapping_=None)
validate_REAL_NUMBER (value)
```

```

class PMML44Super.SupportVectorMachineModel (modelName=None,  functionName=None,
                                              algorithmName=None,  threshold='0',
                                              svmRepresentation='SupportVectors',
                                              classificationMethod='OneAgainstAll',
                                              maxWins=False,  isScorable=True,  MiningSchema=None,  Output=None,  ModelStats=None,  ModelExplanation=None,  Targets=None,  LocalTransformations=None,
                                              LinearKernelType=None,  PolynomialKernelType=None,  RadialBasisKernelType=None,  SigmoidKernelType=None,
                                              VectorDictionary=None,  SupportVectorMachine=None,  ModelVerification=None,
                                              Extension=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension (value)

add_SupportVectorMachine (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile, level, namespace_="", name_='SupportVectorMachineModel', namespacedef="",
        pretty_print=True)

exportAttributes (outfile, level, already_processed, namespace_="",
                  name_='SupportVectorMachineModel')

exportChildren (outfile, level, namespace_="", name_='SupportVectorMachineModel', fromsubclass_=False, pretty_print=True)

exportLiteral (outfile, level, name_='SupportVectorMachineModel')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_Extension ()

get_LinearKernelType ()

get_LocalTransformations ()

get_MiningSchema ()

get_ModelExplanation ()

get_ModelStats ()

get_ModelVerification ()

get_Output ()

get_PolynomialKernelType ()

get_RadialBasisKernelType ()

get_SigmoidKernelType ()

get_SupportVectorMachine ()

```

```
get_Targets ()
get_VectorDictionary ()
get_algorithmName ()
get_classificationMethod ()
get_functionName ()
get_isScorable ()
get_maxWins ()
get_modelName ()
get_svmRepresentation ()
get_threshold ()
hasContent_ ()
insert_Extension_at (index, value)
insert_SupportVectorMachine_at (index, value)
replace_Extension_at (index, value)
replace_SupportVectorMachine_at (index, value)
set_Extension (Extension)
set_LinearKernelType (LinearKernelType)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_PolynomialKernelType (PolynomialKernelType)
set_RadialBasisKernelType (RadialBasisKernelType)
set_SigmoidKernelType (SigmoidKernelType)
set_SupportVectorMachine (SupportVectorMachine)
set_Targets (Targets)
set_VectorDictionary (VectorDictionary)
set_algorithmName (algorithmName)
set_classificationMethod (classificationMethod)
set_functionName (functionName)
set_isScorable (isScorable)
set_maxWins (maxWins)
set_modelName (modelName)
set_svmRepresentation (svmRepresentation)
```

```

    set_threshold(threshold)

    subclass = None

    superclass = None

    to_etree(parent_element=None, name_='SupportVectorMachineModel', mapping_=None)

    validate_MINING_FUNCTION(value)

    validate_REAL_NUMBER(value)

    validate_SVM_CLASSIFICATION_METHOD(value)

    validate_SVM_REPRESENTATION(value)

class PMML44Super.SupportVectors(numberOfSupportVectors=None, numberOfAttributes=None,
                                Extension=None, SupportVector=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    add_SupportVector(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='SupportVectors', namespacedef="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='SupportVectors')
    exportChildren(outfile, level, namespace_="", name_='SupportVectors', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral(outfile, level, name_='SupportVectors')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_SupportVector()
    get_numberOfAttributes()
    get_numberOfSupportVectors()
    hasContent_()
    insert_Extension_at(index, value)
    insert_SupportVector_at(index, value)
    replace_Extension_at(index, value)
    replace_SupportVector_at(index, value)
    set_Extension(Extension)
    set_SupportVector(SupportVector)
    set_numberOfAttributes(numberOfAttributes)
    set_numberOfSupportVectors(numberOfSupportVectors)

```

```
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='SupportVectors', mapping_=None)
    validate_INT_NUMBER (value)

class PMML44Super.TableLocator (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TableLocator', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TableLocator')
    exportChildren (outfile, level, namespace_="", name_='TableLocator', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='TableLocator')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TableLocator', mapping_=None)

class PMML44Super.Target (field=None, optype=None, castInteger=None, min=None, max=None,
                          rescaleConstant=0, rescaleFactor=1, Extension=None, Target-
                          Value=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_TargetValue (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Target', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Target')
    exportChildren (outfile, level, namespace_="", name_='Target', fromsubclass_=False,
                    pretty_print=True)
```

```

exportLiteral (outfile, level, name_='Target')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_TargetValue ()
get_castInteger ()
get_field ()
get_max ()
get_min ()
get_optype ()
get_rescaleConstant ()
get_rescaleFactor ()
hasContent_ ()
insert_Extension_at (index, value)
insert_TargetValue_at (index, value)
replace_Extension_at (index, value)
replace_TargetValue_at (index, value)
set_Extension (Extension)
set_TargetValue (TargetValue)
set_castInteger (castInteger)
set_field (field)
set_max (max)
set_min (min)
set_optype (optype)
set_rescaleConstant (rescaleConstant)
set_rescaleFactor (rescaleFactor)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Target', mapping_=None)
validate_FIELD_NAME (value)
validate_OPTYPE (value)
class PMML44Super.TargetValue (value=None, displayValue=None, priorProbability=None, de-
                                faultValue=None, Extension=None, Partition=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)

```

```
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TargetValue', namespacesdef="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='TargetValue')
exportChildren (outfile, level, namespace_="", name_='TargetValue', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='TargetValue')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Partition ()
get_defaultValue ()
get_displayValue ()
get_priorProbability ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_Partition (Partition)
set_defaultValue (defaultValue)
set_displayValue (displayValue)
set_priorProbability (priorProbability)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TargetValue', mapping_=None)
validate_NUMBER (value)
validate_PROB_NUMBER (value)
class PMML44Super.TargetValueCount (value=None, count=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
```

```

export (outfile, level, namespace="", name_='TargetValueCount', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="", name_='TargetValueCount')
exportChildren (outfile, level, namespace="", name_='TargetValueCount', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='TargetValueCount')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_count()
get_value()
hasContent_()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_count (count)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TargetValueCount', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.TargetValueCounts (Extension=None, TargetValueCount=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_TargetValueCount (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='TargetValueCounts', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="", name_='TargetValueCounts')
    exportChildren (outfile, level, namespace="", name_='TargetValueCounts', fromsubclass_=False,
            pretty_print=True)
    exportLiteral (outfile, level, name_='TargetValueCounts')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)

```

```
get_Extension()
get_TargetValueCount()
hasContent_()
insert_Extension_at(index, value)
insert_TargetValueCount_at(index, value)
replace_Extension_at(index, value)
replace_TargetValueCount_at(index, value)
set_Extension(Extension)
set_TargetValueCount(TargetValueCount)
subclass = None
superclass = None
to_etree(parent_element=None, name_='TargetValueCounts', mapping_=None)
class PMML44Super.TargetValueStat(value=None, Extension=None, AnyDistribution=None,
                                   GaussianDistribution=None, PoissonDistribution=None,
                                   UniformDistribution=None)
Bases: PMML44Super.GeneratedSuper
add_Extension(value)
build(node)
buildAttributes(node, attrs, already_processed)
buildChildren(child_, node, nodeName_, fromsubclass_=False)
export(outfile, level, namespace_="", name_='TargetValueStat', namespacedef="",
        pretty_print=True)
exportAttributes(outfile, level, already_processed, namespace_="", name_='TargetValueStat')
exportChildren(outfile, level, namespace_="", name_='TargetValueStat', fromsubclass_=False,
               pretty_print=True)
exportLiteral(outfile, level, name_='TargetValueStat')
exportLiteralAttributes(outfile, level, already_processed, name_)
exportLiteralChildren(outfile, level, name_)
static factory(*args_, **kwargs_)
get_AnyDistribution()
get_Extension()
get_GaussianDistribution()
get_PoissonDistribution()
get_UniformDistribution()
get_value()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
```

```

    set_AnyDistribution (AnyDistribution)
    set_Extension (Extension)
    set_GaussianDistribution (GaussianDistribution)
    set_PoissonDistribution (PoissonDistribution)
    set_UniformDistribution (UniformDistribution)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TargetValueStat', mapping_=None)
class PMML44Super.TargetValueStats (Extension=None, TargetValueStat=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_TargetValueStat (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TargetValueStats', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TargetValueStats')
    exportChildren (outfile, level, namespace_="", name_='TargetValueStats', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='TargetValueStats')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_TargetValueStat ()
    hasContent_ ()
    insert_Extension_at (index, value)
    insert_TargetValueStat_at (index, value)
    replace_Extension_at (index, value)
    replace_TargetValueStat_at (index, value)
    set_Extension (Extension)
    set_TargetValueStat (TargetValueStat)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TargetValueStats', mapping_=None)

```

```
class PMML44Super.Targets (Extension=None, Target=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    add_Target (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Targets', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Targets')

    exportChildren (outfile, level, namespace_="", name_='Targets', fromsubclass_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='Targets')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_Target ()

    hasContent_ ()

    insert_Extension_at (index, value)

    insert_Target_at (index, value)

    replace_Extension_at (index, value)

    replace_Target_at (index, value)

    set_Extension (Extension)

    set_Target (Target)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='Targets', mapping_=None)

class PMML44Super.Taxonomy (name=None, Extension=None, ChildParent=None)
    Bases: PMML44Super.GeneratedsSuper

    add_ChildParent (value)

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='Taxonomy', namespacedef_="", pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="", name_='Taxonomy')

    exportChildren (outfile, level, namespace_="", name_='Taxonomy', fromsubclass_=False, pretty_print=True)
```

```
exportLiteral (outfile, level, name_='Taxonomy')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_ChildParent ()
get_Extension ()
get_name ()
hasContent_ ()
insert_ChildParent_at (index, value)
insert_Extension_at (index, value)
replace_ChildParent_at (index, value)
replace_Extension_at (index, value)
set_ChildParent (ChildParent)
set_Extension (Extension)
set_name (name)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Taxonomy', mapping_=None)
class PMML44Super.TestDistributions (field=None, testStatistic=None, resetValue='0.0',
                                     windowSize='0', weightField=None, normalization-
                                     Scheme=None, Extension=None, Baseline=None,
                                     Alternate=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TestDistributions', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='TestDistributions')
exportChildren (outfile, level, namespace_="", name_='TestDistributions', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='TestDistributions')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Alternate ()
get_Baseline ()
get_Extension ()
```

```
get_field()
get_normalizationScheme()
get_resetValue()
get_testStatistic()
get_weightField()
get_windowSize()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Alternate(Alternate)
set_Baseline(Baseline)
set_Extension(Extension)
set_field(field)
set_normalizationScheme(normalizationScheme)
set_resetValue(resetValue)
set_testStatistic(testStatistic)
set_weightField(weightField)
set_windowSize(windowSize)
subclass = None
superclass = None
to_etree(parent_element=None, name_='TestDistributions', mapping_=None)
validate_BASELINE_TEST_STATISTIC(value)
validate_FIELD_NAME(value)
validate_INT_NUMBER(value)
validate_REAL_NUMBER(value)
class PMML44Super.TextCorpus(Extension=None, TextDocument=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    add_TextDocument(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='TextCorpus', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='TextCorpus')
    exportChildren(outfile, level, namespace_="", name_='TextCorpus', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='TextCorpus')
```

```

exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_TextDocument ()
hasContent_ ()
insert_Extension_at (index, value)
insert_TextDocument_at (index, value)
replace_Extension_at (index, value)
replace_TextDocument_at (index, value)
set_Extension (Extension)
set_TextDocument (TextDocument)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TextCorpus', mapping_=None)
class PMML44Super.TextDictionary (Extension=None, Taxonomy=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TextDictionary', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TextDictionary')
    exportChildren (outfile, level, namespace_="", name_='TextDictionary', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='TextDictionary')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    get_Taxonomy ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)

```

```
    set_Taxonomy (Taxonomy)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TextDictionary', mapping_=None)
class PMML44Super.TextDocument (id=None, name=None, length=None, file=None, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TextDocument', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TextDocument')
    exportChildren (outfile, level, namespace_="", name_='TextDocument', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='TextDocument')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_file ()
    get_id ()
    get_length ()
    get_name ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_file (file)
    set_id (id)
    set_length (length)
    set_name (name)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TextDocument', mapping_=None)
    validate_INT_NUMBER (value)
```

```

class PMML44Super.TextIndex (textField=None, localTermWeights='termFrequency', isCaseSensitive=False, maxLevenshteinDistance=0, countHits='allHits', wordSeparatorCharacterRE='\s+', tokenize=True, Extension=None, TextIndexNormalization=None, FieldRef=None, Constant=None, NormContinuous=None, NormDiscrete=None, Discretize=None, MapValues=None, TextIndex_member=None, Apply=None, Aggregate=None, Lag=None)

Bases: PMML44Super.GeneratedSuper

add_Extension (value)

add_TextIndexNormalization (value)

build (node)

buildAttributes (node, attrs, already_processed)

buildChildren (child_, node, nodeName_, fromsubclass_=False)

export (outfile, level, namespace_="", name_='TextIndex', namespacedef_="", pretty_print=True)

exportAttributes (outfile, level, already_processed, namespace_="", name_='TextIndex')

exportChildren (outfile, level, namespace_="", name_='TextIndex', fromsubclass_=False, pretty_print=True)

exportLiteral (outfile, level, name_='TextIndex')

exportLiteralAttributes (outfile, level, already_processed, name_)

exportLiteralChildren (outfile, level, name_)

static factory (*args_, **kwargs_)

get_Aggregate ()

get_Apply ()

get_Constant ()

get_Discretize ()

get_Extension ()

get_FieldRef ()

get_Lag ()

get_MapValues ()

get_NormContinuous ()

get_NormDiscrete ()

get_TextIndex ()

get_TextIndexNormalization ()

get_countHits ()

get_isCaseSensitive ()

get_localTermWeights ()

get_maxLevenshteinDistance ()

get_textField ()

get_tokenize ()

```

```
get_wordSeparatorCharacterRE ()
hasContent_ ()
insert_Extension_at (index, value)
insert_TextIndexNormalization_at (index, value)
replace_Extension_at (index, value)
replace_TextIndexNormalization_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
set_TextIndexNormalization (TextIndexNormalization)
set_countHits (countHits)
set_isCaseSensitive (isCaseSensitive)
set_localTermWeights (localTermWeights)
set_maxLevenshteinDistance (maxLevenshteinDistance)
set_textField (textField)
set_tokenize (tokenize)
set_wordSeparatorCharacterRE (wordSeparatorCharacterRE)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TextIndex', mapping_=None)
validate_FIELD_NAME (value)

class PMML44Super.TextIndexNormalization (inField='string', outField='stem', regex-
Field='regex', recursive=False, isCaseSensi-
tive=None, maxLevenshteinDistance=None,
wordSeparatorCharacterRE=None, tok-
enize=None, Extension=None, TableLoca-
tor=None, InlineTable=None)

Bases: PMML44Super.GeneratedsSuper

add_Extension (value)

build (node)
```

```

buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_=", name_='TextIndexNormalization', namespacedef_=",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_=",
        name_='TextIndexNormalization')
exportChildren (outfile, level, namespace_=", name_='TextIndexNormalization', fromsub-
        class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='TextIndexNormalization')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_InlineTable ()
get_TableLocator ()
get_inField ()
get_isCaseSensitive ()
get_maxLevenshteinDistance ()
get_outField ()
get_recursive ()
get_regexField ()
get_tokenize ()
get_wordSeparatorCharacterRE ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_InlineTable (InlineTable)
set_TableLocator (TableLocator)
set_inField (inField)
set_isCaseSensitive (isCaseSensitive)
set_maxLevenshteinDistance (maxLevenshteinDistance)
set_outField (outField)
set_recursive (recursive)
set_regexField (regexField)
set_tokenize (tokenize)
set_wordSeparatorCharacterRE (wordSeparatorCharacterRE)

```

```
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TextIndexNormalization', mapping_=None)
class PMML44Super.TextModel (modelName=None, functionName=None, algorithmName=None,
                             numberOfTerms=None, numberOfDocuments=None, isScorable=True, MiningSchema=None, Output=None, ModelStats=None, ModelExplanation=None, Targets=None, LocalTransformations=None, TextDictionary=None, TextCorpus=None, DocumentTermMatrix=None, TextModelNormalization=None, TextModelSimiliarity=None, ModelVerification=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TextModel', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TextModel')
    exportChildren (outfile, level, namespace_="", name_='TextModel', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='TextModel')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_DocumentTermMatrix ()
    get_Extension ()
    get_LocalTransformations ()
    get_MiningSchema ()
    get_ModelExplanation ()
    get_ModelStats ()
    get_ModelVerification ()
    get_Output ()
    get_Targets ()
    get_TextCorpus ()
    get_TextDictionary ()
    get_TextModelNormalization ()
    get_TextModelSimiliarity ()
    get_algorithmName ()
    get_functionName ()
```

```

    get_isScorable ()
    get_modelName ()
    get_numberOfDocuments ()
    get_numberOfTerms ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_DocumentTermMatrix (DocumentTermMatrix)
    set_Extension (Extension)
    set_LocalTransformations (LocalTransformations)
    set_MiningSchema (MiningSchema)
    set_ModelExplanation (ModelExplanation)
    set_ModelStats (ModelStats)
    set_ModelVerification (ModelVerification)
    set_Output (Output)
    set_Targets (Targets)
    set_TextCorpus (TextCorpus)
    set_TextDictionary (TextDictionary)
    set_TextModelNormalization (TextModelNormalization)
    set_TextModelSimiliarity (TextModelSimiliarity)
    set_algorithmName (algorithmName)
    set_functionName (functionName)
    set_isScorable (isScorable)
    set_modelName (modelName)
    set_numberOfDocuments (numberOfDocuments)
    set_numberOfTerms (numberOfTerms)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TextModel', mapping_=None)
    validate_MINING_FUNCTION (value)
class PMML44Super.TextModelNormalization (localTermWeights='termFrequency',      global-
                                         TermWeights='inverseDocumentFrequency', doc-
                                         umentNormalization='none', Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)

```

```
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TextModelNormalization', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="",
                  name_='TextModelNormalization')
exportChildren (outfile, level, namespace_="", name_='TextModelNormalization', fromsub-
                 class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='TextModelNormalization')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_documentNormalization ()
get_globalTermWeights ()
get_localTermWeights ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_documentNormalization (documentNormalization)
set_globalTermWeights (globalTermWeights)
set_localTermWeights (localTermWeights)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TextModelNormalization', mapping_=None)
class PMML44Super.TextModelSimiliarity (similarityType=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TextModelSimiliarity', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="",
                      name_='TextModelSimiliarity')
    exportChildren (outfile, level, namespace_="", name_='TextModelSimiliarity', fromsub-
                     class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='TextModelSimiliarity')
    exportLiteralAttributes (outfile, level, already_processed, name_)
```

```

exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_similarityType ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_similarityType (similarityType)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TextModelSimiliarity', mapping_=None)
class PMML44Super.Theta (i=None, j=None, theta=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Theta', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Theta')
    exportChildren (outfile, level, namespace_="", name_='Theta', fromsubclass_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='Theta')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_i ()
    get_j ()
    get_theta ()
    hasContent_ ()
    set_i (i)
    set_j (j)
    set_theta (theta)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='Theta', mapping_=None)
    validate_INT_NUMBER (value)

```

```
class PMML44Super.ThetaRecursionState (FinalNoise=None, FinalPredictedNoise=None, Final-  
                                         Theta=None, FinalNu=None)  
    Bases: PMML44Super.GeneratedsSuper  
  
    build (node)  
  
    buildAttributes (node, attrs, already_processed)  
  
    buildChildren (child_, node, nodeName_, fromsubclass_=False)  
  
    export (outfile, level, namespace_="", name_='ThetaRecursionState', namespacedef_="",  
           pretty_print=True)  
  
    exportAttributes (outfile, level, already_processed, namespace_="",  
                      name_='ThetaRecursionState')  
  
    exportChildren (outfile, level, namespace_="", name_='ThetaRecursionState', fromsub-  
                    class_=False, pretty_print=True)  
  
    exportLiteral (outfile, level, name_='ThetaRecursionState')  
  
    exportLiteralAttributes (outfile, level, already_processed, name_)  
  
    exportLiteralChildren (outfile, level, name_)  
  
    static factory (*args_, **kwargs_)  
  
    get_FinalNoise ()  
  
    get_FinalNu ()  
  
    get_FinalPredictedNoise ()  
  
    get_FinalTheta ()  
  
    hasContent_ ()  
  
    set_FinalNoise (FinalNoise)  
  
    set_FinalNu (FinalNu)  
  
    set_FinalPredictedNoise (FinalPredictedNoise)  
  
    set_FinalTheta (FinalTheta)  
  
    subclass = None  
  
    superclass = None  
  
    to_etree (parent_element=None, name_='ThetaRecursionState', mapping_=None)  
  
class PMML44Super.Time (min=None, max=None, mean=None, standardDeviation=None, Exten-  
                        sion=None)  
    Bases: PMML44Super.GeneratedsSuper  
  
    add_Extension (value)  
  
    build (node)  
  
    buildAttributes (node, attrs, already_processed)  
  
    buildChildren (child_, node, nodeName_, fromsubclass_=False)  
  
    export (outfile, level, namespace_="", name_='Time', namespacedef_="", pretty_print=True)  
  
    exportAttributes (outfile, level, already_processed, namespace_="", name_='Time')  
  
    exportChildren (outfile, level, namespace_="", name_='Time', fromsubclass_=False,  
                    pretty_print=True)  
  
    exportLiteral (outfile, level, name_='Time')
```



```

exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_max ()
get_mean ()
get_min ()
get_standardDeviation ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_max (max)
set_mean (mean)
set_min (min)
set_standardDeviation (standardDeviation)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Time', mapping_=None)
validate_NUMBER (value)
class PMML44Super.TimeAnchor (type_=None, offset=None, stepsize=None, displayName=None,
                                TimeCycle=None, TimeException=None)
    Bases: PMML44Super.GeneratedSuper
add_TimeCycle (value)
add_TimeException (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TimeAnchor', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='TimeAnchor')
exportChildren (outfile, level, namespace_="", name_='TimeAnchor', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='TimeAnchor')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_TimeCycle ()

```

```
get_TimeException()
get_displayName()
get_offset()
get_stepsize()
get_type()
hasContent_()
insert_TimeCycle_at(index, value)
insert_TimeException_at(index, value)
replace_TimeCycle_at(index, value)
replace_TimeException_at(index, value)
set_TimeCycle(TimeCycle)
set_TimeException(TimeException)
set_displayName(displayName)
set_offset(offset)
set_stepsize(stepsize)
set_type(type_)
subclass = None
superclass = None
to_etree(parent_element=None, name_='TimeAnchor', mapping_=None)
validate_INT_NUMBER(value)
validate_TIME_ANCHOR(value)

class PMML44Super.TimeCycle(length=None, type_=None, displayName=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_=' ', name_='TimeCycle', namespacedef_=' ', pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_=' ', name_='TimeCycle')
    exportChildren(outfile, level, namespace_=' ', name_='TimeCycle', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='TimeCycle')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    get_displayName()
    get_length()
```

```

    get_type()
    hasContent_()
    set_Array(Array)
    set_displayName(displayName)
    set_length(length)
    set_type(type_)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='TimeCycle', mapping_=None)
    validate_INT_NUMBER(value)
    validate_VALID_TIME_SPEC(value)
class PMML44Super.TimeException(type_=None, count=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='TimeException', namespacesdef_="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='TimeException')
    exportChildren(outfile, level, namespace_="", name_='TimeException', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='TimeException')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    get_count()
    get_type()
    hasContent_()
    set_Array(Array)
    set_count(count)
    set_type(type_)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='TimeException', mapping_=None)
    validate_INT_NUMBER(value)
    validate_TIME_EXCEPTION_TYPE(value)

```

```
class PMML44Super.TimeSeries (usage='original', start_time=None, end_time=None, inter-  
                                polationMethod='none', field=None, TimeAnchor=None,  
                                TimeValue=None)  
    Bases: PMML44Super.GeneratedSuper  
  
    add_TimeValue (value)  
  
    build (node)  
  
    buildAttributes (node, attrs, already_processed)  
  
    buildChildren (child_, node, nodeName_, fromsubclass_=False)  
  
    export (outfile, level, namespace_="", name_='TimeSeries', namespacedef_="", pretty_print=True)  
  
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TimeSeries')  
  
    exportChildren (outfile, level, namespace_="", name_='TimeSeries', fromsubclass_=False,  
                    pretty_print=True)  
  
    exportLiteral (outfile, level, name_='TimeSeries')  
  
    exportLiteralAttributes (outfile, level, already_processed, name_)  
  
    exportLiteralChildren (outfile, level, name_)  
  
    static factory (*args_, **kwargs_)  
  
    get_TimeAnchor ()  
  
    get_TimeValue ()  
  
    get_endTime ()  
  
    get_field ()  
  
    get_interpolationMethod ()  
  
    get_startTime ()  
  
    get_usage ()  
  
    hasContent_ ()  
  
    insert_TimeValue_at (index, value)  
  
    replace_TimeValue_at (index, value)  
  
    set_TimeAnchor (TimeAnchor)  
  
    set_TimeValue (TimeValue)  
  
    set_endTime (endTime)  
  
    set_field (field)  
  
    set_interpolationMethod (interpolationMethod)  
  
    set_startTime (startTime)  
  
    set_usage (usage)  
  
    subclass = None  
  
    superclass = None  
  
    to_etree (parent_element=None, name_='TimeSeries', mapping_=None)  
  
    validate_FIELD_NAME (value)  
  
    validate_INTERPOLATION_METHOD (value)
```

```

validate_REAL_NUMBER (value)
validate_TIMESERIES_USAGE (value)
class PMML44Super.TimeSeriesModel (modelName=None, functionName=None, algorithm-
                                     Name=None, bestFit=None, isScorable=True, Min-
                                     ingSchema=None, Output=None, ModelStats=None,
                                     ModelExplanation=None, LocalTransformations=None,
                                     TimeSeries=None, SpectralAnalysis=None, ARIMA=None,
                                     ExponentialSmoothing=None, SeasonalTrendDecompo-
                                     sition=None, StateSpaceModel=None, GARCH=None,
                                     ModelVerification=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
add_TimeSeries (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace="", name_='TimeSeriesModel', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="", name_='TimeSeriesModel')
exportChildren (outfile, level, namespace="", name_='TimeSeriesModel', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='TimeSeriesModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_ARIMA ()
get_ExponentialSmoothing ()
get_Extension ()
get_GARCH ()
get_LocalTransformations ()
get_MiningSchema ()
get_ModelExplanation ()
get_ModelStats ()
get_ModelVerification ()
get_Output ()
get_SeasonalTrendDecomposition ()
get_SpectralAnalysis ()
get_StateSpaceModel ()
get_TimeSeries ()
get_algorithmName ()

```

```
get_bestFit ()
get_functionName ()
get_isScorable ()
get_modelName ()
hasContent_ ()
insert_Extension_at (index, value)
insert_TimeSeries_at (index, value)
replace_Extension_at (index, value)
replace_TimeSeries_at (index, value)
set_ARIMA (ARIMA)
set_ExponentialSmoothing (ExponentialSmoothing)
set_Extension (Extension)
set_GARCH (GARCH)
set_LocalTransformations (LocalTransformations)
set_MiningSchema (MiningSchema)
set_ModelExplanation (ModelExplanation)
set_ModelStats (ModelStats)
set_ModelVerification (ModelVerification)
set_Output (Output)
set_SeasonalTrendDecomposition (SeasonalTrendDecomposition)
set_SpectralAnalysis (SpectralAnalysis)
set_StateSpaceModel (StateSpaceModel)
set_TimeSeries (TimeSeries)
set_algorithmName (algorithmName)
set_bestFit (bestFit)
set_functionName (functionName)
set_isScorable (isScorable)
set_modelName (modelName)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TimeSeriesModel', mapping_=None)
validate_MINING_FUNCTION (value)
validate_TIMESERIES_ALGORITHM (value)
class PMML44Super.TimeValue (index=None, time=None, value=None, standardError=None, Times-
                             tamp=None)
    Bases: PMML44Super.GeneratedsSuper
    build (node)
```

```

buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TimeValue', namespacesdef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='TimeValue')
exportChildren (outfile, level, namespace_="", name_='TimeValue', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='TimeValue')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Stamp ()
get_index ()
get_standardError ()
get_time ()
get_value ()
hasContent ()
set_Stamp (Timestamp)
set_index (index)
set_standardError (standardError)
set_time (time)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TimeValue', mapping_=None)
validate_INT_NUMBER (value)
validate_NUMBER (value)
validate_REAL_NUMBER (value)
class PMML44Super.Timestamp (Extension=None, valueOf_=None, mixedclass_=None, content_=None)
    Bases: PMML44Super.GeneratedSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Timestamp', namespacesdef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Timestamp')
exportChildren (outfile, level, namespace_="", name_='Timestamp', fromsubclass_=False, pretty_print=True)

```

```
exportLiteral (outfile, level, name_='Timestamp')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_valueOf_ ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_valueOf_ (valueOf_)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Timestamp', mapping_=None)
class PMML44Super.TrainingInstances (isTransformed=False, recordCount=None, field-
                                     Count=None, Extension=None, InstanceFields=None,
                                     TableLocator=None, InlineTable=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TrainingInstances', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='TrainingInstances')
exportChildren (outfile, level, namespace_="", name_='TrainingInstances', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='TrainingInstances')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_InlineTable ()
get_InstanceFields ()
get_TableLocator ()
get_fieldCount ()
get_isTransformed ()
get_recordCount ()
```



```

hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_InlineTable(InlineTable)
set_InstanceFields(InstanceFields)
set_TableLocator(TableLocator)
set_fieldCount(fieldCount)
set_isTransformed(isTransformed)
set_recordCount(recordCount)
subclass = None
superclass = None
to_etree(parent_element=None, name_='TrainingInstances', mapping_=None)
validate_INT_NUMBER(value)
class PMML44Super.TrainingParameters(architectureName=None, dataset=None, frame-
                                     work=None, Extension=None, Losses=None, Met-
                                     rics=None, Optimizers=None)
Bases: PMML44Super.GeneratedSuper
add_Extension(value)
build(node)
buildAttributes(node, attrs, already_processed)
buildChildren(child_, node, nodeName_, fromsubclass_=False)
export(outfile, level, namespace_="", name_='TrainingParameters', namespacedef_="",
        pretty_print=True)
exportAttributes(outfile, level, already_processed, namespace_="",
                 name_='TrainingParameters')
exportChildren(outfile, level, namespace_="", name_='TrainingParameters', fromsubclass_=False,
               pretty_print=True)
exportLiteral(outfile, level, name_='TrainingParameters')
exportLiteralAttributes(outfile, level, already_processed, name_)
exportLiteralChildren(outfile, level, name_)
static factory(*args_, **kwargs_)
get_Extension()
get_Losses()
get_Metrics()
get_Optimizers()
get_architectureName()
get_dataset()
get_framework()

```

```
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_Losses(Losses)
set_Metrics(Metrics)
set_Optimizers(Optimizers)
set_architectureName(architectureName)
set_dataset(dataset)
set_framework(framework)
subclass = None
superclass = None
to_etree(parent_element=None, name_='TrainingParameters', mapping_=None)
class PMML44Super.TransferFunctionValues(Array=None)
    Bases: PMML44Super.GeneratedsSuper
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='TransferFunctionValues', namespacedef="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="",
                     name_='TransferFunctionValues')
    exportChildren(outfile, level, namespace_="", name_='TransferFunctionValues', fromsub-
                   class_=False, pretty_print=True)
    exportLiteral(outfile, level, name_='TransferFunctionValues')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Array()
    hasContent_()
    set_Array(Array)
    subclass = None
    superclass = None
    to_etree(parent_element=None, name_='TransferFunctionValues', mapping_=None)
class PMML44Super.TransformationDictionary(Extension=None, DefineFunction=None, De-
                                           rivedField=None)
    Bases: PMML44Super.GeneratedsSuper
    add_DefineFunction(value)
    add_DerivedField(value)
```

```

add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace="", name_='TransformationDictionary', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="",
        name_='TransformationDictionary')
exportChildren (outfile, level, namespace="", name_='TransformationDictionary', fromsub-
        class_=False, pretty_print=True)
exportLiteral (outfile, level, name_='TransformationDictionary')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_DefineFunction ()
get_DerivedField ()
get_Extension ()
hasContent ()
insert_DefineFunction_at (index, value)
insert_DerivedField_at (index, value)
insert_Extension_at (index, value)
replace_DefineFunction_at (index, value)
replace_DerivedField_at (index, value)
replace_Extension_at (index, value)
set_DefineFunction (DefineFunction)
set_DerivedField (DerivedField)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TransformationDictionary', mapping_=None)
class PMML44Super.TransitionMatrix (Extension=None, Matrix=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='TransitionMatrix', namespacedef="",
            pretty_print=True)

```

```
exportAttributes (outfile, level, already_processed, namespace_="", name_='TransitionMatrix')
exportChildren (outfile, level, namespace_="", name_='TransitionMatrix', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='TransitionMatrix')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_Matrix()
hasContent()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_Matrix (Matrix)
subclass = None
superclass = None
to_etree (parent_element=None, name_='TransitionMatrix', mapping_=None)
class PMML44Super.TreeModel (modelName=None, functionName=None, algorithmName=None,
                              missingValueStrategy='none', missingValuePenalty='1.0',
                              noTrueChildStrategy='returnNullPrediction', splitCharacteristic='multiSplit',
                              isScorable=True, MiningSchema=None, Output=None, ModelStats=None,
                              ModelExplanation=None, Targets=None, LocalTransformations=None,
                              Node=None, ModelVerification=None, Extension=None)
Bases: PMML44Super.GeneratedSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='TreeModel', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='TreeModel')
exportChildren (outfile, level, namespace_="", name_='TreeModel', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='TreeModel')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_LocalTransformations()
```

```
get_MiningSchema()
get_ModelExplanation()
get_ModelStats()
get_ModelVerification()
get_Node()
get_Output()
get_Targets()
get_algorithmName()
get_functionName()
get_isScorable()
get_missingValuePenalty()
get_missingValueStrategy()
get_modelName()
get_noTrueChildStrategy()
get_splitCharacteristic()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_LocalTransformations(LocalTransformations)
set_MiningSchema(MiningSchema)
set_ModelExplanation(ModelExplanation)
set_ModelStats(ModelStats)
set_ModelVerification(ModelVerification)
set_Node(Node)
set_Output(Output)
set_Targets(Targets)
set_algorithmName(algorithmName)
set_functionName(functionName)
set_isScorable(isScorable)
set_missingValuePenalty(missingValuePenalty)
set_missingValueStrategy(missingValueStrategy)
set_modelName(modelName)
set_noTrueChildStrategy(noTrueChildStrategy)
set_splitCharacteristic(splitCharacteristic)
subclass = None
```

```
    superclass = None
    to_etree (parent_element=None, name_='TreeModel', mapping_=None)
    validate_MINING_FUNCTION (value)
    validate_MISSING_VALUE_STRATEGY (value)
    validate_NO_TRUE_CHILD_STRATEGY (value)
    validate_PROB_NUMBER (value)
class PMML44Super.TrendCoefficients (Extension=None, REAL_SparseArray=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='TrendCoefficients', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='TrendCoefficients')
    exportChildren (outfile, level, namespace_="", name_='TrendCoefficients', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='TrendCoefficients')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_REAL_SparseArray ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_REAL_SparseArray (REAL_SparseArray)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='TrendCoefficients', mapping_=None)
class PMML44Super.Trend_ExpoSmooth (trend='additive', gamma=None, phi='1', smoothed-
                                   Value=None, Array=None)
    Bases: PMML44Super.GeneratedSuper
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='Trend_ExpoSmooth', namespacedef_="",
            pretty_print=True)
```

```

exportAttributes (outfile, level, already_processed, namespace="",
                  name_='Trend_ExpoSmooth')
exportChildren (outfile, level, namespace="", name_='Trend_ExpoSmooth', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Trend_ExpoSmooth')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_gamma ()
get_phi ()
get_smoothedValue ()
get_trend ()
hasContent_ ()
set_Array (Array)
set_gamma (gamma)
set_phi (phi)
set_smoothedValue (smoothedValue)
set_trend (trend)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Trend_ExpoSmooth', mapping_=None)
validate_REAL_NUMBER (value)

class PMML44Super.TriangularDistributionForBN (Extension=None, Mean=None,
                                                Lower=None, Upper=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='TriangularDistributionForBN', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="",
                      name_='TriangularDistributionForBN')
    exportChildren (outfile, level, namespace="", name_='TriangularDistributionForBN', fromsub-
                      class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='TriangularDistributionForBN')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)

```

```
static factory(*args_,**kwargs_)
get_Extension()
get_Lower()
get_Mean()
get_Upper()
hasContent_()
insert_Extension_at(index,value)
replace_Extension_at(index,value)
set_Extension(Extension)
set_Lower(Lower)
set_Mean(Mean)
set_Upper(Upper)
subclass = None
superclass = None
to_etree(parent_element=None,name_='TriangularDistributionForBN',mapping_=None)

class PMML44Super.True_(Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node,attrs,already_processed)
    buildChildren(child_,node,nodeName_,fromsubclass_=False)
    export(outfile,level,namespace_=' ',name_='True',namespacedef_=' ',pretty_print=True)
    exportAttributes(outfile,level,already_processed,namespace_=' ',name_='True')
    exportChildren(outfile,level,namespace_=' ',name_='True',fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile,level,name_='True')
    exportLiteralAttributes(outfile,level,already_processed,name_)
    exportLiteralChildren(outfile,level,name_)
    static factory(*args_,**kwargs_)
    get_Extension()
    hasContent_()
    insert_Extension_at(index,value)
    replace_Extension_at(index,value)
    set_Extension(Extension)
    subclass = None
    superclass = None
    to_etree(parent_element=None,name_='True',mapping_=None)
```

```

class PMML44Super.UniformDistribution (lower=None, upper=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='UniformDistribution', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='UniformDistribution')

    exportChildren (outfile, level, namespace_="", name_='UniformDistribution', fromsub-
                    class_=False, pretty_print=True)

    exportLiteral (outfile, level, name_='UniformDistribution')

    exportLiteralAttributes (outfile, level, already_processed, name_)

    exportLiteralChildren (outfile, level, name_)

    static factory (*args_, **kwargs_)

    get_Extension ()

    get_lower ()

    get_upper ()

    hasContent_ ()

    insert_Extension_at (index, value)

    replace_Extension_at (index, value)

    set_Extension (Extension)

    set_lower (lower)

    set_upper (upper)

    subclass = None

    superclass = None

    to_etree (parent_element=None, name_='UniformDistribution', mapping_=None)

    validate_REAL_NUMBER (value)

class PMML44Super.UniformDistributionForBN (Extension=None, Lower=None, Up-
                                           per=None)
    Bases: PMML44Super.GeneratedsSuper

    add_Extension (value)

    build (node)

    buildAttributes (node, attrs, already_processed)

    buildChildren (child_, node, nodeName_, fromsubclass_=False)

    export (outfile, level, namespace_="", name_='UniformDistributionForBN', namespacedef="",
            pretty_print=True)

    exportAttributes (outfile, level, already_processed, namespace_="",
                     name_='UniformDistributionForBN')

```

```
exportChildren (outfile, level, namespace_="", name_='UniformDistributionForBN', fromsubclass_=False, pretty_print=True)
exportLiteral (outfile, level, name_='UniformDistributionForBN')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_Lower ()
get_Upper ()
hasContent _ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_Lower (Lower)
set_Upper (Upper)
subclass = None
superclass = None
to_etree (parent_element=None, name_='UniformDistributionForBN', mapping_=None)
class PMML44Super.UnivariateStats (field=None, weighted='0', Extension=None, Counts=None,
                                   NumericInfo=None, DiscrStats=None, ContStats=None,
                                   Anova=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='UnivariateStats', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='UnivariateStats')
exportChildren (outfile, level, namespace_="", name_='UnivariateStats', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='UnivariateStats')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Anova ()
get_ContStats ()
get_Counts ()
get_DiscrStats ()
```

```

get_Extension()
get_NumericInfo()
get_field()
get_weighted()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Anova(Anova)
set_ContStats(ContStats)
set_Counts(Counts)
set_DiscrStats(DiscrStats)
set_Extension(Extension)
set_NumericInfo(NumericInfo)
set_field(field)
set_weighted(weighted)
subclass = None
superclass = None
to_etree(parent_element=None, name_='UnivariateStats', mapping_=None)
validate_FIELD_NAME(value)
class PMML44Super.Upper(Extension=None, FieldRef=None, Constant=None, NormContinu-
                        ous=None, NormDiscrete=None, Discretize=None, MapValues=None,
                        TextIndex=None, Apply=None, Aggregate=None, Lag=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_="", name_='Upper', namespacedef="", pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_="", name_='Upper')
    exportChildren(outfile, level, namespace_="", name_='Upper', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='Upper')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Aggregate()
    get_Apply()
    get_Constant()

```

```
get_Discretize()
get_Extension()
get_FieldRef()
get_Lag()
get_MapValues()
get_NormContinuous()
get_NormDiscrete()
get_TextIndex()
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Aggregate(Aggregate)
set_Apply(Apply)
set_Constant(Constant)
set_Discretize(Discretize)
set_Extension(Extension)
set_FieldRef(FieldRef)
set_Lag(Lag)
set_MapValues(MapValues)
set_NormContinuous(NormContinuous)
set_NormDiscrete(NormDiscrete)
set_TextIndex(TextIndex)
subclass = None
superclass = None
to_etree(parent_element=None, name_='Upper', mapping_=None)
class PMML44Super.Value(value=None, displayValue=None, property='valid', Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace_=' ', name_='Value', namespacedef_=' ', pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace_=' ', name_='Value')
    exportChildren(outfile, level, namespace_=' ', name_='Value', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral(outfile, level, name_='Value')
    exportLiteralAttributes(outfile, level, already_processed, name_)
```

```

exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_displayValue ()
get_property ()
get_value ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_displayValue (displayValue)
set_property (property)
set_value (value)
subclass = None
superclass = None
to_etree (parent_element=None, name_='Value', mapping_=None)
class PMML44Super.ValueProbability (value=None, probability=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='ValueProbability', namespacedef_="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='ValueProbability')
    exportChildren (outfile, level, namespace_="", name_='ValueProbability', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='ValueProbability')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_probability ()
    get_value ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)

```

```
    set_Extension (Extension)
    set_probability (probability)
    set_value (value)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='ValueProbability', mapping_=None)
    validate_PROB_NUMBER (value)

class PMML44Super.VariableWeight (field=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='VariableWeight', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='VariableWeight')
    exportChildren (outfile, level, namespace_="", name_='VariableWeight', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='VariableWeight')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_field ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_field (field)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='VariableWeight', mapping_=None)
    validate_FIELD_NAME (value)

class PMML44Super.Variance (Extension=None, FieldRef=None, Constant=None, NormContinu-
                             ous=None, NormDiscrete=None, Discretize=None, MapValues=None,
                             TextIndex=None, Apply=None, Aggregate=None, Lag=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
```

```

build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace_="", name_='Variance', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='Variance')
exportChildren (outfile, level, namespace_="", name_='Variance', fromsubclass_=False,
                  pretty_print=True)
exportLiteral (outfile, level, name_='Variance')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Aggregate ()
get_Apply ()
get_Constant ()
get_Discretize ()
get_Extension ()
get_FieldRef ()
get_Lag ()
get_MapValues ()
get_NormContinuous ()
get_NormDiscrete ()
get_TextIndex ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Aggregate (Aggregate)
set_Apply (Apply)
set_Constant (Constant)
set_Discretize (Discretize)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_Lag (Lag)
set_MapValues (MapValues)
set_NormContinuous (NormContinuous)
set_NormDiscrete (NormDiscrete)
set_TextIndex (TextIndex)
subclass = None

```

```
    superclass = None
    to_etree (parent_element=None, name_='Variance', mapping_=None)
class PMML44Super.VarianceCoefficients (Extension=None, PastVariances=None, MACoeffi-
                                         cients=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace="", name_='VarianceCoefficients', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace="",
                     name_='VarianceCoefficients')
    exportChildren (outfile, level, namespace="", name_='VarianceCoefficients', fromsub-
                   class_=False, pretty_print=True)
    exportLiteral (outfile, level, name_='VarianceCoefficients')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_MACoefficients ()
    get_PastVariances ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_MACoefficients (MACoefficients)
    set_PastVariances (PastVariances)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='VarianceCoefficients', mapping_=None)
class PMML44Super.VectorDictionary (numberOfVectors=None, Extension=None, Vector-
                                     Fields=None, VectorInstance=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    add_VectorInstance (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
```



```

export (outfile, level, namespace_="", name_='VectorDictionary', namespacedef_="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='VectorDictionary')
exportChildren (outfile, level, namespace_="", name_='VectorDictionary', fromsubclass_=False,
        pretty_print=True)
exportLiteral (outfile, level, name_='VectorDictionary')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_VectorFields()
get_VectorInstance()
get_numberOfVectors()
hasContent()
insert_Extension_at (index, value)
insert_VectorInstance_at (index, value)
replace_Extension_at (index, value)
replace_VectorInstance_at (index, value)
set_Extension (Extension)
set_VectorFields (VectorFields)
set_VectorInstance (VectorInstance)
set_numberOfVectors (numberOfVectors)
subclass = None
superclass = None
to_etree (parent_element=None, name_='VectorDictionary', mapping_=None)
validate_INT_NUMBER (value)
class PMML44Super.VectorFields (numberOfFields=None, Extension=None, FieldRef=None, Cate-
        goricalPredictor=None)
    Bases: PMML44Super.GeneratedsSuper
    add_CategoricalPredictor (value)
    add_Extension (value)
    add_FieldRef (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='VectorFields', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='VectorFields')

```

```
exportChildren (outfile, level, namespace_="", name_='VectorFields', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='VectorFields')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_CategoricalPredictor ()
get_Extension ()
get_FieldRef ()
get_numberOfFields ()
hasContent_ ()
insert_CategoricalPredictor_at (index, value)
insert_Extension_at (index, value)
insert_FieldRef_at (index, value)
replace_CategoricalPredictor_at (index, value)
replace_Extension_at (index, value)
replace_FieldRef_at (index, value)
set_CategoricalPredictor (CategoricalPredictor)
set_Extension (Extension)
set_FieldRef (FieldRef)
set_numberOfFields (numberOfFields)
subclass = None
superclass = None
to_etree (parent_element=None, name_='VectorFields', mapping_=None)
validate_INT_NUMBER (value)

class PMML44Super.VectorInstance (id=None, Extension=None, REAL_SparseArray=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='VectorInstance', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='VectorInstance')
    exportChildren (outfile, level, namespace_="", name_='VectorInstance', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='VectorInstance')
    exportLiteralAttributes (outfile, level, already_processed, name_)
```

```

exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_Extension ()
get_REAL_SparseArray ()
get_id ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Array (Array)
set_Extension (Extension)
set_REAL_SparseArray (REAL_SparseArray)
set_id (id)
subclass = None
superclass = None
to_etree (parent_element=None, name_='VectorInstance', mapping_=None)
validate_VECTOR_ID (value)

class PMML44Super.VerificationField (field=None, column=None, precision=1e-06,
                                     zeroThreshold=1e-16, Extension=None)
    Bases: PMML44Super.GeneratedSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='VerificationField', namespacedef_="",
           pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='VerificationField')
    exportChildren (outfile, level, namespace_="", name_='VerificationField', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='VerificationField')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    get_column ()
    get_field ()
    get_precision ()
    get_zeroThreshold ()

```

```
hasContent_()
insert_Extension_at(index, value)
replace_Extension_at(index, value)
set_Extension(Extension)
set_column(column)
set_field(field)
set_precision(precision)
set_zeroThreshold(zeroThreshold)
subclass = None
superclass = None
to_etree(parent_element=None, name_='VerificationField', mapping_=None)
validate_FIELD_NAME(value)
class PMML44Super.VerificationFields(Extension=None, VerificationField=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension(value)
    add_VerificationField(value)
    build(node)
    buildAttributes(node, attrs, already_processed)
    buildChildren(child_, node, nodeName_, fromsubclass_=False)
    export(outfile, level, namespace="", name_='VerificationFields', namespacedef="",
           pretty_print=True)
    exportAttributes(outfile, level, already_processed, namespace="", name_='VerificationFields')
    exportChildren(outfile, level, namespace="", name_='VerificationFields', fromsubclass_=False,
                   pretty_print=True)
    exportLiteral(outfile, level, name_='VerificationFields')
    exportLiteralAttributes(outfile, level, already_processed, name_)
    exportLiteralChildren(outfile, level, name_)
    static factory(*args_, **kwargs_)
    get_Extension()
    get_VerificationField()
    hasContent_()
    insert_Extension_at(index, value)
    insert_VerificationField_at(index, value)
    replace_Extension_at(index, value)
    replace_VerificationField_at(index, value)
    set_Extension(Extension)
    set_VerificationField(VerificationField)
```

```

    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='VerificationFields', mapping_=None)
class PMML44Super.XCoordinates (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='XCoordinates', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='XCoordinates')
    exportChildren (outfile, level, namespace_="", name_='XCoordinates', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='XCoordinates')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Array ()
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Array (Array)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='XCoordinates', mapping_=None)
class PMML44Super.YCoordinates (Extension=None, Array=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='YCoordinates', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='YCoordinates')
    exportChildren (outfile, level, namespace_="", name_='YCoordinates', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='YCoordinates')

```

```
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Array ()
get_Extension ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Array (Array)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='YCoordinates', mapping_=None)
class PMML44Super.binarySimilarity (c00_parameter=None, c01_parameter=None,
                                     c10_parameter=None, c11_parameter=None,
                                     d00_parameter=None, d01_parameter=None,
                                     d10_parameter=None, d11_parameter=None, Extension=None)
Bases: PMML44Super.GeneratedsSuper
add_Extension (value)
build (node)
buildAttributes (node, attrs, already_processed)
buildChildren (child_, node, nodeName_, fromsubclass_=False)
export (outfile, level, namespace="", name_='binarySimilarity', namespacedef="",
        pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace="", name_='binarySimilarity')
exportChildren (outfile, level, namespace="", name_='binarySimilarity', fromsubclass_=False,
                pretty_print=True)
exportLiteral (outfile, level, name_='binarySimilarity')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
get_c00_parameter ()
get_c01_parameter ()
get_c10_parameter ()
get_c11_parameter ()
get_d00_parameter ()
get_d01_parameter ()
```

```

    get_d10_parameter ()
    get_d11_parameter ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    set_c00_parameter (c00_parameter)
    set_c01_parameter (c01_parameter)
    set_c10_parameter (c10_parameter)
    set_c11_parameter (c11_parameter)
    set_d00_parameter (d00_parameter)
    set_d01_parameter (d01_parameter)
    set_d10_parameter (d10_parameter)
    set_d11_parameter (d11_parameter)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='binarySimilarity', mapping_=None)
    validate_NUMBER (value)
class PMML44Super.chebychev (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='chebychev', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='chebychev')
    exportChildren (outfile, level, namespace_="", name_='chebychev', fromsubclass_=False,
        pretty_print=True)
    exportLiteral (outfile, level, name_='chebychev')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)

```

```
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='chebychev', mapping_=None)
class PMML44Super.cityBlock (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='cityBlock', namespacesdef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='cityBlock')
    exportChildren (outfile, level, namespace_="", name_='cityBlock', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='cityBlock')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='cityBlock', mapping_=None)
class PMML44Super.euclidean (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='euclidean', namespacesdef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='euclidean')
    exportChildren (outfile, level, namespace_="", name_='euclidean', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='euclidean')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
```



```

    static factory (*args_, **kwargs_)
    get_Extension()
    hasContent_()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='euclidean', mapping_=None)

class PMML44Super.jaccard (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='jaccard', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='jaccard')
    exportChildren (outfile, level, namespace_="", name_='jaccard', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='jaccard')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension()
    hasContent_()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='jaccard', mapping_=None)

class PMML44Super.minkowski (p_parameter=None, Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)

```

```
export (outfile, level, namespace_="", name_='minkowski', namespacedef_="", pretty_print=True)
exportAttributes (outfile, level, already_processed, namespace_="", name_='minkowski')
exportChildren (outfile, level, namespace_="", name_='minkowski', fromsubclass_=False,
                 pretty_print=True)
exportLiteral (outfile, level, name_='minkowski')
exportLiteralAttributes (outfile, level, already_processed, name_)
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension()
get_p_parameter()
hasContent_()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
set_p_parameter (p_parameter)
subclass = None
superclass = None
to_etree (parent_element=None, name_='minkowski', mapping_=None)
validate_NUMBER (value)
class PMML44Super.row (anytypeobjs_=None)
    Bases: PMML44Super.GeneratedsSuper
    add_anytypeobjs_ (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='row', namespacedef_="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='row')
    exportChildren (outfile, level, namespace_="", name_='row', fromsubclass_=False,
                     pretty_print=True)
    exportLiteral (outfile, level, name_='row')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_anytypeobjs_()
    hasContent_()
    insert_anytypeobjs_ (index, value)
    set_anytypeobjs_ (anytypeobjs_)
```

```

    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='row', mapping_=None)
class PMML44Super.simpleMatching (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='simpleMatching', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='simpleMatching')
    exportChildren (outfile, level, namespace_="", name_='simpleMatching', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='simpleMatching')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='simpleMatching', mapping_=None)
class PMML44Super.squaredEuclidean (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='squaredEuclidean', namespacedef="",
            pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='squaredEuclidean')
    exportChildren (outfile, level, namespace_="", name_='squaredEuclidean', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='squaredEuclidean')
    exportLiteralAttributes (outfile, level, already_processed, name_)

```

```
exportLiteralChildren (outfile, level, name_)
static factory (*args_, **kwargs_)
get_Extension ()
hasContent_ ()
insert_Extension_at (index, value)
replace_Extension_at (index, value)
set_Extension (Extension)
subclass = None
superclass = None
to_etree (parent_element=None, name_='squaredEuclidean', mapping_=None)
class PMML44Super.tanimoto (Extension=None)
    Bases: PMML44Super.GeneratedsSuper
    add_Extension (value)
    build (node)
    buildAttributes (node, attrs, already_processed)
    buildChildren (child_, node, nodeName_, fromsubclass_=False)
    export (outfile, level, namespace_="", name_='tanimoto', namespacesdef="", pretty_print=True)
    exportAttributes (outfile, level, already_processed, namespace_="", name_='tanimoto')
    exportChildren (outfile, level, namespace_="", name_='tanimoto', fromsubclass_=False,
                    pretty_print=True)
    exportLiteral (outfile, level, name_='tanimoto')
    exportLiteralAttributes (outfile, level, already_processed, name_)
    exportLiteralChildren (outfile, level, name_)
    static factory (*args_, **kwargs_)
    get_Extension ()
    hasContent_ ()
    insert_Extension_at (index, value)
    replace_Extension_at (index, value)
    set_Extension (Extension)
    subclass = None
    superclass = None
    to_etree (parent_element=None, name_='tanimoto', mapping_=None)
```

CHAPTER 3

arima module

CHAPTER 4

exponential_smoothing module

CHAPTER 5

keras_model_to_pmml module

CHAPTER 6

`lgb_to_pmml` module

CHAPTER 7

metadata module

CHAPTER 8

pre_process module

CHAPTER 9

skl_to_pmml module

CHAPTER 10

xgboost_to_pmml module

m

metadata, [313](#)

p

PMML44Super, [3](#)

A

- AbsoluteExponentialKernel (class in PMML44Super), 7
- Adadelta (class in PMML44Super), 8
- Adagrad (class in PMML44Super), 9
- Adam (class in PMML44Super), 9
- Adamax (class in PMML44Super), 10
- add_Adadelta() (PMML44Super.Optimizers method), 176
- add_Adagrad() (PMML44Super.Optimizers method), 176
- add_Adam() (PMML44Super.Optimizers method), 176
- add_Adamax() (PMML44Super.Optimizers method), 176
- add_Aggregate() (PMML44Super.Apply method), 20
- add_Annotation() (PMML44Super.Header method), 110
- add_AnomalyDetectionModel() (PMML44Super.PMML method), 185
- add_AnovaRow() (PMML44Super.Anova method), 16
- add_anytypeobjs_() (PMML44Super.Extension method), 93
- add_anytypeobjs_() (PMML44Super.row method), 302
- add_Apply() (PMML44Super.Apply method), 20
- add_Array() (PMML44Super.DiscrStats method), 84
- add_Array() (PMML44Super.Matrix method), 139
- add_Array() (PMML44Super.PartitionFieldStats method), 196
- add_AssociationModel() (PMML44Super.PMML method), 185
- add_AssociationRule() (PMML44Super.AssociationModel method), 23
- add_Attribute() (PMML44Super.Characteristic method), 43
- add_BaselineCell() (PMML44Super.BaseCumHazardTables method), 28
- add_BaselineCell() (PMML44Super.BaselineStratum method), 32
- add_BaselineModel() (PMML44Super.PMML method), 185
- add_BaselineStratum() (PMML44Super.BaseCumHazardTables method), 28
- add_BayesianNetworkModel() (PMML44Super.PMML method), 185
- add_BayesInput() (PMML44Super.BayesInputs method), 33
- add_BlockIndicator() (PMML44Super.Lag method), 121
- add_CategoricalPredictor() (PMML44Super.RegressionTable method), 214
- add_CategoricalPredictor() (PMML44Super.VectorFields method), 293
- add_Category() (PMML44Super.Categories method), 41
- add_Characteristic() (PMML44Super.Characteristics method), 44
- add_ChildParent() (PMML44Super.Taxonomy method), 256
- add_Cluster() (PMML44Super.ClusteringModel method), 48
- add_ClusteringField() (PMML44Super.ClusteringModel method), 48
- add_ClusteringModel() (PMML44Super.PMML method), 185
- add_ClusteringModelQuality() (PMML44Super.ModelExplanation method), 149
- add_Coefficient() (PMML44Super.Coefficients method), 51
- add_CompoundPredicate() (PMML44Super.CompoundPredicate method),

56
 add_CompoundRule() (PMML44Super.CompoundRule method), 57
 add_CompoundRule() (PMML44Super.RuleSet method), 219
 add_Con() (PMML44Super.Neuron method), 166
 add_Constant() (PMML44Super.Apply method), 20
 add_ContinuousConditionalProbability() (PMML44Super.ContinuousNode method), 66
 add_ContinuousDistribution() (PMML44Super.ContinuousConditionalProbability method), 64
 add_ContinuousDistribution() (PMML44Super.ContinuousNode method), 66
 add_ContinuousNode() (PMML44Super.BayesianNetworkNodes method), 37
 add_DataField() (PMML44Super.DataDictionary method), 72
 add_Decision() (PMML44Super.Decisions method), 77
 add_DecisionTree() (PMML44Super.MiningModel method), 146
 add_DeepNetwork() (PMML44Super.PMML method), 185
 add_DefineFunction() (PMML44Super.TransformationDictionary method), 278
 add_Delimiter() (PMML44Super.Sequence method), 231
 add_DerivedField() (PMML44Super.ContinuousNode method), 66
 add_DerivedField() (PMML44Super.DiscreteNode method), 86
 add_DerivedField() (PMML44Super.LocalTransformations method), 133
 add_DerivedField() (PMML44Super.TransformationDictionary method), 278
 add_DiscreteConditionalProbability() (PMML44Super.DiscreteNode method), 86
 add_DiscreteNode() (PMML44Super.BayesianNetworkNodes method), 37
 add_Discretize() (PMML44Super.Apply method), 20
 add_DiscretizeBin() (PMML44Super.Discretize method), 87
 add_DynamicRegressor() (PMML44Super.ARIMA method), 5
 add_DynamicRegressor() (PMML44Super.StateSpaceModel method), 243
 add_Extension() (PMML44Super.AbsoluteExponentialKernel method), 7
 add_Extension() (PMML44Super.Adadelta method), 8
 add_Extension() (PMML44Super.Adagrad method), 9
 add_Extension() (PMML44Super.Adam method), 9
 add_Extension() (PMML44Super.Adamax method), 10
 add_Extension() (PMML44Super.Aggregate method), 11
 add_Extension() (PMML44Super.Annotation method), 13
 add_Extension() (PMML44Super.AnomalyDetectionModel method), 14
 add_Extension() (PMML44Super.Anova method), 16
 add_Extension() (PMML44Super.AnovaRow method), 17
 add_Extension() (PMML44Super.AntecedentSequence method), 18
 add_Extension() (PMML44Super.AnyDistribution method), 19
 add_Extension() (PMML44Super.Application method), 19
 add_Extension() (PMML44Super.Apply method), 20
 add_Extension() (PMML44Super.AR method), 3
 add_Extension() (PMML44Super.ARDSquaredExponentialKernel method), 4
 add_Extension() (PMML44Super.ARIMA method), 5
 add_Extension() (PMML44Super.ARMAPart method), 6
 add_Extension() (PMML44Super.AssociationModel method), 23
 add_Extension() (PMML44Super.AssociationRule method), 25
 add_Extension() (PMML44Super.Attribute method), 26
 add_Extension() (PMML44Super.BaseCumHazardTables method), 28
 add_Extension() (PMML44Super.BaselineCell method), 29
 add_Extension() (PMML44Super.BaselineModel method), 30
 add_Extension() (PMML44Super.BaselineStratum method), 32
 add_Extension() (PMML44Super.BayesianNetworkModel method), 35
 add_Extension() (PMML44Super.BayesianNetworkNodes

`method`), 37
`add_Extension()` (*PMML44Super.BayesInput* `method`), 33
`add_Extension()` (*PMML44Super.BayesInputs* `method`), 33
`add_Extension()` (*PMML44Super.BayesOutput* `method`), 34
`add_Extension()` (*PMML44Super.binarySimilarity* `method`), 298
`add_Extension()` (*PMML44Super.BlockIndicator* `method`), 37
`add_Extension()` (*PMML44Super.BoundaryValueMeans* `method`), 38
`add_Extension()` (*PMML44Super.BoundaryValues* `method`), 39
`add_Extension()` (*PMML44Super.CategoricalPredictor* `method`), 40
`add_Extension()` (*PMML44Super.Categories* `method`), 41
`add_Extension()` (*PMML44Super.Category* `method`), 42
`add_Extension()` (*PMML44Super.Characteristic* `method`), 43
`add_Extension()` (*PMML44Super.Characteristics* `method`), 44
`add_Extension()` (*PMML44Super.chebychev* `method`), 299
`add_Extension()` (*PMML44Super.ChildParent* `method`), 44
`add_Extension()` (*PMML44Super.cityBlock* `method`), 300
`add_Extension()` (*PMML44Super.ClassLabels* `method`), 45
`add_Extension()` (*PMML44Super.Cluster* `method`), 46
`add_Extension()` (*PMML44Super.ClusteringField* `method`), 47
`add_Extension()` (*PMML44Super.ClusteringModel* `method`), 48
`add_Extension()` (*PMML44Super.ClusteringModelQuality* `method`), 50
`add_Extension()` (*PMML44Super.Coefficient* `method`), 51
`add_Extension()` (*PMML44Super.Coefficients* `method`), 51
`add_Extension()` (*PMML44Super.ComparisonMeasure* `method`), 52
`add_Extension()` (*PMML44Super.Comparisons* `method`), 54
`add_Extension()` (*PMML44Super.ComplexPartialScore* `method`), 55
`add_Extension()` (*PMML44Super.CompoundPredicate* `method`), 56
`add_Extension()` (*PMML44Super.CompoundRule* `method`), 57
`add_Extension()` (*PMML44Super.Con* `method`), 58
`add_Extension()` (*PMML44Super.ConfusionMatrix* `method`), 59
`add_Extension()` (*PMML44Super.ConsequentSequence* `method`), 60
`add_Extension()` (*PMML44Super.Constraints* `method`), 61
`add_Extension()` (*PMML44Super.ContinuousConditionalProbability* `method`), 64
`add_Extension()` (*PMML44Super.ContinuousDistribution* `method`), 65
`add_Extension()` (*PMML44Super.ContinuousNode* `method`), 66
`add_Extension()` (*PMML44Super.ContStats* `method`), 63
`add_Extension()` (*PMML44Super.CorrelationFields* `method`), 67
`add_Extension()` (*PMML44Super.CorrelationMethods* `method`), 68
`add_Extension()` (*PMML44Super.Correlations* `method`), 69
`add_Extension()` (*PMML44Super.CorrelationValues* `method`), 68
`add_Extension()` (*PMML44Super.COUNT_TABLE_TYPE* `method`), 39
`add_Extension()` (*PMML44Super.Counts* `method`), 70
`add_Extension()` (*PMML44Super.Covariances* `method`), 71
`add_Extension()` (*PMML44Super.CovariateList* `method`), 71
`add_Extension()` (*PMML44Super.DataDictionary* `method`), 72
`add_Extension()` (*PMML44Super.DataField* `method`), 73
`add_Extension()` (*PMML44Super.Decision* `method`), 74
`add_Extension()` (*PMML44Super.Decisions* `method`), 77
`add_Extension()` (*PMML44Super.DecisionTree* `method`), 75
`add_Extension()` (*PMML44Super.DeepNetwork* `method`), 78
`add_Extension()` (*PMML44Super.DefineFunction* `method`), 79
`add_Extension()` (*PMML44Super.Delimiter* `method`), 81
`add_Extension()` (*PMML44Super.Denominator* `method`), 82
`add_Extension()` (*PMML44Super.DerivedField* `method`), 82
`add_Extension()` (*PMML44Super.DiscreteConditionalProbability* `method`), 85

<code>add_Extension()</code> (<i>PMML44Super.DiscreteNode method</i>), 86	<code>add_Extension()</code> (<i>PMML44Super.Discretize method</i>), 87	<code>add_Extension()</code> (<i>PMML44Super.DiscretizeBin method</i>), 88	<code>add_Extension()</code> (<i>PMML44Super.DiscrStats method</i>), 84	<code>add_Extension()</code> (<i>PMML44Super.DocumentTermMatrix method</i>), 89	<code>add_Extension()</code> (<i>PMML44Super.DynamicRegressor method</i>), 90	<code>add_Extension()</code> (<i>PMML44Super.euclidean method</i>), 300	<code>add_Extension()</code> (<i>PMML44Super.EventValues method</i>), 91	<code>add_Extension()</code> (<i>PMML44Super.FactorList method</i>), 93	<code>add_Extension()</code> (<i>PMML44Super.False_ method</i>), 94	<code>add_Extension()</code> (<i>PMML44Super.FieldColumnPair method</i>), 95	<code>add_Extension()</code> (<i>PMML44Super.FieldRef method</i>), 95	<code>add_Extension()</code> (<i>PMML44Super.FieldValue method</i>), 96	<code>add_Extension()</code> (<i>PMML44Super.FieldValueCount method</i>), 97	<code>add_Extension()</code> (<i>PMML44Super.GARCH method</i>), 101	<code>add_Extension()</code> (<i>PMML44Super.GARCHPart method</i>), 102	<code>add_Extension()</code> (<i>PMML44Super.GaussianDistribution method</i>), 103	<code>add_Extension()</code> (<i>PMML44Super.GaussianProcessModel method</i>), 104	<code>add_Extension()</code> (<i>PMML44Super.GeneralizedExponentialKernel method</i>), 109	<code>add_Extension()</code> (<i>PMML44Super.GeneralRegressionModel method</i>), 106	<code>add_Extension()</code> (<i>PMML44Super.Header method</i>), 110	<code>add_Extension()</code> (<i>PMML44Super.InlineTable method</i>), 113	<code>add_Extension()</code> (<i>PMML44Super.InstanceField method</i>), 114	<code>add_Extension()</code> (<i>PMML44Super.InstanceFields method</i>), 114	<code>add_Extension()</code> (<i>PMML44Super.Interval method</i>), 115	<code>add_Extension()</code> (<i>PMML44Super.Item method</i>), 116	<code>add_Extension()</code> (<i>PMML44Super.ItemRef method</i>), 117	<code>add_Extension()</code> (<i>PMML44Super.Itemset method</i>), 118	<code>add_Extension()</code> (<i>PMML44Super.jaccard method</i>), 301	<code>add_Extension()</code> (<i>PMML44Super.KNNInput method</i>), 119	<code>add_Extension()</code> (<i>PMML44Super.KNNInputs method</i>), 119	<code>add_Extension()</code> (<i>PMML44Super.KohonenMap method</i>), 121	<code>add_Extension()</code> (<i>PMML44Super.Lag method</i>), 122	<code>add_Extension()</code> (<i>PMML44Super.Lambda method</i>), 122	<code>add_Extension()</code> (<i>PMML44Super.LayerBias method</i>), 123	<code>add_Extension()</code> (<i>PMML44Super.LayerParameters method</i>), 124	<code>add_Extension()</code> (<i>PMML44Super.LayerWeights method</i>), 128	<code>add_Extension()</code> (<i>PMML44Super.LiftData method</i>), 129	<code>add_Extension()</code> (<i>PMML44Super.LiftGraph method</i>), 130	<code>add_Extension()</code> (<i>PMML44Super.LinearKernelType method</i>), 131	<code>add_Extension()</code> (<i>PMML44Super.LinearNorm method</i>), 132	<code>add_Extension()</code> (<i>PMML44Super.LocalTransformations method</i>), 133	<code>add_Extension()</code> (<i>PMML44Super.LognormalDistributionForBN method</i>), 133	<code>add_Extension()</code> (<i>PMML44Super.Losses method</i>), 134	<code>add_Extension()</code> (<i>PMML44Super.Lower method</i>), 135	<code>add_Extension()</code> (<i>PMML44Super.MA method</i>), 136	<code>add_Extension()</code> (<i>PMML44Super.MACoefficients method</i>), 137	<code>add_Extension()</code> (<i>PMML44Super.MapValues method</i>), 137	<code>add_Extension()</code> (<i>PMML44Super.Mean method</i>), 141	<code>add_Extension()</code> (<i>PMML44Super.MeasurementMatrix method</i>), 142	<code>add_Extension()</code> (<i>PMML44Super.Metrics method</i>), 143	<code>add_Extension()</code> (<i>PMML44Super.MiningBuildTask method</i>), 143	<code>add_Extension()</code> (<i>PMML44Super.MiningField method</i>), 144	<code>add_Extension()</code> (<i>PMML44Super.MiningModel method</i>), 146	<code>add_Extension()</code> (<i>PMML44Super.MiningSchema method</i>), 147
--	--	---	--	--	--	--	---	--	--	---	--	--	---	--	--	---	---	---	---	---	--	--	---	---	---	--	--	--	---	--	---	--	---	--	--	---	---	--	---	---	---	---	---	--	---	---	--	---	--	--	--	--	--	---

<code>add_Extension()</code> (PMML44Super.minkowski method), 301	<code>add_Extension()</code> (PMML44Super.NumericPredictor method), 175
<code>add_Extension()</code> (PMML44Super.MissingValueWeights method), 148	<code>add_Extension()</code> (PMML44Super.Optimizers method), 176
<code>add_Extension()</code> (PMML44Super.ModelExplanation method), 149	<code>add_Extension()</code> (PMML44Super.OptimumLiftGraph method), 178
<code>add_Extension()</code> (PMML44Super.ModelLiftGraph method), 150	<code>add_Extension()</code> (PMML44Super.OutlierEffect method), 178
<code>add_Extension()</code> (PMML44Super.ModelStats method), 150	<code>add_Extension()</code> (PMML44Super.Output method), 179
<code>add_Extension()</code> (PMML44Super.ModelVerification method), 151	<code>add_Extension()</code> (PMML44Super.OutputField method), 180
<code>add_Extension()</code> (PMML44Super.MultivariateStat method), 152	<code>add_Extension()</code> (PMML44Super.PairCounts method), 190
<code>add_Extension()</code> (PMML44Super.MultivariateStats method), 154	<code>add_Extension()</code> (PMML44Super.Parameter method), 192
<code>add_Extension()</code> (PMML44Super.Nadam method), 154	<code>add_Extension()</code> (PMML44Super.ParameterList method), 194
<code>add_Extension()</code> (PMML44Super.NaiveBayesModel method), 155	<code>add_Extension()</code> (PMML44Super.ParamMatrix method), 191
<code>add_Extension()</code> (PMML44Super.NearestNeighborModel method), 157	<code>add_Extension()</code> (PMML44Super.ParentValue method), 194
<code>add_Extension()</code> (PMML44Super.NetworkLayer method), 159	<code>add_Extension()</code> (PMML44Super.Partition method), 195
<code>add_Extension()</code> (PMML44Super.NeuralInput method), 160	<code>add_Extension()</code> (PMML44Super.PartitionFieldStats method), 196
<code>add_Extension()</code> (PMML44Super.NeuralInputs method), 161	<code>add_Extension()</code> (PMML44Super.PastVariances method), 197
<code>add_Extension()</code> (PMML44Super.NeuralLayer method), 162	<code>add_Extension()</code> (PMML44Super.PCell method), 182
<code>add_Extension()</code> (PMML44Super.NeuralNetwork method), 163	<code>add_Extension()</code> (PMML44Super.PCovCell method), 183
<code>add_Extension()</code> (PMML44Super.NeuralOutput method), 165	<code>add_Extension()</code> (PMML44Super.PCovMatrix method), 184
<code>add_Extension()</code> (PMML44Super.NeuralOutputs method), 165	<code>add_Extension()</code> (PMML44Super.PMML method), 185
<code>add_Extension()</code> (PMML44Super.Neuron method), 166	<code>add_Extension()</code> (PMML44Super.PoissonDistribution method), 198
<code>add_Extension()</code> (PMML44Super.Node method), 167	<code>add_Extension()</code> (PMML44Super.PolynomialKernelType method), 198
<code>add_Extension()</code> (PMML44Super.NonseasonalComponent method), 169	<code>add_Extension()</code> (PMML44Super.PPCell method), 189
<code>add_Extension()</code> (PMML44Super.NonseasonalFactor method), 170	<code>add_Extension()</code> (PMML44Super.PPMatrix method), 190
<code>add_Extension()</code> (PMML44Super.NormalDistributionForBN method), 173	<code>add_Extension()</code> (PMML44Super.PredictiveModelQuality method), 199
<code>add_Extension()</code> (PMML44Super.NormContinuous method), 171	<code>add_Extension()</code> (PMML44Super.Predictor method), 201
<code>add_Extension()</code> (PMML44Super.NormDiscrete method), 172	<code>add_Extension()</code> (PMML44Super.PredictorTerm method), 202
<code>add_Extension()</code> (PMML44Super.Numerator method), 173	<code>add_Extension()</code> (PMML44Super.PsiVector method), 203
<code>add_Extension()</code> (PMML44Super.NumericInfo method), 174	<code>add_Extension()</code> (PMML44Super.Quantile method), 204

<code>add_Extension()</code> (<i>PMML44Super.RadialBasisKernel method</i>), 208	<code>add_Extension()</code> (<i>PMML44Super.SetReference method</i>), 238
<code>add_Extension()</code> (<i>PMML44Super.RadialBasisKernelType method</i>), 209	<code>add_Extension()</code> (<i>PMML44Super.SGD method</i>), 222
<code>add_Extension()</code> (<i>PMML44Super.RandomLiftGraph method</i>), 210	<code>add_Extension()</code> (<i>PMML44Super.SigmoidKernelType method</i>), 239
<code>add_Extension()</code> (<i>PMML44Super.Regression method</i>), 211	<code>add_Extension()</code> (<i>PMML44Super.simpleMatching method</i>), 303
<code>add_Extension()</code> (<i>PMML44Super.RegressionModel method</i>), 212	<code>add_Extension()</code> (<i>PMML44Super.SimplePredicate method</i>), 239
<code>add_Extension()</code> (<i>PMML44Super.RegressionTable method</i>), 214	<code>add_Extension()</code> (<i>PMML44Super.SimpleRule method</i>), 240
<code>add_Extension()</code> (<i>PMML44Super.RegressorValues method</i>), 215	<code>add_Extension()</code> (<i>PMML44Super.SimpleSetPredicate method</i>), 242
<code>add_Extension()</code> (<i>PMML44Super.Residuals method</i>), 216	<code>add_Extension()</code> (<i>PMML44Super.squaredEuclidean method</i>), 303
<code>add_Extension()</code> (<i>PMML44Super.ResidualSquareCoefficients method</i>), 216	<code>add_Extension()</code> (<i>PMML44Super.StateSpaceModel method</i>), 243
<code>add_Extension()</code> (<i>PMML44Super.ResultField method</i>), 217	<code>add_Extension()</code> (<i>PMML44Super.StateVector method</i>), 244
<code>add_Extension()</code> (<i>PMML44Super.RMSprop method</i>), 206	<code>add_Extension()</code> (<i>PMML44Super.SupportVector method</i>), 245
<code>add_Extension()</code> (<i>PMML44Super.ROC method</i>), 207	<code>add_Extension()</code> (<i>PMML44Super.SupportVectorMachine method</i>), 245
<code>add_Extension()</code> (<i>PMML44Super.ROCGraph method</i>), 208	<code>add_Extension()</code> (<i>PMML44Super.SupportVectorMachineModel method</i>), 247
<code>add_Extension()</code> (<i>PMML44Super.RuleSelectionMethod method</i>), 218	<code>add_Extension()</code> (<i>PMML44Super.SupportVectors method</i>), 249
<code>add_Extension()</code> (<i>PMML44Super.RuleSet method</i>), 219	<code>add_Extension()</code> (<i>PMML44Super.TableLocator method</i>), 250
<code>add_Extension()</code> (<i>PMML44Super.RuleSetModel method</i>), 220	<code>add_Extension()</code> (<i>PMML44Super.tanimoto method</i>), 304
<code>add_Extension()</code> (<i>PMML44Super.Scorecard method</i>), 223	<code>add_Extension()</code> (<i>PMML44Super.Target method</i>), 250
<code>add_Extension()</code> (<i>PMML44Super.ScoreDistribution method</i>), 222	<code>add_Extension()</code> (<i>PMML44Super.Targets method</i>), 256
<code>add_Extension()</code> (<i>PMML44Super.SeasonalComponent method</i>), 225	<code>add_Extension()</code> (<i>PMML44Super.TargetValue method</i>), 251
<code>add_Extension()</code> (<i>PMML44Super.SeasonalFactor method</i>), 226	<code>add_Extension()</code> (<i>PMML44Super.TargetValueCount method</i>), 252
<code>add_Extension()</code> (<i>PMML44Super.Segment method</i>), 228	<code>add_Extension()</code> (<i>PMML44Super.TargetValueCounts method</i>), 253
<code>add_Extension()</code> (<i>PMML44Super.Segmentation method</i>), 231	<code>add_Extension()</code> (<i>PMML44Super.TargetValueStat method</i>), 254
<code>add_Extension()</code> (<i>PMML44Super.Sequence method</i>), 231	<code>add_Extension()</code> (<i>PMML44Super.TargetValueStats method</i>), 255
<code>add_Extension()</code> (<i>PMML44Super.SequenceModel method</i>), 233	<code>add_Extension()</code> (<i>PMML44Super.Taxonomy method</i>), 256
<code>add_Extension()</code> (<i>PMML44Super.SequenceReference method</i>), 235	<code>add_Extension()</code> (<i>PMML44Super.TestDistributions method</i>), 257
<code>add_Extension()</code> (<i>PMML44Super.SequenceRule method</i>), 236	<code>add_Extension()</code> (<i>PMML44Super.TextCorpus method</i>), 258
<code>add_Extension()</code> (<i>PMML44Super.SetPredicate method</i>), 237	<code>add_Extension()</code> (<i>PMML44Super.TextDictionary method</i>), 259

<code>add_Extension()</code> (PMML44Super.TextDocument method), 260	<code>add_Extension()</code> (PMML44Super.VectorFields method), 293
<code>add_Extension()</code> (PMML44Super.TextIndex method), 261	<code>add_Extension()</code> (PMML44Super.VectorInstance method), 294
<code>add_Extension()</code> (PMML44Super.TextIndexNormalization method), 262	<code>add_Extension()</code> (PMML44Super.VerificationField method), 295
<code>add_Extension()</code> (PMML44Super.TextModel method), 264	<code>add_Extension()</code> (PMML44Super.VerificationFields method), 296
<code>add_Extension()</code> (PMML44Super.TextModelNormalization method), 265	<code>add_Extension()</code> (PMML44Super.XCoordinates method), 297
<code>add_Extension()</code> (PMML44Super.TextModelSimilarity method), 266	<code>add_Extension()</code> (PMML44Super.YCoordinates method), 297
<code>add_Extension()</code> (PMML44Super.Time method), 268	<code>add_False()</code> (PMML44Super.CompoundPredicate method), 56
<code>add_Extension()</code> (PMML44Super.TimeSeriesModel method), 273	<code>add_FieldColumnPair()</code> (PMML44Super.ChildParent method), 44
<code>add_Extension()</code> (PMML44Super.Timestamp method), 275	<code>add_FieldColumnPair()</code> (PMML44Super.MapValues method), 137
<code>add_Extension()</code> (PMML44Super.TrainingInstances method), 276	<code>add_FieldRef()</code> (PMML44Super.Apply method), 20
<code>add_Extension()</code> (PMML44Super.TrainingParameters method), 277	<code>add_FieldRef()</code> (PMML44Super.Baseline method), 29
<code>add_Extension()</code> (PMML44Super.TransformationDictionary method), 278	<code>add_FieldRef()</code> (PMML44Super.PredictorTerm method), 202
<code>add_Extension()</code> (PMML44Super.TransitionMatrix method), 279	<code>add_FieldRef()</code> (PMML44Super.VectorFields method), 293
<code>add_Extension()</code> (PMML44Super.TreeModel method), 280	<code>add_FieldValue()</code> (PMML44Super.COUNT_TABLE_TYPE method), 39
<code>add_Extension()</code> (PMML44Super.TrendCoefficients method), 282	<code>add_FieldValue()</code> (PMML44Super.FieldValue method), 96
<code>add_Extension()</code> (PMML44Super.TriangularDistributionForBN method), 283	<code>add_FieldValueCount()</code> (PMML44Super.COUNT_TABLE_TYPE method), 40
<code>add_Extension()</code> (PMML44Super.True_ method), 284	<code>add_FieldValueCount()</code> (PMML44Super.FieldValue method), 96
<code>add_Extension()</code> (PMML44Super.UniformDistribution method), 285	<code>add_GaussianProcessModel()</code> (PMML44Super.PMML method), 185
<code>add_Extension()</code> (PMML44Super.UniformDistributionForBN method), 285	<code>add_GeneralRegressionModel()</code> (PMML44Super.PMML method), 185
<code>add_Extension()</code> (PMML44Super.UnivariateStats method), 286	<code>add_InstanceField()</code> (PMML44Super.InstanceFields method), 114
<code>add_Extension()</code> (PMML44Super.Upper method), 287	<code>add_Interval()</code> (PMML44Super.ContStats method), 63
<code>add_Extension()</code> (PMML44Super.Value method), 288	<code>add_Interval()</code> (PMML44Super.DataField method), 73
<code>add_Extension()</code> (PMML44Super.ValueProbability method), 289	<code>add_Interval()</code> (PMML44Super.EventValues method), 91
<code>add_Extension()</code> (PMML44Super.VariableWeight method), 290	<code>add_Item()</code> (PMML44Super.AssociationModel method), 23
<code>add_Extension()</code> (PMML44Super.Variance method), 290	<code>add_Item()</code> (PMML44Super.SequenceModel method), 233
<code>add_Extension()</code> (PMML44Super.VarianceCoefficients method), 292	<code>add_ItemRef()</code> (PMML44Super.Itemset method), 118
<code>add_Extension()</code> (PMML44Super.VectorDictionary method), 292	<code>add_Itemset()</code> (PMML44Super.AssociationModel method), 23

<code>add_Itemset()</code> (PMML44Super.SequenceModel method), 233	<code>add_NUM_ARRAY()</code> (PMML44Super.ContStats method), 63
<code>add_KNNInput()</code> (PMML44Super.KNNInputs method), 119	<code>add_NumericPredictor()</code> (PMML44Super.RegressionTable method), 214
<code>add_Lag()</code> (PMML44Super.Apply method), 20	<code>add_OutlierEffect()</code> (PMML44Super.ARIMA method), 5
<code>add_Lambda()</code> (PMML44Super.AbsoluteExponentialKernel method), 7	<code>add_OutputField()</code> (PMML44Super.Output method), 179
<code>add_Lambda()</code> (PMML44Super.ARDSquaredExponentialKernel method), 4	<code>add_PairCounts()</code> (PMML44Super.BayesInput method), 33
<code>add_Lambda()</code> (PMML44Super.GeneralizedExponentialKernel method), 109	<code>add_Parameter()</code> (PMML44Super.ParameterList method), 194
<code>add_LiftData()</code> (PMML44Super.PredictiveModelQuality method), 199	<code>add_ParameterField()</code> (PMML44Super.DefineFunction method), 79
<code>add_LinearNorm()</code> (PMML44Super.NormContinuous method), 171	<code>add_ParentValue()</code> (PMML44Super.ContinuousConditionalProbability method), 64
<code>add_MapValues()</code> (PMML44Super.Apply method), 20	<code>add_ParentValue()</code> (PMML44Super.DiscreteConditionalProbability method), 85
<code>add_MatCell()</code> (PMML44Super.Matrix method), 139	<code>add_PartitionFieldStats()</code> (PMML44Super.Partition method), 195
<code>add_MiningField()</code> (PMML44Super.MiningSchema method), 147	<code>add_PCell()</code> (PMML44Super.ParamMatrix method), 191
<code>add_MiningModel()</code> (PMML44Super.PMML method), 185	<code>add_PCovCell()</code> (PMML44Super.PCovMatrix method), 184
<code>add_MultivariateStat()</code> (PMML44Super.MultivariateStats method), 154	<code>add_PPCell()</code> (PMML44Super.PPMatrix method), 190
<code>add_MultivariateStats()</code> (PMML44Super.ModelStats method), 150	<code>add_PredictiveModelQuality()</code> (PMML44Super.ModelExplanation method), 149
<code>add_Nadam()</code> (PMML44Super.Optimizers method), 176	<code>add_Predictor()</code> (PMML44Super.CovariateList method), 71
<code>add_NaiveBayesModel()</code> (PMML44Super.PMML method), 186	<code>add_Predictor()</code> (PMML44Super.FactorList method), 93
<code>add_NearestNeighborModel()</code> (PMML44Super.PMML method), 186	<code>add_PredictorTerm()</code> (PMML44Super.RegressionTable method), 214
<code>add_NetworkLayer()</code> (PMML44Super.DeepNetwork method), 78	<code>add_Quantile()</code> (PMML44Super.NumericInfo method), 174
<code>add_NeuralInput()</code> (PMML44Super.NeuralInputs method), 161	<code>add_Regression()</code> (PMML44Super.MiningModel method), 146
<code>add_NeuralLayer()</code> (PMML44Super.NeuralNetwork method), 163	<code>add_RegressionModel()</code> (PMML44Super.PMML method), 186
<code>add_NeuralNetwork()</code> (PMML44Super.PMML method), 186	<code>add_RegressionTable()</code> (PMML44Super.Regression method), 211
<code>add_NeuralOutput()</code> (PMML44Super.NeuralOutputs method), 166	<code>add_RegressionTable()</code> (PMML44Super.RegressionModel method), 212
<code>add_Neuron()</code> (PMML44Super.NeuralLayer method), 162	<code>add_ResultField()</code> (PMML44Super.DecisionTree method), 75
<code>add_Node()</code> (PMML44Super.Node method), 167	<code>add_ResultField()</code> (PMML44Super.Regression
<code>add_NormContinuous()</code> (PMML44Super.Apply method), 20	
<code>add_NormDiscrete()</code> (PMML44Super.Apply method), 20	

[method](#)), 211
[add_RMSprop\(\)](#) (*PMML44Super.Optimizers method*), 176
[add_row\(\)](#) (*PMML44Super.InlineTable method*), 113
[add_RuleSelectionMethod\(\)](#) (*PMML44Super.RuleSet method*), 219
[add_RuleSetModel\(\)](#) (*PMML44Super.PMML method*), 186
[add_Scorecard\(\)](#) (*PMML44Super.PMML method*), 186
[add_ScoreDistribution\(\)](#) (*PMML44Super.Node method*), 167
[add_ScoreDistribution\(\)](#) (*PMML44Super.RuleSet method*), 219
[add_ScoreDistribution\(\)](#) (*PMML44Super.SimpleRule method*), 240
[add_Segment\(\)](#) (*PMML44Super.Segmentation method*), 231
[add_Sequence\(\)](#) (*PMML44Super.SequenceModel method*), 233
[add_SequenceModel\(\)](#) (*PMML44Super.PMML method*), 186
[add_SequenceRule\(\)](#) (*PMML44Super.SequenceModel method*), 233
[add_SetPredicate\(\)](#) (*PMML44Super.SequenceModel method*), 233
[add_SetReference\(\)](#) (*PMML44Super.Sequence method*), 232
[add_SGD\(\)](#) (*PMML44Super.Optimizers method*), 176
[add_SimplePredicate\(\)](#) (*PMML44Super.CompoundPredicate method*), 56
[add_SimpleRule\(\)](#) (*PMML44Super.CompoundRule method*), 57
[add_SimpleRule\(\)](#) (*PMML44Super.RuleSet method*), 219
[add_SimpleSetPredicate\(\)](#) (*PMML44Super.CompoundPredicate method*), 56
[add_SupportVector\(\)](#) (*PMML44Super.SupportVectors method*), 249
[add_SupportVectorMachine\(\)](#) (*PMML44Super.SupportVectorMachineModel method*), 247
[add_SupportVectorMachineModel\(\)](#) (*PMML44Super.PMML method*), 186
[add_Target\(\)](#) (*PMML44Super.Targets method*), 256
[add_TargetValue\(\)](#) (*PMML44Super.Target method*), 250
[add_TargetValueCount\(\)](#) (*PMML44Super.TargetValueCounts method*), 253
[add_TargetValueStat\(\)](#) (*PMML44Super.TargetValueStats method*), 255
[add_Taxonomy\(\)](#) (*PMML44Super.DataDictionary method*), 72
[add_TextDocument\(\)](#) (*PMML44Super.TextCorpus method*), 258
[add_TextIndex\(\)](#) (*PMML44Super.Apply method*), 20
[add_TextIndexNormalization\(\)](#) (*PMML44Super.TextIndex method*), 261
[add_TextModel\(\)](#) (*PMML44Super.PMML method*), 186
[add_Theta\(\)](#) (*PMML44Super.FinalTheta method*), 100
[add_TimeCycle\(\)](#) (*PMML44Super.TimeAnchor method*), 269
[add_TimeException\(\)](#) (*PMML44Super.TimeAnchor method*), 269
[add_TimeSeries\(\)](#) (*PMML44Super.TimeSeriesModel method*), 273
[add_TimeSeriesModel\(\)](#) (*PMML44Super.PMML method*), 186
[add_TimeValue\(\)](#) (*PMML44Super.ExponentialSmoothing method*), 92
[add_TimeValue\(\)](#) (*PMML44Super.TimeSeries method*), 272
[add_TreeModel\(\)](#) (*PMML44Super.PMML method*), 186
[add_True\(\)](#) (*PMML44Super.CompoundPredicate method*), 56
[add_UnivariateStats\(\)](#) (*PMML44Super.ModelStats method*), 150
[add_Value\(\)](#) (*PMML44Super.DataField method*), 73
[add_Value\(\)](#) (*PMML44Super.DerivedField method*), 82
[add_Value\(\)](#) (*PMML44Super.EventValues method*), 91
[add_ValueProbability\(\)](#) (*PMML44Super.DiscreteConditionalProbability method*), 85
[add_ValueProbability\(\)](#) (*PMML44Super.DiscreteNode method*), 86
[add_VectorInstance\(\)](#) (*PMML44Super.VectorDictionary method*), 292
[add_VerificationField\(\)](#) (*PMML44Super.VerificationFields method*), 296
[Aggregate](#) (class in *PMML44Super*), 11
[Alternate](#) (class in *PMML44Super*), 12
[Annotation](#) (class in *PMML44Super*), 13
[AnomalyDetectionModel](#) (class in

PMML44Super), 13
 Anova (class in *PMML44Super*), 16
 AnovaRow (class in *PMML44Super*), 17
 AntecedentSequence (class in *PMML44Super*), 18
 AnyDistribution (class in *PMML44Super*), 19
 Application (class in *PMML44Super*), 19
 Apply (class in *PMML44Super*), 20
 AR (class in *PMML44Super*), 3
 ARDSquaredExponentialKernel (class in *PMML44Super*), 4
 ARIMA (class in *PMML44Super*), 5
 ARMAPart (class in *PMML44Super*), 6
 ArrayType (class in *PMML44Super*), 22
 AssociationModel (class in *PMML44Super*), 23
 AssociationRule (class in *PMML44Super*), 25
 Attribute (class in *PMML44Super*), 26

B

BaseCumHazardTables (class in *PMML44Super*), 28
 Baseline (class in *PMML44Super*), 28
 BaselineCell (class in *PMML44Super*), 29
 BaselineModel (class in *PMML44Super*), 30
 BaselineStratum (class in *PMML44Super*), 32
 BayesianNetworkModel (class in *PMML44Super*), 35
 BayesianNetworkNodes (class in *PMML44Super*), 36
 BayesInput (class in *PMML44Super*), 32
 BayesInputs (class in *PMML44Super*), 33
 BayesOutput (class in *PMML44Super*), 34
 binarySimilarity (class in *PMML44Super*), 298
 BlockIndicator (class in *PMML44Super*), 37
 BoundaryValueMeans (class in *PMML44Super*), 38
 BoundaryValues (class in *PMML44Super*), 39
 build() (*PMML44Super.AbsoluteExponentialKernel* method), 7
 build() (*PMML44Super.Adadelta* method), 8
 build() (*PMML44Super.Adagrad* method), 9
 build() (*PMML44Super.Adam* method), 10
 build() (*PMML44Super.Adamax* method), 10
 build() (*PMML44Super.Aggregate* method), 11
 build() (*PMML44Super.Alternate* method), 12
 build() (*PMML44Super.Annotation* method), 13
 build() (*PMML44Super.AnomalyDetectionModel* method), 14
 build() (*PMML44Super.Anova* method), 16
 build() (*PMML44Super.AnovaRow* method), 17
 build() (*PMML44Super.AntecedentSequence* method), 18
 build() (*PMML44Super.AnyDistribution* method), 19
 build() (*PMML44Super.Application* method), 19
 build() (*PMML44Super.Apply* method), 20
 build() (*PMML44Super.AR* method), 3

build() (*PMML44Super.ARDSquaredExponentialKernel* method), 4
 build() (*PMML44Super.ARIMA* method), 5
 build() (*PMML44Super.ARMAPart* method), 6
 build() (*PMML44Super.ArrayType* method), 22
 build() (*PMML44Super.AssociationModel* method), 23
 build() (*PMML44Super.AssociationRule* method), 25
 build() (*PMML44Super.Attribute* method), 27
 build() (*PMML44Super.BaseCumHazardTables* method), 28
 build() (*PMML44Super.Baseline* method), 29
 build() (*PMML44Super.BaselineCell* method), 30
 build() (*PMML44Super.BaselineModel* method), 30
 build() (*PMML44Super.BaselineStratum* method), 32
 build() (*PMML44Super.BayesianNetworkModel* method), 35
 build() (*PMML44Super.BayesianNetworkNodes* method), 37
 build() (*PMML44Super.BayesInput* method), 33
 build() (*PMML44Super.BayesInputs* method), 34
 build() (*PMML44Super.BayesOutput* method), 34
 build() (*PMML44Super.binarySimilarity* method), 298
 build() (*PMML44Super.BlockIndicator* method), 37
 build() (*PMML44Super.BoundaryValueMeans* method), 38
 build() (*PMML44Super.BoundaryValues* method), 39
 build() (*PMML44Super.CategoricalPredictor* method), 40
 build() (*PMML44Super.Categories* method), 41
 build() (*PMML44Super.Category* method), 42
 build() (*PMML44Super.Characteristic* method), 43
 build() (*PMML44Super.Characteristics* method), 44
 build() (*PMML44Super.chebychev* method), 299
 build() (*PMML44Super.ChildParent* method), 44
 build() (*PMML44Super.cityBlock* method), 300
 build() (*PMML44Super.ClassLabels* method), 45
 build() (*PMML44Super.Cluster* method), 46
 build() (*PMML44Super.ClusteringField* method), 47
 build() (*PMML44Super.ClusteringModel* method), 48
 build() (*PMML44Super.ClusteringModelQuality* method), 50
 build() (*PMML44Super.Coefficient* method), 51
 build() (*PMML44Super.Coefficients* method), 52
 build() (*PMML44Super.ComparisonMeasure* method), 52
 build() (*PMML44Super.Comparisons* method), 54
 build() (*PMML44Super.ComplexPartialScore* method), 55
 build() (*PMML44Super.CompoundPredicate* method), 56
 build() (*PMML44Super.CompoundRule* method), 57
 build() (*PMML44Super.Con* method), 58

build() (PMML44Super.ConfusionMatrix method), 59
 build() (PMML44Super.ConsequentSequence method), 60
 build() (PMML44Super.Constant method), 61
 build() (PMML44Super.Constraints method), 61
 build() (PMML44Super.ContinuousConditionalProbability method), 64
 build() (PMML44Super.ContinuousDistribution method), 65
 build() (PMML44Super.ContinuousNode method), 66
 build() (PMML44Super.ContStats method), 63
 build() (PMML44Super.CorrelationFields method), 67
 build() (PMML44Super.CorrelationMethods method), 68
 build() (PMML44Super.Correlations method), 69
 build() (PMML44Super.CorrelationValues method), 68
 build() (PMML44Super.COUNT_TABLE_TYPE method), 40
 build() (PMML44Super.Counts method), 70
 build() (PMML44Super.Covariances method), 71
 build() (PMML44Super.CovariateList method), 71
 build() (PMML44Super.DataDictionary method), 72
 build() (PMML44Super.DataField method), 73
 build() (PMML44Super.Decision method), 74
 build() (PMML44Super.Decisions method), 77
 build() (PMML44Super.DecisionTree method), 75
 build() (PMML44Super.DeepNetwork method), 78
 build() (PMML44Super.DefineFunction method), 79
 build() (PMML44Super.Delimiter method), 81
 build() (PMML44Super.Denominator method), 82
 build() (PMML44Super.DerivedField method), 82
 build() (PMML44Super.DiscreteConditionalProbability method), 85
 build() (PMML44Super.DiscreteNode method), 86
 build() (PMML44Super.Discretize method), 87
 build() (PMML44Super.DiscretizeBin method), 88
 build() (PMML44Super.DiscrStats method), 84
 build() (PMML44Super.DocumentTermMatrix method), 89
 build() (PMML44Super.DynamicRegressor method), 90
 build() (PMML44Super.euclidean method), 300
 build() (PMML44Super.EventValues method), 91
 build() (PMML44Super.ExponentialSmoothing method), 92
 build() (PMML44Super.Extension method), 93
 build() (PMML44Super.FactorList method), 93
 build() (PMML44Super.False_ method), 94
 build() (PMML44Super.FieldColumnPair method), 95
 build() (PMML44Super.FieldRef method), 95
 build() (PMML44Super.FieldValue method), 96
 build() (PMML44Super.FieldValueCount method), 97
 build() (PMML44Super.FinalNoise method), 98
 build() (PMML44Super.FinalNu method), 98
 build() (PMML44Super.FinalOmega method), 99
 build() (PMML44Super.FinalPredictedNoise method), 99
 build() (PMML44Super.FinalStateVector method), 100
 build() (PMML44Super.FinalTheta method), 100
 build() (PMML44Super.GARCH method), 101
 build() (PMML44Super.GARCHPart method), 102
 build() (PMML44Super.GaussianDistribution method), 103
 build() (PMML44Super.GaussianProcessModel method), 104
 build() (PMML44Super.GeneralizedExponentialKernel method), 109
 build() (PMML44Super.GeneralRegressionModel method), 106
 build() (PMML44Super.Header method), 110
 build() (PMML44Super.HVector method), 110
 build() (PMML44Super.Indices method), 113
 build() (PMML44Super.InlineTable method), 113
 build() (PMML44Super.InstanceField method), 114
 build() (PMML44Super.InstanceFields method), 114
 build() (PMML44Super.INT_Entries method), 111
 build() (PMML44Super.INT_SparseArray method), 112
 build() (PMML44Super.Interval method), 115
 build() (PMML44Super.Item method), 116
 build() (PMML44Super.ItemRef method), 117
 build() (PMML44Super.Itemset method), 118
 build() (PMML44Super.jaccard method), 301
 build() (PMML44Super.KalmanState method), 120
 build() (PMML44Super.KNNInput method), 119
 build() (PMML44Super.KNNInputs method), 119
 build() (PMML44Super.KohonenMap method), 121
 build() (PMML44Super.Lag method), 122
 build() (PMML44Super.Lambda method), 122
 build() (PMML44Super.LayerBias method), 123
 build() (PMML44Super.LayerParameters method), 124
 build() (PMML44Super.LayerWeights method), 128
 build() (PMML44Super.Level method), 129
 build() (PMML44Super.LiftData method), 129
 build() (PMML44Super.LiftGraph method), 130
 build() (PMML44Super.LinearKernelType method), 131
 build() (PMML44Super.LinearNorm method), 132
 build() (PMML44Super.LocalTransformations method), 133
 build() (PMML44Super.LognormalDistributionForBN method), 133
 build() (PMML44Super.Losses method), 134
 build() (PMML44Super.Lower method), 135

`build()` (*PMML44Super.MA method*), 136
`build()` (*PMML44Super.MACoefficients method*), 137
`build()` (*PMML44Super.MapValues method*), 137
`build()` (*PMML44Super.MatCell method*), 138
`build()` (*PMML44Super.Matrix method*), 139
`build()` (*PMML44Super.MaximumLikelihoodStat method*), 140
`build()` (*PMML44Super.Mean method*), 141
`build()` (*PMML44Super.MeasurementMatrix method*), 142
`build()` (*PMML44Super.Metrics method*), 143
`build()` (*PMML44Super.MiningBuildTask method*), 144
`build()` (*PMML44Super.MiningField method*), 144
`build()` (*PMML44Super.MiningModel method*), 146
`build()` (*PMML44Super.MiningSchema method*), 147
`build()` (*PMML44Super.minkowski method*), 301
`build()` (*PMML44Super.MissingValueWeights method*), 148
`build()` (*PMML44Super.ModelExplanation method*), 149
`build()` (*PMML44Super.ModelLiftGraph method*), 150
`build()` (*PMML44Super.ModelStats method*), 150
`build()` (*PMML44Super.ModelVerification method*), 151
`build()` (*PMML44Super.MultivariateStat method*), 152
`build()` (*PMML44Super.MultivariateStats method*), 154
`build()` (*PMML44Super.Nadam method*), 154
`build()` (*PMML44Super.NaiveBayesModel method*), 155
`build()` (*PMML44Super.NearestNeighborModel method*), 157
`build()` (*PMML44Super.NetworkLayer method*), 159
`build()` (*PMML44Super.NeuralInput method*), 160
`build()` (*PMML44Super.NeuralInputs method*), 161
`build()` (*PMML44Super.NeuralLayer method*), 162
`build()` (*PMML44Super.NeuralNetwork method*), 163
`build()` (*PMML44Super.NeuralOutput method*), 165
`build()` (*PMML44Super.NeuralOutputs method*), 166
`build()` (*PMML44Super.Neuron method*), 166
`build()` (*PMML44Super.Node method*), 167
`build()` (*PMML44Super.NonseasonalComponent method*), 169
`build()` (*PMML44Super.NonseasonalFactor method*), 170
`build()` (*PMML44Super.NormalDistributionForBN method*), 173
`build()` (*PMML44Super.NormContinuous method*), 171
`build()` (*PMML44Super.NormDiscrete method*), 172
`build()` (*PMML44Super.Numerator method*), 173
`build()` (*PMML44Super.NumericInfo method*), 174
`build()` (*PMML44Super.NumericPredictor method*), 175
`build()` (*PMML44Super.Optimizers method*), 176
`build()` (*PMML44Super.OptimumLiftGraph method*), 178
`build()` (*PMML44Super.OutlierEffect method*), 178
`build()` (*PMML44Super.Output method*), 179
`build()` (*PMML44Super.OutputField method*), 180
`build()` (*PMML44Super.PairCounts method*), 190
`build()` (*PMML44Super.Parameter method*), 192
`build()` (*PMML44Super.ParameterField method*), 193
`build()` (*PMML44Super.ParameterList method*), 194
`build()` (*PMML44Super.ParamMatrix method*), 191
`build()` (*PMML44Super.ParentValue method*), 194
`build()` (*PMML44Super.Partition method*), 195
`build()` (*PMML44Super.PartitionFieldStats method*), 196
`build()` (*PMML44Super.PastVariances method*), 197
`build()` (*PMML44Super.PCell method*), 182
`build()` (*PMML44Super.PCovCell method*), 183
`build()` (*PMML44Super.PCovMatrix method*), 184
`build()` (*PMML44Super.PMML method*), 186
`build()` (*PMML44Super.PoissonDistribution method*), 198
`build()` (*PMML44Super.PolynomialKernelType method*), 198
`build()` (*PMML44Super.PPCell method*), 189
`build()` (*PMML44Super.PPMatrix method*), 190
`build()` (*PMML44Super.PredictiveModelQuality method*), 199
`build()` (*PMML44Super.Predictor method*), 201
`build()` (*PMML44Super.PredictorTerm method*), 202
`build()` (*PMML44Super.PsiVector method*), 203
`build()` (*PMML44Super.Quantile method*), 204
`build()` (*PMML44Super.RadialBasisKernel method*), 208
`build()` (*PMML44Super.RadialBasisKernelType method*), 209
`build()` (*PMML44Super.RandomLiftGraph method*), 210
`build()` (*PMML44Super.REAL_Entries method*), 205
`build()` (*PMML44Super.REAL_SparseArray method*), 205
`build()` (*PMML44Super.Regression method*), 211
`build()` (*PMML44Super.RegressionModel method*), 212
`build()` (*PMML44Super.RegressionTable method*), 214
`build()` (*PMML44Super.RegressorValues method*), 215
`build()` (*PMML44Super.Residuals method*), 217
`build()` (*PMML44Super.ResidualSquareCoefficients method*), 216

- `build()` (*PMML44Super.ResultField* method), 217
- `build()` (*PMML44Super.RMSprop* method), 206
- `build()` (*PMML44Super.ROC* method), 207
- `build()` (*PMML44Super.ROCGraph* method), 208
- `build()` (*PMML44Super.row* method), 302
- `build()` (*PMML44Super.RuleSelectionMethod* method), 218
- `build()` (*PMML44Super.RuleSet* method), 219
- `build()` (*PMML44Super.RuleSetModel* method), 220
- `build()` (*PMML44Super.Scorecard* method), 223
- `build()` (*PMML44Super.ScoreDistribution* method), 222
- `build()` (*PMML44Super.SeasonalComponent* method), 225
- `build()` (*PMML44Super.SeasonalFactor* method), 226
- `build()` (*PMML44Super.Seasonality_ExpoSmooth* method), 227
- `build()` (*PMML44Super.SeasonalTrendDecomposition* method), 227
- `build()` (*PMML44Super.Segment* method), 228
- `build()` (*PMML44Super.Segmentation* method), 231
- `build()` (*PMML44Super.Sequence* method), 232
- `build()` (*PMML44Super.SequenceModel* method), 233
- `build()` (*PMML44Super.SequenceReference* method), 235
- `build()` (*PMML44Super.SequenceRule* method), 236
- `build()` (*PMML44Super.SetPredicate* method), 237
- `build()` (*PMML44Super.SetReference* method), 238
- `build()` (*PMML44Super.SGD* method), 222
- `build()` (*PMML44Super.SigmoidKernelType* method), 239
- `build()` (*PMML44Super.simpleMatching* method), 303
- `build()` (*PMML44Super.SimplePredicate* method), 239
- `build()` (*PMML44Super.SimpleRule* method), 240
- `build()` (*PMML44Super.SimpleSetPredicate* method), 242
- `build()` (*PMML44Super.SpectralAnalysis* method), 242
- `build()` (*PMML44Super.squaredEuclidean* method), 303
- `build()` (*PMML44Super.StateSpaceModel* method), 243
- `build()` (*PMML44Super.StateVector* method), 244
- `build()` (*PMML44Super.SupportVector* method), 245
- `build()` (*PMML44Super.SupportVectorMachine* method), 245
- `build()` (*PMML44Super.SupportVectorMachineModel* method), 247
- `build()` (*PMML44Super.SupportVectors* method), 249
- `build()` (*PMML44Super.TableLocator* method), 250
- `build()` (*PMML44Super.tanimoto* method), 304
- `build()` (*PMML44Super.Target* method), 250
- `build()` (*PMML44Super.Targets* method), 256
- `build()` (*PMML44Super.TargetValue* method), 251
- `build()` (*PMML44Super.TargetValueCount* method), 252
- `build()` (*PMML44Super.TargetValueCounts* method), 253
- `build()` (*PMML44Super.TargetValueStat* method), 254
- `build()` (*PMML44Super.TargetValueStats* method), 255
- `build()` (*PMML44Super.Taxonomy* method), 256
- `build()` (*PMML44Super.TestDistributions* method), 257
- `build()` (*PMML44Super.TextCorpus* method), 258
- `build()` (*PMML44Super.TextDictionary* method), 259
- `build()` (*PMML44Super.TextDocument* method), 260
- `build()` (*PMML44Super.TextIndex* method), 261
- `build()` (*PMML44Super.TextIndexNormalization* method), 262
- `build()` (*PMML44Super.TextModel* method), 264
- `build()` (*PMML44Super.TextModelNormalization* method), 265
- `build()` (*PMML44Super.TextModelSimilarity* method), 266
- `build()` (*PMML44Super.Theta* method), 267
- `build()` (*PMML44Super.ThetaRecursionState* method), 268
- `build()` (*PMML44Super.Time* method), 268
- `build()` (*PMML44Super.TimeAnchor* method), 269
- `build()` (*PMML44Super.TimeCycle* method), 270
- `build()` (*PMML44Super.TimeException* method), 271
- `build()` (*PMML44Super.TimeSeries* method), 272
- `build()` (*PMML44Super.TimeSeriesModel* method), 273
- `build()` (*PMML44Super.Timestamp* method), 275
- `build()` (*PMML44Super.TimeValue* method), 274
- `build()` (*PMML44Super.TrainingInstances* method), 276
- `build()` (*PMML44Super.TrainingParameters* method), 277
- `build()` (*PMML44Super.TransferFunctionValues* method), 278
- `build()` (*PMML44Super.TransformationDictionary* method), 279
- `build()` (*PMML44Super.TransitionMatrix* method), 279
- `build()` (*PMML44Super.TreeModel* method), 280
- `build()` (*PMML44Super.Trend_ExpoSmooth* method), 282
- `build()` (*PMML44Super.TrendCoefficients* method), 282
- `build()` (*PMML44Super.TriangularDistributionForBN* method), 283
- `build()` (*PMML44Super.True_* method), 284
- `build()` (*PMML44Super.UniformDistribution* method), 285

`build()` (*PMML44Super.UniformDistributionForBN method*), 285
`build()` (*PMML44Super.UnivariateStats method*), 286
`build()` (*PMML44Super.Upper method*), 287
`build()` (*PMML44Super.Value method*), 288
`build()` (*PMML44Super.ValueProbability method*), 289
`build()` (*PMML44Super.VariableWeight method*), 290
`build()` (*PMML44Super.Variance method*), 290
`build()` (*PMML44Super.VarianceCoefficients method*), 292
`build()` (*PMML44Super.VectorDictionary method*), 292
`build()` (*PMML44Super.VectorFields method*), 293
`build()` (*PMML44Super.VectorInstance method*), 294
`build()` (*PMML44Super.VerificationField method*), 295
`build()` (*PMML44Super.VerificationFields method*), 296
`build()` (*PMML44Super.XCoordinates method*), 297
`build()` (*PMML44Super.YCoordinates method*), 297
`buildAttributes()` (*PMML44Super.AbsoluteExponentialKernel method*), 7
`buildAttributes()` (*PMML44Super.Adadelta method*), 8
`buildAttributes()` (*PMML44Super.Adagrad method*), 9
`buildAttributes()` (*PMML44Super.Adam method*), 10
`buildAttributes()` (*PMML44Super.Adamax method*), 10
`buildAttributes()` (*PMML44Super.Aggregate method*), 11
`buildAttributes()` (*PMML44Super.Alternate method*), 12
`buildAttributes()` (*PMML44Super.Annotation method*), 13
`buildAttributes()` (*PMML44Super.AnomalyDetectionModel method*), 14
`buildAttributes()` (*PMML44Super.Anova method*), 16
`buildAttributes()` (*PMML44Super.AnovaRow method*), 17
`buildAttributes()` (*PMML44Super.AntecedentSequence method*), 18
`buildAttributes()` (*PMML44Super.AnyDistribution method*), 19
`buildAttributes()` (*PMML44Super.Application method*), 19
`buildAttributes()` (*PMML44Super.Apply method*), 21
`buildAttributes()` (*PMML44Super.AR method*), 3
`buildAttributes()` (*PMML44Super.ARDSquaredExponentialKernel method*), 4
`buildAttributes()` (*PMML44Super.ARIMA method*), 5
`buildAttributes()` (*PMML44Super.ARMAPart method*), 6
`buildAttributes()` (*PMML44Super.ArrayType method*), 22
`buildAttributes()` (*PMML44Super.AssociationModel method*), 23
`buildAttributes()` (*PMML44Super.AssociationRule method*), 25
`buildAttributes()` (*PMML44Super.Attribute method*), 27
`buildAttributes()` (*PMML44Super.BaseCumHazardTables method*), 28
`buildAttributes()` (*PMML44Super.Baseline method*), 29
`buildAttributes()` (*PMML44Super.BaselineCell method*), 30
`buildAttributes()` (*PMML44Super.BaselineModel method*), 30
`buildAttributes()` (*PMML44Super.BaselineStratum method*), 32
`buildAttributes()` (*PMML44Super.BayesianNetworkModel method*), 35
`buildAttributes()` (*PMML44Super.BayesianNetworkNodes method*), 37
`buildAttributes()` (*PMML44Super.BayesInput method*), 33
`buildAttributes()` (*PMML44Super.BayesInputs method*), 34
`buildAttributes()` (*PMML44Super.BayesOutput method*), 34
`buildAttributes()` (*PMML44Super.binarySimilarity method*), 298
`buildAttributes()` (*PMML44Super.BlockIndicator method*), 37
`buildAttributes()` (*PMML44Super.BoundaryValueMeans method*), 38
`buildAttributes()` (*PMML44Super.BoundaryValues method*),

39	59
<code>buildAttributes()</code> (<i>PMML44Super.CategoricalPredictor</i> method), 41	<code>buildAttributes()</code> (<i>PMML44Super.ConsequentSequence</i> method), 60
<code>buildAttributes()</code> (<i>PMML44Super.Categories</i> method), 41	<code>buildAttributes()</code> (<i>PMML44Super.Constant</i> method), 61
<code>buildAttributes()</code> (<i>PMML44Super.Category</i> method), 42	<code>buildAttributes()</code> (<i>PMML44Super.Constraints</i> method), 61
<code>buildAttributes()</code> (<i>PMML44Super.Characteristic</i> method), 43	<code>buildAttributes()</code> (<i>PMML44Super.ContinuousConditionalProbability</i> method), 64
<code>buildAttributes()</code> (<i>PMML44Super.Characteristics</i> method), 44	<code>buildAttributes()</code> (<i>PMML44Super.ContinuousDistribution</i> method), 65
<code>buildAttributes()</code> (<i>PMML44Super.chebychev</i> method), 299	<code>buildAttributes()</code> (<i>PMML44Super.ContinuousNode</i> method), 66
<code>buildAttributes()</code> (<i>PMML44Super.ChildParent</i> method), 44	<code>buildAttributes()</code> (<i>PMML44Super.ContStats</i> method), 63
<code>buildAttributes()</code> (<i>PMML44Super.cityBlock</i> method), 300	<code>buildAttributes()</code> (<i>PMML44Super.CorrelationFields</i> method), 67
<code>buildAttributes()</code> (<i>PMML44Super.ClassLabels</i> method), 46	<code>buildAttributes()</code> (<i>PMML44Super.CorrelationMethods</i> method), 68
<code>buildAttributes()</code> (<i>PMML44Super.Cluster</i> method), 46	<code>buildAttributes()</code> (<i>PMML44Super.Correlations</i> method), 69
<code>buildAttributes()</code> (<i>PMML44Super.ClusteringField</i> method), 47	<code>buildAttributes()</code> (<i>PMML44Super.CorrelationValues</i> method), 68
<code>buildAttributes()</code> (<i>PMML44Super.ClusteringModel</i> method), 48	<code>buildAttributes()</code> (<i>PMML44Super.COUNT_TABLE_TYPE</i> method), 40
<code>buildAttributes()</code> (<i>PMML44Super.ClusteringModelQuality</i> method), 50	<code>buildAttributes()</code> (<i>PMML44Super.Counts</i> method), 70
<code>buildAttributes()</code> (<i>PMML44Super.Coefficient</i> method), 51	<code>buildAttributes()</code> (<i>PMML44Super.Covariances</i> method), 71
<code>buildAttributes()</code> (<i>PMML44Super.Coefficients</i> method), 52	<code>buildAttributes()</code> (<i>PMML44Super.CovariateList</i> method), 71
<code>buildAttributes()</code> (<i>PMML44Super.ComparisonMeasure</i> method), 52	<code>buildAttributes()</code> (<i>PMML44Super.DataDictionary</i> method), 72
<code>buildAttributes()</code> (<i>PMML44Super.Comparisons</i> method), 54	<code>buildAttributes()</code> (<i>PMML44Super.DataField</i> method), 73
<code>buildAttributes()</code> (<i>PMML44Super.ComplexPartialScore</i> method), 55	<code>buildAttributes()</code> (<i>PMML44Super.Decision</i> method), 74
<code>buildAttributes()</code> (<i>PMML44Super.CompoundPredicate</i> method), 56	<code>buildAttributes()</code> (<i>PMML44Super.Decisions</i> method), 77
<code>buildAttributes()</code> (<i>PMML44Super.CompoundRule</i> method), 57	<code>buildAttributes()</code> (<i>PMML44Super.DecisionTree</i> method), 75
<code>buildAttributes()</code> (<i>PMML44Super.Con</i> method), 58	<code>buildAttributes()</code> (<i>PMML44Super.DeepNetwork</i> method), 78
<code>buildAttributes()</code> (<i>PMML44Super.ConfusionMatrix</i> method),	<code>buildAttributes()</code> (<i>PMML44Super.DefineFunction</i> method), 79
	<code>buildAttributes()</code> (<i>PMML44Super.Delimiter</i>

`method)`, 81
`buildAttributes()` (*PMML44Super.Denominator* `method)`, 82
`buildAttributes()` (*PMML44Super.DerivedField* `method)`, 83
`buildAttributes()` (*PMML44Super.DiscreteConditionalProbability* `method)`, 85
`buildAttributes()` (*PMML44Super.DiscreteNode* `method)`, 86
`buildAttributes()` (*PMML44Super.Discretize* `method)`, 87
`buildAttributes()` (*PMML44Super.DiscretizeBin* `method)`, 88
`buildAttributes()` (*PMML44Super.DiscrStats* `method)`, 84
`buildAttributes()` (*PMML44Super.DocumentTermMatrix* `method)`, 89
`buildAttributes()` (*PMML44Super.DynamicRegressor* `method)`, 90
`buildAttributes()` (*PMML44Super.euclidean* `method)`, 300
`buildAttributes()` (*PMML44Super.EventValues* `method)`, 91
`buildAttributes()` (*PMML44Super.ExponentialSmoothing* `method)`, 92
`buildAttributes()` (*PMML44Super.Extension* `method)`, 93
`buildAttributes()` (*PMML44Super.FactorList* `method)`, 93
`buildAttributes()` (*PMML44Super.False_* `method)`, 94
`buildAttributes()` (*PMML44Super.FieldColumnPair* `method)`, 95
`buildAttributes()` (*PMML44Super.FieldRef* `method)`, 95
`buildAttributes()` (*PMML44Super.FieldValue* `method)`, 96
`buildAttributes()` (*PMML44Super.FieldValueCount* `method)`, 97
`buildAttributes()` (*PMML44Super.FinalNoise* `method)`, 98
`buildAttributes()` (*PMML44Super.FinalNu* `method)`, 98
`buildAttributes()` (*PMML44Super.FinalOmega* `method)`, 99
`buildAttributes()` (*PMML44Super.FinalPredictedNoise* `method)`, 99
`buildAttributes()` (*PMML44Super.FinalStateVector* `method)`, 100
`buildAttributes()` (*PMML44Super.FinalTheta* `method)`, 101
`buildAttributes()` (*PMML44Super.GARCH* `method)`, 101
`buildAttributes()` (*PMML44Super.GARCHPart* `method)`, 102
`buildAttributes()` (*PMML44Super.GaussianDistribution* `method)`, 103
`buildAttributes()` (*PMML44Super.GaussianProcessModel* `method)`, 104
`buildAttributes()` (*PMML44Super.GeneralizedExponentialKernel* `method)`, 109
`buildAttributes()` (*PMML44Super.GeneralRegressionModel* `method)`, 106
`buildAttributes()` (*PMML44Super.Header* `method)`, 110
`buildAttributes()` (*PMML44Super.HVector* `method)`, 110
`buildAttributes()` (*PMML44Super.Indices* `method)`, 113
`buildAttributes()` (*PMML44Super.InlineTable* `method)`, 113
`buildAttributes()` (*PMML44Super.InstanceField* `method)`, 114
`buildAttributes()` (*PMML44Super.InstanceFields* `method)`, 115
`buildAttributes()` (*PMML44Super.INT_Entries* `method)`, 111
`buildAttributes()` (*PMML44Super.INT_SparseArray* `method)`, 112
`buildAttributes()` (*PMML44Super.Interval* `method)`, 115
`buildAttributes()` (*PMML44Super.Item* `method)`, 116
`buildAttributes()` (*PMML44Super.ItemRef* `method)`, 117
`buildAttributes()` (*PMML44Super.Itemset* `method)`, 118
`buildAttributes()` (*PMML44Super.jaccard* `method)`, 301
`buildAttributes()` (*PMML44Super.KalmanState* `method)`, 120
`buildAttributes()` (*PMML44Super.KNNInput* `method)`, 119
`buildAttributes()` (*PMML44Super.KNNInputs* `method)`, 119

`buildAttributes()` (*PMML44Super.KohonenMap method*), 121
`buildAttributes()` (*PMML44Super.Lag method*), 122
`buildAttributes()` (*PMML44Super.Lambda method*), 122
`buildAttributes()` (*PMML44Super.LayerBias method*), 123
`buildAttributes()` (*PMML44Super.LayerParameters method*), 124
`buildAttributes()` (*PMML44Super.LayerWeights method*), 128
`buildAttributes()` (*PMML44Super.Level method*), 129
`buildAttributes()` (*PMML44Super.LiftData method*), 129
`buildAttributes()` (*PMML44Super.LiftGraph method*), 130
`buildAttributes()` (*PMML44Super.LinearKernelType method*), 131
`buildAttributes()` (*PMML44Super.LinearNorm method*), 132
`buildAttributes()` (*PMML44Super.LocalTransformations method*), 133
`buildAttributes()` (*PMML44Super.LognormalDistributionForBN method*), 133
`buildAttributes()` (*PMML44Super.Losses method*), 134
`buildAttributes()` (*PMML44Super.Lower method*), 135
`buildAttributes()` (*PMML44Super.MA method*), 136
`buildAttributes()` (*PMML44Super.MACoefficients method*), 137
`buildAttributes()` (*PMML44Super.MapValues method*), 137
`buildAttributes()` (*PMML44Super.MatCell method*), 138
`buildAttributes()` (*PMML44Super.Matrix method*), 139
`buildAttributes()` (*PMML44Super.MaximumLikelihoodStat method*), 140
`buildAttributes()` (*PMML44Super.Mean method*), 141
`buildAttributes()` (*PMML44Super.MeasurementMatrix method*), 142
`buildAttributes()` (*PMML44Super.Metrics method*), 143
`buildAttributes()` (*PMML44Super.MiningBuildTask method*), 144
`buildAttributes()` (*PMML44Super.MiningField method*), 144
`buildAttributes()` (*PMML44Super.MiningModel method*), 146
`buildAttributes()` (*PMML44Super.MiningSchema method*), 147
`buildAttributes()` (*PMML44Super.minkowski method*), 301
`buildAttributes()` (*PMML44Super.MissingValueWeights method*), 148
`buildAttributes()` (*PMML44Super.ModelExplanation method*), 149
`buildAttributes()` (*PMML44Super.ModelLiftGraph method*), 150
`buildAttributes()` (*PMML44Super.ModelStats method*), 150
`buildAttributes()` (*PMML44Super.ModelVerification method*), 151
`buildAttributes()` (*PMML44Super.MultivariateStat method*), 152
`buildAttributes()` (*PMML44Super.MultivariateStats method*), 154
`buildAttributes()` (*PMML44Super.Nadam method*), 154
`buildAttributes()` (*PMML44Super.NaiveBayesModel method*), 155
`buildAttributes()` (*PMML44Super.NearestNeighborModel method*), 157
`buildAttributes()` (*PMML44Super.NetworkLayer method*), 159
`buildAttributes()` (*PMML44Super.NeuralInput method*), 160
`buildAttributes()` (*PMML44Super.NeuralInputs method*), 161
`buildAttributes()` (*PMML44Super.NeuralLayer method*), 162
`buildAttributes()` (*PMML44Super.NeuralNetwork method*), 163
`buildAttributes()` (*PMML44Super.NeuralOutput method*), 165

<code>buildAttributes()</code> (<i>PMML44Super.NeuralOutputs</i> method), 166	<code>buildAttributes()</code> (<i>PMML44Super.PartitionFieldStats</i> method), 196
<code>buildAttributes()</code> (<i>PMML44Super.Neuron</i> method), 166	<code>buildAttributes()</code> (<i>PMML44Super.PastVariances</i> method), 197
<code>buildAttributes()</code> (<i>PMML44Super.Node</i> method), 167	<code>buildAttributes()</code> (<i>PMML44Super.PCell</i> method), 182
<code>buildAttributes()</code> (<i>PMML44Super.NonseasonalComponent</i> method), 169	<code>buildAttributes()</code> (<i>PMML44Super.PCovCell</i> method), 183
<code>buildAttributes()</code> (<i>PMML44Super.NonseasonalFactor</i> method), 170	<code>buildAttributes()</code> (<i>PMML44Super.PCovMatrix</i> method), 184
<code>buildAttributes()</code> (<i>PMML44Super.NormalDistributionForBN</i> method), 173	<code>buildAttributes()</code> (<i>PMML44Super.PMML</i> method), 186
<code>buildAttributes()</code> (<i>PMML44Super.NormContinuous</i> method), 171	<code>buildAttributes()</code> (<i>PMML44Super.PoissonDistribution</i> method), 198
<code>buildAttributes()</code> (<i>PMML44Super.NormDiscrete</i> method), 172	<code>buildAttributes()</code> (<i>PMML44Super.PolynomialKernelType</i> method), 198
<code>buildAttributes()</code> (<i>PMML44Super.Numerator</i> method), 173	<code>buildAttributes()</code> (<i>PMML44Super.PPCell</i> method), 189
<code>buildAttributes()</code> (<i>PMML44Super.NumericInfo</i> method), 174	<code>buildAttributes()</code> (<i>PMML44Super.PPMatrix</i> method), 190
<code>buildAttributes()</code> (<i>PMML44Super.NumericPredictor</i> method), 175	<code>buildAttributes()</code> (<i>PMML44Super.PredictiveModelQuality</i> method), 199
<code>buildAttributes()</code> (<i>PMML44Super.Optimizers</i> method), 176	<code>buildAttributes()</code> (<i>PMML44Super.Predictor</i> method), 201
<code>buildAttributes()</code> (<i>PMML44Super.OptimumLiftGraph</i> method), 178	<code>buildAttributes()</code> (<i>PMML44Super.PredictorTerm</i> method), 202
<code>buildAttributes()</code> (<i>PMML44Super.OutlierEffect</i> method), 178	<code>buildAttributes()</code> (<i>PMML44Super.PsiVector</i> method), 203
<code>buildAttributes()</code> (<i>PMML44Super.Output</i> method), 179	<code>buildAttributes()</code> (<i>PMML44Super.Quantile</i> method), 204
<code>buildAttributes()</code> (<i>PMML44Super.OutputField</i> method), 180	<code>buildAttributes()</code> (<i>PMML44Super.RadialBasisKernel</i> method), 208
<code>buildAttributes()</code> (<i>PMML44Super.PairCounts</i> method), 191	<code>buildAttributes()</code> (<i>PMML44Super.RadialBasisKernelType</i> method), 209
<code>buildAttributes()</code> (<i>PMML44Super.Parameter</i> method), 192	<code>buildAttributes()</code> (<i>PMML44Super.RandomLiftGraph</i> method), 210
<code>buildAttributes()</code> (<i>PMML44Super.ParameterField</i> method), 193	<code>buildAttributes()</code> (<i>PMML44Super.REAL_Entries</i> method), 205
<code>buildAttributes()</code> (<i>PMML44Super.ParameterList</i> method), 194	<code>buildAttributes()</code> (<i>PMML44Super.REAL_SparseArray</i> method), 205
<code>buildAttributes()</code> (<i>PMML44Super.ParamMatrix</i> method), 191	<code>buildAttributes()</code> (<i>PMML44Super.Regression</i> method), 211
<code>buildAttributes()</code> (<i>PMML44Super.ParentValue</i> method), 194	<code>buildAttributes()</code> (<i>PMML44Super.RegressionModel</i> method), 212
<code>buildAttributes()</code> (<i>PMML44Super.Partition</i> method), 195	<code>buildAttributes()</code> (<i>PMML44Super.RegressionTable</i> method),
<code>buildAttributes()</code>	

- 214
 buildAttributes() (PMML44Super.RegressorValues method), 215
- buildAttributes() (PMML44Super.Residuals method), 217
- buildAttributes() (PMML44Super.ResidualSquareCoefficients method), 216
- buildAttributes() (PMML44Super.ResultField method), 217
- buildAttributes() (PMML44Super.RMSprop method), 206
- buildAttributes() (PMML44Super.ROC method), 207
- buildAttributes() (PMML44Super.ROCGraph method), 208
- buildAttributes() (PMML44Super.row method), 302
- buildAttributes() (PMML44Super.RuleSelectionMethod method), 218
- buildAttributes() (PMML44Super.RuleSet method), 219
- buildAttributes() (PMML44Super.RuleSetModel method), 220
- buildAttributes() (PMML44Super.Scorecard method), 224
- buildAttributes() (PMML44Super.ScoreDistribution method), 223
- buildAttributes() (PMML44Super.SeasonalComponent method), 225
- buildAttributes() (PMML44Super.SeasonalFactor method), 226
- buildAttributes() (PMML44Super.Seasonality_ExpoSmooth method), 227
- buildAttributes() (PMML44Super.SeasonalTrendDecomposition method), 227
- buildAttributes() (PMML44Super.Segment method), 228
- buildAttributes() (PMML44Super.Segmentation method), 231
- buildAttributes() (PMML44Super.Sequence method), 232
- buildAttributes() (PMML44Super.SequenceModel method), 233
- buildAttributes() (PMML44Super.SequenceReference method), 235
- buildAttributes() (PMML44Super.SequenceRule method), 236
- buildAttributes() (PMML44Super.SetPredicate method), 237
- buildAttributes() (PMML44Super.SetReference method), 238
- buildAttributes() (PMML44Super.SGD method), 222
- buildAttributes() (PMML44Super.SigmoidKernelType method), 239
- buildAttributes() (PMML44Super.simpleMatching method), 303
- buildAttributes() (PMML44Super.SimplePredicate method), 239
- buildAttributes() (PMML44Super.SimpleRule method), 240
- buildAttributes() (PMML44Super.SimpleSetPredicate method), 242
- buildAttributes() (PMML44Super.SpectralAnalysis method), 242
- buildAttributes() (PMML44Super.squaredEuclidean method), 303
- buildAttributes() (PMML44Super.StateSpaceModel method), 243
- buildAttributes() (PMML44Super.StateVector method), 244
- buildAttributes() (PMML44Super.SupportVector method), 245
- buildAttributes() (PMML44Super.SupportVectorMachine method), 245
- buildAttributes() (PMML44Super.SupportVectorMachineModel method), 247
- buildAttributes() (PMML44Super.SupportVectors method), 249
- buildAttributes() (PMML44Super.TableLocator method), 250
- buildAttributes() (PMML44Super.tanimoto method), 304
- buildAttributes() (PMML44Super.Target method), 250
- buildAttributes() (PMML44Super.Targets method), 256
- buildAttributes() (PMML44Super.TargetValue method), 256

method), 251

buildAttributes() (PMML44Super.TargetValueCount method), 252

buildAttributes() (PMML44Super.TargetValueCounts method), 253

buildAttributes() (PMML44Super.TargetValueStat method), 254

buildAttributes() (PMML44Super.TargetValueStats method), 255

buildAttributes() (PMML44Super.Taxonomy method), 256

buildAttributes() (PMML44Super.TestDistributions method), 257

buildAttributes() (PMML44Super.TextCorpus method), 258

buildAttributes() (PMML44Super.TextDictionary method), 259

buildAttributes() (PMML44Super.TextDocument method), 260

buildAttributes() (PMML44Super.TextIndex method), 261

buildAttributes() (PMML44Super.TextIndexNormalization method), 262

buildAttributes() (PMML44Super.TextModel method), 264

buildAttributes() (PMML44Super.TextModelNormalization method), 265

buildAttributes() (PMML44Super.TextModelSimiliarity method), 266

buildAttributes() (PMML44Super.Theta method), 267

buildAttributes() (PMML44Super.ThetaRecursionState method), 268

buildAttributes() (PMML44Super.Time method), 268

buildAttributes() (PMML44Super.TimeAnchor method), 269

buildAttributes() (PMML44Super.TimeCycle method), 270

buildAttributes() (PMML44Super.TimeException method), 271

buildAttributes() (PMML44Super.TimeSeries method), 272

buildAttributes() (PMML44Super.TimeSeriesModel method), 273

buildAttributes() (PMML44Super.Timestamp method), 275

buildAttributes() (PMML44Super.TimeValue method), 274

buildAttributes() (PMML44Super.TrainingInstances method), 276

buildAttributes() (PMML44Super.TrainingParameters method), 277

buildAttributes() (PMML44Super.TransferFunctionValues method), 278

buildAttributes() (PMML44Super.TransformationDictionary method), 279

buildAttributes() (PMML44Super.TransitionMatrix method), 279

buildAttributes() (PMML44Super.TreeModel method), 280

buildAttributes() (PMML44Super.Trend_ExpoSmooth method), 282

buildAttributes() (PMML44Super.TrendCoefficients method), 282

buildAttributes() (PMML44Super.TriangularDistributionForBN method), 283

buildAttributes() (PMML44Super.True_ method), 284

buildAttributes() (PMML44Super.UniformDistribution method), 285

buildAttributes() (PMML44Super.UniformDistributionForBN method), 285

buildAttributes() (PMML44Super.UnivariateStats method), 286

buildAttributes() (PMML44Super.Upper method), 287

buildAttributes() (PMML44Super.Value method), 288

buildAttributes() (PMML44Super.ValueProbability method), 289

buildAttributes() (PMML44Super.VariableWeight method), 290

buildAttributes() (PMML44Super.Variance method), 291

buildAttributes ()	5	
(PMML44Super.VarianceCoefficients method),	buildChildren ()	(PMML44Super.ARMAPart
292	method), 6	
buildAttributes ()	buildChildren ()	(PMML44Super.ArrayType
(PMML44Super.VectorDictionary method),	method), 22	
292	buildChildren ()	(PMML44Super.AssociationModel
buildAttributes () (PMML44Super.VectorFields	method), 23	
method), 293	buildChildren ()	(PMML44Super.AssociationRule
buildAttributes () (PMML44Super.VectorInstance	method), 25	
method), 294	buildChildren ()	(PMML44Super.Attribute
buildAttributes ()	method), 27	
(PMML44Super.VerificationField method),	buildChildren ()	(PMML44Super.BaseCumHazardTables
295	method), 28	
buildAttributes ()	buildChildren ()	(PMML44Super.Baseline
(PMML44Super.VerificationFields method),	method), 29	
296	buildChildren ()	(PMML44Super.BaselineCell
buildAttributes () (PMML44Super.XCoordinates	method), 30	
method), 297	buildChildren ()	(PMML44Super.BaselineModel
buildAttributes () (PMML44Super.YCoordinates	method), 30	
method), 297	buildChildren ()	(PMML44Super.BaselineStratum
buildChildren () (PMML44Super.AbsoluteExponentialKernel	method), 32	
method), 7	buildChildren ()	(PMML44Super.BayesianNetworkModel
buildChildren ()	method), 35	
(PMML44Super.Adadelta	buildChildren ()	(PMML44Super.BayesianNetworkNodes
method), 8	method), 37	
buildChildren ()	buildChildren ()	(PMML44Super.BayesInput
(PMML44Super.Adagrad	method), 33	
method), 9	buildChildren ()	(PMML44Super.BayesInputs
buildChildren () (PMML44Super.Adam method),	method), 34	
10	buildChildren ()	(PMML44Super.BayesOutput
buildChildren () (PMML44Super.Adamax method),	method), 34	
10	buildChildren ()	(PMML44Super.binarySimilarity
buildChildren ()	method), 298	
(PMML44Super.Aggregate	buildChildren ()	(PMML44Super.BlockIndicator
method), 11	method), 38	
buildChildren ()	buildChildren ()	(PMML44Super.BoundaryValueMeans
(PMML44Super.Alternate	method), 38	
method), 12	buildChildren ()	(PMML44Super.BoundaryValues
buildChildren ()	method), 39	
(PMML44Super.Annotation	buildChildren ()	(PMML44Super.CategoricalPredictor
method), 13	method), 41	
buildChildren () (PMML44Super.AnomalyDetectionModel	buildChildren ()	(PMML44Super.Categories
method), 14	method), 41	
buildChildren () (PMML44Super.Anova method),	buildChildren ()	(PMML44Super.Category
16	method), 42	
buildChildren ()	buildChildren ()	(PMML44Super.Characteristic
(PMML44Super.AnovaRow	method), 43	
method), 17	buildChildren ()	(PMML44Super.Characteristics
buildChildren () (PMML44Super.AntecedentSequence	method), 44	
method), 18	buildChildren ()	(PMML44Super.chebychev
buildChildren ()	method), 299	
(PMML44Super.AnyDistribution	buildChildren ()	(PMML44Super.ChildParent
method), 19	method), 45	
buildChildren ()	buildChildren ()	(PMML44Super.cityBlock
(PMML44Super.Application		
method), 19		
buildChildren () (PMML44Super.Apply method),		
21		
buildChildren () (PMML44Super.AR method), 3		
buildChildren () (PMML44Super.ARDSquaredExponentialKernel		
method), 4		
buildChildren () (PMML44Super.ARIMA method),		

<i>method</i>), 300		<i>buildChildren()</i> (PMML44Super.Covariances <i>method</i>), 71	
<i>buildChildren()</i> (PMML44Super.ClassLabels <i>method</i>), 46		<i>buildChildren()</i> (PMML44Super.CovariateList <i>method</i>), 71	
<i>buildChildren()</i> (PMML44Super.Cluster <i>method</i>), 46		<i>buildChildren()</i> (PMML44Super.DataDictionary <i>method</i>), 72	
<i>buildChildren()</i> (PMML44Super.ClusteringField <i>method</i>), 47		<i>buildChildren()</i> (PMML44Super.DataField <i>method</i>), 73	
<i>buildChildren()</i> (PMML44Super.ClusteringModel <i>method</i>), 48		<i>buildChildren()</i> (PMML44Super.Decision <i>method</i>), 74	
<i>buildChildren()</i> (PMML44Super.ClusteringModelQuality <i>method</i>), 50		<i>buildChildren()</i> (PMML44Super.Decisions <i>method</i>), 77	
<i>buildChildren()</i> (PMML44Super.Coefficient <i>method</i>), 51		<i>buildChildren()</i> (PMML44Super.DecisionTree <i>method</i>), 75	
<i>buildChildren()</i> (PMML44Super.Coefficients <i>method</i>), 52		<i>buildChildren()</i> (PMML44Super.DeepNetwork <i>method</i>), 78	
<i>buildChildren()</i> (PMML44Super.ComparisonMeasure <i>method</i>), 53		<i>buildChildren()</i> (PMML44Super.DefineFunction <i>method</i>), 79	
<i>buildChildren()</i> (PMML44Super.Comparisons <i>method</i>), 54		<i>buildChildren()</i> (PMML44Super.Delimiter <i>method</i>), 81	
<i>buildChildren()</i> (PMML44Super.ComplexPartialScore <i>method</i>), 55		<i>buildChildren()</i> (PMML44Super.Denominator <i>method</i>), 82	
<i>buildChildren()</i> (PMML44Super.CompoundPredicate <i>method</i>), 56		<i>buildChildren()</i> (PMML44Super.DerivedField <i>method</i>), 83	
<i>buildChildren()</i> (PMML44Super.CompoundRule <i>method</i>), 57		<i>buildChildren()</i> (PMML44Super.DiscreteConditionalProbability <i>method</i>), 85	
<i>buildChildren()</i> (PMML44Super.Con <i>method</i>), 58		<i>buildChildren()</i> (PMML44Super.DiscreteNode <i>method</i>), 86	
<i>buildChildren()</i> (PMML44Super.ConfusionMatrix <i>method</i>), 59		<i>buildChildren()</i> (PMML44Super.Discretize <i>method</i>), 87	
<i>buildChildren()</i> (PMML44Super.ConsequentSequence <i>method</i>), 60		<i>buildChildren()</i> (PMML44Super.DiscretizeBin <i>method</i>), 88	
<i>buildChildren()</i> (PMML44Super.Constant <i>method</i>), 61		<i>buildChildren()</i> (PMML44Super.DiscrStats <i>method</i>), 84	
<i>buildChildren()</i> (PMML44Super.Constraints <i>method</i>), 61		<i>buildChildren()</i> (PMML44Super.DocumentTermMatrix <i>method</i>), 89	
<i>buildChildren()</i> (PMML44Super.ContinuousConditionalProbability <i>method</i>), 64		<i>buildChildren()</i> (PMML44Super.DynamicRegressor <i>method</i>), 90	
<i>buildChildren()</i> (PMML44Super.ContinuousDistribution <i>method</i>), 65		<i>buildChildren()</i> (PMML44Super.euclidean <i>method</i>), 300	
<i>buildChildren()</i> (PMML44Super.ContinuousNode <i>method</i>), 66		<i>buildChildren()</i> (PMML44Super.EventValues <i>method</i>), 91	
<i>buildChildren()</i> (PMML44Super.ContStats <i>method</i>), 63		<i>buildChildren()</i> (PMML44Super.ExponentialSmoothing <i>method</i>), 92	
<i>buildChildren()</i> (PMML44Super.CorrelationFields <i>method</i>), 67		<i>buildChildren()</i> (PMML44Super.Extension <i>method</i>), 93	
<i>buildChildren()</i> (PMML44Super.CorrelationMethods <i>method</i>), 68		<i>buildChildren()</i> (PMML44Super.FactorList <i>method</i>), 93	
<i>buildChildren()</i> (PMML44Super.Correlations <i>method</i>), 69		<i>buildChildren()</i> (PMML44Super.False_ <i>method</i>), 94	
<i>buildChildren()</i> (PMML44Super.CorrelationValues <i>method</i>), 68		<i>buildChildren()</i> (PMML44Super.FieldColumnPair <i>method</i>), 95	
<i>buildChildren()</i> (PMML44Super.COUNT_TABLE_TYPE <i>method</i>), 40		<i>buildChildren()</i> (PMML44Super.FieldRef <i>method</i>), 95	
<i>buildChildren()</i> (PMML44Super.Counts <i>method</i>), 70			

<code>buildChildren()</code> (PMML44Super.FieldValue method), 96	<code>buildChildren()</code> (PMML44Super.KalmanState method), 120
<code>buildChildren()</code> (PMML44Super.FieldValueCount method), 97	<code>buildChildren()</code> (PMML44Super.KNNInput method), 119
<code>buildChildren()</code> (PMML44Super.FinalNoise method), 98	<code>buildChildren()</code> (PMML44Super.KNNInputs method), 119
<code>buildChildren()</code> (PMML44Super.FinalNu method), 98	<code>buildChildren()</code> (PMML44Super.KohonenMap method), 121
<code>buildChildren()</code> (PMML44Super.FinalOmega method), 99	<code>buildChildren()</code> (PMML44Super.Lag method), 122
<code>buildChildren()</code> (PMML44Super.FinalPredictedNoise method), 99	<code>buildChildren()</code> (PMML44Super.Lambda method), 122
<code>buildChildren()</code> (PMML44Super.FinalStateVector method), 100	<code>buildChildren()</code> (PMML44Super.LayerBias method), 123
<code>buildChildren()</code> (PMML44Super.FinalTheta method), 101	<code>buildChildren()</code> (PMML44Super.LayerParameters method), 124
<code>buildChildren()</code> (PMML44Super.GARCH method), 101	<code>buildChildren()</code> (PMML44Super.LayerWeights method), 128
<code>buildChildren()</code> (PMML44Super.GARCHPart method), 102	<code>buildChildren()</code> (PMML44Super.Level method), 129
<code>buildChildren()</code> (PMML44Super.GaussianDistribution method), 103	<code>buildChildren()</code> (PMML44Super.LiftData method), 130
<code>buildChildren()</code> (PMML44Super.GaussianProcessModel method), 104	<code>buildChildren()</code> (PMML44Super.LiftGraph method), 130
<code>buildChildren()</code> (PMML44Super.GeneralizedExponentialKernel method), 109	<code>buildChildren()</code> (PMML44Super.LinearKernelType method), 131
<code>buildChildren()</code> (PMML44Super.GeneralRegressionModel method), 106	<code>buildChildren()</code> (PMML44Super.LinearNorm method), 132
<code>buildChildren()</code> (PMML44Super.Header method), 110	<code>buildChildren()</code> (PMML44Super.LocalTransformations method), 133
<code>buildChildren()</code> (PMML44Super.HVector method), 110	<code>buildChildren()</code> (PMML44Super.LognormalDistributionForBN method), 133
<code>buildChildren()</code> (PMML44Super.Indices method), 113	<code>buildChildren()</code> (PMML44Super.Losses method), 134
<code>buildChildren()</code> (PMML44Super.InlineTable method), 113	<code>buildChildren()</code> (PMML44Super.Lower method), 135
<code>buildChildren()</code> (PMML44Super.InstanceField method), 114	<code>buildChildren()</code> (PMML44Super.MA method), 136
<code>buildChildren()</code> (PMML44Super.InstanceFields method), 115	<code>buildChildren()</code> (PMML44Super.MACoefficients method), 137
<code>buildChildren()</code> (PMML44Super.INT_Entries method), 111	<code>buildChildren()</code> (PMML44Super.MapValues method), 137
<code>buildChildren()</code> (PMML44Super.INT_SparseArray method), 112	<code>buildChildren()</code> (PMML44Super.MatCell method), 138
<code>buildChildren()</code> (PMML44Super.Interval method), 115	<code>buildChildren()</code> (PMML44Super.Matrix method), 139
<code>buildChildren()</code> (PMML44Super.Item method), 116	<code>buildChildren()</code> (PMML44Super.MaximumLikelihoodStat method), 140
<code>buildChildren()</code> (PMML44Super.ItemRef method), 117	<code>buildChildren()</code> (PMML44Super.Mean method), 141
<code>buildChildren()</code> (PMML44Super.Itemset method), 118	<code>buildChildren()</code> (PMML44Super.MeasurementMatrix method), 142
<code>buildChildren()</code> (PMML44Super.jaccard method), 301	<code>buildChildren()</code> (PMML44Super.Metrics method), 143
	<code>buildChildren()</code> (PMML44Super.MiningBuildTask method), 144

<code>buildChildren()</code> (PMML44Super.MiningField method), 144	<code>buildChildren()</code> (PMML44Super.NormDiscrete method), 172
<code>buildChildren()</code> (PMML44Super.MiningModel method), 146	<code>buildChildren()</code> (PMML44Super.Numerator method), 173
<code>buildChildren()</code> (PMML44Super.MiningSchema method), 147	<code>buildChildren()</code> (PMML44Super.NumericInfo method), 174
<code>buildChildren()</code> (PMML44Super.minkowski method), 301	<code>buildChildren()</code> (PMML44Super.NumericPredictor method), 175
<code>buildChildren()</code> (PMML44Super.MissingValueWeights method), 148	<code>buildChildren()</code> (PMML44Super.Optimizers method), 176
<code>buildChildren()</code> (PMML44Super.ModelExplanation method), 149	<code>buildChildren()</code> (PMML44Super.OptimumLiftGraph method), 178
<code>buildChildren()</code> (PMML44Super.ModelLiftGraph method), 150	<code>buildChildren()</code> (PMML44Super.OutlierEffect method), 178
<code>buildChildren()</code> (PMML44Super.ModelStats method), 150	<code>buildChildren()</code> (PMML44Super.Output method), 179
<code>buildChildren()</code> (PMML44Super.ModelVerification method), 151	<code>buildChildren()</code> (PMML44Super.OutputField method), 180
<code>buildChildren()</code> (PMML44Super.MultivariateStat method), 152	<code>buildChildren()</code> (PMML44Super.PairCounts method), 191
<code>buildChildren()</code> (PMML44Super.MultivariateStats method), 154	<code>buildChildren()</code> (PMML44Super.Parameter method), 192
<code>buildChildren()</code> (PMML44Super.Nadam method), 155	<code>buildChildren()</code> (PMML44Super.ParameterField method), 193
<code>buildChildren()</code> (PMML44Super.NaiveBayesModel method), 155	<code>buildChildren()</code> (PMML44Super.ParameterList method), 194
<code>buildChildren()</code> (PMML44Super.NearestNeighborModel method), 157	<code>buildChildren()</code> (PMML44Super.ParamMatrix method), 191
<code>buildChildren()</code> (PMML44Super.NetworkLayer method), 159	<code>buildChildren()</code> (PMML44Super.ParentValue method), 194
<code>buildChildren()</code> (PMML44Super.NeuralInput method), 160	<code>buildChildren()</code> (PMML44Super.Partition method), 195
<code>buildChildren()</code> (PMML44Super.NeuralInputs method), 161	<code>buildChildren()</code> (PMML44Super.PartitionFieldStats method), 196
<code>buildChildren()</code> (PMML44Super.NeuralLayer method), 162	<code>buildChildren()</code> (PMML44Super.PastVariances method), 197
<code>buildChildren()</code> (PMML44Super.NeuralNetwork method), 163	<code>buildChildren()</code> (PMML44Super.PCell method), 183
<code>buildChildren()</code> (PMML44Super.NeuralOutput method), 165	<code>buildChildren()</code> (PMML44Super.PCovCell method), 183
<code>buildChildren()</code> (PMML44Super.NeuralOutputs method), 166	<code>buildChildren()</code> (PMML44Super.PCovMatrix method), 184
<code>buildChildren()</code> (PMML44Super.Neuron method), 166	<code>buildChildren()</code> (PMML44Super.PMML method), 186
<code>buildChildren()</code> (PMML44Super.Node method), 168	<code>buildChildren()</code> (PMML44Super.PoissonDistribution method), 198
<code>buildChildren()</code> (PMML44Super.NonseasonalComponent method), 169	<code>buildChildren()</code> (PMML44Super.PolynomialKernelType method), 198
<code>buildChildren()</code> (PMML44Super.NonseasonalFactor method), 170	<code>buildChildren()</code> (PMML44Super.PPCell method), 189
<code>buildChildren()</code> (PMML44Super.NormalDistributionForBN method), 173	<code>buildChildren()</code> (PMML44Super.PPMatrix method), 190
<code>buildChildren()</code> (PMML44Super.NormContinuous method), 171	<code>buildChildren()</code> (PMML44Super.PredictiveModelQuality method), 199

`buildChildren()` (PMML44Super.Predictor method), 227
`buildChildren()` (PMML44Super.PredictorTerm method), 202
`buildChildren()` (PMML44Super.PsiVector method), 203
`buildChildren()` (PMML44Super.Quantile method), 204
`buildChildren()` (PMML44Super.RadialBasisKernel method), 208
`buildChildren()` (PMML44Super.RadialBasisKernelType method), 209
`buildChildren()` (PMML44Super.RandomLiftGraph method), 210
`buildChildren()` (PMML44Super.REAL_Entries method), 205
`buildChildren()` (PMML44Super.REAL_SparseArray method), 205
`buildChildren()` (PMML44Super.Regression method), 211
`buildChildren()` (PMML44Super.RegressionModel method), 212
`buildChildren()` (PMML44Super.RegressionTable method), 214
`buildChildren()` (PMML44Super.RegressorValues method), 215
`buildChildren()` (PMML44Super.Residuals method), 217
`buildChildren()` (PMML44Super.ResidualSquareCoefficients method), 216
`buildChildren()` (PMML44Super.ResultField method), 217
`buildChildren()` (PMML44Super.RMSprop method), 206
`buildChildren()` (PMML44Super.ROC method), 207
`buildChildren()` (PMML44Super.ROCGraph method), 208
`buildChildren()` (PMML44Super.row method), 302
`buildChildren()` (PMML44Super.RuleSelectionMethod method), 218
`buildChildren()` (PMML44Super.RuleSet method), 219
`buildChildren()` (PMML44Super.RuleSetModel method), 220
`buildChildren()` (PMML44Super.Scorecard method), 224
`buildChildren()` (PMML44Super.ScoreDistribution method), 223
`buildChildren()` (PMML44Super.SeasonalComponent method), 225
`buildChildren()` (PMML44Super.SeasonalFactor method), 226
`buildChildren()` (PMML44Super.Seasonality_ExpoSmooth method), 227
`buildChildren()` (PMML44Super.SeasonalTrendDecomposition method), 227
`buildChildren()` (PMML44Super.Segment method), 228
`buildChildren()` (PMML44Super.Segmentation method), 231
`buildChildren()` (PMML44Super.Sequence method), 232
`buildChildren()` (PMML44Super.SequenceModel method), 233
`buildChildren()` (PMML44Super.SequenceReference method), 235
`buildChildren()` (PMML44Super.SequenceRule method), 236
`buildChildren()` (PMML44Super.SetPredicate method), 237
`buildChildren()` (PMML44Super.SetReference method), 238
`buildChildren()` (PMML44Super.SGD method), 222
`buildChildren()` (PMML44Super.SigmoidKernelType method), 239
`buildChildren()` (PMML44Super.simpleMatching method), 303
`buildChildren()` (PMML44Super.SimplePredicate method), 239
`buildChildren()` (PMML44Super.SimpleRule method), 240
`buildChildren()` (PMML44Super.SimpleSetPredicate method), 242
`buildChildren()` (PMML44Super.SpectralAnalysis method), 242
`buildChildren()` (PMML44Super.squaredEuclidean method), 303
`buildChildren()` (PMML44Super.StateSpaceModel method), 243
`buildChildren()` (PMML44Super.StateVector method), 244
`buildChildren()` (PMML44Super.SupportVector method), 245
`buildChildren()` (PMML44Super.SupportVectorMachine method), 246
`buildChildren()` (PMML44Super.SupportVectorMachineModel method), 247
`buildChildren()` (PMML44Super.SupportVectors method), 249
`buildChildren()` (PMML44Super.TableLocator method), 250
`buildChildren()` (PMML44Super.tanimoto method), 304
`buildChildren()` (PMML44Super.Target method), 250
`buildChildren()` (PMML44Super.Targets method), 250

256

`buildChildren()` (*PMML44Super.TargetValue method*), 252

`buildChildren()` (*PMML44Super.TargetValueCount method*), 252

`buildChildren()` (*PMML44Super.TargetValueCounts method*), 253

`buildChildren()` (*PMML44Super.TargetValueStat method*), 254

`buildChildren()` (*PMML44Super.TargetValueStats method*), 255

`buildChildren()` (*PMML44Super.Taxonomy method*), 256

`buildChildren()` (*PMML44Super.TestDistributions method*), 257

`buildChildren()` (*PMML44Super.TextCorpus method*), 258

`buildChildren()` (*PMML44Super.TextDictionary method*), 259

`buildChildren()` (*PMML44Super.TextDocument method*), 260

`buildChildren()` (*PMML44Super.TextIndex method*), 261

`buildChildren()` (*PMML44Super.TextIndexNormalization method*), 263

`buildChildren()` (*PMML44Super.TextModel method*), 264

`buildChildren()` (*PMML44Super.TextModelNormalization method*), 265

`buildChildren()` (*PMML44Super.TextModelSimilarity method*), 266

`buildChildren()` (*PMML44Super.Theta method*), 267

`buildChildren()` (*PMML44Super.ThetaRecursionState method*), 268

`buildChildren()` (*PMML44Super.Time method*), 268

`buildChildren()` (*PMML44Super.TimeAnchor method*), 269

`buildChildren()` (*PMML44Super.TimeCycle method*), 270

`buildChildren()` (*PMML44Super.TimeException method*), 271

`buildChildren()` (*PMML44Super.TimeSeries method*), 272

`buildChildren()` (*PMML44Super.TimeSeriesModel method*), 273

`buildChildren()` (*PMML44Super.Timestamp method*), 275

`buildChildren()` (*PMML44Super.TimeValue method*), 275

`buildChildren()` (*PMML44Super.TrainingInstances method*), 276

`buildChildren()` (*PMML44Super.TrainingParameters method*), 277

`buildChildren()` (*PMML44Super.TransferFunctionValues method*), 278

`buildChildren()` (*PMML44Super.TransformationDictionary method*), 279

`buildChildren()` (*PMML44Super.TransitionMatrix method*), 279

`buildChildren()` (*PMML44Super.TreeModel method*), 280

`buildChildren()` (*PMML44Super.Trend_ExpoSmooth method*), 282

`buildChildren()` (*PMML44Super.TrendCoefficients method*), 282

`buildChildren()` (*PMML44Super.TriangularDistributionForBN method*), 283

`buildChildren()` (*PMML44Super.True_ method*), 284

`buildChildren()` (*PMML44Super.UniformDistribution method*), 285

`buildChildren()` (*PMML44Super.UniformDistributionForBN method*), 285

`buildChildren()` (*PMML44Super.UnivariateStats method*), 286

`buildChildren()` (*PMML44Super.Upper method*), 287

`buildChildren()` (*PMML44Super.Value method*), 288

`buildChildren()` (*PMML44Super.ValueProbability method*), 289

`buildChildren()` (*PMML44Super.VariableWeight method*), 290

`buildChildren()` (*PMML44Super.Variance method*), 291

`buildChildren()` (*PMML44Super.VarianceCoefficients method*), 292

`buildChildren()` (*PMML44Super.VectorDictionary method*), 292

`buildChildren()` (*PMML44Super.VectorFields method*), 293

`buildChildren()` (*PMML44Super.VectorInstance method*), 294

`buildChildren()` (*PMML44Super.VerificationField method*), 295

`buildChildren()` (*PMML44Super.VerificationFields method*), 296

`buildChildren()` (*PMML44Super.XCoordinates method*), 297

`buildChildren()` (*PMML44Super.YCoordinates method*), 297

C

`CategoricalPredictor` (class in *PMML44Super*), 40

`Categories` (class in *PMML44Super*), 41

Category (class in *PMML44Super*), 42
 Characteristic (class in *PMML44Super*), 43
 Characteristics (class in *PMML44Super*), 44
 chebychev (class in *PMML44Super*), 299
 ChildParent (class in *PMML44Super*), 44
 cityBlock (class in *PMML44Super*), 300
 ClassLabels (class in *PMML44Super*), 45
 Cluster (class in *PMML44Super*), 46
 ClusteringField (class in *PMML44Super*), 47
 ClusteringModel (class in *PMML44Super*), 48
 ClusteringModelQuality (class in *PMML44Super*), 50
 Coefficient (class in *PMML44Super*), 51
 Coefficients (class in *PMML44Super*), 51
 ComparisonMeasure (class in *PMML44Super*), 52
 Comparisons (class in *PMML44Super*), 54
 ComplexPartialScore (class in *PMML44Super*), 54
 CompoundPredicate (class in *PMML44Super*), 56
 CompoundRule (class in *PMML44Super*), 57
 Con (class in *PMML44Super*), 58
 ConfusionMatrix (class in *PMML44Super*), 59
 ConsequentSequence (class in *PMML44Super*), 60
 Constant (class in *PMML44Super*), 61
 Constraints (class in *PMML44Super*), 61
 ContinuousConditionalProbability (class in *PMML44Super*), 64
 ContinuousDistribution (class in *PMML44Super*), 65
 ContinuousNode (class in *PMML44Super*), 66
 ContStats (class in *PMML44Super*), 63
 CorrelationFields (class in *PMML44Super*), 67
 CorrelationMethods (class in *PMML44Super*), 68
 Correlations (class in *PMML44Super*), 69
 CorrelationValues (class in *PMML44Super*), 68
 COUNT_TABLE_TYPE (class in *PMML44Super*), 39
 Counts (class in *PMML44Super*), 70
 Covariances (class in *PMML44Super*), 71
 CovariateList (class in *PMML44Super*), 71

D

DataDictionary (class in *PMML44Super*), 72
 DataField (class in *PMML44Super*), 73
 Decision (class in *PMML44Super*), 74
 Decisions (class in *PMML44Super*), 77
 DecisionTree (class in *PMML44Super*), 75
 DeepNetwork (class in *PMML44Super*), 77
 DefineFunction (class in *PMML44Super*), 79
 Delimiter (class in *PMML44Super*), 81
 Denominator (class in *PMML44Super*), 82
 DerivedField (class in *PMML44Super*), 82
 DiscreteConditionalProbability (class in *PMML44Super*), 85
 DiscreteNode (class in *PMML44Super*), 86

Discretize (class in *PMML44Super*), 87
 DiscretizeBin (class in *PMML44Super*), 88
 DiscrStats (class in *PMML44Super*), 84
 DocumentTermMatrix (class in *PMML44Super*), 89
 DynamicRegressor (class in *PMML44Super*), 90

E

euclidean (class in *PMML44Super*), 300
 EventValues (class in *PMML44Super*), 91
 ExponentialSmoothing (class in *PMML44Super*), 92
 export () (*PMML44Super.AbsoluteExponentialKernel* method), 7
 export () (*PMML44Super.Adadelata* method), 8
 export () (*PMML44Super.Adagrad* method), 9
 export () (*PMML44Super.Adam* method), 10
 export () (*PMML44Super.Adamax* method), 11
 export () (*PMML44Super.Aggregate* method), 11
 export () (*PMML44Super.Alternate* method), 12
 export () (*PMML44Super.Annotation* method), 13
 export () (*PMML44Super.AnomalyDetectionModel* method), 14
 export () (*PMML44Super.Anova* method), 16
 export () (*PMML44Super.AnovaRow* method), 17
 export () (*PMML44Super.AntecedentSequence* method), 18
 export () (*PMML44Super.AnyDistribution* method), 19
 export () (*PMML44Super.Application* method), 20
 export () (*PMML44Super.Apply* method), 21
 export () (*PMML44Super.AR* method), 3
 export () (*PMML44Super.ARDSquaredExponentialKernel* method), 4
 export () (*PMML44Super.ARIMA* method), 5
 export () (*PMML44Super.ARMAPart* method), 6
 export () (*PMML44Super.ArrayType* method), 22
 export () (*PMML44Super.AssociationModel* method), 23
 export () (*PMML44Super.AssociationRule* method), 25
 export () (*PMML44Super.Attribute* method), 27
 export () (*PMML44Super.BaseCumHazardTables* method), 28
 export () (*PMML44Super.Baseline* method), 29
 export () (*PMML44Super.BaselineCell* method), 30
 export () (*PMML44Super.BaselineModel* method), 30
 export () (*PMML44Super.BaselineStratum* method), 32
 export () (*PMML44Super.BayesianNetworkModel* method), 35
 export () (*PMML44Super.BayesianNetworkNodes* method), 37
 export () (*PMML44Super.BayesInput* method), 33
 export () (*PMML44Super.BayesInputs* method), 34

`export () (PMML44Super.BayesOutput method), 34`
`export () (PMML44Super.binarySimilarity method), 298`
`export () (PMML44Super.BlockIndicator method), 38`
`export () (PMML44Super.BoundaryValueMeans method), 38`
`export () (PMML44Super.BoundaryValues method), 39`
`export () (PMML44Super.CategoricalPredictor method), 41`
`export () (PMML44Super.Categories method), 41`
`export () (PMML44Super.Category method), 42`
`export () (PMML44Super.Characteristic method), 43`
`export () (PMML44Super.Characteristics method), 44`
`export () (PMML44Super.chebychev method), 299`
`export () (PMML44Super.ChildParent method), 45`
`export () (PMML44Super.cityBlock method), 300`
`export () (PMML44Super.ClassLabels method), 46`
`export () (PMML44Super.Cluster method), 46`
`export () (PMML44Super.ClusteringField method), 47`
`export () (PMML44Super.ClusteringModel method), 48`
`export () (PMML44Super.ClusteringModelQuality method), 50`
`export () (PMML44Super.Coefficient method), 51`
`export () (PMML44Super.Coefficients method), 52`
`export () (PMML44Super.ComparisonMeasure method), 53`
`export () (PMML44Super.Comparisons method), 54`
`export () (PMML44Super.ComplexPartialScore method), 55`
`export () (PMML44Super.CompoundPredicate method), 56`
`export () (PMML44Super.CompoundRule method), 57`
`export () (PMML44Super.Con method), 58`
`export () (PMML44Super.ConfusionMatrix method), 59`
`export () (PMML44Super.ConsequentSequence method), 60`
`export () (PMML44Super.Constant method), 61`
`export () (PMML44Super.Constraints method), 61`
`export () (PMML44Super.ContinuousConditionalProbability method), 64`
`export () (PMML44Super.ContinuousDistribution method), 65`
`export () (PMML44Super.ContinuousNode method), 66`
`export () (PMML44Super.ContStats method), 63`
`export () (PMML44Super.CorrelationFields method), 67`
`export () (PMML44Super.CorrelationMethods method), 68`
`export () (PMML44Super.Correlations method), 69`
`export () (PMML44Super.CorrelationValues method), 68`
`export () (PMML44Super.COUNT_TABLE_TYPE method), 40`
`export () (PMML44Super.Counts method), 70`
`export () (PMML44Super.Covariances method), 71`
`export () (PMML44Super.CovariateList method), 71`
`export () (PMML44Super.DataDictionary method), 72`
`export () (PMML44Super.DataField method), 73`
`export () (PMML44Super.Decision method), 74`
`export () (PMML44Super.Decisions method), 77`
`export () (PMML44Super.DecisionTree method), 75`
`export () (PMML44Super.DeepNetwork method), 78`
`export () (PMML44Super.DefineFunction method), 79`
`export () (PMML44Super.Delimiter method), 81`
`export () (PMML44Super.Denominator method), 82`
`export () (PMML44Super.DerivedField method), 83`
`export () (PMML44Super.DiscreteConditionalProbability method), 85`
`export () (PMML44Super.DiscreteNode method), 86`
`export () (PMML44Super.Discretize method), 87`
`export () (PMML44Super.DiscretizeBin method), 88`
`export () (PMML44Super.DiscrStats method), 84`
`export () (PMML44Super.DocumentTermMatrix method), 89`
`export () (PMML44Super.DynamicRegressor method), 90`
`export () (PMML44Super.euclidean method), 300`
`export () (PMML44Super.EventValues method), 91`
`export () (PMML44Super.ExponentialSmoothing method), 92`
`export () (PMML44Super.Extension method), 93`
`export () (PMML44Super.FactorList method), 93`
`export () (PMML44Super.False_ method), 94`
`export () (PMML44Super.FieldColumnPair method), 95`
`export () (PMML44Super.FieldRef method), 95`
`export () (PMML44Super.FieldValue method), 96`
`export () (PMML44Super.FieldValueCount method), 97`
`export () (PMML44Super.FinalNoise method), 98`
`export () (PMML44Super.FinalNu method), 98`
`export () (PMML44Super.FinalOmega method), 99`
`export () (PMML44Super.FinalPredictedNoise method), 99`
`export () (PMML44Super.FinalStateVector method), 100`
`export () (PMML44Super.FinalTheta method), 101`
`export () (PMML44Super.GARCH method), 101`
`export () (PMML44Super.GARCHPart method), 102`
`export () (PMML44Super.GaussianDistribution method), 103`
`export () (PMML44Super.GaussianProcessModel method), 104`

`export () (PMML44Super.GeneralizedExponentialKernel method), 109`
`export () (PMML44Super.GeneralRegressionModel method), 106`
`export () (PMML44Super.Header method), 110`
`export () (PMML44Super.HVector method), 110`
`export () (PMML44Super.Indices method), 113`
`export () (PMML44Super.InlineTable method), 113`
`export () (PMML44Super.InstanceField method), 114`
`export () (PMML44Super.InstanceFields method), 115`
`export () (PMML44Super.INT_Entries method), 111`
`export () (PMML44Super.INT_SparseArray method), 112`
`export () (PMML44Super.Interval method), 115`
`export () (PMML44Super.Item method), 116`
`export () (PMML44Super.ItemRef method), 117`
`export () (PMML44Super.Itemset method), 118`
`export () (PMML44Super.jaccard method), 301`
`export () (PMML44Super.KalmanState method), 120`
`export () (PMML44Super.KNNInput method), 119`
`export () (PMML44Super.KNNInputs method), 120`
`export () (PMML44Super.KohonenMap method), 121`
`export () (PMML44Super.Lag method), 122`
`export () (PMML44Super.Lambda method), 123`
`export () (PMML44Super.LayerBias method), 123`
`export () (PMML44Super.LayerParameters method), 124`
`export () (PMML44Super.LayerWeights method), 128`
`export () (PMML44Super.Level method), 129`
`export () (PMML44Super.LiftData method), 130`
`export () (PMML44Super.LiftGraph method), 131`
`export () (PMML44Super.LinearKernelType method), 131`
`export () (PMML44Super.LinearNorm method), 132`
`export () (PMML44Super.LocalTransformations method), 133`
`export () (PMML44Super.LognormalDistributionForBN method), 133`
`export () (PMML44Super.Losses method), 134`
`export () (PMML44Super.Lower method), 135`
`export () (PMML44Super.MA method), 136`
`export () (PMML44Super.MACoefficients method), 137`
`export () (PMML44Super.MapValues method), 137`
`export () (PMML44Super.MatCell method), 139`
`export () (PMML44Super.Matrix method), 139`
`export () (PMML44Super.MaximumLikelihoodStat method), 140`
`export () (PMML44Super.Mean method), 141`
`export () (PMML44Super.MeasurementMatrix method), 142`
`export () (PMML44Super.Metrics method), 143`
`export () (PMML44Super.MiningBuildTask method), 144`
`export () (PMML44Super.MiningField method), 144`
`export () (PMML44Super.MiningModel method), 146`
`export () (PMML44Super.MiningSchema method), 147`
`export () (PMML44Super.minkowski method), 301`
`export () (PMML44Super.MissingValueWeights method), 148`
`export () (PMML44Super.ModelExplanation method), 149`
`export () (PMML44Super.ModelLiftGraph method), 150`
`export () (PMML44Super.ModelStats method), 150`
`export () (PMML44Super.ModelVerification method), 151`
`export () (PMML44Super.MultivariateStat method), 152`
`export () (PMML44Super.MultivariateStats method), 154`
`export () (PMML44Super.Nadam method), 155`
`export () (PMML44Super.NaiveBayesModel method), 156`
`export () (PMML44Super.NearestNeighborModel method), 157`
`export () (PMML44Super.NetworkLayer method), 159`
`export () (PMML44Super.NeuralInput method), 160`
`export () (PMML44Super.NeuralInputs method), 161`
`export () (PMML44Super.NeuralLayer method), 162`
`export () (PMML44Super.NeuralNetwork method), 163`
`export () (PMML44Super.NeuralOutput method), 165`
`export () (PMML44Super.NeuralOutputs method), 166`
`export () (PMML44Super.Neuron method), 166`
`export () (PMML44Super.Node method), 168`
`export () (PMML44Super.NonseasonalComponent method), 169`
`export () (PMML44Super.NonseasonalFactor method), 170`
`export () (PMML44Super.NormalDistributionForBN method), 173`
`export () (PMML44Super.NormContinuous method), 171`
`export () (PMML44Super.NormDiscrete method), 172`
`export () (PMML44Super.Numerator method), 173`
`export () (PMML44Super.NumericInfo method), 174`
`export () (PMML44Super.NumericPredictor method), 175`
`export () (PMML44Super.Optimizers method), 176`
`export () (PMML44Super.OptimumLiftGraph method), 178`
`export () (PMML44Super.OutlierEffect method), 178`
`export () (PMML44Super.Output method), 179`
`export () (PMML44Super.OutputField method), 180`
`export () (PMML44Super.PairCounts method), 191`

- `export ()` (*PMML44Super.Parameter* method), 192
- `export ()` (*PMML44Super.ParameterField* method), 193
- `export ()` (*PMML44Super.ParameterList* method), 194
- `export ()` (*PMML44Super.ParamMatrix* method), 191
- `export ()` (*PMML44Super.ParentValue* method), 194
- `export ()` (*PMML44Super.Partition* method), 195
- `export ()` (*PMML44Super.PartitionFieldStats* method), 196
- `export ()` (*PMML44Super.PastVariances* method), 197
- `export ()` (*PMML44Super.PCell* method), 183
- `export ()` (*PMML44Super.PCovCell* method), 183
- `export ()` (*PMML44Super.PCovMatrix* method), 184
- `export ()` (*PMML44Super.PMML* method), 186
- `export ()` (*PMML44Super.PoissonDistribution* method), 198
- `export ()` (*PMML44Super.PolynomialKernelType* method), 198
- `export ()` (*PMML44Super.PPCell* method), 189
- `export ()` (*PMML44Super.PPMatrix* method), 190
- `export ()` (*PMML44Super.PredictiveModelQuality* method), 199
- `export ()` (*PMML44Super.Predictor* method), 201
- `export ()` (*PMML44Super.PredictorTerm* method), 202
- `export ()` (*PMML44Super.PsiVector* method), 203
- `export ()` (*PMML44Super.Quantile* method), 204
- `export ()` (*PMML44Super.RadialBasisKernel* method), 208
- `export ()` (*PMML44Super.RadialBasisKernelType* method), 209
- `export ()` (*PMML44Super.RandomLiftGraph* method), 210
- `export ()` (*PMML44Super.REAL_Entries* method), 205
- `export ()` (*PMML44Super.REAL_SparseArray* method), 205
- `export ()` (*PMML44Super.Regression* method), 211
- `export ()` (*PMML44Super.RegressionModel* method), 212
- `export ()` (*PMML44Super.RegressionTable* method), 214
- `export ()` (*PMML44Super.RegressorValues* method), 215
- `export ()` (*PMML44Super.Residuals* method), 217
- `export ()` (*PMML44Super.ResidualSquareCoefficients* method), 216
- `export ()` (*PMML44Super.ResultField* method), 217
- `export ()` (*PMML44Super.RMSprop* method), 206
- `export ()` (*PMML44Super.ROC* method), 207
- `export ()` (*PMML44Super.ROCGraph* method), 208
- `export ()` (*PMML44Super.row* method), 302
- `export ()` (*PMML44Super.RuleSelectionMethod* method), 218
- `export ()` (*PMML44Super.RuleSet* method), 219
- `export ()` (*PMML44Super.RuleSetModel* method), 220
- `export ()` (*PMML44Super.Scorecard* method), 224
- `export ()` (*PMML44Super.ScoreDistribution* method), 223
- `export ()` (*PMML44Super.SeasonalComponent* method), 225
- `export ()` (*PMML44Super.SeasonalFactor* method), 226
- `export ()` (*PMML44Super.Seasonality_ExpoSmooth* method), 227
- `export ()` (*PMML44Super.SeasonalTrendDecomposition* method), 227
- `export ()` (*PMML44Super.Segment* method), 228
- `export ()` (*PMML44Super.Segmentation* method), 231
- `export ()` (*PMML44Super.Sequence* method), 232
- `export ()` (*PMML44Super.SequenceModel* method), 233
- `export ()` (*PMML44Super.SequenceReference* method), 235
- `export ()` (*PMML44Super.SequenceRule* method), 236
- `export ()` (*PMML44Super.SetPredicate* method), 237
- `export ()` (*PMML44Super.SetReference* method), 238
- `export ()` (*PMML44Super.SGD* method), 222
- `export ()` (*PMML44Super.SigmoidKernelType* method), 239
- `export ()` (*PMML44Super.simpleMatching* method), 303
- `export ()` (*PMML44Super.SimplePredicate* method), 239
- `export ()` (*PMML44Super.SimpleRule* method), 240
- `export ()` (*PMML44Super.SimpleSetPredicate* method), 242
- `export ()` (*PMML44Super.SpectralAnalysis* method), 243
- `export ()` (*PMML44Super.squaredEuclidean* method), 303
- `export ()` (*PMML44Super.StateSpaceModel* method), 243
- `export ()` (*PMML44Super.StateVector* method), 244
- `export ()` (*PMML44Super.SupportVector* method), 245
- `export ()` (*PMML44Super.SupportVectorMachine* method), 246
- `export ()` (*PMML44Super.SupportVectorMachineModel* method), 247
- `export ()` (*PMML44Super.SupportVectors* method), 249
- `export ()` (*PMML44Super.TableLocator* method), 250
- `export ()` (*PMML44Super.tanimoto* method), 304
- `export ()` (*PMML44Super.Target* method), 250
- `export ()` (*PMML44Super.Targets* method), 256
- `export ()` (*PMML44Super.TargetValue* method), 252
- `export ()` (*PMML44Super.TargetValueCount* method), 252
- `export ()` (*PMML44Super.TargetValueCounts* method), 253

[export \(\) \(PMML44Super.TargetValueStat method\), 254](#)
[export \(\) \(PMML44Super.TargetValueStats method\), 255](#)
[export \(\) \(PMML44Super.Taxonomy method\), 256](#)
[export \(\) \(PMML44Super.TestDistributions method\), 257](#)
[export \(\) \(PMML44Super.TextCorpus method\), 258](#)
[export \(\) \(PMML44Super.TextDictionary method\), 259](#)
[export \(\) \(PMML44Super.TextDocument method\), 260](#)
[export \(\) \(PMML44Super.TextIndex method\), 261](#)
[export \(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[export \(\) \(PMML44Super.TextModel method\), 264](#)
[export \(\) \(PMML44Super.TextModelNormalization method\), 266](#)
[export \(\) \(PMML44Super.TextModelSimilarity method\), 266](#)
[export \(\) \(PMML44Super.Theta method\), 267](#)
[export \(\) \(PMML44Super.ThetaRecursionState method\), 268](#)
[export \(\) \(PMML44Super.Time method\), 268](#)
[export \(\) \(PMML44Super.TimeAnchor method\), 269](#)
[export \(\) \(PMML44Super.TimeCycle method\), 270](#)
[export \(\) \(PMML44Super.TimeException method\), 271](#)
[export \(\) \(PMML44Super.TimeSeries method\), 272](#)
[export \(\) \(PMML44Super.TimeSeriesModel method\), 273](#)
[export \(\) \(PMML44Super.Timestamp method\), 275](#)
[export \(\) \(PMML44Super.TimeValue method\), 275](#)
[export \(\) \(PMML44Super.TrainingInstances method\), 276](#)
[export \(\) \(PMML44Super.TrainingParameters method\), 277](#)
[export \(\) \(PMML44Super.TransferFunctionValues method\), 278](#)
[export \(\) \(PMML44Super.TransformationDictionary method\), 279](#)
[export \(\) \(PMML44Super.TransitionMatrix method\), 279](#)
[export \(\) \(PMML44Super.TreeModel method\), 280](#)
[export \(\) \(PMML44Super.Trend_ExpoSmooth method\), 282](#)
[export \(\) \(PMML44Super.TrendCoefficients method\), 282](#)
[export \(\) \(PMML44Super.TriangularDistributionForBN method\), 283](#)
[export \(\) \(PMML44Super.True_ method\), 284](#)
[export \(\) \(PMML44Super.UniformDistribution method\), 285](#)
[export \(\) \(PMML44Super.UniformDistributionForBN method\), 285](#)
[export \(\) \(PMML44Super.UnivariateStats method\), 286](#)
[export \(\) \(PMML44Super.Upper method\), 287](#)
[export \(\) \(PMML44Super.Value method\), 288](#)
[export \(\) \(PMML44Super.ValueProbability method\), 289](#)
[export \(\) \(PMML44Super.VariableWeight method\), 290](#)
[export \(\) \(PMML44Super.Variance method\), 291](#)
[export \(\) \(PMML44Super.VarianceCoefficients method\), 292](#)
[export \(\) \(PMML44Super.VectorDictionary method\), 292](#)
[export \(\) \(PMML44Super.VectorFields method\), 293](#)
[export \(\) \(PMML44Super.VectorInstance method\), 294](#)
[export \(\) \(PMML44Super.VerificationField method\), 295](#)
[export \(\) \(PMML44Super.VerificationFields method\), 296](#)
[export \(\) \(PMML44Super.XCoordinates method\), 297](#)
[export \(\) \(PMML44Super.YCoordinates method\), 297](#)
[exportAttributes \(\) \(PMML44Super.AbsoluteExponentialKernel method\), 7](#)
[exportAttributes \(\) \(PMML44Super.Adadelta method\), 8](#)
[exportAttributes \(\) \(PMML44Super.Adagrad method\), 9](#)
[exportAttributes \(\) \(PMML44Super.Adam method\), 10](#)
[exportAttributes \(\) \(PMML44Super.Adamax method\), 11](#)
[exportAttributes \(\) \(PMML44Super.Aggregate method\), 11](#)
[exportAttributes \(\) \(PMML44Super.Alternate method\), 12](#)
[exportAttributes \(\) \(PMML44Super.Annotation method\), 13](#)
[exportAttributes \(\) \(PMML44Super.AnomalyDetectionModel method\), 14](#)
[exportAttributes \(\) \(PMML44Super.Anova method\), 16](#)
[exportAttributes \(\) \(PMML44Super.AnovaRow method\), 17](#)
[exportAttributes \(\) \(PMML44Super.AntecedentSequence method\), 18](#)
[exportAttributes \(\) \(PMML44Super.AnyDistribution method\), 19](#)
[exportAttributes \(\) \(PMML44Super.Application method\), 20](#)

exportAttributes ()	(PMML44Super.Apply	exportAttributes ()	
method), 21		(PMML44Super.BoundaryValues	method),
exportAttributes ()	(PMML44Super.AR	39	
3		exportAttributes ()	
exportAttributes ()	(PMML44Super.ARDSquaredExponentialKernel	41	
method), 4		exportAttributes ()	(PMML44Super.Categories
exportAttributes ()	(PMML44Super.ARIMA	method), 41	
method), 5		exportAttributes ()	(PMML44Super.Category
exportAttributes ()	(PMML44Super.ARMAPart	method), 42	
method), 6		exportAttributes ()	(PMML44Super.Characteristic
exportAttributes ()	(PMML44Super.ArrayType	method),	
method), 23		43	
exportAttributes ()	(PMML44Super.AssociationModel	method),	
method), 23		exportAttributes ()	(PMML44Super.Characteristics
exportAttributes ()	(PMML44Super.AssociationRule	method),	
26		44	
exportAttributes ()	(PMML44Super.Attribute	exportAttributes ()	(PMML44Super.chebychev
method), 27		method), 299	
exportAttributes ()	(PMML44Super.BaseCumHazardTables	exportAttributes ()	(PMML44Super.ChildParent
method), 28		method), 45	
exportAttributes ()	(PMML44Super.Baseline	exportAttributes ()	(PMML44Super.cityBlock
method), 29		method), 300	
exportAttributes ()	(PMML44Super.BaselineCell	exportAttributes ()	(PMML44Super.ClassLabels
method), 30		method), 46	
exportAttributes ()	(PMML44Super.BaselineModel	method), 46	
method), 30		exportAttributes ()	(PMML44Super.Cluster
exportAttributes ()	(PMML44Super.BaselineStratum	method), 46	
method), 32		exportAttributes ()	(PMML44Super.ClusteringField
exportAttributes ()	(PMML44Super.BayesianNetworkModel	method),	
method), 35		47	
exportAttributes ()	(PMML44Super.BayesianNetworkNodes	exportAttributes ()	(PMML44Super.ClusteringModel
method), 37		method),	
exportAttributes ()	(PMML44Super.BayesInput	48	
method), 33		exportAttributes ()	(PMML44Super.ClusteringModelQuality
exportAttributes ()	(PMML44Super.BayesInputs	method), 50	
method), 34		exportAttributes ()	(PMML44Super.Coefficient
exportAttributes ()	(PMML44Super.BayesOutput	method), 51	
method), 34		exportAttributes ()	(PMML44Super.Coefficients
exportAttributes ()	(PMML44Super.binarySimilarity	method), 52	
method),		exportAttributes ()	(PMML44Super.ComparisonMeasure
298		method),	
exportAttributes ()	(PMML44Super.BlockIndicator	53	
method),		exportAttributes ()	(PMML44Super.Comparisons
38		method), 54	
exportAttributes ()	(PMML44Super.BoundaryValueMeans	exportAttributes ()	(PMML44Super.ComplexPartialScore
method), 38		method),	
		55	
		exportAttributes ()	(PMML44Super.CompoundPredicate
		method),	
		56	
		exportAttributes ()	(PMML44Super.CompoundRule
		method),	
		57	
		exportAttributes ()	(PMML44Super.Con

`method)`, 59
`exportAttributes()` (`PMML44Super.ConfusionMatrix method`), 59
`exportAttributes()` (`PMML44Super.ConsequentSequence method`), 60
`exportAttributes()` (`PMML44Super.Constant method`), 61
`exportAttributes()` (`PMML44Super.Constraints method`), 61
`exportAttributes()` (`PMML44Super.ContinuousConditionalProbability method`), 64
`exportAttributes()` (`PMML44Super.ContinuousDistribution method`), 65
`exportAttributes()` (`PMML44Super.ContinuousNode method`), 66
`exportAttributes()` (`PMML44Super.ContStats method`), 63
`exportAttributes()` (`PMML44Super.CorrelationFields method`), 67
`exportAttributes()` (`PMML44Super.CorrelationMethods method`), 68
`exportAttributes()` (`PMML44Super.Correlations method`), 69
`exportAttributes()` (`PMML44Super.CorrelationValues method`), 68
`exportAttributes()` (`PMML44Super.COUNT_TABLE_TYPE method`), 40
`exportAttributes()` (`PMML44Super.Counts method`), 70
`exportAttributes()` (`PMML44Super.Covariances method`), 71
`exportAttributes()` (`PMML44Super.CovariateList method`), 71
`exportAttributes()` (`PMML44Super.DataDictionary method`), 72
`exportAttributes()` (`PMML44Super.DataField method`), 73
`exportAttributes()` (`PMML44Super.Decision method`), 74
`exportAttributes()` (`PMML44Super.Decisions method`), 77
`exportAttributes()` (`PMML44Super.DecisionTree method`), 75
`exportAttributes()` (`PMML44Super.DeepNetwork method`), 78
`exportAttributes()` (`PMML44Super.DefineFunction method`), 80
`exportAttributes()` (`PMML44Super.Delimiter method`), 81
`exportAttributes()` (`PMML44Super.Denominator method`), 82
`exportAttributes()` (`PMML44Super.DerivedField method`), 83
`exportAttributes()` (`PMML44Super.DiscreteConditionalProbability method`), 85
`exportAttributes()` (`PMML44Super.DiscreteNode method`), 86
`exportAttributes()` (`PMML44Super.Discretize method`), 87
`exportAttributes()` (`PMML44Super.DiscretizeBin method`), 88
`exportAttributes()` (`PMML44Super.DiscrStats method`), 84
`exportAttributes()` (`PMML44Super.DocumentTermMatrix method`), 89
`exportAttributes()` (`PMML44Super.DynamicRegressor method`), 90
`exportAttributes()` (`PMML44Super.euclidean method`), 300
`exportAttributes()` (`PMML44Super.EventValues method`), 91
`exportAttributes()` (`PMML44Super.ExponentialSmoothing method`), 92
`exportAttributes()` (`PMML44Super.Extension method`), 93
`exportAttributes()` (`PMML44Super.FactorList method`), 93
`exportAttributes()` (`PMML44Super.False_ method`), 94
`exportAttributes()` (`PMML44Super.FieldColumnPair method`), 95
`exportAttributes()` (`PMML44Super.FieldRef method`), 95
`exportAttributes()` (`PMML44Super.FieldValue method`), 96
`exportAttributes()` (`PMML44Super.FieldValueCount method`), 97
`exportAttributes()` (`PMML44Super.FinalNoise method`), 98
`exportAttributes()` (`PMML44Super.FinalNu method`), 99
`exportAttributes()` (`PMML44Super.FinalOmega`

[method](#)), 99
[exportAttributes\(\)](#) ([PMML44Super.FinalPredictedNoise](#) [method](#)), 100
[exportAttributes\(\)](#) ([PMML44Super.FinalStateVector](#) [method](#)), 100
[exportAttributes\(\)](#) ([PMML44Super.FinalTheta](#) [method](#)), 101
[exportAttributes\(\)](#) ([PMML44Super.GARCH](#) [method](#)), 101
[exportAttributes\(\)](#) ([PMML44Super.GARCHPart](#) [method](#)), 102
[exportAttributes\(\)](#) ([PMML44Super.GaussianDistribution](#) [method](#)), 103
[exportAttributes\(\)](#) ([PMML44Super.GaussianProcessModel](#) [method](#)), 104
[exportAttributes\(\)](#) ([PMML44Super.GeneralizedExponentialKernel](#) [method](#)), 109
[exportAttributes\(\)](#) ([PMML44Super.GeneralRegressionModel](#) [method](#)), 106
[exportAttributes\(\)](#) ([PMML44Super.Header](#) [method](#)), 110
[exportAttributes\(\)](#) ([PMML44Super.HVector](#) [method](#)), 110
[exportAttributes\(\)](#) ([PMML44Super.Indices](#) [method](#)), 113
[exportAttributes\(\)](#) ([PMML44Super.InlineTable](#) [method](#)), 113
[exportAttributes\(\)](#) ([PMML44Super.InstanceField](#) [method](#)), 114
[exportAttributes\(\)](#) ([PMML44Super.InstanceFields](#) [method](#)), 115
[exportAttributes\(\)](#) ([PMML44Super.INT_Entries](#) [method](#)), 111
[exportAttributes\(\)](#) ([PMML44Super.INT_SparseArray](#) [method](#)), 112
[exportAttributes\(\)](#) ([PMML44Super.Interval](#) [method](#)), 115
[exportAttributes\(\)](#) ([PMML44Super.Item](#) [method](#)), 116
[exportAttributes\(\)](#) ([PMML44Super.ItemRef](#) [method](#)), 117
[exportAttributes\(\)](#) ([PMML44Super.Itemset](#) [method](#)), 118
[exportAttributes\(\)](#) ([PMML44Super.jaccard](#) [method](#)), 301
[exportAttributes\(\)](#) ([PMML44Super.KalmanState](#) [method](#)), 120
[exportAttributes\(\)](#) ([PMML44Super.KNNInput](#) [method](#)), 119
[exportAttributes\(\)](#) ([PMML44Super.KNNInputs](#) [method](#)), 120
[exportAttributes\(\)](#) ([PMML44Super.KohonenMap](#) [method](#)), 121
[exportAttributes\(\)](#) ([PMML44Super.Lag](#) [method](#)), 122
[exportAttributes\(\)](#) ([PMML44Super.Lambda](#) [method](#)), 123
[exportAttributes\(\)](#) ([PMML44Super.LayerBias](#) [method](#)), 123
[exportAttributes\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 124
[exportAttributes\(\)](#) ([PMML44Super.LayerWeights](#) [method](#)), 128
[exportAttributes\(\)](#) ([PMML44Super.Level](#) [method](#)), 129
[exportAttributes\(\)](#) ([PMML44Super.LiftData](#) [method](#)), 130
[exportAttributes\(\)](#) ([PMML44Super.LiftGraph](#) [method](#)), 131
[exportAttributes\(\)](#) ([PMML44Super.LinearKernelType](#) [method](#)), 131
[exportAttributes\(\)](#) ([PMML44Super.LinearNorm](#) [method](#)), 132
[exportAttributes\(\)](#) ([PMML44Super.LocalTransformations](#) [method](#)), 133
[exportAttributes\(\)](#) ([PMML44Super.LognormalDistributionForBN](#) [method](#)), 134
[exportAttributes\(\)](#) ([PMML44Super.Losses](#) [method](#)), 134
[exportAttributes\(\)](#) ([PMML44Super.Lower](#) [method](#)), 135
[exportAttributes\(\)](#) ([PMML44Super.MA](#) [method](#)), 136
[exportAttributes\(\)](#) ([PMML44Super.MACoefficients](#) [method](#)), 137
[exportAttributes\(\)](#) ([PMML44Super.MapValues](#) [method](#)), 138
[exportAttributes\(\)](#) ([PMML44Super.MatCell](#) [method](#)), 139
[exportAttributes\(\)](#) ([PMML44Super.Matrix](#) [method](#)), 139
[exportAttributes\(\)](#) ([PMML44Super.MaximumLikelihoodStat](#) [method](#)), 140
[exportAttributes\(\)](#) ([PMML44Super.Mean](#)

[method](#)), [141](#)
[exportAttributes\(\)](#) ([PMML44Super.MeasurementMatrix](#) [method](#)), [142](#)
[exportAttributes\(\)](#) ([PMML44Super.Metrics](#) [method](#)), [143](#)
[exportAttributes\(\)](#) ([PMML44Super.MiningBuildTask](#) [method](#)), [144](#)
[exportAttributes\(\)](#) ([PMML44Super.MiningField](#) [method](#)), [144](#)
[exportAttributes\(\)](#) ([PMML44Super.MiningModel](#) [method](#)), [146](#)
[exportAttributes\(\)](#) ([PMML44Super.MiningSchema](#) [method](#)), [147](#)
[exportAttributes\(\)](#) ([PMML44Super.minkowski](#) [method](#)), [302](#)
[exportAttributes\(\)](#) ([PMML44Super.MissingValueWeights](#) [method](#)), [148](#)
[exportAttributes\(\)](#) ([PMML44Super.ModelExplanation](#) [method](#)), [149](#)
[exportAttributes\(\)](#) ([PMML44Super.ModelLiftGraph](#) [method](#)), [150](#)
[exportAttributes\(\)](#) ([PMML44Super.ModelStats](#) [method](#)), [150](#)
[exportAttributes\(\)](#) ([PMML44Super.ModelVerification](#) [method](#)), [151](#)
[exportAttributes\(\)](#) ([PMML44Super.MultivariateStat](#) [method](#)), [152](#)
[exportAttributes\(\)](#) ([PMML44Super.MultivariateStats](#) [method](#)), [154](#)
[exportAttributes\(\)](#) ([PMML44Super.Nadam](#) [method](#)), [155](#)
[exportAttributes\(\)](#) ([PMML44Super.NaiveBayesModel](#) [method](#)), [156](#)
[exportAttributes\(\)](#) ([PMML44Super.NearestNeighborModel](#) [method](#)), [157](#)
[exportAttributes\(\)](#) ([PMML44Super.NetworkLayer](#) [method](#)), [159](#)
[exportAttributes\(\)](#) ([PMML44Super.NeuralInput](#) [method](#)), [160](#)
[exportAttributes\(\)](#) ([PMML44Super.NeuralInputs](#) [method](#)), [161](#)
[exportAttributes\(\)](#) ([PMML44Super.NeuralLayer](#) [method](#)), [162](#)
[exportAttributes\(\)](#) ([PMML44Super.NeuralNetwork](#) [method](#)), [163](#)
[exportAttributes\(\)](#) ([PMML44Super.NeuralOutput](#) [method](#)), [165](#)
[exportAttributes\(\)](#) ([PMML44Super.NeuralOutputs](#) [method](#)), [166](#)
[exportAttributes\(\)](#) ([PMML44Super.Neuron](#) [method](#)), [166](#)
[exportAttributes\(\)](#) ([PMML44Super.Node](#) [method](#)), [168](#)
[exportAttributes\(\)](#) ([PMML44Super.NonseasonalComponent](#) [method](#)), [169](#)
[exportAttributes\(\)](#) ([PMML44Super.NonseasonalFactor](#) [method](#)), [170](#)
[exportAttributes\(\)](#) ([PMML44Super.NormalDistributionForBN](#) [method](#)), [173](#)
[exportAttributes\(\)](#) ([PMML44Super.NormContinuous](#) [method](#)), [171](#)
[exportAttributes\(\)](#) ([PMML44Super.NormDiscrete](#) [method](#)), [172](#)
[exportAttributes\(\)](#) ([PMML44Super.Numerator](#) [method](#)), [173](#)
[exportAttributes\(\)](#) ([PMML44Super.NumericInfo](#) [method](#)), [174](#)
[exportAttributes\(\)](#) ([PMML44Super.NumericPredictor](#) [method](#)), [175](#)
[exportAttributes\(\)](#) ([PMML44Super.Optimizers](#) [method](#)), [176](#)
[exportAttributes\(\)](#) ([PMML44Super.OptimumLiftGraph](#) [method](#)), [178](#)
[exportAttributes\(\)](#) ([PMML44Super.OutlierEffect](#) [method](#)), [179](#)
[exportAttributes\(\)](#) ([PMML44Super.Output](#) [method](#)), [179](#)
[exportAttributes\(\)](#) ([PMML44Super.OutputField](#) [method](#)), [180](#)
[exportAttributes\(\)](#) ([PMML44Super.PairCounts](#) [method](#)), [191](#)
[exportAttributes\(\)](#) ([PMML44Super.Parameter](#) [method](#)), [192](#)
[exportAttributes\(\)](#) ([PMML44Super.ParameterField](#) [method](#)), [193](#)

<code>exportAttributes()</code> (<i>PMML44Super.ParameterList</i> <i>method</i>), 194	<code>exportAttributes()</code> (<i>PMML44Super.REAL_Entries</i> <i>method</i>), 205
<code>exportAttributes()</code> (<i>PMML44Super.ParamMatrix</i> <i>method</i>), 191	<code>exportAttributes()</code> (<i>PMML44Super.REAL_SparseArray</i> <i>method</i>), 205
<code>exportAttributes()</code> (<i>PMML44Super.ParentValue</i> <i>method</i>), 194	<code>exportAttributes()</code> (<i>PMML44Super.Regression</i> <i>method</i>), 211
<code>exportAttributes()</code> (<i>PMML44Super.Partition</i> <i>method</i>), 195	<code>exportAttributes()</code> (<i>PMML44Super.RegressionModel</i> <i>method</i>), 212
<code>exportAttributes()</code> (<i>PMML44Super.PartitionFieldStats</i> <i>method</i>), 196	<code>exportAttributes()</code> (<i>PMML44Super.RegressionTable</i> <i>method</i>), 214
<code>exportAttributes()</code> (<i>PMML44Super.PastVariances</i> <i>method</i>), 197	<code>exportAttributes()</code> (<i>PMML44Super.RegressorValues</i> <i>method</i>), 215
<code>exportAttributes()</code> (<i>PMML44Super.PCell</i> <i>method</i>), 183	<code>exportAttributes()</code> (<i>PMML44Super.Residuals</i> <i>method</i>), 217
<code>exportAttributes()</code> (<i>PMML44Super.PCovCell</i> <i>method</i>), 183	<code>exportAttributes()</code> (<i>PMML44Super.ResidualSquareCoefficients</i> <i>method</i>), 216
<code>exportAttributes()</code> (<i>PMML44Super.PCovMatrix</i> <i>method</i>), 184	<code>exportAttributes()</code> (<i>PMML44Super.ResultField</i> <i>method</i>), 217
<code>exportAttributes()</code> (<i>PMML44Super.PMML</i> <i>method</i>), 186	<code>exportAttributes()</code> (<i>PMML44Super.RMSprop</i> <i>method</i>), 206
<code>exportAttributes()</code> (<i>PMML44Super.PoissonDistribution</i> <i>method</i>), 198	<code>exportAttributes()</code> (<i>PMML44Super.ROC</i> <i>method</i>), 207
<code>exportAttributes()</code> (<i>PMML44Super.PolynomialKernelType</i> <i>method</i>), 198	<code>exportAttributes()</code> (<i>PMML44Super.ROCGraph</i> <i>method</i>), 208
<code>exportAttributes()</code> (<i>PMML44Super.PPCell</i> <i>method</i>), 189	<code>exportAttributes()</code> (<i>PMML44Super.row</i> <i>method</i>), 302
<code>exportAttributes()</code> (<i>PMML44Super.PPMatrix</i> <i>method</i>), 190	<code>exportAttributes()</code> (<i>PMML44Super.RuleSelectionMethod</i> <i>method</i>), 218
<code>exportAttributes()</code> (<i>PMML44Super.PredictiveModelQuality</i> <i>method</i>), 200	<code>exportAttributes()</code> (<i>PMML44Super.RuleSet</i> <i>method</i>), 219
<code>exportAttributes()</code> (<i>PMML44Super.Predictor</i> <i>method</i>), 201	<code>exportAttributes()</code> (<i>PMML44Super.RuleSetModel</i> <i>method</i>), 220
<code>exportAttributes()</code> (<i>PMML44Super.PredictorTerm</i> <i>method</i>), 202	<code>exportAttributes()</code> (<i>PMML44Super.Scorecard</i> <i>method</i>), 224
<code>exportAttributes()</code> (<i>PMML44Super.PsiVector</i> <i>method</i>), 203	<code>exportAttributes()</code> (<i>PMML44Super.ScoreDistribution</i> <i>method</i>), 223
<code>exportAttributes()</code> (<i>PMML44Super.Quantile</i> <i>method</i>), 204	<code>exportAttributes()</code> (<i>PMML44Super.SeasonalComponent</i> <i>method</i>), 225
<code>exportAttributes()</code> (<i>PMML44Super.RadialBasisKernel</i> <i>method</i>), 209	<code>exportAttributes()</code> (<i>PMML44Super.SeasonalFactor</i> <i>method</i>), 226
<code>exportAttributes()</code> (<i>PMML44Super.RadialBasisKernelType</i> <i>method</i>), 209	<code>exportAttributes()</code> (<i>PMML44Super.Seasonality_ExpoSmooth</i> <i>method</i>), 227
<code>exportAttributes()</code> (<i>PMML44Super.RandomLiftGraph</i> <i>method</i>), 210	

`exportAttributes()`
 (*PMML44Super.SeasonalTrendDecomposition*
 method), 227
`exportAttributes()` (*PMML44Super.Segment*
 method), 229
`exportAttributes()`
 (*PMML44Super.Segmentation method*), 231
`exportAttributes()` (*PMML44Super.Sequence*
 method), 232
`exportAttributes()`
 (*PMML44Super.SequenceModel method*),
 233
`exportAttributes()`
 (*PMML44Super.SequenceReference method*),
 235
`exportAttributes()`
 (*PMML44Super.SequenceRule method*),
 236
`exportAttributes()` (*PMML44Super.SetPredicate*
 method), 237
`exportAttributes()` (*PMML44Super.SetReference*
 method), 238
`exportAttributes()` (*PMML44Super.SGD*
 method), 222
`exportAttributes()`
 (*PMML44Super.SigmoidKernelType method*),
 239
`exportAttributes()`
 (*PMML44Super.simpleMatching method*),
 303
`exportAttributes()`
 (*PMML44Super.SimplePredicate method*),
 240
`exportAttributes()` (*PMML44Super.SimpleRule*
 method), 240
`exportAttributes()`
 (*PMML44Super.SimpleSetPredicate method*),
 242
`exportAttributes()`
 (*PMML44Super.SpectralAnalysis method*),
 243
`exportAttributes()`
 (*PMML44Super.squaredEuclidean method*),
 303
`exportAttributes()`
 (*PMML44Super.StateSpaceModel method*),
 243
`exportAttributes()` (*PMML44Super.StateVector*
 method), 244
`exportAttributes()`
 (*PMML44Super.SupportVector method*),
 245
`exportAttributes()`
 (*PMML44Super.SupportVectorMachine*
 method), 246
`exportAttributes()`
 (*PMML44Super.SupportVectorMachineModel*
 method), 247
`exportAttributes()`
 (*PMML44Super.SupportVectors method*),
 249
`exportAttributes()` (*PMML44Super.TableLocator*
 method), 250
`exportAttributes()` (*PMML44Super.tanimoto*
 method), 304
`exportAttributes()` (*PMML44Super.Target*
 method), 250
`exportAttributes()` (*PMML44Super.Targets*
 method), 256
`exportAttributes()` (*PMML44Super.TargetValue*
 method), 252
`exportAttributes()`
 (*PMML44Super.TargetValueCount method*),
 253
`exportAttributes()`
 (*PMML44Super.TargetValueCounts method*),
 253
`exportAttributes()`
 (*PMML44Super.TargetValueStat method*),
 254
`exportAttributes()`
 (*PMML44Super.TargetValueStats method*),
 255
`exportAttributes()` (*PMML44Super.Taxonomy*
 method), 256
`exportAttributes()`
 (*PMML44Super.TestDistributions method*),
 257
`exportAttributes()` (*PMML44Super.TextCorpus*
 method), 258
`exportAttributes()`
 (*PMML44Super.TextDictionary method*),
 259
`exportAttributes()`
 (*PMML44Super.TextDocument method*),
 260
`exportAttributes()` (*PMML44Super.TextIndex*
 method), 261
`exportAttributes()`
 (*PMML44Super.TextIndexNormalization*
 method), 263
`exportAttributes()` (*PMML44Super.TextModel*
 method), 264
`exportAttributes()`
 (*PMML44Super.TextModelNormalization*
 method), 266
`exportAttributes()`
 (*PMML44Super.TextModelSimiliarity method*),

- 266
`exportAttributes()` (*PMML44Super.Theta* method), 267
`exportAttributes()` (*PMML44Super.ThetaRecursionState* method), 268
`exportAttributes()` (*PMML44Super.Time* method), 268
`exportAttributes()` (*PMML44Super.TimeAnchor* method), 269
`exportAttributes()` (*PMML44Super.TimeCycle* method), 270
`exportAttributes()` (*PMML44Super.TimeException* method), 271
`exportAttributes()` (*PMML44Super.TimeSeries* method), 272
`exportAttributes()` (*PMML44Super.TimeSeriesModel* method), 273
`exportAttributes()` (*PMML44Super.Timestamp* method), 275
`exportAttributes()` (*PMML44Super.TimeValue* method), 275
`exportAttributes()` (*PMML44Super.TrainingInstances* method), 276
`exportAttributes()` (*PMML44Super.TrainingParameters* method), 277
`exportAttributes()` (*PMML44Super.TransferFunctionValues* method), 278
`exportAttributes()` (*PMML44Super.TransformationDictionary* method), 279
`exportAttributes()` (*PMML44Super.TransitionMatrix* method), 279
`exportAttributes()` (*PMML44Super.TreeModel* method), 280
`exportAttributes()` (*PMML44Super.Trend_ExpoSmooth* method), 283
`exportAttributes()` (*PMML44Super.TrendCoefficients* method), 282
`exportAttributes()` (*PMML44Super.TriangularDistributionForBN* method), 283
`exportAttributes()` (*PMML44Super.True_* method), 284
`exportAttributes()` (*PMML44Super.UniformDistribution* method), 285
`exportAttributes()` (*PMML44Super.UniformDistributionForBN* method), 285
`exportAttributes()` (*PMML44Super.UnivariateStats* method), 286
`exportAttributes()` (*PMML44Super.Upper* method), 287
`exportAttributes()` (*PMML44Super.Value* method), 288
`exportAttributes()` (*PMML44Super.ValueProbability* method), 289
`exportAttributes()` (*PMML44Super.VariableWeight* method), 290
`exportAttributes()` (*PMML44Super.Variance* method), 291
`exportAttributes()` (*PMML44Super.VarianceCoefficients* method), 292
`exportAttributes()` (*PMML44Super.VectorDictionary* method), 293
`exportAttributes()` (*PMML44Super.VectorFields* method), 293
`exportAttributes()` (*PMML44Super.VectorInstance* method), 294
`exportAttributes()` (*PMML44Super.VerificationField* method), 295
`exportAttributes()` (*PMML44Super.VerificationFields* method), 296
`exportAttributes()` (*PMML44Super.XCoordinates* method), 297
`exportAttributes()` (*PMML44Super.YCoordinates* method), 297
`exportChildren()` (*PMML44Super.AbsoluteExponentialKernel* method), 7
`exportChildren()` (*PMML44Super.Adadelta* method), 8
`exportChildren()` (*PMML44Super.Adagrad* method), 9
`exportChildren()` (*PMML44Super.Adam* method), 10
`exportChildren()` (*PMML44Super.Adamax* method), 11
`exportChildren()` (*PMML44Super.Aggregate* method), 11
`exportChildren()` (*PMML44Super.Alternate*

<i>method</i>), 12		<code>exportChildren()</code> (PMML44Super.BlockIndicator <i>method</i>), 38	
<code>exportChildren()</code> (PMML44Super.Annotation <i>method</i>), 13		<code>exportChildren()</code> (PMML44Super.BoundaryValueMeans <i>method</i>), 38	
<code>exportChildren()</code> (PMML44Super.AnomalyDetectionModel <i>method</i>), 14		<code>exportChildren()</code> (PMML44Super.BoundaryValues <i>method</i>), 39	
<code>exportChildren()</code> (PMML44Super.Anova <i>method</i>), 16		<code>exportChildren()</code> (PMML44Super.CategoricalPredictor <i>method</i>), 41	
<code>exportChildren()</code> (PMML44Super.AnovaRow <i>method</i>), 17		<code>exportChildren()</code> (PMML44Super.Categories <i>method</i>), 41	
<code>exportChildren()</code> (PMML44Super.AntecedentSequence <i>method</i>), 18		<code>exportChildren()</code> (PMML44Super.Category <i>method</i>), 42	
<code>exportChildren()</code> (PMML44Super.AnyDistribution <i>method</i>), 19		<code>exportChildren()</code> (PMML44Super.Characteristic <i>method</i>), 43	
<code>exportChildren()</code> (PMML44Super.Application <i>method</i>), 20		<code>exportChildren()</code> (PMML44Super.Characteristics <i>method</i>), 44	
<code>exportChildren()</code> (PMML44Super.Apply <i>method</i>), 21		<code>exportChildren()</code> (PMML44Super.chebychev <i>method</i>), 299	
<code>exportChildren()</code> (PMML44Super.AR <i>method</i>), 3		<code>exportChildren()</code> (PMML44Super.ChildParent <i>method</i>), 45	
<code>exportChildren()</code> (PMML44Super.ARDSquaredExponentialKernel <i>method</i>), 4		<code>exportChildren()</code> (PMML44Super.cityBlock <i>method</i>), 300	
<code>exportChildren()</code> (PMML44Super.ARIMA <i>method</i>), 5		<code>exportChildren()</code> (PMML44Super.ClassLabels <i>method</i>), 46	
<code>exportChildren()</code> (PMML44Super.ARMAPart <i>method</i>), 6		<code>exportChildren()</code> (PMML44Super.Cluster <i>method</i>), 46	
<code>exportChildren()</code> (PMML44Super.ArrayType <i>method</i>), 23		<code>exportChildren()</code> (PMML44Super.ClusteringField <i>method</i>), 47	
<code>exportChildren()</code> (PMML44Super.AssociationModel <i>method</i>), 23		<code>exportChildren()</code> (PMML44Super.ClusteringModel <i>method</i>), 48	
<code>exportChildren()</code> (PMML44Super.AssociationRule <i>method</i>), 26		<code>exportChildren()</code> (PMML44Super.ClusteringModelQuality <i>method</i>), 50	
<code>exportChildren()</code> (PMML44Super.Attribute <i>method</i>), 27		<code>exportChildren()</code> (PMML44Super.Coefficient <i>method</i>), 51	
<code>exportChildren()</code> (PMML44Super.BaseCumHazardTables <i>method</i>), 28		<code>exportChildren()</code> (PMML44Super.Coefficients <i>method</i>), 52	
<code>exportChildren()</code> (PMML44Super.Baseline <i>method</i>), 29		<code>exportChildren()</code> (PMML44Super.ComparisonMeasure <i>method</i>), 53	
<code>exportChildren()</code> (PMML44Super.BaselineCell <i>method</i>), 30		<code>exportChildren()</code> (PMML44Super.Comparisons <i>method</i>), 54	
<code>exportChildren()</code> (PMML44Super.BaselineModel <i>method</i>), 30		<code>exportChildren()</code> (PMML44Super.ComplexPartialScore <i>method</i>), 55	
<code>exportChildren()</code> (PMML44Super.BaselineStratum <i>method</i>), 32		<code>exportChildren()</code> (PMML44Super.CompoundPredicate <i>method</i>), 56	
<code>exportChildren()</code> (PMML44Super.BayesianNetworkModel <i>method</i>), 35		<code>exportChildren()</code> (PMML44Super.CompoundRule <i>method</i>), 57	
<code>exportChildren()</code> (PMML44Super.BayesianNetworkNodes <i>method</i>), 37		<code>exportChildren()</code> (PMML44Super.Con <i>method</i>), 59	
<code>exportChildren()</code> (PMML44Super.BayesInput <i>method</i>), 33		<code>exportChildren()</code> (PMML44Super.ConfusionMatrix <i>method</i>), 59	
<code>exportChildren()</code> (PMML44Super.BayesInputs <i>method</i>), 34		<code>exportChildren()</code> (PMML44Super.ConsequentSequence <i>method</i>), 60	
<code>exportChildren()</code> (PMML44Super.BayesOutput <i>method</i>), 34		<code>exportChildren()</code> (PMML44Super.Constant <i>method</i>), 61	
<code>exportChildren()</code> (PMML44Super.binarySimilarity <i>method</i>), 298			

<code>exportChildren()</code> (<i>method</i>), 61	(<i>PMML44Super.Constraints</i>	<code>exportChildren()</code> (<i>method</i>), 84	(<i>PMML44Super.DiscrStats</i>
<code>exportChildren()</code> (<i>method</i>), 64	(<i>PMML44Super.ContinuousConditionalProbability</i>	<code>exportChildren()</code> (<i>method</i>), 89	(<i>PMML44Super.DocumentTermMatrix</i>
<code>exportChildren()</code> (<i>method</i>), 65	(<i>PMML44Super.ContinuousDistribution</i>	<code>exportChildren()</code> (<i>method</i>), 90	(<i>PMML44Super.DynamicRegressor</i>
<code>exportChildren()</code> (<i>method</i>), 66	(<i>PMML44Super.ContinuousNode</i>	<code>exportChildren()</code> (<i>method</i>), 300	(<i>PMML44Super.euclidean</i>
<code>exportChildren()</code> (<i>method</i>), 63	(<i>PMML44Super.ContStats</i>	<code>exportChildren()</code> (<i>method</i>), 91	(<i>PMML44Super.EventValues</i>
<code>exportChildren()</code> (<i>method</i>), 67	(<i>PMML44Super.CorrelationFields</i>	<code>exportChildren()</code> (<i>method</i>), 92	(<i>PMML44Super.ExponentialSmoothing</i>
<code>exportChildren()</code> (<i>method</i>), 68	(<i>PMML44Super.CorrelationMethods</i>	<code>exportChildren()</code> (<i>method</i>), 93	(<i>PMML44Super.Extension</i>
<code>exportChildren()</code> (<i>method</i>), 69	(<i>PMML44Super.Correlations</i>	<code>exportChildren()</code> (<i>method</i>), 93	(<i>PMML44Super.FactorList</i>
<code>exportChildren()</code> (<i>method</i>), 68	(<i>PMML44Super.CorrelationValues</i>	<code>exportChildren()</code> (<i>method</i>), 94	(<i>PMML44Super.False_ method</i>),
<code>exportChildren()</code> (<i>method</i>), 40	(<i>PMML44Super.COUNT_TABLE_TYPE</i>	<code>exportChildren()</code> (<i>method</i>), 95	(<i>PMML44Super.FieldColumnPair</i>
<code>exportChildren()</code> (<i>method</i>), 70	(<i>PMML44Super.Counts</i>	<code>exportChildren()</code> (<i>method</i>), 95	(<i>PMML44Super.FieldRef</i>
<code>exportChildren()</code> (<i>method</i>), 71	(<i>PMML44Super.Covariances</i>	<code>exportChildren()</code> (<i>method</i>), 96	(<i>PMML44Super.FieldValue</i>
<code>exportChildren()</code> (<i>method</i>), 71	(<i>PMML44Super.CovariateList</i>	<code>exportChildren()</code> (<i>method</i>), 97	(<i>PMML44Super.FieldValueCount</i>
<code>exportChildren()</code> (<i>method</i>), 72	(<i>PMML44Super.DataDictionary</i>	<code>exportChildren()</code> (<i>method</i>), 98	(<i>PMML44Super.FinalNoise</i>
<code>exportChildren()</code> (<i>method</i>), 73	(<i>PMML44Super.DataField</i>	<code>exportChildren()</code> (<i>method</i>), 99	(<i>PMML44Super.FinalNu</i>
<code>exportChildren()</code> (<i>method</i>), 74	(<i>PMML44Super.Decision</i>	<code>exportChildren()</code> (<i>method</i>), 99	(<i>PMML44Super.FinalOmega</i>
<code>exportChildren()</code> (<i>method</i>), 77	(<i>PMML44Super.Decisions</i>	<code>exportChildren()</code> (<i>method</i>), 100	(<i>PMML44Super.FinalPredictedNoise</i>
<code>exportChildren()</code> (<i>method</i>), 75	(<i>PMML44Super.DecisionTree</i>	<code>exportChildren()</code> (<i>method</i>), 100	(<i>PMML44Super.FinalStateVector</i>
<code>exportChildren()</code> (<i>method</i>), 78	(<i>PMML44Super.DeepNetwork</i>	<code>exportChildren()</code> (<i>method</i>), 101	(<i>PMML44Super.FinalTheta</i>
<code>exportChildren()</code> (<i>method</i>), 80	(<i>PMML44Super.DefineFunction</i>	<code>exportChildren()</code> (<i>method</i>), 101	(<i>PMML44Super.GARCH</i>
<code>exportChildren()</code> (<i>method</i>), 81	(<i>PMML44Super.Delimiter</i>	<code>exportChildren()</code> (<i>method</i>), 102	(<i>PMML44Super.GARCHPart</i>
<code>exportChildren()</code> (<i>method</i>), 82	(<i>PMML44Super.Denominator</i>	<code>exportChildren()</code> (<i>method</i>), 103	(<i>PMML44Super.GaussianDistribution</i>
<code>exportChildren()</code> (<i>method</i>), 83	(<i>PMML44Super.DerivedField</i>	<code>exportChildren()</code> (<i>method</i>), 104	(<i>PMML44Super.GaussianProcessModel</i>
<code>exportChildren()</code> (<i>method</i>), 85	(<i>PMML44Super.DiscreteConditionalProbability</i>	<code>exportChildren()</code> (<i>method</i>), 109	(<i>PMML44Super.GeneralizedExponentialKernel</i>
<code>exportChildren()</code> (<i>method</i>), 86	(<i>PMML44Super.DiscreteNode</i>	<code>exportChildren()</code> (<i>method</i>), 106	(<i>PMML44Super.GeneralRegressionModel</i>
<code>exportChildren()</code> (<i>method</i>), 87	(<i>PMML44Super.Discretize</i>	<code>exportChildren()</code> (<i>method</i>), 110	(<i>PMML44Super.Header</i>
<code>exportChildren()</code> (<i>method</i>), 88	(<i>PMML44Super.DiscretizeBin</i>	<code>exportChildren()</code> (<i>method</i>), 110	(<i>PMML44Super.HVector</i>

<code>exportChildren()</code> (<i>method</i>), 113	(<i>PMML44Super.Indices</i>	<code>exportChildren()</code> (<i>PMML44Super.Losses method</i>), 134
<code>exportChildren()</code> (<i>method</i>), 113	(<i>PMML44Super.InlineTable</i>	<code>exportChildren()</code> (<i>PMML44Super.Lower method</i>), 135
<code>exportChildren()</code> (<i>method</i>), 114	(<i>PMML44Super.InstanceField</i>	<code>exportChildren()</code> (<i>PMML44Super.MA method</i>), 136
<code>exportChildren()</code> (<i>method</i>), 115	(<i>PMML44Super.InstanceFields</i>	<code>exportChildren()</code> (<i>PMML44Super.MACoefficients</i> <i>method</i>), 137
<code>exportChildren()</code> (<i>method</i>), 111	(<i>PMML44Super.INT_Entries</i>	<code>exportChildren()</code> (<i>PMML44Super.MapValues</i> <i>method</i>), 138
<code>exportChildren()</code> (<i>PMML44Super.INT_SparseArray</i> <i>method</i>), 112		<code>exportChildren()</code> (<i>PMML44Super.MatCell</i> <i>method</i>), 139
<code>exportChildren()</code> (<i>method</i>), 115	(<i>PMML44Super.Interval</i>	<code>exportChildren()</code> (<i>PMML44Super.Matrix method</i>), 139
<code>exportChildren()</code> (<i>PMML44Super.Item method</i>), 116		<code>exportChildren()</code> (<i>PMML44Super.MaximumLikelihoodStat</i> <i>method</i>), 140
<code>exportChildren()</code> (<i>method</i>), 117	(<i>PMML44Super.ItemRef</i>	<code>exportChildren()</code> (<i>PMML44Super.Mean method</i>), 141
<code>exportChildren()</code> (<i>method</i>), 118	(<i>PMML44Super.Itemset</i>	<code>exportChildren()</code> (<i>PMML44Super.MeasurementMatrix</i> <i>method</i>), 142
<code>exportChildren()</code> (<i>method</i>), 301	(<i>PMML44Super.jaccard</i>	<code>exportChildren()</code> (<i>PMML44Super.Metrics</i> <i>method</i>), 143
<code>exportChildren()</code> (<i>method</i>), 120	(<i>PMML44Super.KalmanState</i>	<code>exportChildren()</code> (<i>PMML44Super.MiningBuildTask</i> <i>method</i>), 144
<code>exportChildren()</code> (<i>method</i>), 119	(<i>PMML44Super.KNNInput</i>	<code>exportChildren()</code> (<i>PMML44Super.MiningField</i> <i>method</i>), 144
<code>exportChildren()</code> (<i>method</i>), 120	(<i>PMML44Super.KNNInputs</i>	<code>exportChildren()</code> (<i>PMML44Super.MiningModel</i> <i>method</i>), 146
<code>exportChildren()</code> (<i>method</i>), 121	(<i>PMML44Super.KohonenMap</i>	<code>exportChildren()</code> (<i>PMML44Super.MiningSchema</i> <i>method</i>), 147
<code>exportChildren()</code> (<i>PMML44Super.Lag method</i>), 122		<code>exportChildren()</code> (<i>PMML44Super.minkowski</i> <i>method</i>), 302
<code>exportChildren()</code> (<i>method</i>), 123	(<i>PMML44Super.Lambda</i>	<code>exportChildren()</code> (<i>PMML44Super.MissingValueWeights</i> <i>method</i>), 148
<code>exportChildren()</code> (<i>method</i>), 123	(<i>PMML44Super.LayerBias</i>	<code>exportChildren()</code> (<i>PMML44Super.ModelExplanation</i> <i>method</i>), 149
<code>exportChildren()</code> (<i>PMML44Super.LayerParameters</i> <i>method</i>), 124		<code>exportChildren()</code> (<i>PMML44Super.ModelLiftGraph</i> <i>method</i>), 150
<code>exportChildren()</code> (<i>method</i>), 128	(<i>PMML44Super.LayerWeights</i>	<code>exportChildren()</code> (<i>PMML44Super.ModelStats</i> <i>method</i>), 150
<code>exportChildren()</code> (<i>PMML44Super.Level method</i>), 129		<code>exportChildren()</code> (<i>PMML44Super.ModelVerification</i> <i>method</i>), 151
<code>exportChildren()</code> (<i>method</i>), 130	(<i>PMML44Super.LiftData</i>	<code>exportChildren()</code> (<i>PMML44Super.MultivariateStat</i> <i>method</i>), 152
<code>exportChildren()</code> (<i>method</i>), 131	(<i>PMML44Super.LiftGraph</i>	<code>exportChildren()</code> (<i>PMML44Super.MultivariateStats</i> <i>method</i>), 154
<code>exportChildren()</code> (<i>PMML44Super.LinearKernelType</i> <i>method</i>), 131		<code>exportChildren()</code> (<i>PMML44Super.Nadam</i> <i>method</i>), 155
<code>exportChildren()</code> (<i>method</i>), 132	(<i>PMML44Super.LinearNorm</i>	<code>exportChildren()</code> (<i>PMML44Super.NaiveBayesModel</i> <i>method</i>), 156
<code>exportChildren()</code> (<i>PMML44Super.LocalTransformations</i> <i>method</i>), 133		<code>exportChildren()</code> (<i>PMML44Super.NearestNeighborModel</i> <i>method</i>), 157
<code>exportChildren()</code> (<i>PMML44Super.LognormalDistributionForBN</i> <i>method</i>), 134		<code>exportChildren()</code> (<i>PMML44Super.NetworkLayer</i> <i>method</i>), 159

<code>exportChildren()</code> <i>method</i>), 160	(<i>PMML44Super.NeuralInput</i>	<code>exportChildren()</code> <i>method</i>), 195	(<i>PMML44Super.Partition</i>
<code>exportChildren()</code> <i>method</i>), 161	(<i>PMML44Super.NeuralInputs</i>	<code>exportChildren()</code> <i>method</i>), 196	(<i>PMML44Super.PartitionFieldStats</i>
<code>exportChildren()</code> <i>method</i>), 162	(<i>PMML44Super.NeuralLayer</i>	<code>exportChildren()</code> <i>method</i>), 197	(<i>PMML44Super.PastVariances</i>
<code>exportChildren()</code> <i>method</i>), 163	(<i>PMML44Super.NeuralNetwork</i>	<code>exportChildren()</code> <i>method</i>), 183	(<i>PMML44Super.PCell</i>
<code>exportChildren()</code> <i>method</i>), 165	(<i>PMML44Super.NeuralOutput</i>	<code>exportChildren()</code> <i>method</i>), 183	(<i>PMML44Super.PCovCell</i>
<code>exportChildren()</code> <i>method</i>), 166	(<i>PMML44Super.NeuralOutputs</i>	<code>exportChildren()</code> <i>method</i>), 184	(<i>PMML44Super.PCovMatrix</i>
<code>exportChildren()</code> <i>method</i>), 166	(<i>PMML44Super.Neuron</i>	<code>exportChildren()</code> <i>method</i>), 186	(<i>PMML44Super.PMML</i>
<code>exportChildren()</code> <i>method</i>), 168	(<i>PMML44Super.Node</i>	<code>exportChildren()</code> <i>method</i>), 198	(<i>PMML44Super.PoissonDistribution</i>
<code>exportChildren()</code> <i>method</i>), 169	(<i>PMML44Super.NonseasonalComponent</i>	<code>exportChildren()</code> <i>method</i>), 198	(<i>PMML44Super.PolynomialKernelType</i>
<code>exportChildren()</code> <i>method</i>), 170	(<i>PMML44Super.NonseasonalFactor</i>	<code>exportChildren()</code> <i>method</i>), 189	(<i>PMML44Super.PPCell</i>
<code>exportChildren()</code> <i>method</i>), 173	(<i>PMML44Super.NormalDistributionForBN</i>	<code>exportChildren()</code> <i>method</i>), 190	(<i>PMML44Super.PPMatrix</i>
<code>exportChildren()</code> <i>method</i>), 171	(<i>PMML44Super.NormContinuous</i>	<code>exportChildren()</code> <i>method</i>), 200	(<i>PMML44Super.PredictiveModelQuality</i>
<code>exportChildren()</code> <i>method</i>), 172	(<i>PMML44Super.NormDiscrete</i>	<code>exportChildren()</code> <i>method</i>), 201	(<i>PMML44Super.Predictor</i>
<code>exportChildren()</code> <i>method</i>), 173	(<i>PMML44Super.Numerator</i>	<code>exportChildren()</code> <i>method</i>), 202	(<i>PMML44Super.PredictorTerm</i>
<code>exportChildren()</code> <i>method</i>), 174	(<i>PMML44Super.NumericInfo</i>	<code>exportChildren()</code> <i>method</i>), 203	(<i>PMML44Super.PsiVector</i>
<code>exportChildren()</code> <i>method</i>), 175	(<i>PMML44Super.NumericPredictor</i>	<code>exportChildren()</code> <i>method</i>), 204	(<i>PMML44Super.Quantile</i>
<code>exportChildren()</code> <i>method</i>), 176	(<i>PMML44Super.Optimizers</i>	<code>exportChildren()</code> <i>method</i>), 209	(<i>PMML44Super.RadialBasisKernel</i>
<code>exportChildren()</code> <i>method</i>), 178	(<i>PMML44Super.OptimumLiftGraph</i>	<code>exportChildren()</code> <i>method</i>), 209	(<i>PMML44Super.RadialBasisKernelType</i>
<code>exportChildren()</code> <i>method</i>), 179	(<i>PMML44Super.OutlierEffect</i>	<code>exportChildren()</code> <i>method</i>), 210	(<i>PMML44Super.RandomLiftGraph</i>
<code>exportChildren()</code> <i>method</i>), 179	(<i>PMML44Super.Output</i>	<code>exportChildren()</code> <i>method</i>), 205	(<i>PMML44Super.REAL_Entries</i>
<code>exportChildren()</code> <i>method</i>), 180	(<i>PMML44Super.OutputField</i>	<code>exportChildren()</code> <i>method</i>), 205	(<i>PMML44Super.REAL_SparseArray</i>
<code>exportChildren()</code> <i>method</i>), 191	(<i>PMML44Super.PairCounts</i>	<code>exportChildren()</code> <i>method</i>), 211	(<i>PMML44Super.Regression</i>
<code>exportChildren()</code> <i>method</i>), 192	(<i>PMML44Super.Parameter</i>	<code>exportChildren()</code> <i>method</i>), 212	(<i>PMML44Super.RegressionModel</i>
<code>exportChildren()</code> <i>method</i>), 193	(<i>PMML44Super.ParameterField</i>	<code>exportChildren()</code> <i>method</i>), 214	(<i>PMML44Super.RegressionTable</i>
<code>exportChildren()</code> <i>method</i>), 194	(<i>PMML44Super.ParameterList</i>	<code>exportChildren()</code> <i>method</i>), 215	(<i>PMML44Super.RegressorValues</i>
<code>exportChildren()</code> <i>method</i>), 191	(<i>PMML44Super.ParamMatrix</i>	<code>exportChildren()</code> <i>method</i>), 217	(<i>PMML44Super.Residuals</i>
<code>exportChildren()</code> <i>method</i>), 194	(<i>PMML44Super.ParentValue</i>	<code>exportChildren()</code> <i>method</i>), 216	(<i>PMML44Super.ResidualSquareCoefficients</i>

<code>exportChildren()</code> (<i>PMML44Super.ResultField</i> <i>method</i>), 217	<code>exportChildren()</code> (<i>PMML44Super.SimpleSetPredicate</i> <i>method</i>), 242
<code>exportChildren()</code> (<i>PMML44Super.RMSprop</i> <i>method</i>), 206	<code>exportChildren()</code> (<i>PMML44Super.SpectralAnalysis</i> <i>method</i>), 243
<code>exportChildren()</code> (<i>PMML44Super.ROC</i> <i>method</i>), 207	<code>exportChildren()</code> (<i>PMML44Super.squaredEuclidean</i> <i>method</i>), 303
<code>exportChildren()</code> (<i>PMML44Super.ROCGraph</i> <i>method</i>), 208	<code>exportChildren()</code> (<i>PMML44Super.StateSpaceModel</i> <i>method</i>), 243
<code>exportChildren()</code> (<i>PMML44Super.row</i> <i>method</i>), 302	<code>exportChildren()</code> (<i>PMML44Super.StateVector</i> <i>method</i>), 244
<code>exportChildren()</code> (<i>PMML44Super.RuleSelectionMethod</i> <i>method</i>), 218	<code>exportChildren()</code> (<i>PMML44Super.SupportVector</i> <i>method</i>), 245
<code>exportChildren()</code> (<i>PMML44Super.RuleSet</i> <i>method</i>), 219	<code>exportChildren()</code> (<i>PMML44Super.SupportVectorMachine</i> <i>method</i>), 246
<code>exportChildren()</code> (<i>PMML44Super.RuleSetModel</i> <i>method</i>), 220	<code>exportChildren()</code> (<i>PMML44Super.SupportVectorMachineModel</i> <i>method</i>), 247
<code>exportChildren()</code> (<i>PMML44Super.Scorecard</i> <i>method</i>), 224	<code>exportChildren()</code> (<i>PMML44Super.SupportVectors</i> <i>method</i>), 249
<code>exportChildren()</code> (<i>PMML44Super.ScoreDistribution</i> <i>method</i>), 223	<code>exportChildren()</code> (<i>PMML44Super.TableLocator</i> <i>method</i>), 250
<code>exportChildren()</code> (<i>PMML44Super.SeasonalComponent</i> <i>method</i>), 225	<code>exportChildren()</code> (<i>PMML44Super.tanimoto</i> <i>method</i>), 304
<code>exportChildren()</code> (<i>PMML44Super.SeasonalFactor</i> <i>method</i>), 226	<code>exportChildren()</code> (<i>PMML44Super.Target</i> <i>method</i>), 250
<code>exportChildren()</code> (<i>PMML44Super.Seasonality_Exponential</i> <i>method</i>), 228	<code>exportChildren()</code> (<i>PMML44Super.Targets</i> <i>method</i>), 256
<code>exportChildren()</code> (<i>PMML44Super.SeasonalTrendDecomposition</i> <i>method</i>), 227	<code>exportChildren()</code> (<i>PMML44Super.TargetValue</i> <i>method</i>), 252
<code>exportChildren()</code> (<i>PMML44Super.Segment</i> <i>method</i>), 229	<code>exportChildren()</code> (<i>PMML44Super.TargetValueCount</i> <i>method</i>), 253
<code>exportChildren()</code> (<i>PMML44Super.Segmentation</i> <i>method</i>), 231	<code>exportChildren()</code> (<i>PMML44Super.TargetValueCounts</i> <i>method</i>), 253
<code>exportChildren()</code> (<i>PMML44Super.Sequence</i> <i>method</i>), 232	<code>exportChildren()</code> (<i>PMML44Super.TargetValueStat</i> <i>method</i>), 254
<code>exportChildren()</code> (<i>PMML44Super.SequenceModel</i> <i>method</i>), 233	<code>exportChildren()</code> (<i>PMML44Super.TargetValueStats</i> <i>method</i>), 255
<code>exportChildren()</code> (<i>PMML44Super.SequenceReference</i> <i>method</i>), 235	<code>exportChildren()</code> (<i>PMML44Super.Taxonomy</i> <i>method</i>), 256
<code>exportChildren()</code> (<i>PMML44Super.SequenceRule</i> <i>method</i>), 236	<code>exportChildren()</code> (<i>PMML44Super.TestDistributions</i> <i>method</i>), 257
<code>exportChildren()</code> (<i>PMML44Super.SetPredicate</i> <i>method</i>), 237	<code>exportChildren()</code> (<i>PMML44Super.TextCorpus</i> <i>method</i>), 258
<code>exportChildren()</code> (<i>PMML44Super.SetReference</i> <i>method</i>), 238	<code>exportChildren()</code> (<i>PMML44Super.TextDictionary</i> <i>method</i>), 259
<code>exportChildren()</code> (<i>PMML44Super.SGD</i> <i>method</i>), 222	<code>exportChildren()</code> (<i>PMML44Super.TextDocument</i> <i>method</i>), 260
<code>exportChildren()</code> (<i>PMML44Super.SigmoidKernelType</i> <i>method</i>), 239	<code>exportChildren()</code> (<i>PMML44Super.TextIndex</i> <i>method</i>), 261
<code>exportChildren()</code> (<i>PMML44Super.simpleMatching</i> <i>method</i>), 303	<code>exportChildren()</code> (<i>PMML44Super.TextIndexNormalization</i> <i>method</i>), 263
<code>exportChildren()</code> (<i>PMML44Super.SimplePredicate</i> <i>method</i>), 240	<code>exportChildren()</code> (<i>PMML44Super.TextModel</i> <i>method</i>), 264
<code>exportChildren()</code> (<i>PMML44Super.SimpleRule</i> <i>method</i>), 240	<code>exportChildren()</code> (<i>PMML44Super.TextModelNormalization</i> <i>method</i>), 266

<code>exportChildren()</code> (PMML44Super.TextModelSimilarity method), 266	<code>exportChildren()</code> (PMML44Super.VariableWeight method), 290
<code>exportChildren()</code> (PMML44Super.Theta method), 267	<code>exportChildren()</code> (PMML44Super.Variance method), 291
<code>exportChildren()</code> (PMML44Super.ThetaRecursionState method), 268	<code>exportChildren()</code> (PMML44Super.VarianceCoefficients method), 292
<code>exportChildren()</code> (PMML44Super.Time method), 268	<code>exportChildren()</code> (PMML44Super.VectorDictionary method), 293
<code>exportChildren()</code> (PMML44Super.TimeAnchor method), 269	<code>exportChildren()</code> (PMML44Super.VectorFields method), 293
<code>exportChildren()</code> (PMML44Super.TimeCycle method), 270	<code>exportChildren()</code> (PMML44Super.VectorInstance method), 294
<code>exportChildren()</code> (PMML44Super.TimeException method), 271	<code>exportChildren()</code> (PMML44Super.VerificationField method), 295
<code>exportChildren()</code> (PMML44Super.TimeSeries method), 272	<code>exportChildren()</code> (PMML44Super.VerificationFields method), 296
<code>exportChildren()</code> (PMML44Super.TimeSeriesModel method), 273	<code>exportChildren()</code> (PMML44Super.XCoordinates method), 297
<code>exportChildren()</code> (PMML44Super.Timestamp method), 275	<code>exportChildren()</code> (PMML44Super.YCoordinates method), 297
<code>exportChildren()</code> (PMML44Super.TimeValue method), 275	<code>exportLiteral()</code> (PMML44Super.AbsoluteExponentialKernel method), 7
<code>exportChildren()</code> (PMML44Super.TrainingInstances method), 276	<code>exportLiteral()</code> (PMML44Super.Adadelta method), 8
<code>exportChildren()</code> (PMML44Super.TrainingParameters method), 277	<code>exportLiteral()</code> (PMML44Super.Adagrad method), 9
<code>exportChildren()</code> (PMML44Super.TransferFunctionValues method), 278	<code>exportLiteral()</code> (PMML44Super.Adam method), 10
<code>exportChildren()</code> (PMML44Super.TransformationDictionary method), 279	<code>exportLiteral()</code> (PMML44Super.Adamax method), 11
<code>exportChildren()</code> (PMML44Super.TransitionMatrix method), 280	<code>exportLiteral()</code> (PMML44Super.Aggregate method), 12
<code>exportChildren()</code> (PMML44Super.TreeModel method), 280	<code>exportLiteral()</code> (PMML44Super.Alternate method), 12
<code>exportChildren()</code> (PMML44Super.Trend_ExpoSmooth method), 283	<code>exportLiteral()</code> (PMML44Super.Annotation method), 13
<code>exportChildren()</code> (PMML44Super.TrendCoefficients method), 282	<code>exportLiteral()</code> (PMML44Super.AnomalyDetectionModel method), 14
<code>exportChildren()</code> (PMML44Super.TriangularDistributionForBN method), 283	<code>exportLiteral()</code> (PMML44Super.Anova method), 16
<code>exportChildren()</code> (PMML44Super.True_ method), 284	<code>exportLiteral()</code> (PMML44Super.AnovaRow method), 17
<code>exportChildren()</code> (PMML44Super.UniformDistribution method), 285	<code>exportLiteral()</code> (PMML44Super.AntecedentSequence method), 18
<code>exportChildren()</code> (PMML44Super.UniformDistributionForBN method), 286	<code>exportLiteral()</code> (PMML44Super.AnyDistribution method), 19
<code>exportChildren()</code> (PMML44Super.UnivariateStats method), 286	<code>exportLiteral()</code> (PMML44Super.Application method), 20
<code>exportChildren()</code> (PMML44Super.Upper method), 287	<code>exportLiteral()</code> (PMML44Super.Apply method), 21
<code>exportChildren()</code> (PMML44Super.Value method), 288	<code>exportLiteral()</code> (PMML44Super.AR method), 3
<code>exportChildren()</code> (PMML44Super.ValueProbability method), 289	<code>exportLiteral()</code> (PMML44Super.ARDSquaredExponentialKernel method), 4
	<code>exportLiteral()</code> (PMML44Super.ARIMA method),

5

`exportLiteral()` (PMML44Super.ARMAPart *method*), 300

`exportLiteral()` (PMML44Super.ArrayType *method*), 6

`exportLiteral()` (PMML44Super.AssociationModel *method*), 23

`exportLiteral()` (PMML44Super.AssociationRule *method*), 24

`exportLiteral()` (PMML44Super.Attribute *method*), 26

`exportLiteral()` (PMML44Super.BaseCumHazardTables *method*), 27

`exportLiteral()` (PMML44Super.Baseline *method*), 28

`exportLiteral()` (PMML44Super.BaselineCell *method*), 29

`exportLiteral()` (PMML44Super.BaselineModel *method*), 30

`exportLiteral()` (PMML44Super.BaselineStratum *method*), 30

`exportLiteral()` (PMML44Super.BayesianNetworkModel *method*), 32

`exportLiteral()` (PMML44Super.BayesianNetworkNodes *method*), 35

`exportLiteral()` (PMML44Super.BayesInput *method*), 37

`exportLiteral()` (PMML44Super.BayesInputs *method*), 33

`exportLiteral()` (PMML44Super.BayesOutput *method*), 34

`exportLiteral()` (PMML44Super.binarySimilarity *method*), 298

`exportLiteral()` (PMML44Super.BlockIndicator *method*), 38

`exportLiteral()` (PMML44Super.BoundaryValueMeans *method*), 38

`exportLiteral()` (PMML44Super.BoundaryValues *method*), 39

`exportLiteral()` (PMML44Super.CategoricalPredictor *method*), 41

`exportLiteral()` (PMML44Super.Categories *method*), 42

`exportLiteral()` (PMML44Super.Category *method*), 42

`exportLiteral()` (PMML44Super.Characteristic *method*), 43

`exportLiteral()` (PMML44Super.Characteristics *method*), 44

`exportLiteral()` (PMML44Super.chebychev *method*), 299

`exportLiteral()` (PMML44Super.ChildParent *method*), 45

`exportLiteral()` (PMML44Super.cityBlock *method*), 300

`exportLiteral()` (PMML44Super.ClassLabels *method*), 46

`exportLiteral()` (PMML44Super.Cluster *method*), 46

`exportLiteral()` (PMML44Super.ClusteringField *method*), 47

`exportLiteral()` (PMML44Super.ClusteringModel *method*), 49

`exportLiteral()` (PMML44Super.ClusteringModelQuality *method*), 50

`exportLiteral()` (PMML44Super.Coefficient *method*), 51

`exportLiteral()` (PMML44Super.Coefficients *method*), 52

`exportLiteral()` (PMML44Super.ComparisonMeasure *method*), 53

`exportLiteral()` (PMML44Super.Comparisons *method*), 54

`exportLiteral()` (PMML44Super.ComplexPartialScore *method*), 55

`exportLiteral()` (PMML44Super.CompoundPredicate *method*), 56

`exportLiteral()` (PMML44Super.CompoundRule *method*), 57

`exportLiteral()` (PMML44Super.Con *method*), 59

`exportLiteral()` (PMML44Super.ConfusionMatrix *method*), 59

`exportLiteral()` (PMML44Super.ConsequentSequence *method*), 60

`exportLiteral()` (PMML44Super.Constant *method*), 61

`exportLiteral()` (PMML44Super.Constraints *method*), 62

`exportLiteral()` (PMML44Super.ContinuousConditionalProbability *method*), 64

`exportLiteral()` (PMML44Super.ContinuousDistribution *method*), 65

`exportLiteral()` (PMML44Super.ContinuousNode *method*), 66

`exportLiteral()` (PMML44Super.ContStats *method*), 63

`exportLiteral()` (PMML44Super.CorrelationFields *method*), 67

`exportLiteral()` (PMML44Super.CorrelationMethods *method*), 68

`exportLiteral()` (PMML44Super.Correlations *method*), 69

`exportLiteral()` (PMML44Super.CorrelationValues *method*), 68

`exportLiteral()` (PMML44Super.COUNT_TABLE_TYPE *method*), 40

`exportLiteral()` (PMML44Super.Counts *method*), 70

<code>exportLiteral()</code> (<i>method</i>), 71	(<i>PMML44Super.Covariances</i>	<code>exportLiteral()</code> (<i>method</i>), 96	(<i>PMML44Super.FieldValue</i>
<code>exportLiteral()</code> (<i>method</i>), 71	(<i>PMML44Super.CovariateList</i>	<code>exportLiteral()</code> (<i>method</i>), 97	(<i>PMML44Super.FieldValueCount</i>
<code>exportLiteral()</code> (<i>method</i>), 72	(<i>PMML44Super.DataDictionary</i>	<code>exportLiteral()</code> (<i>method</i>), 98	(<i>PMML44Super.FinalNoise</i>
<code>exportLiteral()</code> (<i>method</i>), 73	(<i>PMML44Super.DataField</i>	<code>exportLiteral()</code> (<i>method</i>), 99	(<i>PMML44Super.FinalNu</i>
<code>exportLiteral()</code> (<i>method</i>), 74	(<i>PMML44Super.Decision</i>	<code>exportLiteral()</code> (<i>method</i>), 99	(<i>PMML44Super.FinalOmega</i>
<code>exportLiteral()</code> (<i>method</i>), 77	(<i>PMML44Super.Decisions</i>	<code>exportLiteral()</code> (<i>method</i>), 100	(<i>PMML44Super.FinalPredictedNoise</i>
<code>exportLiteral()</code> (<i>method</i>), 75	(<i>PMML44Super.DecisionTree</i>	<code>exportLiteral()</code> (<i>method</i>), 100	(<i>PMML44Super.FinalStateVector</i>
<code>exportLiteral()</code> (<i>method</i>), 78	(<i>PMML44Super.DeepNetwork</i>	<code>exportLiteral()</code> (<i>method</i>), 101	(<i>PMML44Super.FinalTheta</i>
<code>exportLiteral()</code> (<i>method</i>), 80	(<i>PMML44Super.DefineFunction</i>	<code>exportLiteral()</code> (<i>method</i>), 101	(<i>PMML44Super.GARCH</i>
<code>exportLiteral()</code> (<i>method</i>), 81	(<i>PMML44Super.Delimiter</i>	<code>exportLiteral()</code> (<i>method</i>), 102	(<i>PMML44Super.GARCHPart</i>
<code>exportLiteral()</code> (<i>method</i>), 82	(<i>PMML44Super.Denominator</i>	<code>exportLiteral()</code> (<i>method</i>), 103	(<i>PMML44Super.GaussianDistribution</i>
<code>exportLiteral()</code> (<i>method</i>), 83	(<i>PMML44Super.DerivedField</i>	<code>exportLiteral()</code> (<i>method</i>), 104	(<i>PMML44Super.GaussianProcessModel</i>
<code>exportLiteral()</code> (<i>method</i>), 85	(<i>PMML44Super.DiscreteConditionalProbability</i>	<code>exportLiteral()</code> (<i>method</i>), 109	(<i>PMML44Super.GeneralizedExponentialKernel</i>
<code>exportLiteral()</code> (<i>method</i>), 86	(<i>PMML44Super.DiscreteNode</i>	<code>exportLiteral()</code> (<i>method</i>), 106	(<i>PMML44Super.GeneralRegressionModel</i>
<code>exportLiteral()</code> (<i>method</i>), 87	(<i>PMML44Super.Discretize</i>	<code>exportLiteral()</code> (<i>method</i>), 111	(<i>PMML44Super.Header</i>
<code>exportLiteral()</code> (<i>method</i>), 88	(<i>PMML44Super.DiscretizeBin</i>	<code>exportLiteral()</code> (<i>method</i>), 110	(<i>PMML44Super.HVector</i>
<code>exportLiteral()</code> (<i>method</i>), 84	(<i>PMML44Super.DiscrStats</i>	<code>exportLiteral()</code> (<i>method</i>), 113	(<i>PMML44Super.Indices</i>
<code>exportLiteral()</code> (<i>method</i>), 89	(<i>PMML44Super.DocumentTermMatrix</i>	<code>exportLiteral()</code> (<i>method</i>), 113	(<i>PMML44Super.InlineTable</i>
<code>exportLiteral()</code> (<i>method</i>), 90	(<i>PMML44Super.DynamicRegressor</i>	<code>exportLiteral()</code> (<i>method</i>), 114	(<i>PMML44Super.InstanceField</i>
<code>exportLiteral()</code> (<i>method</i>), 300	(<i>PMML44Super.euclidean</i>	<code>exportLiteral()</code> (<i>method</i>), 115	(<i>PMML44Super.InstanceFields</i>
<code>exportLiteral()</code> (<i>method</i>), 91	(<i>PMML44Super.EventValues</i>	<code>exportLiteral()</code> (<i>method</i>), 111	(<i>PMML44Super.INT_Entries</i>
<code>exportLiteral()</code> (<i>method</i>), 92	(<i>PMML44Super.ExponentialSmoothing</i>	<code>exportLiteral()</code> (<i>method</i>), 112	(<i>PMML44Super.INT_SparseArray</i>
<code>exportLiteral()</code> (<i>method</i>), 93	(<i>PMML44Super.Extension</i>	<code>exportLiteral()</code> (<i>method</i>), 115	(<i>PMML44Super.Interval</i>
<code>exportLiteral()</code> (<i>method</i>), 94	(<i>PMML44Super.FactorList</i>	<code>exportLiteral()</code> (<i>method</i>), 116	(<i>PMML44Super.Item</i>
<code>exportLiteral()</code> (<i>method</i>), 94	(<i>PMML44Super.False_</i>	<code>exportLiteral()</code> (<i>method</i>), 117	(<i>PMML44Super.ItemRef</i>
<code>exportLiteral()</code> (<i>method</i>), 95	(<i>PMML44Super.FieldColumnPair</i>	<code>exportLiteral()</code> (<i>method</i>), 118	(<i>PMML44Super.Itemset</i>
<code>exportLiteral()</code> (<i>method</i>), 96	(<i>PMML44Super.FieldRef</i>	<code>exportLiteral()</code> (<i>method</i>), 301	(<i>PMML44Super.jaccard</i>

<code>exportLiteral()</code> (PMML44Super.KalmanState method), 120	<code>exportLiteral()</code> (PMML44Super.MiningField method), 144
<code>exportLiteral()</code> (PMML44Super.KNNInput method), 119	<code>exportLiteral()</code> (PMML44Super.MiningModel method), 146
<code>exportLiteral()</code> (PMML44Super.KNNInputs method), 120	<code>exportLiteral()</code> (PMML44Super.MiningSchema method), 147
<code>exportLiteral()</code> (PMML44Super.KohonenMap method), 121	<code>exportLiteral()</code> (PMML44Super.minkowski method), 302
<code>exportLiteral()</code> (PMML44Super.Lag method), 122	<code>exportLiteral()</code> (PMML44Super.MissingValueWeights method), 148
<code>exportLiteral()</code> (PMML44Super.Lambda method), 123	<code>exportLiteral()</code> (PMML44Super.ModelExplanation method), 149
<code>exportLiteral()</code> (PMML44Super.LayerBias method), 123	<code>exportLiteral()</code> (PMML44Super.ModelLiftGraph method), 150
<code>exportLiteral()</code> (PMML44Super.LayerParameters method), 124	<code>exportLiteral()</code> (PMML44Super.ModelStats method), 150
<code>exportLiteral()</code> (PMML44Super.LayerWeights method), 128	<code>exportLiteral()</code> (PMML44Super.ModelVerification method), 151
<code>exportLiteral()</code> (PMML44Super.Level method), 129	<code>exportLiteral()</code> (PMML44Super.MultivariateStat method), 152
<code>exportLiteral()</code> (PMML44Super.LiftData method), 130	<code>exportLiteral()</code> (PMML44Super.MultivariateStats method), 154
<code>exportLiteral()</code> (PMML44Super.LiftGraph method), 131	<code>exportLiteral()</code> (PMML44Super.Nadam method), 155
<code>exportLiteral()</code> (PMML44Super.LinearKernelType method), 131	<code>exportLiteral()</code> (PMML44Super.NaiveBayesModel method), 156
<code>exportLiteral()</code> (PMML44Super.LinearNorm method), 132	<code>exportLiteral()</code> (PMML44Super.NearestNeighborModel method), 157
<code>exportLiteral()</code> (PMML44Super.LocalTransformations method), 133	<code>exportLiteral()</code> (PMML44Super.NetworkLayer method), 160
<code>exportLiteral()</code> (PMML44Super.LognormalDistributionForBN method), 134	<code>exportLiteral()</code> (PMML44Super.NeuralInput method), 161
<code>exportLiteral()</code> (PMML44Super.Losses method), 134	<code>exportLiteral()</code> (PMML44Super.NeuralInputs method), 161
<code>exportLiteral()</code> (PMML44Super.Lower method), 135	<code>exportLiteral()</code> (PMML44Super.NeuralLayer method), 162
<code>exportLiteral()</code> (PMML44Super.MA method), 136	<code>exportLiteral()</code> (PMML44Super.NeuralNetwork method), 163
<code>exportLiteral()</code> (PMML44Super.MACoefficients method), 137	<code>exportLiteral()</code> (PMML44Super.NeuralOutput method), 165
<code>exportLiteral()</code> (PMML44Super.MapValues method), 138	<code>exportLiteral()</code> (PMML44Super.NeuralOutputs method), 166
<code>exportLiteral()</code> (PMML44Super.MatCell method), 139	<code>exportLiteral()</code> (PMML44Super.Neuron method), 167
<code>exportLiteral()</code> (PMML44Super.Matrix method), 139	<code>exportLiteral()</code> (PMML44Super.Node method), 168
<code>exportLiteral()</code> (PMML44Super.MaximumLikelihoodStep method), 140	<code>exportLiteral()</code> (PMML44Super.NonseasonalComponent method), 169
<code>exportLiteral()</code> (PMML44Super.Mean method), 141	<code>exportLiteral()</code> (PMML44Super.NonseasonalFactor method), 170
<code>exportLiteral()</code> (PMML44Super.MeasurementMatrix method), 142	<code>exportLiteral()</code> (PMML44Super.NormalDistributionForBN method), 173
<code>exportLiteral()</code> (PMML44Super.Metrics method), 143	<code>exportLiteral()</code> (PMML44Super.NormContinuous method), 171
<code>exportLiteral()</code> (PMML44Super.MiningBuildTask method), 144	

<code>exportLiteral()</code> (PMML44Super.NormDiscrete method), 172	<code>exportLiteral()</code> (PMML44Super.Predictor method), 202
<code>exportLiteral()</code> (PMML44Super.Numerator method), 174	<code>exportLiteral()</code> (PMML44Super.PredictorTerm method), 202
<code>exportLiteral()</code> (PMML44Super.NumericInfo method), 174	<code>exportLiteral()</code> (PMML44Super.PsiVector method), 203
<code>exportLiteral()</code> (PMML44Super.NumericPredictor method), 175	<code>exportLiteral()</code> (PMML44Super.Quantile method), 204
<code>exportLiteral()</code> (PMML44Super.Optimizers method), 176	<code>exportLiteral()</code> (PMML44Super.RadialBasisKernel method), 209
<code>exportLiteral()</code> (PMML44Super.OptimumLiftGraph method), 178	<code>exportLiteral()</code> (PMML44Super.RadialBasisKernelType method), 209
<code>exportLiteral()</code> (PMML44Super.OutlierEffect method), 179	<code>exportLiteral()</code> (PMML44Super.RandomLiftGraph method), 210
<code>exportLiteral()</code> (PMML44Super.Output method), 179	<code>exportLiteral()</code> (PMML44Super.REAL_Entries method), 205
<code>exportLiteral()</code> (PMML44Super.OutputField method), 180	<code>exportLiteral()</code> (PMML44Super.REAL_SparseArray method), 205
<code>exportLiteral()</code> (PMML44Super.PairCounts method), 191	<code>exportLiteral()</code> (PMML44Super.Regression method), 211
<code>exportLiteral()</code> (PMML44Super.Parameter method), 192	<code>exportLiteral()</code> (PMML44Super.RegressionModel method), 212
<code>exportLiteral()</code> (PMML44Super.ParameterField method), 193	<code>exportLiteral()</code> (PMML44Super.RegressionTable method), 214
<code>exportLiteral()</code> (PMML44Super.ParameterList method), 194	<code>exportLiteral()</code> (PMML44Super.RegressorValues method), 215
<code>exportLiteral()</code> (PMML44Super.ParamMatrix method), 191	<code>exportLiteral()</code> (PMML44Super.Residuals method), 217
<code>exportLiteral()</code> (PMML44Super.ParentValue method), 194	<code>exportLiteral()</code> (PMML44Super.ResidualSquareCoefficients method), 216
<code>exportLiteral()</code> (PMML44Super.Partition method), 195	<code>exportLiteral()</code> (PMML44Super.ResultField method), 217
<code>exportLiteral()</code> (PMML44Super.PartitionFieldStats method), 196	<code>exportLiteral()</code> (PMML44Super.RMSprop method), 206
<code>exportLiteral()</code> (PMML44Super.PastVariances method), 197	<code>exportLiteral()</code> (PMML44Super.ROC method), 207
<code>exportLiteral()</code> (PMML44Super.PCell method), 183	<code>exportLiteral()</code> (PMML44Super.ROCGraph method), 208
<code>exportLiteral()</code> (PMML44Super.PCovCell method), 184	<code>exportLiteral()</code> (PMML44Super.row method), 302
<code>exportLiteral()</code> (PMML44Super.PCovMatrix method), 185	<code>exportLiteral()</code> (PMML44Super.RuleSelectionMethod method), 218
<code>exportLiteral()</code> (PMML44Super.PMML method), 186	<code>exportLiteral()</code> (PMML44Super.RuleSet method), 219
<code>exportLiteral()</code> (PMML44Super.PoissonDistribution method), 198	<code>exportLiteral()</code> (PMML44Super.RuleSetModel method), 221
<code>exportLiteral()</code> (PMML44Super.PolynomialKernelType method), 198	<code>exportLiteral()</code> (PMML44Super.Scorecard method), 224
<code>exportLiteral()</code> (PMML44Super.PPCell method), 189	<code>exportLiteral()</code> (PMML44Super.ScoreDistribution method), 223
<code>exportLiteral()</code> (PMML44Super.PPMatrix method), 190	<code>exportLiteral()</code> (PMML44Super.SeasonalComponent method), 225
<code>exportLiteral()</code> (PMML44Super.PredictiveModelQuality method), 200	<code>exportLiteral()</code> (PMML44Super.SeasonalFactor method), 226
	<code>exportLiteral()</code> (PMML44Super.Seasonality_ExpoSmooth method), 226

`method`), 228
`exportLiteral()` (`PMML44Super.SeasonalTrendDecomposition`
`method`), 227
`exportLiteral()` (`PMML44Super.Segment` `method`),
229
`exportLiteral()` (`PMML44Super.Segmentation`
`method`), 231
`exportLiteral()` (`PMML44Super.Sequence`
`method`), 232
`exportLiteral()` (`PMML44Super.SequenceModel`
`method`), 233
`exportLiteral()` (`PMML44Super.SequenceReference`
`method`), 235
`exportLiteral()` (`PMML44Super.SequenceRule`
`method`), 236
`exportLiteral()` (`PMML44Super.SetPredicate`
`method`), 237
`exportLiteral()` (`PMML44Super.SetReference`
`method`), 238
`exportLiteral()` (`PMML44Super.SGD` `method`),
222
`exportLiteral()` (`PMML44Super.SigmoidKernelType`
`method`), 239
`exportLiteral()` (`PMML44Super.simpleMatching`
`method`), 303
`exportLiteral()` (`PMML44Super.SimplePredicate`
`method`), 240
`exportLiteral()` (`PMML44Super.SimpleRule`
`method`), 240
`exportLiteral()` (`PMML44Super.SimpleSetPredicate`
`method`), 242
`exportLiteral()` (`PMML44Super.SpectralAnalysis`
`method`), 243
`exportLiteral()` (`PMML44Super.squaredEuclidean`
`method`), 303
`exportLiteral()` (`PMML44Super.StateSpaceModel`
`method`), 243
`exportLiteral()` (`PMML44Super.StateVector`
`method`), 244
`exportLiteral()` (`PMML44Super.SupportVector`
`method`), 245
`exportLiteral()` (`PMML44Super.SupportVectorMachine`
`method`), 246
`exportLiteral()` (`PMML44Super.SupportVectorMachineModel`
`method`), 247
`exportLiteral()` (`PMML44Super.SupportVectors`
`method`), 249
`exportLiteral()` (`PMML44Super.TableLocator`
`method`), 250
`exportLiteral()` (`PMML44Super.tanimoto`
`method`), 304
`exportLiteral()` (`PMML44Super.Target` `method`),
251
`exportLiteral()` (`PMML44Super.Targets` `method`),
256
`exportLiteral()` (`PMML44Super.TargetValue`
`method`), 252
`exportLiteral()` (`PMML44Super.TargetValueCount`
`method`), 253
`exportLiteral()` (`PMML44Super.TargetValueCounts`
`method`), 253
`exportLiteral()` (`PMML44Super.TargetValueStat`
`method`), 254
`exportLiteral()` (`PMML44Super.TargetValueStats`
`method`), 255
`exportLiteral()` (`PMML44Super.Taxonomy`
`method`), 257
`exportLiteral()` (`PMML44Super.TestDistributions`
`method`), 257
`exportLiteral()` (`PMML44Super.TextCorpus`
`method`), 258
`exportLiteral()` (`PMML44Super.TextDictionary`
`method`), 259
`exportLiteral()` (`PMML44Super.TextDocument`
`method`), 260
`exportLiteral()` (`PMML44Super.TextIndex`
`method`), 261
`exportLiteral()` (`PMML44Super.TextIndexNormalization`
`method`), 263
`exportLiteral()` (`PMML44Super.TextModel`
`method`), 264
`exportLiteral()` (`PMML44Super.TextModelNormalization`
`method`), 266
`exportLiteral()` (`PMML44Super.TextModelSimilarity`
`method`), 266
`exportLiteral()` (`PMML44Super.Theta` `method`),
267
`exportLiteral()` (`PMML44Super.ThetaRecursionState`
`method`), 268
`exportLiteral()` (`PMML44Super.Time` `method`),
268
`exportLiteral()` (`PMML44Super.TimeAnchor`
`method`), 269
`exportLiteral()` (`PMML44Super.TimeCycle`
`method`), 270
`exportLiteral()` (`PMML44Super.TimeException`
`method`), 271
`exportLiteral()` (`PMML44Super.TimeSeries`
`method`), 272
`exportLiteral()` (`PMML44Super.TimeSeriesModel`
`method`), 273
`exportLiteral()` (`PMML44Super.Timestamp`
`method`), 275
`exportLiteral()` (`PMML44Super.TimeValue`
`method`), 275
`exportLiteral()` (`PMML44Super.TrainingInstances`
`method`), 276
`exportLiteral()` (`PMML44Super.TrainingParameters`

`method`), 277
`exportLiteral()` (*PMML44Super.TransferFunctionValues* `method`), 278
`exportLiteral()` (*PMML44Super.TransformationDictionary* `method`), 279
`exportLiteral()` (*PMML44Super.TransitionMatrix* `method`), 280
`exportLiteral()` (*PMML44Super.TreeModel* `method`), 280
`exportLiteral()` (*PMML44Super.Trend_ExpoSmooth* `method`), 283
`exportLiteral()` (*PMML44Super.TrendCoefficients* `method`), 282
`exportLiteral()` (*PMML44Super.TriangularDistributionForBN* `method`), 283
`exportLiteral()` (*PMML44Super.True_* `method`), 284
`exportLiteral()` (*PMML44Super.UniformDistribution* `method`), 285
`exportLiteral()` (*PMML44Super.UniformDistributionForBN* `method`), 286
`exportLiteral()` (*PMML44Super.UnivariateStats* `method`), 286
`exportLiteral()` (*PMML44Super.Upper* `method`), 287
`exportLiteral()` (*PMML44Super.Value* `method`), 288
`exportLiteral()` (*PMML44Super.ValueProbability* `method`), 289
`exportLiteral()` (*PMML44Super.VariableWeight* `method`), 290
`exportLiteral()` (*PMML44Super.Variance* `method`), 291
`exportLiteral()` (*PMML44Super.VarianceCoefficients* `method`), 292
`exportLiteral()` (*PMML44Super.VectorDictionary* `method`), 293
`exportLiteral()` (*PMML44Super.VectorFields* `method`), 294
`exportLiteral()` (*PMML44Super.VectorInstance* `method`), 294
`exportLiteral()` (*PMML44Super.VerificationField* `method`), 295
`exportLiteral()` (*PMML44Super.VerificationFields* `method`), 296
`exportLiteral()` (*PMML44Super.XCoordinates* `method`), 297
`exportLiteral()` (*PMML44Super.YCoordinates* `method`), 297
`exportLiteralAttributes()` (*PMML44Super.AbsoluteExponentialKernel* `method`), 7
`exportLiteralAttributes()` (*PMML44Super.Adadelata* `method`), 8
`exportLiteralAttributes()` (*PMML44Super.Adagrad* `method`), 9
`exportLiteralAttributes()` (*PMML44Super.Adam* `method`), 10
`exportLiteralAttributes()` (*PMML44Super.Adamax* `method`), 11
`exportLiteralAttributes()` (*PMML44Super.Aggregate* `method`), 12
`exportLiteralAttributes()` (*PMML44Super.Alternate* `method`), 12
`exportLiteralAttributes()` (*PMML44Super.Annotation* `method`), 13
`exportLiteralAttributes()` (*PMML44Super.AnomalyDetectionModel* `method`), 14
`exportLiteralAttributes()` (*PMML44Super.Anova* `method`), 16
`exportLiteralAttributes()` (*PMML44Super.AnovaRow* `method`), 17
`exportLiteralAttributes()` (*PMML44Super.AntecedentSequence* `method`), 18
`exportLiteralAttributes()` (*PMML44Super.AnyDistribution* `method`), 19
`exportLiteralAttributes()` (*PMML44Super.Application* `method`), 20
`exportLiteralAttributes()` (*PMML44Super.Apply* `method`), 21
`exportLiteralAttributes()` (*PMML44Super.AR* `method`), 3
`exportLiteralAttributes()` (*PMML44Super.ARDSquaredExponentialKernel* `method`), 4
`exportLiteralAttributes()` (*PMML44Super.ARIMA* `method`), 5
`exportLiteralAttributes()` (*PMML44Super.ARMAPart* `method`), 6
`exportLiteralAttributes()` (*PMML44Super.ArrayType* `method`), 23
`exportLiteralAttributes()` (*PMML44Super.AssociationModel* `method`), 24
`exportLiteralAttributes()` (*PMML44Super.AssociationRule* `method`), 26
`exportLiteralAttributes()` (*PMML44Super.Attribute* `method`), 27
`exportLiteralAttributes()` (*PMML44Super.BaseCumHazardTables* `method`), 28
`exportLiteralAttributes()` (*PMML44Super.Baseline* `method`), 29
`exportLiteralAttributes()` (*PMML44Super.BaselineCell* `method`), 30

[exportLiteralAttributes\(\)](#)
 (PMML44Super.BaselineModel method), [31](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BaselineStratum method), [32](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BayesianNetworkModel method), [35](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BayesianNetworkNodes method), [37](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BayesInput method), [33](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BayesInputs method), [34](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BayesOutput method), [34](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.binarySimilarity method), [298](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BlockIndicator method), [38](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BoundaryValueMeans method), [38](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.BoundaryValues method), [39](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.CategoricalPredictor method), [41](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Categories method), [42](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Category method), [42](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Characteristic method), [43](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Characteristics method), [44](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.chebychev method), [299](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ChildParent method), [45](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.cityBlock method), [300](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ClassLabels method), [46](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Cluster method), [46](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ClusteringField method), [47](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ClusteringModel method), [49](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ClusteringModelQuality method), [50](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Coefficient method), [51](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Coefficients method), [52](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ComparisonMeasure method), [53](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Comparisons method), [54](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ComplexPartialScore method), [55](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.CompoundPredicate method), [56](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.CompoundRule method), [58](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Con method), [59](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ConfusionMatrix method), [59](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ConsequentSequence method), [60](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Constant method), [61](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.Constraints method), [62](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ContinuousConditionalProbability method), [64](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ContinuousDistribution method), [65](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ContinuousNode method), [66](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.ContStats method), [63](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.CorrelationFields method), [67](#)
[exportLiteralAttributes\(\)](#)
 (PMML44Super.CorrelationMethods method), [68](#)

`exportLiteralAttributes()`
 (*PMML44Super.Correlations method*), 69
`exportLiteralAttributes()`
 (*PMML44Super.CorrelationValues method*), 69
`exportLiteralAttributes()`
 (*PMML44Super.COUNT_TABLE_TYPE method*), 40
`exportLiteralAttributes()`
 (*PMML44Super.Counts method*), 70
`exportLiteralAttributes()`
 (*PMML44Super.Covariances method*), 71
`exportLiteralAttributes()`
 (*PMML44Super.CovariateList method*), 71
`exportLiteralAttributes()`
 (*PMML44Super.DataDictionary method*), 72
`exportLiteralAttributes()`
 (*PMML44Super.DataField method*), 73
`exportLiteralAttributes()`
 (*PMML44Super.Decision method*), 74
`exportLiteralAttributes()`
 (*PMML44Super.Decisions method*), 77
`exportLiteralAttributes()`
 (*PMML44Super.DecisionTree method*), 75
`exportLiteralAttributes()`
 (*PMML44Super.DeepNetwork method*), 78
`exportLiteralAttributes()`
 (*PMML44Super.DefineFunction method*), 80
`exportLiteralAttributes()`
 (*PMML44Super.Delimiter method*), 81
`exportLiteralAttributes()`
 (*PMML44Super.Denominator method*), 82
`exportLiteralAttributes()`
 (*PMML44Super.DerivedField method*), 83
`exportLiteralAttributes()`
 (*PMML44Super.DiscreteConditionalProbability method*), 85
`exportLiteralAttributes()`
 (*PMML44Super.DiscreteNode method*), 86
`exportLiteralAttributes()`
 (*PMML44Super.Discretize method*), 87
`exportLiteralAttributes()`
 (*PMML44Super.DiscretizeBin method*), 88
`exportLiteralAttributes()`
 (*PMML44Super.DiscrStats method*), 84
`exportLiteralAttributes()`
 (*PMML44Super.DocumentTermMatrix method*), 89
`exportLiteralAttributes()`
 (*PMML44Super.DynamicRegressor method*), 90
`exportLiteralAttributes()`
 (*PMML44Super.euclidean method*), 300
`exportLiteralAttributes()`
 (*PMML44Super.EventValues method*), 91
`exportLiteralAttributes()`
 (*PMML44Super.ExponentialSmoothing method*), 92
`exportLiteralAttributes()`
 (*PMML44Super.Extension method*), 93
`exportLiteralAttributes()`
 (*PMML44Super.FactorList method*), 94
`exportLiteralAttributes()`
 (*PMML44Super.False_ method*), 94
`exportLiteralAttributes()`
 (*PMML44Super.FieldColumnPair method*), 95
`exportLiteralAttributes()`
 (*PMML44Super.FieldRef method*), 96
`exportLiteralAttributes()`
 (*PMML44Super.FieldValue method*), 96
`exportLiteralAttributes()`
 (*PMML44Super.FieldValueCount method*), 97
`exportLiteralAttributes()`
 (*PMML44Super.FinalNoise method*), 98
`exportLiteralAttributes()`
 (*PMML44Super.FinalNu method*), 99
`exportLiteralAttributes()`
 (*PMML44Super.FinalOmega method*), 99
`exportLiteralAttributes()`
 (*PMML44Super.FinalPredictedNoise method*), 100
`exportLiteralAttributes()`
 (*PMML44Super.FinalStateVector method*), 100
`exportLiteralAttributes()`
 (*PMML44Super.FinalTheta method*), 101
`exportLiteralAttributes()`
 (*PMML44Super.GARCH method*), 101
`exportLiteralAttributes()`
 (*PMML44Super.GARCHPart method*), 102
`exportLiteralAttributes()`
 (*PMML44Super.GaussianDistribution method*), 103
`exportLiteralAttributes()`
 (*PMML44Super.GaussianProcessModel method*), 104
`exportLiteralAttributes()`
 (*PMML44Super.GeneralizedExponentialKernel method*), 109
`exportLiteralAttributes()`
 (*PMML44Super.GeneralRegressionModel method*), 106
`exportLiteralAttributes()`
 (*PMML44Super.Header method*), 111

`exportLiteralAttributes()`
 (*PMML44Super.HVector method*), 110
`exportLiteralAttributes()`
 (*PMML44Super.Indices method*), 113
`exportLiteralAttributes()`
 (*PMML44Super.InlineTable method*), 113
`exportLiteralAttributes()`
 (*PMML44Super.InstanceField method*), 114
`exportLiteralAttributes()`
 (*PMML44Super.InstanceFields method*),
 115
`exportLiteralAttributes()`
 (*PMML44Super.INT_Entries method*), 112
`exportLiteralAttributes()`
 (*PMML44Super.INT_SparseArray method*),
 112
`exportLiteralAttributes()`
 (*PMML44Super.Interval method*), 115
`exportLiteralAttributes()`
 (*PMML44Super.Item method*), 116
`exportLiteralAttributes()`
 (*PMML44Super.ItemRef method*), 117
`exportLiteralAttributes()`
 (*PMML44Super.Itemset method*), 118
`exportLiteralAttributes()`
 (*PMML44Super.jaccard method*), 301
`exportLiteralAttributes()`
 (*PMML44Super.KalmanState method*), 120
`exportLiteralAttributes()`
 (*PMML44Super.KNNInput method*), 119
`exportLiteralAttributes()`
 (*PMML44Super.KNNInputs method*), 120
`exportLiteralAttributes()`
 (*PMML44Super.KohonenMap method*), 121
`exportLiteralAttributes()`
 (*PMML44Super.Lag method*), 122
`exportLiteralAttributes()`
 (*PMML44Super.Lambda method*), 123
`exportLiteralAttributes()`
 (*PMML44Super.LayerBias method*), 123
`exportLiteralAttributes()`
 (*PMML44Super.LayerParameters method*),
 125
`exportLiteralAttributes()`
 (*PMML44Super.LayerWeights method*), 128
`exportLiteralAttributes()`
 (*PMML44Super.Level method*), 129
`exportLiteralAttributes()`
 (*PMML44Super.LiftData method*), 130
`exportLiteralAttributes()`
 (*PMML44Super.LiftGraph method*), 131
`exportLiteralAttributes()`
 (*PMML44Super.LinearKernelType method*),
 131
`exportLiteralAttributes()`
 (*PMML44Super.LinearNorm method*), 132
`exportLiteralAttributes()`
 (*PMML44Super.LocalTransformations*
 method), 133
`exportLiteralAttributes()`
 (*PMML44Super.LognormalDistributionForBN*
 method), 134
`exportLiteralAttributes()`
 (*PMML44Super.Losses method*), 134
`exportLiteralAttributes()`
 (*PMML44Super.Lower method*), 135
`exportLiteralAttributes()`
 (*PMML44Super.MA method*), 136
`exportLiteralAttributes()`
 (*PMML44Super.MACoefficients method*),
 137
`exportLiteralAttributes()`
 (*PMML44Super.MapValues method*), 138
`exportLiteralAttributes()`
 (*PMML44Super.MatCell method*), 139
`exportLiteralAttributes()`
 (*PMML44Super.Matrix method*), 139
`exportLiteralAttributes()`
 (*PMML44Super.MaximumLikelihoodStat*
 method), 140
`exportLiteralAttributes()`
 (*PMML44Super.Mean method*), 141
`exportLiteralAttributes()`
 (*PMML44Super.MeasurementMatrix method*),
 142
`exportLiteralAttributes()`
 (*PMML44Super.Metrics method*), 143
`exportLiteralAttributes()`
 (*PMML44Super.MiningBuildTask method*),
 144
`exportLiteralAttributes()`
 (*PMML44Super.MiningField method*), 144
`exportLiteralAttributes()`
 (*PMML44Super.MiningModel method*), 146
`exportLiteralAttributes()`
 (*PMML44Super.MiningSchema method*),
 147
`exportLiteralAttributes()`
 (*PMML44Super.minkowski method*), 302
`exportLiteralAttributes()`
 (*PMML44Super.MissingValueWeights method*),
 148
`exportLiteralAttributes()`
 (*PMML44Super.ModelExplanation method*),
 149
`exportLiteralAttributes()`
 (*PMML44Super.ModelLiftGraph method*),
 150

`exportLiteralAttributes()`
 (*PMML44Super.ModelStats method*), 151
`exportLiteralAttributes()`
 (*PMML44Super.ModelVerification method*), 151
`exportLiteralAttributes()`
 (*PMML44Super.MultivariateStat method*), 152
`exportLiteralAttributes()`
 (*PMML44Super.MultivariateStats method*), 154
`exportLiteralAttributes()`
 (*PMML44Super.Nadam method*), 155
`exportLiteralAttributes()`
 (*PMML44Super.NaiveBayesModel method*), 156
`exportLiteralAttributes()`
 (*PMML44Super.NearestNeighborModel method*), 157
`exportLiteralAttributes()`
 (*PMML44Super.NetworkLayer method*), 159
`exportLiteralAttributes()`
 (*PMML44Super.NeuralInput method*), 160
`exportLiteralAttributes()`
 (*PMML44Super.NeuralInputs method*), 161
`exportLiteralAttributes()`
 (*PMML44Super.NeuralLayer method*), 162
`exportLiteralAttributes()`
 (*PMML44Super.NeuralNetwork method*), 163
`exportLiteralAttributes()`
 (*PMML44Super.NeuralOutput method*), 165
`exportLiteralAttributes()`
 (*PMML44Super.NeuralOutputs method*), 166
`exportLiteralAttributes()`
 (*PMML44Super.Neuron method*), 167
`exportLiteralAttributes()`
 (*PMML44Super.Node method*), 168
`exportLiteralAttributes()`
 (*PMML44Super.NonseasonalComponent method*), 169
`exportLiteralAttributes()`
 (*PMML44Super.NonseasonalFactor method*), 170
`exportLiteralAttributes()`
 (*PMML44Super.NormalDistributionForBN method*), 173
`exportLiteralAttributes()`
 (*PMML44Super.NormContinuous method*), 171
`exportLiteralAttributes()`
 (*PMML44Super.NormDiscrete method*), 172
`exportLiteralAttributes()`
 (*PMML44Super.Numerator method*), 174
`exportLiteralAttributes()`
 (*PMML44Super.NumericInfo method*), 174
`exportLiteralAttributes()`
 (*PMML44Super.NumericPredictor method*), 175
`exportLiteralAttributes()`
 (*PMML44Super.Optimizers method*), 176
`exportLiteralAttributes()`
 (*PMML44Super.OptimumLiftGraph method*), 178
`exportLiteralAttributes()`
 (*PMML44Super.OutlierEffect method*), 179
`exportLiteralAttributes()`
 (*PMML44Super.Output method*), 179
`exportLiteralAttributes()`
 (*PMML44Super.OutputField method*), 180
`exportLiteralAttributes()`
 (*PMML44Super.PairCounts method*), 191
`exportLiteralAttributes()`
 (*PMML44Super.Parameter method*), 192
`exportLiteralAttributes()`
 (*PMML44Super.ParameterField method*), 193
`exportLiteralAttributes()`
 (*PMML44Super.ParameterList method*), 194
`exportLiteralAttributes()`
 (*PMML44Super.ParamMatrix method*), 191
`exportLiteralAttributes()`
 (*PMML44Super.ParentValue method*), 194
`exportLiteralAttributes()`
 (*PMML44Super.Partition method*), 195
`exportLiteralAttributes()`
 (*PMML44Super.PartitionFieldStats method*), 196
`exportLiteralAttributes()`
 (*PMML44Super.PastVariances method*), 197
`exportLiteralAttributes()`
 (*PMML44Super.PCell method*), 183
`exportLiteralAttributes()`
 (*PMML44Super.PCovCell method*), 184
`exportLiteralAttributes()`
 (*PMML44Super.PCovMatrix method*), 185
`exportLiteralAttributes()`
 (*PMML44Super.PMML method*), 186
`exportLiteralAttributes()`
 (*PMML44Super.PoissonDistribution method*), 198
`exportLiteralAttributes()`

(PMML44Super.PolynomialKernelType
 method), 199
 exportLiteralAttributes()
 (PMML44Super.PPCell method), 189
 exportLiteralAttributes()
 (PMML44Super.PPMatrix method), 190
 exportLiteralAttributes()
 (PMML44Super.PredictiveModelQuality
 method), 200
 exportLiteralAttributes()
 (PMML44Super.Predictor method), 202
 exportLiteralAttributes()
 (PMML44Super.PredictorTerm method),
 202
 exportLiteralAttributes()
 (PMML44Super.PsiVector method), 203
 exportLiteralAttributes()
 (PMML44Super.Quantile method), 204
 exportLiteralAttributes()
 (PMML44Super.RadialBasisKernel method),
 209
 exportLiteralAttributes()
 (PMML44Super.RadialBasisKernelType
 method), 209
 exportLiteralAttributes()
 (PMML44Super.RandomLiftGraph method),
 210
 exportLiteralAttributes()
 (PMML44Super.REAL_Entries method),
 205
 exportLiteralAttributes()
 (PMML44Super.REAL_SparseArray method),
 205
 exportLiteralAttributes()
 (PMML44Super.Regression method), 211
 exportLiteralAttributes()
 (PMML44Super.RegressionModel method),
 212
 exportLiteralAttributes()
 (PMML44Super.RegressionTable method),
 214
 exportLiteralAttributes()
 (PMML44Super.RegressorValues method),
 215
 exportLiteralAttributes()
 (PMML44Super.Residuals method), 217
 exportLiteralAttributes()
 (PMML44Super.ResidualSquareCoefficients
 method), 216
 exportLiteralAttributes()
 (PMML44Super.ResultField method), 217
 exportLiteralAttributes()
 (PMML44Super.RMSprop method), 206
 exportLiteralAttributes()
 (PMML44Super.ROC method), 207
 exportLiteralAttributes()
 (PMML44Super.ROCGraph method), 208
 exportLiteralAttributes()
 (PMML44Super.row method), 302
 exportLiteralAttributes()
 (PMML44Super.RuleSelectionMethod method),
 218
 exportLiteralAttributes()
 (PMML44Super.RuleSet method), 219
 exportLiteralAttributes()
 (PMML44Super.RuleSetModel method),
 221
 exportLiteralAttributes()
 (PMML44Super.Scorecard method), 224
 exportLiteralAttributes()
 (PMML44Super.ScoreDistribution method),
 223
 exportLiteralAttributes()
 (PMML44Super.SeasonalComponent method),
 225
 exportLiteralAttributes()
 (PMML44Super.SeasonalFactor method),
 226
 exportLiteralAttributes()
 (PMML44Super.Seasonality_ExpoSmooth
 method), 228
 exportLiteralAttributes()
 (PMML44Super.SeasonalTrendDecomposition
 method), 227
 exportLiteralAttributes()
 (PMML44Super.Segment method), 229
 exportLiteralAttributes()
 (PMML44Super.Segmentation method), 231
 exportLiteralAttributes()
 (PMML44Super.Sequence method), 232
 exportLiteralAttributes()
 (PMML44Super.SequenceModel method),
 233
 exportLiteralAttributes()
 (PMML44Super.SequenceReference method),
 235
 exportLiteralAttributes()
 (PMML44Super.SequenceRule method),
 236
 exportLiteralAttributes()
 (PMML44Super.SetPredicate method), 237
 exportLiteralAttributes()
 (PMML44Super.SetReference method), 238
 exportLiteralAttributes()
 (PMML44Super.SGD method), 222
 exportLiteralAttributes()
 (PMML44Super.SigmoidKernelType method),
 239

`exportLiteralAttributes()` (*PMML44Super.simpleMatching method*), 303
`exportLiteralAttributes()` (*PMML44Super.SimplePredicate method*), 240
`exportLiteralAttributes()` (*PMML44Super.SimpleRule method*), 240
`exportLiteralAttributes()` (*PMML44Super.SimpleSetPredicate method*), 242
`exportLiteralAttributes()` (*PMML44Super.SpectralAnalysis method*), 243
`exportLiteralAttributes()` (*PMML44Super.squaredEuclidean method*), 303
`exportLiteralAttributes()` (*PMML44Super.StateSpaceModel method*), 243
`exportLiteralAttributes()` (*PMML44Super.StateVector method*), 244
`exportLiteralAttributes()` (*PMML44Super.SupportVector method*), 245
`exportLiteralAttributes()` (*PMML44Super.SupportVectorMachine method*), 246
`exportLiteralAttributes()` (*PMML44Super.SupportVectorMachineModel method*), 247
`exportLiteralAttributes()` (*PMML44Super.SupportVectors method*), 249
`exportLiteralAttributes()` (*PMML44Super.TableLocator method*), 250
`exportLiteralAttributes()` (*PMML44Super.tanimoto method*), 304
`exportLiteralAttributes()` (*PMML44Super.Target method*), 251
`exportLiteralAttributes()` (*PMML44Super.Targets method*), 256
`exportLiteralAttributes()` (*PMML44Super.TargetValue method*), 252
`exportLiteralAttributes()` (*PMML44Super.TargetValueCount method*), 253
`exportLiteralAttributes()` (*PMML44Super.TargetValueCounts method*), 253
`exportLiteralAttributes()` (*PMML44Super.TargetValueStat method*), 254
`exportLiteralAttributes()` (*PMML44Super.TargetValueStats method*), 255
`exportLiteralAttributes()` (*PMML44Super.Taxonomy method*), 257
`exportLiteralAttributes()` (*PMML44Super.TestDistributions method*), 257
`exportLiteralAttributes()` (*PMML44Super.TextCorpus method*), 258
`exportLiteralAttributes()` (*PMML44Super.TextDictionary method*), 259
`exportLiteralAttributes()` (*PMML44Super.TextDocument method*), 260
`exportLiteralAttributes()` (*PMML44Super.TextIndex method*), 261
`exportLiteralAttributes()` (*PMML44Super.TextIndexNormalization method*), 263
`exportLiteralAttributes()` (*PMML44Super.TextModel method*), 264
`exportLiteralAttributes()` (*PMML44Super.TextModelNormalization method*), 266
`exportLiteralAttributes()` (*PMML44Super.TextModelSimiliarity method*), 266
`exportLiteralAttributes()` (*PMML44Super.Theta method*), 267
`exportLiteralAttributes()` (*PMML44Super.ThetaRecursionState method*), 268
`exportLiteralAttributes()` (*PMML44Super.Time method*), 268
`exportLiteralAttributes()` (*PMML44Super.TimeAnchor method*), 269
`exportLiteralAttributes()` (*PMML44Super.TimeCycle method*), 270
`exportLiteralAttributes()` (*PMML44Super.TimeException method*), 271
`exportLiteralAttributes()` (*PMML44Super.TimeSeries method*), 272
`exportLiteralAttributes()` (*PMML44Super.TimeSeriesModel method*), 273
`exportLiteralAttributes()` (*PMML44Super.Timestamp method*), 276
`exportLiteralAttributes()` (*PMML44Super.TimeValue method*), 275
`exportLiteralAttributes()` (*PMML44Super.TrainingInstances method*), 276

<code>exportLiteralAttributes()</code> (<i>PMML44Super.TrainingParameters</i> method), 277	<code>exportLiteralAttributes()</code> (<i>PMML44Super.VectorInstance</i> method), 294
<code>exportLiteralAttributes()</code> (<i>PMML44Super.TransferFunctionValues</i> method), 278	<code>exportLiteralAttributes()</code> (<i>PMML44Super.VerificationField</i> method), 295
<code>exportLiteralAttributes()</code> (<i>PMML44Super.TransformationDictionary</i> method), 279	<code>exportLiteralAttributes()</code> (<i>PMML44Super.VerificationFields</i> method), 296
<code>exportLiteralAttributes()</code> (<i>PMML44Super.TransitionMatrix</i> method), 280	<code>exportLiteralAttributes()</code> (<i>PMML44Super.XCoordinates</i> method), 297
<code>exportLiteralAttributes()</code> (<i>PMML44Super.TreeModel</i> method), 280	<code>exportLiteralAttributes()</code> (<i>PMML44Super.YCoordinates</i> method), 297
<code>exportLiteralAttributes()</code> (<i>PMML44Super.Trend_ExpoSmooth</i> method), 283	<code>exportLiteralChildren()</code> (<i>PMML44Super.AbsoluteExponentialKernel</i> method), 7
<code>exportLiteralAttributes()</code> (<i>PMML44Super.TrendCoefficients</i> method), 282	<code>exportLiteralChildren()</code> (<i>PMML44Super.Adadelta</i> method), 8
<code>exportLiteralAttributes()</code> (<i>PMML44Super.TriangularDistributionForBN</i> method), 283	<code>exportLiteralChildren()</code> (<i>PMML44Super.Adagrad</i> method), 9
<code>exportLiteralAttributes()</code> (<i>PMML44Super.True_</i> method), 284	<code>exportLiteralChildren()</code> (<i>PMML44Super.Adam</i> method), 10
<code>exportLiteralAttributes()</code> (<i>PMML44Super.UniformDistribution</i> method), 285	<code>exportLiteralChildren()</code> (<i>PMML44Super.Adamax</i> method), 11
<code>exportLiteralAttributes()</code> (<i>PMML44Super.UniformDistributionForBN</i> method), 286	<code>exportLiteralChildren()</code> (<i>PMML44Super.Aggregate</i> method), 12
<code>exportLiteralAttributes()</code> (<i>PMML44Super.UnivariateStats</i> method), 286	<code>exportLiteralChildren()</code> (<i>PMML44Super.Alternate</i> method), 12
<code>exportLiteralAttributes()</code> (<i>PMML44Super.UnivariateStats</i> method), 286	<code>exportLiteralChildren()</code> (<i>PMML44Super.Annotation</i> method), 13
<code>exportLiteralAttributes()</code> (<i>PMML44Super.Upper</i> method), 287	<code>exportLiteralChildren()</code> (<i>PMML44Super.AnomalyDetectionModel</i> method), 14
<code>exportLiteralAttributes()</code> (<i>PMML44Super.Value</i> method), 288	<code>exportLiteralChildren()</code> (<i>PMML44Super.Anova</i> method), 16
<code>exportLiteralAttributes()</code> (<i>PMML44Super.ValueProbability</i> method), 289	<code>exportLiteralChildren()</code> (<i>PMML44Super.AnovaRow</i> method), 17
<code>exportLiteralAttributes()</code> (<i>PMML44Super.VariableWeight</i> method), 290	<code>exportLiteralChildren()</code> (<i>PMML44Super.AntecedentSequence</i> method), 18
<code>exportLiteralAttributes()</code> (<i>PMML44Super.Variance</i> method), 291	<code>exportLiteralChildren()</code> (<i>PMML44Super.AnyDistribution</i> method), 19
<code>exportLiteralAttributes()</code> (<i>PMML44Super.VarianceCoefficients</i> method), 292	<code>exportLiteralChildren()</code> (<i>PMML44Super.Application</i> method), 20
<code>exportLiteralAttributes()</code> (<i>PMML44Super.VectorDictionary</i> method), 293	<code>exportLiteralChildren()</code> (<i>PMML44Super.Apply</i> method), 21
<code>exportLiteralAttributes()</code> (<i>PMML44Super.VectorFields</i> method), 294	<code>exportLiteralChildren()</code> (<i>PMML44Super.AR</i> method), 3
	<code>exportLiteralChildren()</code> (<i>PMML44Super.ARDSquaredExponentialKernel</i> method), 4
	<code>exportLiteralChildren()</code>

(PMML44Super.ARIMA method), 5
 exportLiteralChildren()
 (PMML44Super.ARMAPart method), 6
 exportLiteralChildren()
 (PMML44Super.ArrayType method), 23
 exportLiteralChildren()
 (PMML44Super.AssociationModel method), 24
 exportLiteralChildren()
 (PMML44Super.AssociationRule method), 26
 exportLiteralChildren()
 (PMML44Super.Attribute method), 27
 exportLiteralChildren()
 (PMML44Super.BaseCumHazardTables method), 28
 exportLiteralChildren()
 (PMML44Super.Baseline method), 29
 exportLiteralChildren()
 (PMML44Super.BaselineCell method), 30
 exportLiteralChildren()
 (PMML44Super.BaselineModel method), 31
 exportLiteralChildren()
 (PMML44Super.BaselineStratum method), 32
 exportLiteralChildren()
 (PMML44Super.BayesianNetworkModel method), 35
 exportLiteralChildren()
 (PMML44Super.BayesianNetworkNodes method), 37
 exportLiteralChildren()
 (PMML44Super.BayesInput method), 33
 exportLiteralChildren()
 (PMML44Super.BayesInputs method), 34
 exportLiteralChildren()
 (PMML44Super.BayesOutput method), 34
 exportLiteralChildren()
 (PMML44Super.binarySimilarity method), 298
 exportLiteralChildren()
 (PMML44Super.BlockIndicator method), 38
 exportLiteralChildren()
 (PMML44Super.BoundaryValueMeans method), 38
 exportLiteralChildren()
 (PMML44Super.BoundaryValues method), 39
 exportLiteralChildren()
 (PMML44Super.CategoricalPredictor method), 41
 exportLiteralChildren()
 (PMML44Super.Categories method), 42
 exportLiteralChildren()
 (PMML44Super.Category method), 42
 exportLiteralChildren()
 (PMML44Super.Characteristic method), 43
 exportLiteralChildren()
 (PMML44Super.Characteristics method), 44
 exportLiteralChildren()
 (PMML44Super.chebychev method), 299
 exportLiteralChildren()
 (PMML44Super.ChildParent method), 45
 exportLiteralChildren()
 (PMML44Super.cityBlock method), 300
 exportLiteralChildren()
 (PMML44Super.ClassLabels method), 46
 exportLiteralChildren()
 (PMML44Super.Cluster method), 46
 exportLiteralChildren()
 (PMML44Super.ClusteringField method), 47
 exportLiteralChildren()
 (PMML44Super.ClusteringModel method), 49
 exportLiteralChildren()
 (PMML44Super.ClusteringModelQuality method), 50
 exportLiteralChildren()
 (PMML44Super.Coefficient method), 51
 exportLiteralChildren()
 (PMML44Super.Coefficients method), 52
 exportLiteralChildren()
 (PMML44Super.ComparisonMeasure method), 53
 exportLiteralChildren()
 (PMML44Super.Comparisons method), 54
 exportLiteralChildren()
 (PMML44Super.ComplexPartialScore method), 55
 exportLiteralChildren()
 (PMML44Super.CompoundPredicate method), 56
 exportLiteralChildren()
 (PMML44Super.CompoundRule method), 58
 exportLiteralChildren() (PMML44Super.Con method), 59
 exportLiteralChildren()
 (PMML44Super.ConfusionMatrix method), 59
 exportLiteralChildren()
 (PMML44Super.ConsequentSequence method), 60
 exportLiteralChildren()

(PMML44Super.Constant method), 61
 exportLiteralChildren()
 (PMML44Super.Constraints method), 62
 exportLiteralChildren()
 (PMML44Super.ContinuousConditionalProbability
 method), 64
 exportLiteralChildren()
 (PMML44Super.ContinuousDistribution
 method), 65
 exportLiteralChildren()
 (PMML44Super.ContinuousNode method),
 66
 exportLiteralChildren()
 (PMML44Super.ContStats method), 63
 exportLiteralChildren()
 (PMML44Super.CorrelationFields method), 67
 exportLiteralChildren()
 (PMML44Super.CorrelationMethods method),
 68
 exportLiteralChildren()
 (PMML44Super.Correlations method), 69
 exportLiteralChildren()
 (PMML44Super.CorrelationValues method), 69
 exportLiteralChildren()
 (PMML44Super.COUNT_TABLE_TYPE
 method), 40
 exportLiteralChildren()
 (PMML44Super.Counts method), 70
 exportLiteralChildren()
 (PMML44Super.Covariances method), 71
 exportLiteralChildren()
 (PMML44Super.CovariateList method), 71
 exportLiteralChildren()
 (PMML44Super.DataDictionary method),
 72
 exportLiteralChildren()
 (PMML44Super.DataField method), 73
 exportLiteralChildren()
 (PMML44Super.Decision method), 75
 exportLiteralChildren()
 (PMML44Super.Decisions method), 77
 exportLiteralChildren()
 (PMML44Super.DecisionTree method), 75
 exportLiteralChildren()
 (PMML44Super.DeepNetwork method),
 78
 exportLiteralChildren()
 (PMML44Super.DefineFunction method),
 80
 exportLiteralChildren()
 (PMML44Super.Delimiter method), 81
 exportLiteralChildren()
 (PMML44Super.Denominator method), 82
 exportLiteralChildren()
 (PMML44Super.DerivedField method), 83
 exportLiteralChildren()
 (PMML44Super.DiscreteConditionalProbability
 method), 85
 exportLiteralChildren()
 (PMML44Super.DiscreteNode method), 86
 exportLiteralChildren()
 (PMML44Super.Discretize method), 87
 exportLiteralChildren()
 (PMML44Super.DiscretizeBin method), 88
 exportLiteralChildren()
 (PMML44Super.DiscrStats method), 84
 exportLiteralChildren()
 (PMML44Super.DocumentTermMatrix
 method), 89
 exportLiteralChildren()
 (PMML44Super.DynamicRegressor method),
 90
 exportLiteralChildren()
 (PMML44Super.euclidean method), 300
 exportLiteralChildren()
 (PMML44Super.EventValues method), 91
 exportLiteralChildren()
 (PMML44Super.ExponentialSmoothing
 method), 92
 exportLiteralChildren()
 (PMML44Super.Extension method), 93
 exportLiteralChildren()
 (PMML44Super.FactorList method), 94
 exportLiteralChildren()
 (PMML44Super.False_ method), 94
 exportLiteralChildren()
 (PMML44Super.FieldColumnPair method),
 95
 exportLiteralChildren()
 (PMML44Super.FieldRef method), 96
 exportLiteralChildren()
 (PMML44Super.FieldValue method), 96
 exportLiteralChildren()
 (PMML44Super.FieldValueCount method),
 97
 exportLiteralChildren()
 (PMML44Super.FinalNoise method), 98
 exportLiteralChildren()
 (PMML44Super.FinalNu method), 99
 exportLiteralChildren()
 (PMML44Super.FinalOmega method), 99
 exportLiteralChildren()
 (PMML44Super.FinalPredictedNoise method),
 100
 exportLiteralChildren()
 (PMML44Super.FinalStateVector method),
 100
 exportLiteralChildren()

(*PMML44Super.FinalTheta method*), 101
`exportLiteralChildren()`
(*PMML44Super.GARCH method*), 101
`exportLiteralChildren()`
(*PMML44Super.GARCHPart method*), 102
`exportLiteralChildren()`
(*PMML44Super.GaussianDistribution method*), 103
`exportLiteralChildren()`
(*PMML44Super.GaussianProcessModel method*), 104
`exportLiteralChildren()`
(*PMML44Super.GeneralizedExponentialKernel method*), 109
`exportLiteralChildren()`
(*PMML44Super.GeneralRegressionModel method*), 106
`exportLiteralChildren()`
(*PMML44Super.Header method*), 111
`exportLiteralChildren()`
(*PMML44Super.HVector method*), 110
`exportLiteralChildren()`
(*PMML44Super.Indices method*), 113
`exportLiteralChildren()`
(*PMML44Super.InlineTable method*), 113
`exportLiteralChildren()`
(*PMML44Super.InstanceField method*), 114
`exportLiteralChildren()`
(*PMML44Super.InstanceFields method*), 115
`exportLiteralChildren()`
(*PMML44Super.INT_Entries method*), 112
`exportLiteralChildren()`
(*PMML44Super.INT_SparseArray method*), 112
`exportLiteralChildren()`
(*PMML44Super.Interval method*), 115
`exportLiteralChildren()` (*PMML44Super.Item method*), 116
`exportLiteralChildren()`
(*PMML44Super.ItemRef method*), 117
`exportLiteralChildren()`
(*PMML44Super.Itemset method*), 118
`exportLiteralChildren()`
(*PMML44Super.jaccard method*), 301
`exportLiteralChildren()`
(*PMML44Super.KalmanState method*), 120
`exportLiteralChildren()`
(*PMML44Super.KNNInput method*), 119
`exportLiteralChildren()`
(*PMML44Super.KNNInputs method*), 120
`exportLiteralChildren()`
(*PMML44Super.KohonenMap method*), 121
`exportLiteralChildren()` (*PMML44Super.Lag method*), 122
`exportLiteralChildren()`
(*PMML44Super.Lambda method*), 123
`exportLiteralChildren()`
(*PMML44Super.LayerBias method*), 123
`exportLiteralChildren()`
(*PMML44Super.LayerParameters method*), 125
`exportLiteralChildren()`
(*PMML44Super.LayerWeights method*), 128
`exportLiteralChildren()` (*PMML44Super.Level method*), 129
`exportLiteralChildren()`
(*PMML44Super.LiftData method*), 130
`exportLiteralChildren()`
(*PMML44Super.LiftGraph method*), 131
`exportLiteralChildren()`
(*PMML44Super.LinearKernelType method*), 131
`exportLiteralChildren()`
(*PMML44Super.LinearNorm method*), 132
`exportLiteralChildren()`
(*PMML44Super.LocalTransformations method*), 133
`exportLiteralChildren()`
(*PMML44Super.LognormalDistributionForBN method*), 134
`exportLiteralChildren()`
(*PMML44Super.Losses method*), 134
`exportLiteralChildren()`
(*PMML44Super.Lower method*), 135
`exportLiteralChildren()` (*PMML44Super.MA method*), 136
`exportLiteralChildren()`
(*PMML44Super.MACoefficients method*), 137
`exportLiteralChildren()`
(*PMML44Super.MapValues method*), 138
`exportLiteralChildren()`
(*PMML44Super.MatCell method*), 139
`exportLiteralChildren()`
(*PMML44Super.Matrix method*), 139
`exportLiteralChildren()`
(*PMML44Super.MaximumLikelihoodStat method*), 140
`exportLiteralChildren()` (*PMML44Super.Mean method*), 141
`exportLiteralChildren()`
(*PMML44Super.MeasurementMatrix method*), 142
`exportLiteralChildren()`
(*PMML44Super.Metrics method*), 143
`exportLiteralChildren()`
(*PMML44Super.MiningBuildTask method*),

[144](#)
`exportLiteralChildren()`
 (*PMML44Super.MiningField method*), [144](#)
`exportLiteralChildren()`
 (*PMML44Super.MiningModel method*), [146](#)
`exportLiteralChildren()`
 (*PMML44Super.MiningSchema method*), [148](#)
`exportLiteralChildren()`
 (*PMML44Super.minkowski method*), [302](#)
`exportLiteralChildren()`
 (*PMML44Super.MissingValueWeights method*), [148](#)
`exportLiteralChildren()`
 (*PMML44Super.ModelExplanation method*), [149](#)
`exportLiteralChildren()`
 (*PMML44Super.ModelLiftGraph method*), [150](#)
`exportLiteralChildren()`
 (*PMML44Super.ModelStats method*), [151](#)
`exportLiteralChildren()`
 (*PMML44Super.ModelVerification method*), [151](#)
`exportLiteralChildren()`
 (*PMML44Super.MultivariateStat method*), [152](#)
`exportLiteralChildren()`
 (*PMML44Super.MultivariateStats method*), [154](#)
`exportLiteralChildren()`
 (*PMML44Super.Nadam method*), [155](#)
`exportLiteralChildren()`
 (*PMML44Super.NaiveBayesModel method*), [156](#)
`exportLiteralChildren()`
 (*PMML44Super.NearestNeighborModel method*), [157](#)
`exportLiteralChildren()`
 (*PMML44Super.NetworkLayer method*), [159](#)
`exportLiteralChildren()`
 (*PMML44Super.NeuralInput method*), [160](#)
`exportLiteralChildren()`
 (*PMML44Super.NeuralInputs method*), [161](#)
`exportLiteralChildren()`
 (*PMML44Super.NeuralLayer method*), [162](#)
`exportLiteralChildren()`
 (*PMML44Super.NeuralNetwork method*), [163](#)
`exportLiteralChildren()`
 (*PMML44Super.NeuralOutput method*), [165](#)
`exportLiteralChildren()`
 (*PMML44Super.NeuralOutputs method*), [166](#)
`exportLiteralChildren()`
 (*PMML44Super.Neuron method*), [167](#)
`exportLiteralChildren()` (*PMML44Super.Node method*), [168](#)
`exportLiteralChildren()`
 (*PMML44Super.NonseasonalComponent method*), [169](#)
`exportLiteralChildren()`
 (*PMML44Super.NonseasonalFactor method*), [170](#)
`exportLiteralChildren()`
 (*PMML44Super.NormalDistributionForBN method*), [173](#)
`exportLiteralChildren()`
 (*PMML44Super.NormContinuous method*), [171](#)
`exportLiteralChildren()`
 (*PMML44Super.NormDiscrete method*), [172](#)
`exportLiteralChildren()`
 (*PMML44Super.Numerator method*), [174](#)
`exportLiteralChildren()`
 (*PMML44Super.NumericInfo method*), [174](#)
`exportLiteralChildren()`
 (*PMML44Super.NumericPredictor method*), [175](#)
`exportLiteralChildren()`
 (*PMML44Super.Optimizers method*), [176](#)
`exportLiteralChildren()`
 (*PMML44Super.OptimumLiftGraph method*), [178](#)
`exportLiteralChildren()`
 (*PMML44Super.OutlierEffect method*), [179](#)
`exportLiteralChildren()`
 (*PMML44Super.Output method*), [179](#)
`exportLiteralChildren()`
 (*PMML44Super.OutputField method*), [180](#)
`exportLiteralChildren()`
 (*PMML44Super.PairCounts method*), [191](#)
`exportLiteralChildren()`
 (*PMML44Super.Parameter method*), [192](#)
`exportLiteralChildren()`
 (*PMML44Super.ParameterField method*), [193](#)
`exportLiteralChildren()`
 (*PMML44Super.ParameterList method*), [194](#)
`exportLiteralChildren()`
 (*PMML44Super.ParamMatrix method*), [191](#)
`exportLiteralChildren()`
 (*PMML44Super.ParentValue method*), [195](#)
`exportLiteralChildren()`

(PMML44Super.Partition method), 195
 exportLiteralChildren() (PMML44Super.PartitionFieldStats method), 196
 exportLiteralChildren() (PMML44Super.PastVariances method), 197
 exportLiteralChildren() (PMML44Super.PCell method), 183
 exportLiteralChildren() (PMML44Super.PCovCell method), 184
 exportLiteralChildren() (PMML44Super.PCovMatrix method), 185
 exportLiteralChildren() (PMML44Super.PMML method), 186
 exportLiteralChildren() (PMML44Super.PoissonDistribution method), 198
 exportLiteralChildren() (PMML44Super.PolynomialKernelType method), 199
 exportLiteralChildren() (PMML44Super.PPCell method), 189
 exportLiteralChildren() (PMML44Super.PPMatrix method), 190
 exportLiteralChildren() (PMML44Super.PredictiveModelQuality method), 200
 exportLiteralChildren() (PMML44Super.Predictor method), 202
 exportLiteralChildren() (PMML44Super.PredictorTerm method), 202
 exportLiteralChildren() (PMML44Super.PsiVector method), 203
 exportLiteralChildren() (PMML44Super.Quantile method), 204
 exportLiteralChildren() (PMML44Super.RadialBasisKernel method), 209
 exportLiteralChildren() (PMML44Super.RadialBasisKernelType method), 210
 exportLiteralChildren() (PMML44Super.RandomLiftGraph method), 210
 exportLiteralChildren() (PMML44Super.REAL_Entries method), 205
 exportLiteralChildren() (PMML44Super.REAL_SparseArray method), 205
 exportLiteralChildren() (PMML44Super.Regression method), 211
 exportLiteralChildren() (PMML44Super.RegressionModel method), 212
 exportLiteralChildren() (PMML44Super.RegressionTable method), 214
 exportLiteralChildren() (PMML44Super.RegressorValues method), 215
 exportLiteralChildren() (PMML44Super.Residuals method), 217
 exportLiteralChildren() (PMML44Super.ResidualSquareCoefficients method), 216
 exportLiteralChildren() (PMML44Super.ResultField method), 217
 exportLiteralChildren() (PMML44Super.RMSprop method), 206
 exportLiteralChildren() (PMML44Super.ROC method), 207
 exportLiteralChildren() (PMML44Super.ROCGraph method), 208
 exportLiteralChildren() (PMML44Super.row method), 302
 exportLiteralChildren() (PMML44Super.RuleSelectionMethod method), 218
 exportLiteralChildren() (PMML44Super.RuleSet method), 219
 exportLiteralChildren() (PMML44Super.RuleSetModel method), 221
 exportLiteralChildren() (PMML44Super.Scorecard method), 224
 exportLiteralChildren() (PMML44Super.ScoreDistribution method), 223
 exportLiteralChildren() (PMML44Super.SeasonalComponent method), 225
 exportLiteralChildren() (PMML44Super.SeasonalFactor method), 226
 exportLiteralChildren() (PMML44Super.Seasonality_ExpoSmooth method), 228
 exportLiteralChildren() (PMML44Super.SeasonalTrendDecomposition method), 227
 exportLiteralChildren() (PMML44Super.Segment method), 229
 exportLiteralChildren() (PMML44Super.Segmentation method), 231
 exportLiteralChildren()

(*PMML44Super.Sequence method*), 232
 exportLiteralChildren() (*PMML44Super.SequenceModel method*), 233
 exportLiteralChildren() (*PMML44Super.SequenceReference method*), 235
 exportLiteralChildren() (*PMML44Super.SequenceRule method*), 236
 exportLiteralChildren() (*PMML44Super.SetPredicate method*), 237
 exportLiteralChildren() (*PMML44Super.SetReference method*), 238
 exportLiteralChildren() (*PMML44Super.SGD method*), 222
 exportLiteralChildren() (*PMML44Super.SigmoidKernelType method*), 239
 exportLiteralChildren() (*PMML44Super.simpleMatching method*), 303
 exportLiteralChildren() (*PMML44Super.SimplePredicate method*), 240
 exportLiteralChildren() (*PMML44Super.SimpleRule method*), 241
 exportLiteralChildren() (*PMML44Super.SimpleSetPredicate method*), 242
 exportLiteralChildren() (*PMML44Super.SpectralAnalysis method*), 243
 exportLiteralChildren() (*PMML44Super.squaredEuclidean method*), 304
 exportLiteralChildren() (*PMML44Super.StateSpaceModel method*), 243
 exportLiteralChildren() (*PMML44Super.StateVector method*), 244
 exportLiteralChildren() (*PMML44Super.SupportVector method*), 245
 exportLiteralChildren() (*PMML44Super.SupportVectorMachine method*), 246
 exportLiteralChildren() (*PMML44Super.SupportVectorMachineModel method*), 247
 exportLiteralChildren() (*PMML44Super.SupportVectors method*), 249
 exportLiteralChildren() (*PMML44Super.TableLocator method*), 250
 exportLiteralChildren() (*PMML44Super.tanimoto method*), 304
 exportLiteralChildren() (*PMML44Super.Target method*), 251
 exportLiteralChildren() (*PMML44Super.Targets method*), 256
 exportLiteralChildren() (*PMML44Super.TargetValue method*), 252
 exportLiteralChildren() (*PMML44Super.TargetValueCount method*), 253
 exportLiteralChildren() (*PMML44Super.TargetValueCounts method*), 253
 exportLiteralChildren() (*PMML44Super.TargetValueStat method*), 254
 exportLiteralChildren() (*PMML44Super.TargetValueStats method*), 255
 exportLiteralChildren() (*PMML44Super.Taxonomy method*), 257
 exportLiteralChildren() (*PMML44Super.TestDistributions method*), 257
 exportLiteralChildren() (*PMML44Super.TextCorpus method*), 259
 exportLiteralChildren() (*PMML44Super.TextDictionary method*), 259
 exportLiteralChildren() (*PMML44Super.TextDocument method*), 260
 exportLiteralChildren() (*PMML44Super.TextIndex method*), 261
 exportLiteralChildren() (*PMML44Super.TextIndexNormalization method*), 263
 exportLiteralChildren() (*PMML44Super.TextModel method*), 264
 exportLiteralChildren() (*PMML44Super.TextModelNormalization method*), 266
 exportLiteralChildren() (*PMML44Super.TextModelSimiliarity method*), 266
 exportLiteralChildren() (*PMML44Super.Theta method*), 267
 exportLiteralChildren() (*PMML44Super.ThetaRecursionState method*), 268
 exportLiteralChildren() (*PMML44Super.Time method*), 269

`exportLiteralChildren()`
 (*PMML44Super.TimeAnchor method*), 269
`exportLiteralChildren()`
 (*PMML44Super.TimeCycle method*), 270
`exportLiteralChildren()`
 (*PMML44Super.TimeException method*),
 271
`exportLiteralChildren()`
 (*PMML44Super.TimeSeries method*), 272
`exportLiteralChildren()`
 (*PMML44Super.TimeSeriesModel method*),
 273
`exportLiteralChildren()`
 (*PMML44Super.Timestamp method*), 276
`exportLiteralChildren()`
 (*PMML44Super.TimeValue method*), 275
`exportLiteralChildren()`
 (*PMML44Super.TrainingInstances method*),
 276
`exportLiteralChildren()`
 (*PMML44Super.TrainingParameters method*),
 277
`exportLiteralChildren()`
 (*PMML44Super.TransferFunctionValues*
 method), 278
`exportLiteralChildren()`
 (*PMML44Super.TransformationDictionary*
 method), 279
`exportLiteralChildren()`
 (*PMML44Super.TransitionMatrix method*),
 280
`exportLiteralChildren()`
 (*PMML44Super.TreeModel method*), 280
`exportLiteralChildren()`
 (*PMML44Super.Trend_ExpoSmooth method*),
 283
`exportLiteralChildren()`
 (*PMML44Super.TrendCoefficients method*),
 282
`exportLiteralChildren()`
 (*PMML44Super.TriangularDistributionForBN*
 method), 283
`exportLiteralChildren()` (*PMML44Super.True_*
 method), 284
`exportLiteralChildren()`
 (*PMML44Super.UniformDistribution method*),
 285
`exportLiteralChildren()`
 (*PMML44Super.UniformDistributionForBN*
 method), 286
`exportLiteralChildren()`
 (*PMML44Super.UnivariateStats method*),
 286
`exportLiteralChildren()`
 (*PMML44Super.Upper method*), 287
`exportLiteralChildren()` (*PMML44Super.Value*
 method), 288
`exportLiteralChildren()`
 (*PMML44Super.ValueProbability method*),
 289
`exportLiteralChildren()`
 (*PMML44Super.VariableWeight method*),
 290
`exportLiteralChildren()`
 (*PMML44Super.Variance method*), 291
`exportLiteralChildren()`
 (*PMML44Super.VarianceCoefficients method*),
 292
`exportLiteralChildren()`
 (*PMML44Super.VectorDictionary method*),
 293
`exportLiteralChildren()`
 (*PMML44Super.VectorFields method*), 294
`exportLiteralChildren()`
 (*PMML44Super.VectorInstance method*),
 294
`exportLiteralChildren()`
 (*PMML44Super.VerificationField method*),
 295
`exportLiteralChildren()`
 (*PMML44Super.VerificationFields method*),
 296
`exportLiteralChildren()`
 (*PMML44Super.XCoordinates method*),
 297
`exportLiteralChildren()`
 (*PMML44Super.YCoordinates method*), 298
 Extension (class in *PMML44Super*), 92

F

FactorList (class in *PMML44Super*), 93
 factory() (*PMML44Super.AbsoluteExponentialKernel*
 static method), 7
 factory() (*PMML44Super.Adadelta static method*), 8
 factory() (*PMML44Super.Adagrad static method*), 9
 factory() (*PMML44Super.Adam static method*), 10
 factory() (*PMML44Super.Adamax static method*), 11
 factory() (*PMML44Super.Aggregate static method*),
 12
 factory() (*PMML44Super.Alternate static method*),
 12
 factory() (*PMML44Super.Annotation static method*),
 13
 factory() (*PMML44Super.AnomalyDetectionModel*
 static method), 14
 factory() (*PMML44Super.Anova static method*), 16
 factory() (*PMML44Super.AnovaRow static method*),
 17

<code>factory()</code> (<i>PMML44Super.AntecedentSequence static method</i>), 18	<code>factory()</code> (<i>PMML44Super.Characteristic static method</i>), 43
<code>factory()</code> (<i>PMML44Super.AnyDistribution static method</i>), 19	<code>factory()</code> (<i>PMML44Super.Characteristics static method</i>), 44
<code>factory()</code> (<i>PMML44Super.Application static method</i>), 20	<code>factory()</code> (<i>PMML44Super.chebychev static method</i>), 299
<code>factory()</code> (<i>PMML44Super.Apply static method</i>), 21	<code>factory()</code> (<i>PMML44Super.ChildParent static method</i>), 45
<code>factory()</code> (<i>PMML44Super.AR static method</i>), 3	<code>factory()</code> (<i>PMML44Super.cityBlock static method</i>), 300
<code>factory()</code> (<i>PMML44Super.ARDSquaredExponentialKernel static method</i>), 4	<code>factory()</code> (<i>PMML44Super.ClassLabels static method</i>), 46
<code>factory()</code> (<i>PMML44Super.ARIMA static method</i>), 5	<code>factory()</code> (<i>PMML44Super.Cluster static method</i>), 46
<code>factory()</code> (<i>PMML44Super.ARMAPart static method</i>), 6	<code>factory()</code> (<i>PMML44Super.ClusteringField static method</i>), 47
<code>factory()</code> (<i>PMML44Super.ArrayType static method</i>), 23	<code>factory()</code> (<i>PMML44Super.ClusteringModel static method</i>), 49
<code>factory()</code> (<i>PMML44Super.AssociationModel static method</i>), 24	<code>factory()</code> (<i>PMML44Super.ClusteringModelQuality static method</i>), 50
<code>factory()</code> (<i>PMML44Super.AssociationRule static method</i>), 26	<code>factory()</code> (<i>PMML44Super.Coefficient static method</i>), 51
<code>factory()</code> (<i>PMML44Super.Attribute static method</i>), 27	<code>factory()</code> (<i>PMML44Super.Coefficients static method</i>), 52
<code>factory()</code> (<i>PMML44Super.BaseCumHazardTables static method</i>), 28	<code>factory()</code> (<i>PMML44Super.ComparisonMeasure static method</i>), 53
<code>factory()</code> (<i>PMML44Super.Baseline static method</i>), 29	<code>factory()</code> (<i>PMML44Super.Comparisons static method</i>), 54
<code>factory()</code> (<i>PMML44Super.BaselineCell static method</i>), 30	<code>factory()</code> (<i>PMML44Super.ComplexPartialScore static method</i>), 55
<code>factory()</code> (<i>PMML44Super.BaselineModel static method</i>), 31	<code>factory()</code> (<i>PMML44Super.CompoundPredicate static method</i>), 56
<code>factory()</code> (<i>PMML44Super.BaselineStratum static method</i>), 32	<code>factory()</code> (<i>PMML44Super.CompoundRule static method</i>), 58
<code>factory()</code> (<i>PMML44Super.BayesianNetworkModel static method</i>), 35	<code>factory()</code> (<i>PMML44Super.Con static method</i>), 59
<code>factory()</code> (<i>PMML44Super.BayesianNetworkNodes static method</i>), 37	<code>factory()</code> (<i>PMML44Super.ConfusionMatrix static method</i>), 59
<code>factory()</code> (<i>PMML44Super.BayesInput static method</i>), 33	<code>factory()</code> (<i>PMML44Super.ConsequentSequence static method</i>), 60
<code>factory()</code> (<i>PMML44Super.BayesInputs static method</i>), 34	<code>factory()</code> (<i>PMML44Super.Constant static method</i>), 61
<code>factory()</code> (<i>PMML44Super.BayesOutput static method</i>), 34	<code>factory()</code> (<i>PMML44Super.Constraints static method</i>), 62
<code>factory()</code> (<i>PMML44Super.binarySimilarity static method</i>), 298	<code>factory()</code> (<i>PMML44Super.ContinuousConditionalProbability static method</i>), 64
<code>factory()</code> (<i>PMML44Super.BlockIndicator static method</i>), 38	<code>factory()</code> (<i>PMML44Super.ContinuousDistribution static method</i>), 65
<code>factory()</code> (<i>PMML44Super.BoundaryValueMeans static method</i>), 38	<code>factory()</code> (<i>PMML44Super.ContinuousNode static method</i>), 66
<code>factory()</code> (<i>PMML44Super.BoundaryValues static method</i>), 39	<code>factory()</code> (<i>PMML44Super.ContStats static method</i>), 63
<code>factory()</code> (<i>PMML44Super.CategoricalPredictor static method</i>), 41	<code>factory()</code> (<i>PMML44Super.CorrelationFields static method</i>), 67
<code>factory()</code> (<i>PMML44Super.Categories static method</i>), 42	<code>factory()</code> (<i>PMML44Super.CorrelationMethods static method</i>), 68
<code>factory()</code> (<i>PMML44Super.Category static method</i>), 42	

<code>factory()</code> (PMML44Super.Correlations static method), 69	94
<code>factory()</code> (PMML44Super.CorrelationValues static method), 69	<code>factory()</code> (PMML44Super.False_ static method), 94
<code>factory()</code> (PMML44Super.COUNT_TABLE_TYPE static method), 40	<code>factory()</code> (PMML44Super.FieldColumnPair static method), 95
<code>factory()</code> (PMML44Super.Counts static method), 70	<code>factory()</code> (PMML44Super.FieldRef static method), 96
<code>factory()</code> (PMML44Super.Covariances static method), 71	<code>factory()</code> (PMML44Super.FieldValue static method), 96
<code>factory()</code> (PMML44Super.CovariateList static method), 72	<code>factory()</code> (PMML44Super.FieldValueCount static method), 97
<code>factory()</code> (PMML44Super.DataDictionary static method), 72	<code>factory()</code> (PMML44Super.FinalNoise static method), 98
<code>factory()</code> (PMML44Super.DataField static method), 73	<code>factory()</code> (PMML44Super.FinalNu static method), 99
<code>factory()</code> (PMML44Super.Decision static method), 75	<code>factory()</code> (PMML44Super.FinalOmega static method), 99
<code>factory()</code> (PMML44Super.Decisions static method), 77	<code>factory()</code> (PMML44Super.FinalPredictedNoise static method), 100
<code>factory()</code> (PMML44Super.DecisionTree static method), 75	<code>factory()</code> (PMML44Super.FinalStateVector static method), 100
<code>factory()</code> (PMML44Super.DeepNetwork static method), 78	<code>factory()</code> (PMML44Super.FinalTheta static method), 101
<code>factory()</code> (PMML44Super.DefineFunction static method), 80	<code>factory()</code> (PMML44Super.GARCH static method), 101
<code>factory()</code> (PMML44Super.Delimiter static method), 81	<code>factory()</code> (PMML44Super.GARCHPart static method), 102
<code>factory()</code> (PMML44Super.Denominator static method), 82	<code>factory()</code> (PMML44Super.GaussianDistribution static method), 103
<code>factory()</code> (PMML44Super.DerivedField static method), 83	<code>factory()</code> (PMML44Super.GaussianProcessModel static method), 104
<code>factory()</code> (PMML44Super.DiscreteConditionalProbability static method), 85	<code>factory()</code> (PMML44Super.GeneralizedExponentialKernel static method), 109
<code>factory()</code> (PMML44Super.DiscreteNode static method), 86	<code>factory()</code> (PMML44Super.GeneralRegressionModel static method), 106
<code>factory()</code> (PMML44Super.Discretize static method), 88	<code>factory()</code> (PMML44Super.Header static method), 111
<code>factory()</code> (PMML44Super.DiscretizeBin static method), 88	<code>factory()</code> (PMML44Super.HVector static method), 110
<code>factory()</code> (PMML44Super.DiscrStats static method), 85	<code>factory()</code> (PMML44Super.Indices static method), 113
<code>factory()</code> (PMML44Super.DocumentTermMatrix static method), 89	<code>factory()</code> (PMML44Super.InlineTable static method), 113
<code>factory()</code> (PMML44Super.DynamicRegressor static method), 90	<code>factory()</code> (PMML44Super.InstanceField static method), 114
<code>factory()</code> (PMML44Super.euclidean static method), 300	<code>factory()</code> (PMML44Super.InstanceFields static method), 115
<code>factory()</code> (PMML44Super.EventValues static method), 91	<code>factory()</code> (PMML44Super.INT_Entries static method), 112
<code>factory()</code> (PMML44Super.ExponentialSmoothing static method), 92	<code>factory()</code> (PMML44Super.INT_SparseArray static method), 112
<code>factory()</code> (PMML44Super.Extension static method), 93	<code>factory()</code> (PMML44Super.Interval static method), 115
<code>factory()</code> (PMML44Super.FactorList static method), 94	<code>factory()</code> (PMML44Super.Item static method), 116
	<code>factory()</code> (PMML44Super.ItemRef static method), 117
	<code>factory()</code> (PMML44Super.Itemset static method), 118
	<code>factory()</code> (PMML44Super.jaccard static method), 118

301
 factory() (PMML44Super.KalmanState static method), 120
 factory() (PMML44Super.KNNInput static method), 119
 factory() (PMML44Super.KNNInputs static method), 120
 factory() (PMML44Super.KohonenMap static method), 121
 factory() (PMML44Super.Lag static method), 122
 factory() (PMML44Super.Lambda static method), 123
 factory() (PMML44Super.LayerBias static method), 123
 factory() (PMML44Super.LayerParameters static method), 125
 factory() (PMML44Super.LayerWeights static method), 128
 factory() (PMML44Super.Level static method), 129
 factory() (PMML44Super.LiftData static method), 130
 factory() (PMML44Super.LiftGraph static method), 131
 factory() (PMML44Super.LinearKernelType static method), 132
 factory() (PMML44Super.LinearNorm static method), 132
 factory() (PMML44Super.LocalTransformations static method), 133
 factory() (PMML44Super.LognormalDistributionForBN static method), 134
 factory() (PMML44Super.Losses static method), 134
 factory() (PMML44Super.Lower static method), 135
 factory() (PMML44Super.MA static method), 136
 factory() (PMML44Super.MACoefficients static method), 137
 factory() (PMML44Super.MapValues static method), 138
 factory() (PMML44Super.MatCell static method), 139
 factory() (PMML44Super.Matrix static method), 139
 factory() (PMML44Super.MaximumLikelihoodStat static method), 140
 factory() (PMML44Super.Mean static method), 141
 factory() (PMML44Super.MeasurementMatrix static method), 142
 factory() (PMML44Super.Metrics static method), 143
 factory() (PMML44Super.MiningBuildTask static method), 144
 factory() (PMML44Super.MiningField static method), 144
 factory() (PMML44Super.MiningModel static method), 146
 factory() (PMML44Super.MiningSchema static method), 148
 factory() (PMML44Super.minkowski static method), 302
 factory() (PMML44Super.MissingValueWeights static method), 148
 factory() (PMML44Super.ModelExplanation static method), 149
 factory() (PMML44Super.ModelLiftGraph static method), 150
 factory() (PMML44Super.ModelStats static method), 151
 factory() (PMML44Super.ModelVerification static method), 151
 factory() (PMML44Super.MultivariateStat static method), 152
 factory() (PMML44Super.MultivariateStats static method), 154
 factory() (PMML44Super.Nadam static method), 155
 factory() (PMML44Super.NaiveBayesModel static method), 156
 factory() (PMML44Super.NearestNeighborModel static method), 157
 factory() (PMML44Super.NetworkLayer static method), 159
 factory() (PMML44Super.NeuralInput static method), 160
 factory() (PMML44Super.NeuralInputs static method), 161
 factory() (PMML44Super.NeuralLayer static method), 162
 factory() (PMML44Super.NeuralNetwork static method), 163
 factory() (PMML44Super.NeuralOutput static method), 165
 factory() (PMML44Super.NeuralOutputs static method), 166
 factory() (PMML44Super.Neuron static method), 167
 factory() (PMML44Super.Node static method), 168
 factory() (PMML44Super.NonseasonalComponent static method), 169
 factory() (PMML44Super.NonseasonalFactor static method), 170
 factory() (PMML44Super.NormalDistributionForBN static method), 173
 factory() (PMML44Super.NormContinuous static method), 171
 factory() (PMML44Super.NormDiscrete static method), 172
 factory() (PMML44Super.Numerator static method), 174
 factory() (PMML44Super.NumericInfo static method), 174
 factory() (PMML44Super.NumericPredictor static

- method*), 175
- `factory()` (*PMML44Super.Optimizers static method*), 176
- `factory()` (*PMML44Super.OptimumLiftGraph static method*), 178
- `factory()` (*PMML44Super.OutlierEffect static method*), 179
- `factory()` (*PMML44Super.Output static method*), 180
- `factory()` (*PMML44Super.OutputField static method*), 180
- `factory()` (*PMML44Super.PairCounts static method*), 191
- `factory()` (*PMML44Super.Parameter static method*), 192
- `factory()` (*PMML44Super.ParameterField static method*), 193
- `factory()` (*PMML44Super.ParameterList static method*), 194
- `factory()` (*PMML44Super.ParamMatrix static method*), 191
- `factory()` (*PMML44Super.ParentValue static method*), 195
- `factory()` (*PMML44Super.Partition static method*), 195
- `factory()` (*PMML44Super.PartitionFieldStats static method*), 196
- `factory()` (*PMML44Super.PastVariances static method*), 197
- `factory()` (*PMML44Super.PCell static method*), 183
- `factory()` (*PMML44Super.PCovCell static method*), 184
- `factory()` (*PMML44Super.PCovMatrix static method*), 185
- `factory()` (*PMML44Super.PMML static method*), 186
- `factory()` (*PMML44Super.PoissonDistribution static method*), 198
- `factory()` (*PMML44Super.PolynomialKernelType static method*), 199
- `factory()` (*PMML44Super.PPCell static method*), 189
- `factory()` (*PMML44Super.PPMatrix static method*), 190
- `factory()` (*PMML44Super.PredictiveModelQuality static method*), 200
- `factory()` (*PMML44Super.Predictor static method*), 202
- `factory()` (*PMML44Super.PredictorTerm static method*), 202
- `factory()` (*PMML44Super.PsiVector static method*), 203
- `factory()` (*PMML44Super.Quantile static method*), 204
- `factory()` (*PMML44Super.RadialBasisKernel static method*), 209
- `factory()` (*PMML44Super.RadialBasisKernelType static method*), 210
- `factory()` (*PMML44Super.RandomLiftGraph static method*), 210
- `factory()` (*PMML44Super.REAL_Entries static method*), 205
- `factory()` (*PMML44Super.REAL_SparseArray static method*), 205
- `factory()` (*PMML44Super.Regression static method*), 211
- `factory()` (*PMML44Super.RegressionModel static method*), 213
- `factory()` (*PMML44Super.RegressionTable static method*), 214
- `factory()` (*PMML44Super.RegressorValues static method*), 215
- `factory()` (*PMML44Super.Residuals static method*), 217
- `factory()` (*PMML44Super.ResidualSquareCoefficients static method*), 216
- `factory()` (*PMML44Super.ResultField static method*), 217
- `factory()` (*PMML44Super.RMSprop static method*), 206
- `factory()` (*PMML44Super.ROC static method*), 207
- `factory()` (*PMML44Super.ROCGraph static method*), 208
- `factory()` (*PMML44Super.row static method*), 302
- `factory()` (*PMML44Super.RuleSelectionMethod static method*), 218
- `factory()` (*PMML44Super.RuleSet static method*), 219
- `factory()` (*PMML44Super.RuleSetModel static method*), 221
- `factory()` (*PMML44Super.Scorecard static method*), 224
- `factory()` (*PMML44Super.ScoreDistribution static method*), 223
- `factory()` (*PMML44Super.SeasonalComponent static method*), 225
- `factory()` (*PMML44Super.SeasonalFactor static method*), 226
- `factory()` (*PMML44Super.Seasonality_ExpoSmooth static method*), 228
- `factory()` (*PMML44Super.SeasonalTrendDecomposition static method*), 227
- `factory()` (*PMML44Super.Segment static method*), 229
- `factory()` (*PMML44Super.Segmentation static method*), 231
- `factory()` (*PMML44Super.Sequence static method*), 232
- `factory()` (*PMML44Super.SequenceModel static method*), 233
- `factory()` (*PMML44Super.SequenceReference static method*), 233

- method*), 235
- `factory()` (*PMML44Super.SequenceRule static method*), 236
- `factory()` (*PMML44Super.SetPredicate static method*), 237
- `factory()` (*PMML44Super.SetReference static method*), 238
- `factory()` (*PMML44Super.SGD static method*), 222
- `factory()` (*PMML44Super.SigmoidKernelType static method*), 239
- `factory()` (*PMML44Super.simpleMatching static method*), 303
- `factory()` (*PMML44Super.SimplePredicate static method*), 240
- `factory()` (*PMML44Super.SimpleRule static method*), 241
- `factory()` (*PMML44Super.SimpleSetPredicate static method*), 242
- `factory()` (*PMML44Super.SpectralAnalysis static method*), 243
- `factory()` (*PMML44Super.squaredEuclidean static method*), 304
- `factory()` (*PMML44Super.StateSpaceModel static method*), 243
- `factory()` (*PMML44Super.StateVector static method*), 244
- `factory()` (*PMML44Super.SupportVector static method*), 245
- `factory()` (*PMML44Super.SupportVectorMachine static method*), 246
- `factory()` (*PMML44Super.SupportVectorMachineModel static method*), 247
- `factory()` (*PMML44Super.SupportVectors static method*), 249
- `factory()` (*PMML44Super.TableLocator static method*), 250
- `factory()` (*PMML44Super.tanimoto static method*), 304
- `factory()` (*PMML44Super.Target static method*), 251
- `factory()` (*PMML44Super.Targets static method*), 256
- `factory()` (*PMML44Super.TargetValue static method*), 252
- `factory()` (*PMML44Super.TargetValueCount static method*), 253
- `factory()` (*PMML44Super.TargetValueCounts static method*), 253
- `factory()` (*PMML44Super.TargetValueStat static method*), 254
- `factory()` (*PMML44Super.TargetValueStats static method*), 255
- `factory()` (*PMML44Super.Taxonomy static method*), 257
- `factory()` (*PMML44Super.TestDistributions static method*), 257
- `factory()` (*PMML44Super.TextCorpus static method*), 259
- `factory()` (*PMML44Super.TextDictionary static method*), 259
- `factory()` (*PMML44Super.TextDocument static method*), 260
- `factory()` (*PMML44Super.TextIndex static method*), 261
- `factory()` (*PMML44Super.TextIndexNormalization static method*), 263
- `factory()` (*PMML44Super.TextModel static method*), 264
- `factory()` (*PMML44Super.TextModelNormalization static method*), 266
- `factory()` (*PMML44Super.TextModelSimiliarity static method*), 267
- `factory()` (*PMML44Super.Theta static method*), 267
- `factory()` (*PMML44Super.ThetaRecursionState static method*), 268
- `factory()` (*PMML44Super.Time static method*), 269
- `factory()` (*PMML44Super.TimeAnchor static method*), 269
- `factory()` (*PMML44Super.TimeCycle static method*), 270
- `factory()` (*PMML44Super.TimeException static method*), 271
- `factory()` (*PMML44Super.TimeSeries static method*), 272
- `factory()` (*PMML44Super.TimeSeriesModel static method*), 273
- `factory()` (*PMML44Super.Timestamp static method*), 276
- `factory()` (*PMML44Super.TimeValue static method*), 275
- `factory()` (*PMML44Super.TrainingInstances static method*), 276
- `factory()` (*PMML44Super.TrainingParameters static method*), 277
- `factory()` (*PMML44Super.TransferFunctionValues static method*), 278
- `factory()` (*PMML44Super.TransformationDictionary static method*), 279
- `factory()` (*PMML44Super.TransitionMatrix static method*), 280
- `factory()` (*PMML44Super.TreeModel static method*), 280
- `factory()` (*PMML44Super.Trend_ExpoSmooth static method*), 283
- `factory()` (*PMML44Super.TrendCoefficients static method*), 282
- `factory()` (*PMML44Super.TriangularDistributionForBN static method*), 283
- `factory()` (*PMML44Super.True_ static method*), 284
- `factory()` (*PMML44Super.UniformDistribution static method*), 284

method), 285
 factory() (PMML44Super.UniformDistributionForBN static method), 286
 factory() (PMML44Super.UnivariateStats static method), 286
 factory() (PMML44Super.Upper static method), 287
 factory() (PMML44Super.Value static method), 289
 factory() (PMML44Super.ValueProbability static method), 289
 factory() (PMML44Super.VariableWeight static method), 290
 factory() (PMML44Super.Variance static method), 291
 factory() (PMML44Super.VarianceCoefficients static method), 292
 factory() (PMML44Super.VectorDictionary static method), 293
 factory() (PMML44Super.VectorFields static method), 294
 factory() (PMML44Super.VectorInstance static method), 295
 factory() (PMML44Super.VerificationField static method), 295
 factory() (PMML44Super.VerificationFields static method), 296
 factory() (PMML44Super.XCoordinates static method), 297
 factory() (PMML44Super.YCoordinates static method), 298
 False_ (class in PMML44Super), 94
 FieldColumnPair (class in PMML44Super), 95
 FieldRef (class in PMML44Super), 95
 FieldValue (class in PMML44Super), 96
 FieldValueCount (class in PMML44Super), 97
 FinalNoise (class in PMML44Super), 98
 FinalNu (class in PMML44Super), 98
 FinalOmega (class in PMML44Super), 99
 FinalPredictedNoise (class in PMML44Super), 99
 FinalStateVector (class in PMML44Super), 100
 FinalTheta (class in PMML44Super), 100

G

GARCH (class in PMML44Super), 101
 GARCHPart (class in PMML44Super), 102
 GaussianDistribution (class in PMML44Super), 103
 GaussianProcessModel (class in PMML44Super), 103
 GeneralizedExponentialKernel (class in PMML44Super), 109
 GeneralRegressionModel (class in PMML44Super), 105

get_AbsoluteExponentialKernel() (PMML44Super.GaussianProcessModel method), 104
 get_absoluteValue() (PMML44Super.Coefficients method), 52
 get_activationFunction() (PMML44Super.LayerParameters method), 125
 get_activationFunction() (PMML44Super.NeuralLayer method), 162
 get_activationFunction() (PMML44Super.NeuralNetwork method), 164
 get_activityRegularizerL1() (PMML44Super.LayerParameters method), 125
 get_activityRegularizerL2() (PMML44Super.LayerParameters method), 125
 get_Adadelta() (PMML44Super.Optimizers method), 176
 get_Adagrad() (PMML44Super.Optimizers method), 177
 get_Adam() (PMML44Super.Optimizers method), 177
 get_Adamax() (PMML44Super.Optimizers method), 177
 get_adj_r_squared() (PMML44Super.PredictiveModelQuality method), 200
 get_affinity() (PMML44Super.AssociationRule method), 26
 get_Aggregate() (PMML44Super.Apply method), 21
 get_Aggregate() (PMML44Super.ComplexPartialScore method), 55
 get_Aggregate() (PMML44Super.DefineFunction method), 80
 get_Aggregate() (PMML44Super.DerivedField method), 83
 get_aggregate() (PMML44Super.Lag method), 122
 get_Aggregate() (PMML44Super.Lower method), 135
 get_Aggregate() (PMML44Super.Mean method), 141
 get_Aggregate() (PMML44Super.OutputField method), 180
 get_Aggregate() (PMML44Super.TextIndex method), 261
 get_Aggregate() (PMML44Super.Upper method), 287
 get_Aggregate() (PMML44Super.Variance method), 291
 get_AIC() (PMML44Super.PredictiveModelQuality method), 200

[get_AICc\(\)](#) (*PMML44Super.PredictiveModelQuality method*), 200
[get_algorithm\(\)](#) (*PMML44Super.OutputField method*), 181
[get_algorithmName\(\)](#) (*PMML44Super.AnomalyDetectionModel method*), 15
[get_algorithmName\(\)](#) (*PMML44Super.AssociationModel method*), 24
[get_algorithmName\(\)](#) (*PMML44Super.BaselineModel method*), 31
[get_algorithmName\(\)](#) (*PMML44Super.BayesianNetworkModel method*), 36
[get_algorithmName\(\)](#) (*PMML44Super.ClusteringModel method*), 49
[get_algorithmName\(\)](#) (*PMML44Super.DecisionTree method*), 76
[get_algorithmName\(\)](#) (*PMML44Super.DeepNetwork method*), 78
[get_algorithmName\(\)](#) (*PMML44Super.GaussianProcessModel method*), 104
[get_algorithmName\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 107
[get_algorithmName\(\)](#) (*PMML44Super.MiningModel method*), 146
[get_algorithmName\(\)](#) (*PMML44Super.NaiveBayesModel method*), 156
[get_algorithmName\(\)](#) (*PMML44Super.NearestNeighborModel method*), 158
[get_algorithmName\(\)](#) (*PMML44Super.NeuralNetwork method*), 164
[get_algorithmName\(\)](#) (*PMML44Super.Regression method*), 211
[get_algorithmName\(\)](#) (*PMML44Super.RegressionModel method*), 213
[get_algorithmName\(\)](#) (*PMML44Super.RuleSetModel method*), 221
[get_algorithmName\(\)](#) (*PMML44Super.Scorecard method*), 224
[get_algorithmName\(\)](#) (*PMML44Super.SequenceModel method*), 234
[get_algorithmName\(\)](#) (*PMML44Super.SupportVectorMachineModel method*), 248
[get_algorithmName\(\)](#) (*PMML44Super.TextModel method*), 264
[get_algorithmName\(\)](#) (*PMML44Super.TimeSeriesModel method*), 273
[get_algorithmName\(\)](#) (*PMML44Super.TreeModel method*), 281
[get_algorithmType\(\)](#) (*PMML44Super.AnomalyDetectionModel method*), 15
[get_alpha\(\)](#) (*PMML44Super.Level method*), 129
[get_alphaDropoutRate\(\)](#) (*PMML44Super.LayerParameters method*), 125
[get_alphaDropoutSeed\(\)](#) (*PMML44Super.LayerParameters method*), 125
[get_Alternate\(\)](#) (*PMML44Super.TestDistributions method*), 257
[get_alternateTargetCategory\(\)](#) (*PMML44Super.SupportVectorMachine method*), 246
[get_altitude\(\)](#) (*PMML44Super.NeuralLayer method*), 162
[get_altitude\(\)](#) (*PMML44Super.NeuralNetwork method*), 164
[get_altitude\(\)](#) (*PMML44Super.Neuron method*), 167
[get_Annotation\(\)](#) (*PMML44Super.Header method*), 111
[get_AnomalyDetectionModel\(\)](#) (*PMML44Super.AnomalyDetectionModel method*), 14
[get_AnomalyDetectionModel\(\)](#) (*PMML44Super.PMML method*), 186
[get_AnomalyDetectionModel\(\)](#) (*PMML44Super.Segment method*), 229
[get_Anova\(\)](#) (*PMML44Super.UnivariateStats method*), 286
[get_AnovaRow\(\)](#) (*PMML44Super.Anova method*), 16
[get_antecedent\(\)](#) (*PMML44Super.AssociationRule method*), 26
[get_AntecedentSequence\(\)](#) (*PMML44Super.SequenceRule method*), 236
[get_AnyDistribution\(\)](#) (*PMML44Super.Alternate method*), 12
[get_AnyDistribution\(\)](#) (*PMML44Super.Baseline method*), 29
[get_AnyDistribution\(\)](#) (*PMML44Super.TargetValueStat method*), 254

`get_anytypeobjs_()` (PMML44Super.Extension method), 93
`get_anytypeobjs_()` (PMML44Super.row method), 302
`get_Application()` (PMML44Super.Header method), 111
`get_Apply()` (PMML44Super.Apply method), 21
`get_Apply()` (PMML44Super.ComplexPartialScore method), 55
`get_Apply()` (PMML44Super.DefineFunction method), 80
`get_Apply()` (PMML44Super.DerivedField method), 83
`get_Apply()` (PMML44Super.Lower method), 135
`get_Apply()` (PMML44Super.Mean method), 141
`get_Apply()` (PMML44Super.OutputField method), 180
`get_Apply()` (PMML44Super.TextIndex method), 261
`get_Apply()` (PMML44Super.Upper method), 287
`get_Apply()` (PMML44Super.Variance method), 291
`get_AR()` (PMML44Super.ARMAPart method), 6
`get_AR()` (PMML44Super.NonseasonalComponent method), 169
`get_AR()` (PMML44Super.SeasonalComponent method), 225
`get_architectureName()` (PMML44Super.DerivedField method), 83
`get_architectureName()` (PMML44Super.TrainingParameters method), 277
`get_ARDSquaredExponentialKernel()` (PMML44Super.GaussianProcessModel method), 104
`get_ARIMA()` (PMML44Super.TimeSeriesModel method), 273
`get_ARMAPart()` (PMML44Super.GARCH method), 101
`get_Array()` (PMML44Super.AR method), 3
`get_Array()` (PMML44Super.BoundaryValueMeans method), 38
`get_Array()` (PMML44Super.BoundaryValues method), 39
`get_Array()` (PMML44Super.ClassLabels method), 46
`get_Array()` (PMML44Super.Cluster method), 46
`get_Array()` (PMML44Super.CorrelationFields method), 67
`get_Array()` (PMML44Super.DiscrStats method), 85
`get_Array()` (PMML44Super.FinalNoise method), 98
`get_Array()` (PMML44Super.FinalNu method), 99
`get_Array()` (PMML44Super.FinalPredictedNoise method), 100
`get_Array()` (PMML44Super.FinalStateVector method), 100
`get_Array()` (PMML44Super.HVector method), 110
`get_Array()` (PMML44Super.Lambda method), 123
`get_Array()` (PMML44Super.MACoefficients method), 137
`get_Array()` (PMML44Super.Matrix method), 139
`get_Array()` (PMML44Super.MissingValueWeights method), 148
`get_Array()` (PMML44Super.NonseasonalFactor method), 170
`get_Array()` (PMML44Super.PartitionFieldStats method), 196
`get_Array()` (PMML44Super.PastVariances method), 197
`get_Array()` (PMML44Super.PsiVector method), 203
`get_Array()` (PMML44Super.Residuals method), 217
`get_Array()` (PMML44Super.SeasonalFactor method), 226
`get_Array()` (PMML44Super.Seasonality_ExpoSmooth method), 228
`get_Array()` (PMML44Super.SetPredicate method), 237
`get_Array()` (PMML44Super.SimpleSetPredicate method), 242
`get_Array()` (PMML44Super.StateVector method), 244
`get_Array()` (PMML44Super.TextDictionary method), 259
`get_Array()` (PMML44Super.TimeCycle method), 270
`get_Array()` (PMML44Super.TimeException method), 271
`get_Array()` (PMML44Super.TransferFunctionValues method), 278
`get_Array()` (PMML44Super.Trend_ExpoSmooth method), 283
`get_Array()` (PMML44Super.VectorInstance method), 295
`get_Array()` (PMML44Super.XCoordinates method), 297
`get_Array()` (PMML44Super.YCoordinates method), 298
`get_AssociationModel()` (PMML44Super.AnomalyDetectionModel method), 14
`get_AssociationModel()` (PMML44Super.PMML method), 186
`get_AssociationModel()` (PMML44Super.Segment method), 229
`get_AssociationRule()` (PMML44Super.AssociationModel method), 24
`get_Attribute()` (PMML44Super.Characteristic method), 43
`get_avgNumberOfItemsPerTA()` (PMML44Super.AssociationModel method), 24

`get_avgNumberOfItemsPerTransaction()`
 (*PMML44Super.SequenceModel method*), 234
`get_avgNumberOfTAsPerTAGroup()`
 (*PMML44Super.SequenceModel method*), 234
`get_BaseCumHazardTables()`
 (*PMML44Super.GeneralRegressionModel method*), 106
`get_Baseline()` (*PMML44Super.TestDistributions method*), 257
`get_BaselineCell()`
 (*PMML44Super.BaseCumHazardTables method*), 28
`get_BaselineCell()`
 (*PMML44Super.BaselineStratum method*), 32
`get_baselineMethod()` (*PMML44Super.Scorecard method*), 224
`get_BaselineModel()`
 (*PMML44Super.AnomalyDetectionModel method*), 14
`get_BaselineModel()` (*PMML44Super.PMML method*), 186
`get_BaselineModel()` (*PMML44Super.Segment method*), 229
`get_baselineScore()`
 (*PMML44Super.Characteristic method*), 43
`get_baselineScore()` (*PMML44Super.Scorecard method*), 224
`get_baselineStrataVariable()`
 (*PMML44Super.GeneralRegressionModel method*), 107
`get_BaselineStratum()`
 (*PMML44Super.BaseCumHazardTables method*), 28
`get_batchNormalizationAxis()`
 (*PMML44Super.LayerParameters method*), 125
`get_batchNormalizationCenter()`
 (*PMML44Super.LayerParameters method*), 125
`get_batchNormalizationEpsilon()`
 (*PMML44Super.LayerParameters method*), 125
`get_batchNormalizationMomentum()`
 (*PMML44Super.LayerParameters method*), 125
`get_batchNormalizationScale()`
 (*PMML44Super.LayerParameters method*), 125
`get_batchSize()` (*PMML44Super.LayerParameters method*), 125
`get_BayesianNetworkModel()`
 (*PMML44Super.AnomalyDetectionModel method*), 14
`get_BayesianNetworkModel()`
 (*PMML44Super.PMML method*), 186
`get_BayesianNetworkModel()`
 (*PMML44Super.Segment method*), 229
`get_BayesianNetworkNodes()`
 (*PMML44Super.BayesianNetworkModel method*), 35
`get_BayesInput()` (*PMML44Super.BayesInputs method*), 34
`get_BayesInputs()`
 (*PMML44Super.NaiveBayesModel method*), 156
`get_BayesOutput()`
 (*PMML44Super.NaiveBayesModel method*), 156
`get_bestFit()` (*PMML44Super.TimeSeriesModel method*), 273
`get_beta()` (*PMML44Super.PCell method*), 183
`get_beta_1()` (*PMML44Super.Adam method*), 10
`get_beta_1()` (*PMML44Super.Adamax method*), 11
`get_beta_1()` (*PMML44Super.Nadam method*), 155
`get_beta_2()` (*PMML44Super.Adam method*), 10
`get_beta_2()` (*PMML44Super.Adamax method*), 11
`get_beta_2()` (*PMML44Super.Nadam method*), 155
`get_betaConstraint()`
 (*PMML44Super.LayerParameters method*), 125
`get_betaInitializer()`
 (*PMML44Super.LayerParameters method*), 125
`get_betaRegularizer()`
 (*PMML44Super.LayerParameters method*), 125
`get_bias()` (*PMML44Super.Neuron method*), 167
`get_biasConstraint()`
 (*PMML44Super.LayerParameters method*), 125
`get_biasFlattenAxis()`
 (*PMML44Super.LayerBias method*), 123
`get_biasInitializer()`
 (*PMML44Super.LayerParameters method*), 125
`get_biasRegularizer()`
 (*PMML44Super.LayerParameters method*), 125
`get_biasShape()` (*PMML44Super.LayerBias method*), 123
`get_BIC()` (*PMML44Super.PredictiveModelQuality method*), 200
`get_binarySimilarity()`
 (*PMML44Super.ComparisonMeasure method*), 53

`get_binValue()` (PMML44Super.DiscretizeBin method), 89
`get_BlockIndicator()` (PMML44Super.Lag method), 122
`get_booleanOperator()` (PMML44Super.CompoundPredicate method), 56
`get_booleanOperator()` (PMML44Super.SimpleSetPredicate method), 242
`get_BoundaryValueMeans()` (PMML44Super.LiftGraph method), 131
`get_BoundaryValues()` (PMML44Super.LiftGraph method), 131
`get_BoundaryValues()` (PMML44Super.ROCGraph method), 208
`get_businessProblem()` (PMML44Super.Decisions method), 77
`get_c00_parameter()` (PMML44Super.binarySimilarity method), 298
`get_c01_parameter()` (PMML44Super.binarySimilarity method), 298
`get_c10_parameter()` (PMML44Super.binarySimilarity method), 298
`get_c11_parameter()` (PMML44Super.binarySimilarity method), 298
`get_cardinality()` (PMML44Super.Counts method), 70
`get_castInteger()` (PMML44Super.Target method), 251
`get_CategoricalPredictor()` (PMML44Super.RegressionTable method), 214
`get_CategoricalPredictor()` (PMML44Super.VectorFields method), 294
`get_categoricalScoringMethod()` (PMML44Super.NearestNeighborModel method), 158
`get_Categories()` (PMML44Super.Predictor method), 202
`get_Category()` (PMML44Super.Categories method), 42
`get_category()` (PMML44Super.Item method), 116
`get_category()` (PMML44Super.MultivariateStat method), 152
`get_Characteristic()` (PMML44Super.Characteristics method), 44
`get_Characteristics()` (PMML44Super.Scorecard method), 224
`get_chebychev()` (PMML44Super.ComparisonMeasure method), 53
`get_childField()` (PMML44Super.ChildParent method), 45
`get_ChildParent()` (PMML44Super.Taxonomy method), 257
`get_chiSquareValue()` (PMML44Super.MultivariateStat method), 152
`get_cityBlock()` (PMML44Super.ComparisonMeasure method), 53
`get_classificationMethod()` (PMML44Super.SupportVectorMachineModel method), 248
`get_ClassLabels()` (PMML44Super.ConfusionMatrix method), 59
`get_clipnorm()` (PMML44Super.Optimizers method), 177
`get_clipvalue()` (PMML44Super.Optimizers method), 177
`get_closure()` (PMML44Super.Interval method), 116
`get_Cluster()` (PMML44Super.ClusteringModel method), 49
`get_ClusteringField()` (PMML44Super.ClusteringModel method), 49
`get_ClusteringModel()` (PMML44Super.AnomalyDetectionModel method), 14
`get_ClusteringModel()` (PMML44Super.PMML method), 186
`get_ClusteringModel()` (PMML44Super.Segment method), 229
`get_ClusteringModelQuality()` (PMML44Super.ModelExplanation method), 149
`get_coef0()` (PMML44Super.PolynomialKernelType method), 199
`get_coef0()` (PMML44Super.SigmoidKernelType method), 239
`get_coefficient()` (PMML44Super.CategoricalPredictor method), 41
`get_Coefficient()` (PMML44Super.Coefficients method), 52
`get_coefficient()` (PMML44Super.NumericPredictor method), 175
`get_coefficient()` (PMML44Super.PredictorTerm method), 203
`get_Coefficients()` (PMML44Super.SupportVectorMachine

`method`), 246
`get_col()` (*PMML44Super.MatCell* `method`), 139
`get_column()` (*PMML44Super.FieldColumnPair* `method`), 95
`get_column()` (*PMML44Super.InstanceField* `method`), 114
`get_column()` (*PMML44Super.VerificationField* `method`), 295
`get_compareFunction()` (*PMML44Super.ClusteringField* `method`), 48
`get_compareFunction()` (*PMML44Super.ComparisonMeasure* `method`), 53
`get_compareFunction()` (*PMML44Super.KNNInput* `method`), 119
`get_ComparisonMeasure()` (*PMML44Super.ClusteringModel* `method`), 49
`get_ComparisonMeasure()` (*PMML44Super.NearestNeighborModel* `method`), 157
`get_Comparisons()` (*PMML44Super.ClusteringField* `method`), 47
`get_ComplexPartialScore()` (*PMML44Super.Attribute* `method`), 27
`get_CompoundPredicate()` (*PMML44Super.Attribute* `method`), 27
`get_CompoundPredicate()` (*PMML44Super.CompoundPredicate* `method`), 56
`get_CompoundPredicate()` (*PMML44Super.CompoundRule* `method`), 58
`get_CompoundPredicate()` (*PMML44Super.Node* `method`), 168
`get_CompoundPredicate()` (*PMML44Super.Segment* `method`), 229
`get_CompoundPredicate()` (*PMML44Super.SimpleRule* `method`), 241
`get_CompoundRule()` (*PMML44Super.CompoundRule* `method`), 58
`get_CompoundRule()` (*PMML44Super.RuleSet* `method`), 219
`get_Con()` (*PMML44Super.Neuron* `method`), 167
`get_confidence()` (*PMML44Super.AssociationRule* `method`), 26
`get_confidence()` (*PMML44Super.ScoreDistribution* `method`), 223
`get_confidence()` (*PMML44Super.SequenceRule* `method`), 236
`get_confidence()` (*PMML44Super.SimpleRule* `method`), 241
`get_confidenceLevel()` (*PMML44Super.MultivariateStat* `method`), 152
`get_confidenceLowerBound()` (*PMML44Super.MultivariateStat* `method`), 153
`get_confidenceUpperBound()` (*PMML44Super.MultivariateStat* `method`), 153
`get_ConfusionMatrix()` (*PMML44Super.PredictiveModelQuality* `method`), 200
`get_connectionLayerId()` (*PMML44Super.NetworkLayer* `method`), 159
`get_consequent()` (*PMML44Super.AssociationRule* `method`), 26
`get_ConsequentSequence()` (*PMML44Super.SequenceRule* `method`), 236
`get_Constant()` (*PMML44Super.Apply* `method`), 21
`get_constant()` (*PMML44Super.ARMAPart* `method`), 6
`get_Constant()` (*PMML44Super.ComplexPartialScore* `method`), 55
`get_Constant()` (*PMML44Super.DefineFunction* `method`), 80
`get_Constant()` (*PMML44Super.DerivedField* `method`), 83
`get_constant()` (*PMML44Super.GARCHPart* `method`), 102
`get_Constant()` (*PMML44Super.Lower* `method`), 135
`get_Constant()` (*PMML44Super.Mean* `method`), 141
`get_Constant()` (*PMML44Super.OutputField* `method`), 180
`get_Constant()` (*PMML44Super.TextIndex* `method`), 261
`get_Constant()` (*PMML44Super.Upper* `method`), 287
`get_Constant()` (*PMML44Super.Variance* `method`), 291
`get_constantTerm()` (*PMML44Super.ARIMA* `method`), 5
`get_Constraints()` (*PMML44Super.SequenceModel* `method`), 233
`get_ContinuousConditionalProbability()` (*PMML44Super.ContinuousNode* `method`), 66
`get_ContinuousDistribution()` (*PMML44Super.ContinuousConditionalProbability* `method`), 64
`get_ContinuousDistribution()`

(PMML44Super.ContinuousNode method), 66
 get_ContinuousNode() (PMML44Super.BayesianNetworkNodes method), 37
 get_continuousScoringMethod() (PMML44Super.NearestNeighborModel method), 158
 get_contrastMatrixType() (PMML44Super.Predictor method), 202
 get_ContStats() (PMML44Super.UnivariateStats method), 286
 get_coord1() (PMML44Super.KohonenMap method), 121
 get_coord2() (PMML44Super.KohonenMap method), 121
 get_coord3() (PMML44Super.KohonenMap method), 121
 get_copyright() (PMML44Super.Header method), 111
 get_CorrelationFields() (PMML44Super.Correlations method), 69
 get_CorrelationMethods() (PMML44Super.Correlations method), 69
 get_Correlations() (PMML44Super.ModelExplanation method), 149
 get_CorrelationValues() (PMML44Super.Correlations method), 69
 get_count() (PMML44Super.ContinuousConditionalProbability method), 64
 get_count() (PMML44Super.ContinuousNode method), 66
 get_count() (PMML44Super.DiscreteConditionalProbability method), 85
 get_count() (PMML44Super.DiscreteNode method), 87
 get_count() (PMML44Super.FieldValueCount method), 97
 get_count() (PMML44Super.TargetValueCount method), 253
 get_count() (PMML44Super.TimeException method), 271
 get_countHits() (PMML44Super.TextIndex method), 261
 get_Counts() (PMML44Super.PartitionFieldStats method), 196
 get_Counts() (PMML44Super.UnivariateStats method), 286
 get_CountTable() (PMML44Super.Baseline method), 29
 get_Covariances() (PMML44Super.Cluster method), 46
 get_CovariateList() (PMML44Super.GeneralRegressionModel method), 106
 get_criterion() (PMML44Super.RuleSelectionMethod method), 219
 get_croppingDims() (PMML44Super.LayerParameters method), 125
 get_cumHazard() (PMML44Super.BaselineCell method), 30
 get_cumulativeLink() (PMML44Super.GeneralRegressionModel method), 107
 get_d() (PMML44Super.NonseasonalComponent method), 169
 get_D() (PMML44Super.SeasonalComponent method), 225
 get_d00_parameter() (PMML44Super.binarySimilarity method), 298
 get_d01_parameter() (PMML44Super.binarySimilarity method), 298
 get_d10_parameter() (PMML44Super.binarySimilarity method), 298
 get_d11_parameter() (PMML44Super.binarySimilarity method), 299
 get_dampingCoefficient() (PMML44Super.OutlierEffect method), 179
 get_DataDictionary() (PMML44Super.PMML method), 186
 get_DataField() (PMML44Super.DataDictionary method), 72
 get_dataName() (PMML44Super.ClusteringModelQuality method), 50
 get_dataName() (PMML44Super.PredictiveModelQuality method), 200
 get_dataset() (PMML44Super.TrainingParameters method), 277
 get_datasetName() (PMML44Super.DerivedField method), 83
 get_dataType() (PMML44Super.Constant method), 61
 get_dataType() (PMML44Super.DataField method), 73
 get_dataType() (PMML44Super.DefineFunction method), 80
 get_dataType() (PMML44Super.DerivedField method), 83
 get_dataType() (PMML44Super.Discretize method), 88
 get_dataType() (PMML44Super.MapValues method), 138

`get_dataType()` (PMML44Super.OutputField method), 181
`get_dataType()` (PMML44Super.ParameterField method), 193
`get_dataType()` (PMML44Super.ResultField method), 217
`get_dataUsage()` (PMML44Super.PredictiveModelQuality method), 200
`get_decayRate()` (PMML44Super.Adadelta method), 8
`get_decayRate()` (PMML44Super.Adagrad method), 9
`get_decayRate()` (PMML44Super.Adam method), 10
`get_decayRate()` (PMML44Super.Adamax method), 11
`get_decayRate()` (PMML44Super.RMSprop method), 206
`get_decayRate()` (PMML44Super.SGD method), 222
`get_Decision()` (PMML44Super.Decisions method), 77
`get_Decisions()` (PMML44Super.OutputField method), 180
`get_DecisionTree()` (PMML44Super.MiningModel method), 146
`get_DecisionTree()` (PMML44Super.Node method), 168
`get_DeepNetwork()` (PMML44Super.AnomalyDetectionModel method), 14
`get_DeepNetwork()` (PMML44Super.PMML method), 186
`get_DeepNetwork()` (PMML44Super.Segment method), 229
`get_defaultChild()` (PMML44Super.Node method), 168
`get_defaultConfidence()` (PMML44Super.RuleSet method), 219
`get_defaultScore()` (PMML44Super.RuleSet method), 219
`get_defaultValue()` (PMML44Super.Apply method), 21
`get_defaultValue()` (PMML44Super.Discretize method), 88
`get_defaultValue()` (PMML44Super.INT_SparseArray method), 112
`get_defaultValue()` (PMML44Super.MapValues method), 138
`get_defaultValue()` (PMML44Super.REAL_SparseArray method), 205
`get_defaultValue()` (PMML44Super.TargetValue method), 252
`get_DefineFunction()` (PMML44Super.TransformationDictionary method), 279
`get_degree()` (PMML44Super.GeneralizedExponentialKernel method), 109
`get_degree()` (PMML44Super.PolynomialKernelType method), 199
`get_degreesOfFreedom()` (PMML44Super.AnovaRow method), 17
`get_degreesOfFreedom()` (PMML44Super.PredictiveModelQuality method), 200
`get_delay()` (PMML44Super.DynamicRegressor method), 90
`get_delimiter()` (PMML44Super.Delimiter method), 81
`get_Delimiter()` (PMML44Super.Sequence method), 232
`get_Delimiter()` (PMML44Super.SequenceRule method), 236
`get_delta()` (PMML44Super.Seasonality_ExpoSmooth method), 228
`get_Denominator()` (PMML44Super.DynamicRegressor method), 90
`get_depthMultiplier()` (PMML44Super.LayerParameters method), 125
`get_depthwiseConstraint()` (PMML44Super.LayerParameters method), 125
`get_DerivedField()` (PMML44Super.BayesInput method), 33
`get_DerivedField()` (PMML44Super.ContinuousNode method), 66
`get_DerivedField()` (PMML44Super.DiscreteNode method), 86
`get_DerivedField()` (PMML44Super.LocalTransformations method), 133
`get_DerivedField()` (PMML44Super.NeuralInput method), 160
`get_DerivedField()` (PMML44Super.NeuralOutput method), 165
`get_DerivedField()` (PMML44Super.TransformationDictionary method), 279
`get_description()` (PMML44Super.AbsoluteExponentialKernel method), 7
`get_description()`

(PMML44Super.ARDSquaredExponentialKernel
 method), 4
 get_description() (PMML44Super.Decision
 method), 75
 get_description() (PMML44Super.Decisions
 method), 77
 get_description() (PMML44Super.GeneralizedExponentialKernel
 method), 109
 get_description() (PMML44Super.Header
 method), 111
 get_description() (PMML44Super.LinearKernelType
 method), 132
 get_description() (PMML44Super.PolynomialKernelType
 method), 199
 get_description() (PMML44Super.RadialBasisKernel
 method), 209
 get_description() (PMML44Super.RadialBasisKernelType
 method), 210
 get_description() (PMML44Super.SigmoidKernelType
 method), 239
 get_df() (PMML44Super.MultivariateStat
 method), 153
 get_df() (PMML44Super.PCell
 method), 183
 get_diagDefault() (PMML44Super.Matrix
 method), 139
 get_difference() (PMML44Super.NonseasonalFactor
 method), 170
 get_difference() (PMML44Super.SeasonalFactor
 method), 226
 get_dilationRate() (PMML44Super.LayerParameters
 method), 125
 get_DiscreteConditionalProbability() (PMML44Super.DiscreteNode
 method), 86
 get_DiscreteNode() (PMML44Super.BayesianNetworkNodes
 method), 37
 get_Discretize() (PMML44Super.Apply
 method), 21
 get_Discretize() (PMML44Super.ComplexPartialScore
 method), 55
 get_Discretize() (PMML44Super.DefineFunction
 method), 80
 get_Discretize() (PMML44Super.DerivedField
 method), 83
 get_Discretize() (PMML44Super.Lower
 method), 135
 get_Discretize() (PMML44Super.Mean
 method), 141
 get_Discretize() (PMML44Super.OutputField
 method), 181
 get_Discretize() (PMML44Super.TextIndex
 method), 261
 get_Discretize() (PMML44Super.Upper
 method), 287
 get_Discretize() (PMML44Super.Variance
 method), 291
 get_DiscretizeBin() (PMML44Super.Discretize
 method), 88
 get_DiscrStats() (PMML44Super.UnivariateStats
 method), 286
 get_displayName() (PMML44Super.DataField
 method), 73
 get_displayName() (PMML44Super.DerivedField
 method), 83
 get_displayName() (PMML44Super.OutputField
 method), 181
 get_displayName() (PMML44Super.ParameterField
 method), 193
 get_displayName() (PMML44Super.ResultField
 method), 218
 get_displayName() (PMML44Super.TimeAnchor
 method), 270
 get_displayName() (PMML44Super.TimeCycle
 method), 270
 get_displayValue() (PMML44Super.Decision
 method), 75
 get_displayValue() (PMML44Super.TargetValue
 method), 252
 get_displayValue() (PMML44Super.Value
 method), 289
 get_distParameter() (PMML44Super.GeneralRegressionModel
 method), 107
 get_distribution() (PMML44Super.GeneralRegressionModel
 method), 107
 get_documentNormalization() (PMML44Super.TextModelNormalization
 method), 266
 get_DocumentTermMatrix() (PMML44Super.TextModel
 method), 264
 get_dropoutNoiseShape() (PMML44Super.LayerParameters
 method), 125
 get_dropoutRate() (PMML44Super.LayerParameters
 method), 125
 get_dropoutSeed() (PMML44Super.LayerParameters
 method), 125

`get_DynamicRegressor()` (PMML44Super.ARIMA method), 5
`get_DynamicRegressor()` (PMML44Super.StateSpaceModel method), 243
`get_endTime()` (PMML44Super.TimeSeries method), 272
`get_endTimeVariable()` (PMML44Super.GeneralRegressionModel method), 107
`get_epsilon()` (PMML44Super.Adadelta method), 8
`get_epsilon()` (PMML44Super.Adagrad method), 9
`get_epsilon()` (PMML44Super.Adam method), 10
`get_epsilon()` (PMML44Super.Adamax method), 11
`get_epsilon()` (PMML44Super.Nadam method), 155
`get_epsilon()` (PMML44Super.RMSprop method), 206
`get_euclidean()` (PMML44Super.ComparisonMeasure method), 53
`get_EventValues()` (PMML44Super.GeneralRegressionModel method), 106
`get_exponent()` (PMML44Super.MultivariateStat method), 153
`get_exponent()` (PMML44Super.NumericPredictor method), 175
`get_ExponentialSmoothing()` (PMML44Super.TimeSeriesModel method), 273
`get_extender()` (PMML44Super.Extension method), 93
`get_Extension()` (PMML44Super.AbsoluteExponentialKernel method), 7
`get_Extension()` (PMML44Super.Adadelta method), 8
`get_Extension()` (PMML44Super.Adagrad method), 9
`get_Extension()` (PMML44Super.Adam method), 10
`get_Extension()` (PMML44Super.Adamax method), 11
`get_Extension()` (PMML44Super.Aggregate method), 12
`get_Extension()` (PMML44Super.Annotation method), 13
`get_Extension()` (PMML44Super.AnomalyDetectionModel method), 14
`get_Extension()` (PMML44Super.Anova method), 16
`get_Extension()` (PMML44Super.AnovaRow method), 17
`get_Extension()` (PMML44Super.AntecedentSequence method), 18
`get_Extension()` (PMML44Super.AnyDistribution method), 19
`get_Extension()` (PMML44Super.Application method), 20
`get_Extension()` (PMML44Super.Apply method), 21
`get_Extension()` (PMML44Super.AR method), 3
`get_Extension()` (PMML44Super.ARDSquaredExponentialKernel method), 4
`get_Extension()` (PMML44Super.ARIMA method), 5
`get_Extension()` (PMML44Super.ARMAPart method), 6
`get_Extension()` (PMML44Super.AssociationModel method), 24
`get_Extension()` (PMML44Super.AssociationRule method), 26
`get_Extension()` (PMML44Super.Attribute method), 27
`get_Extension()` (PMML44Super.BaseCumHazardTables method), 28
`get_Extension()` (PMML44Super.BaselineCell method), 30
`get_Extension()` (PMML44Super.BaselineModel method), 31
`get_Extension()` (PMML44Super.BaselineStratum method), 32
`get_Extension()` (PMML44Super.BayesianNetworkModel method), 35
`get_Extension()` (PMML44Super.BayesianNetworkNodes method), 37
`get_Extension()` (PMML44Super.BayesInput method), 33
`get_Extension()` (PMML44Super.BayesInputs method), 34
`get_Extension()` (PMML44Super.BayesOutput method), 35
`get_Extension()` (PMML44Super.binarySimilarity method), 298
`get_Extension()` (PMML44Super.BlockIndicator method), 38
`get_Extension()` (PMML44Super.BoundaryValueMeans method), 38
`get_Extension()` (PMML44Super.BoundaryValues method), 39
`get_Extension()` (PMML44Super.CategoricalPredictor method), 41
`get_Extension()` (PMML44Super.Categories method), 42
`get_Extension()` (PMML44Super.Category method), 42
`get_Extension()` (PMML44Super.Characteristic method), 43
`get_Extension()` (PMML44Super.Characteristics method), 44

<code>get_Extension()</code>	(<i>PMML44Super.Chebyshev</i>	<i>method</i>), 40	
<i>method</i>), 299			
<code>get_Extension()</code>	(<i>PMML44Super.ChildParent</i>	<i>method</i>), 45	<code>get_Extension()</code> (<i>PMML44Super.Counts</i> <i>method</i>),
			70
<code>get_Extension()</code>	(<i>PMML44Super.cityBlock</i>	<i>method</i>), 300	<code>get_Extension()</code> (<i>PMML44Super.Covariances</i>
			<i>method</i>), 71
<code>get_Extension()</code>	(<i>PMML44Super.ClassLabels</i>	<i>method</i>), 46	<code>get_Extension()</code> (<i>PMML44Super.CovariateList</i>
			<i>method</i>), 72
<code>get_Extension()</code>	(<i>PMML44Super.Cluster</i> <i>method</i>),	46	<code>get_Extension()</code> (<i>PMML44Super.DataDictionary</i>
			<i>method</i>), 72
<code>get_Extension()</code>	(<i>PMML44Super.ClusteringField</i>	<i>method</i>), 47	<code>get_Extension()</code> (<i>PMML44Super.DataField</i>
			<i>method</i>), 73
<code>get_Extension()</code>	(<i>PMML44Super.ClusteringModel</i>	<i>method</i>), 49	<code>get_Extension()</code> (<i>PMML44Super.Decision</i>
			<i>method</i>), 75
<code>get_Extension()</code>	(<i>PMML44Super.ClusteringModelQuality</i>	<i>method</i>), 50	<code>get_Extension()</code> (<i>PMML44Super.Decisions</i>
			<i>method</i>), 77
<code>get_Extension()</code>	(<i>PMML44Super.Coefficient</i>	<i>method</i>), 51	<code>get_Extension()</code> (<i>PMML44Super.DecisionTree</i>
			<i>method</i>), 75
<code>get_Extension()</code>	(<i>PMML44Super.Coefficients</i>	<i>method</i>), 52	<code>get_Extension()</code> (<i>PMML44Super.DeepNetwork</i>
			<i>method</i>), 78
<code>get_Extension()</code>	(<i>PMML44Super.ComparisonMeasure</i>	<i>method</i>), 53	<code>get_Extension()</code> (<i>PMML44Super.DefineFunction</i>
			<i>method</i>), 80
<code>get_Extension()</code>	(<i>PMML44Super.Comparisons</i>	<i>method</i>), 54	<code>get_Extension()</code> (<i>PMML44Super.Delimiter</i>
			<i>method</i>), 81
<code>get_Extension()</code>	(<i>PMML44Super.ComplexPartialScore</i>	<i>method</i>), 55	<code>get_Extension()</code> (<i>PMML44Super.Denominator</i>
			<i>method</i>), 82
<code>get_Extension()</code>	(<i>PMML44Super.CompoundPredicate</i>	<i>method</i>), 56	<code>get_Extension()</code> (<i>PMML44Super.DerivedField</i>
			<i>method</i>), 83
<code>get_Extension()</code>	(<i>PMML44Super.CompoundRule</i>	<i>method</i>), 58	<code>get_Extension()</code> (<i>PMML44Super.DiscreteConditionalProbability</i>
			<i>method</i>), 85
<code>get_Extension()</code>	(<i>PMML44Super.Con</i> <i>method</i>), 59		<code>get_Extension()</code> (<i>PMML44Super.DiscreteNode</i>
			<i>method</i>), 86
<code>get_Extension()</code>	(<i>PMML44Super.ConfusionMatrix</i>	<i>method</i>), 59	<code>get_Extension()</code> (<i>PMML44Super.Discretize</i>
			<i>method</i>), 88
<code>get_Extension()</code>	(<i>PMML44Super.ConsequentSequence</i>	<i>method</i>), 60	<code>get_Extension()</code> (<i>PMML44Super.DiscretizeBin</i>
			<i>method</i>), 89
<code>get_Extension()</code>	(<i>PMML44Super.Constraints</i>	<i>method</i>), 62	<code>get_Extension()</code> (<i>PMML44Super.DiscrStats</i>
			<i>method</i>), 85
<code>get_Extension()</code>	(<i>PMML44Super.ContinuousConditionalProbability</i>	<i>method</i>), 64	<code>get_Extension()</code> (<i>PMML44Super.DocumentTermMatrix</i>
			<i>method</i>), 89
<code>get_Extension()</code>	(<i>PMML44Super.ContinuousDistribution</i>	<i>method</i>), 65	<code>get_Extension()</code> (<i>PMML44Super.DynamicRegressor</i>
			<i>method</i>), 90
<code>get_Extension()</code>	(<i>PMML44Super.ContinuousNode</i>	<i>method</i>), 66	<code>get_Extension()</code> (<i>PMML44Super.euclidean</i>
			<i>method</i>), 301
<code>get_Extension()</code>	(<i>PMML44Super.ContStats</i>	<i>method</i>), 63	<code>get_Extension()</code> (<i>PMML44Super.EventValues</i>
			<i>method</i>), 91
<code>get_Extension()</code>	(<i>PMML44Super.CorrelationFields</i>	<i>method</i>), 67	<code>get_Extension()</code> (<i>PMML44Super.FactorList</i>
			<i>method</i>), 94
<code>get_Extension()</code>	(<i>PMML44Super.CorrelationMethods</i>	<i>method</i>), 68	<code>get_Extension()</code> (<i>PMML44Super.False_</i> <i>method</i>),
			94
<code>get_Extension()</code>	(<i>PMML44Super.Correlations</i>	<i>method</i>), 69	<code>get_Extension()</code> (<i>PMML44Super.FieldColumnPair</i>
			<i>method</i>), 95
<code>get_Extension()</code>	(<i>PMML44Super.CorrelationValues</i>	<i>method</i>), 69	<code>get_Extension()</code> (<i>PMML44Super.FieldRef</i> <i>method</i>),
			96
<code>get_Extension()</code>	(<i>PMML44Super.COUNT_TABLE_TYPE</i>	<i>method</i>), 69	<code>get_Extension()</code> (<i>PMML44Super.FieldValue</i>

- method*), 96
- `get_Extension()` (*PMML44Super.FieldValueCount method*), 97
- `get_Extension()` (*PMML44Super.GARCH method*), 101
- `get_Extension()` (*PMML44Super.GARCHPart method*), 102
- `get_Extension()` (*PMML44Super.GaussianDistribution method*), 103
- `get_Extension()` (*PMML44Super.GaussianProcessModel method*), 104
- `get_Extension()` (*PMML44Super.GeneralizedExponentialKernel method*), 109
- `get_Extension()` (*PMML44Super.GeneralRegressionModel method*), 106
- `get_Extension()` (*PMML44Super.Header method*), 111
- `get_Extension()` (*PMML44Super.InlineTable method*), 113
- `get_Extension()` (*PMML44Super.InstanceField method*), 114
- `get_Extension()` (*PMML44Super.InstanceFields method*), 115
- `get_Extension()` (*PMML44Super.Interval method*), 115
- `get_Extension()` (*PMML44Super.Item method*), 116
- `get_Extension()` (*PMML44Super.ItemRef method*), 117
- `get_Extension()` (*PMML44Super.Itemset method*), 118
- `get_Extension()` (*PMML44Super.jaccard method*), 301
- `get_Extension()` (*PMML44Super.KNNInput method*), 119
- `get_Extension()` (*PMML44Super.KNNInputs method*), 120
- `get_Extension()` (*PMML44Super.KohonenMap method*), 121
- `get_Extension()` (*PMML44Super.Lag method*), 122
- `get_Extension()` (*PMML44Super.Lambda method*), 123
- `get_Extension()` (*PMML44Super.LayerBias method*), 123
- `get_Extension()` (*PMML44Super.LayerParameters method*), 125
- `get_Extension()` (*PMML44Super.LayerWeights method*), 128
- `get_Extension()` (*PMML44Super.LiftData method*), 130
- `get_Extension()` (*PMML44Super.LiftGraph method*), 131
- `get_Extension()` (*PMML44Super.LinearKernelType method*), 132
- `get_Extension()` (*PMML44Super.LinearNorm method*), 132
- `get_Extension()` (*PMML44Super.LocalTransformations method*), 133
- `get_Extension()` (*PMML44Super.LognormalDistributionForBN method*), 134
- `get_Extension()` (*PMML44Super.Losses method*), 134
- `get_Extension()` (*PMML44Super.Lower method*), 135
- `get_Extension()` (*PMML44Super.MA method*), 136
- `get_Extension()` (*PMML44Super.MACoefficients method*), 137
- `get_Extension()` (*PMML44Super.MapValues method*), 138
- `get_Extension()` (*PMML44Super.Mean method*), 141
- `get_Extension()` (*PMML44Super.MeasurementMatrix method*), 142
- `get_Extension()` (*PMML44Super.Metrics method*), 143
- `get_Extension()` (*PMML44Super.MiningBuildTask method*), 144
- `get_Extension()` (*PMML44Super.MiningField method*), 144
- `get_Extension()` (*PMML44Super.MiningModel method*), 146
- `get_Extension()` (*PMML44Super.MiningSchema method*), 148
- `get_Extension()` (*PMML44Super.minkowski method*), 302
- `get_Extension()` (*PMML44Super.MissingValueWeights method*), 148
- `get_Extension()` (*PMML44Super.ModelExplanation method*), 149
- `get_Extension()` (*PMML44Super.ModelLiftGraph method*), 150
- `get_Extension()` (*PMML44Super.ModelStats method*), 151
- `get_Extension()` (*PMML44Super.ModelVerification method*), 151
- `get_Extension()` (*PMML44Super.MultivariateStat method*), 152
- `get_Extension()` (*PMML44Super.MultivariateStats method*), 154
- `get_Extension()` (*PMML44Super.Nadam method*), 155
- `get_Extension()` (*PMML44Super.NaiveBayesModel method*), 156
- `get_Extension()` (*PMML44Super.NearestNeighborModel method*), 157
- `get_Extension()` (*PMML44Super.NetworkLayer method*), 159
- `get_Extension()` (*PMML44Super.NeuralInput*

<i>method</i>), 160		<i>method</i>), 196	
get_Extension() (PMML44Super.NeuralInputs		get_Extension() (PMML44Super.PastVariances	
<i>method</i>), 161		<i>method</i>), 197	
get_Extension() (PMML44Super.NeuralLayer		get_Extension() (PMML44Super.PCell	<i>method</i>),
<i>method</i>), 162		183	
get_Extension() (PMML44Super.NeuralNetwork		get_Extension() (PMML44Super.PCovCell	
<i>method</i>), 163		<i>method</i>), 184	
get_Extension() (PMML44Super.NeuralOutput		get_Extension() (PMML44Super.PCovMatrix	
<i>method</i>), 165		<i>method</i>), 185	
get_Extension() (PMML44Super.NeuralOutputs		get_Extension() (PMML44Super.PMML	<i>method</i>),
<i>method</i>), 166		186	
get_Extension() (PMML44Super.Neuron	<i>method</i>),	get_Extension() (PMML44Super.PoissonDistribution	
167		<i>method</i>), 198	
get_Extension() (PMML44Super.Node	<i>method</i>),	get_Extension() (PMML44Super.PolynomialKernelType	
168		<i>method</i>), 199	
get_Extension() (PMML44Super.NonseasonalComponent	<i>method</i>), 169	get_Extension() (PMML44Super.PPCell	<i>method</i>),
		189	
get_Extension() (PMML44Super.NonseasonalFactor	<i>method</i>), 170	get_Extension() (PMML44Super.PPMatrix	
		<i>method</i>), 190	
get_Extension() (PMML44Super.NormalDistributionForBN	<i>method</i>), 173	get_Extension() (PMML44Super.PredictiveModelQuality	
		<i>method</i>), 200	
get_Extension() (PMML44Super.NormContinuous	<i>method</i>), 171	get_Extension() (PMML44Super.Predictor	
		<i>method</i>), 202	
get_Extension() (PMML44Super.NormDiscrete	<i>method</i>), 172	get_Extension() (PMML44Super.PredictorTerm	
		<i>method</i>), 203	
get_Extension() (PMML44Super.Numerator	<i>method</i>), 174	get_Extension() (PMML44Super.PsiVector	
		<i>method</i>), 203	
get_Extension() (PMML44Super.NumericInfo	<i>method</i>), 174	get_Extension() (PMML44Super.Quantile	
		<i>method</i>), 204	
get_Extension() (PMML44Super.NumericPredictor	<i>method</i>), 175	get_Extension() (PMML44Super.RadialBasisKernel	
		<i>method</i>), 209	
get_Extension() (PMML44Super.Optimizers	<i>method</i>), 177	get_Extension() (PMML44Super.RadialBasisKernelType	
		<i>method</i>), 210	
get_Extension() (PMML44Super.OptimumLiftGraph	<i>method</i>), 178	get_Extension() (PMML44Super.RandomLiftGraph	
		<i>method</i>), 210	
get_Extension() (PMML44Super.OutlierEffect	<i>method</i>), 179	get_Extension() (PMML44Super.Regression	
		<i>method</i>), 211	
get_Extension() (PMML44Super.Output	<i>method</i>),	get_Extension() (PMML44Super.RegressionModel	
180		<i>method</i>), 213	
get_Extension() (PMML44Super.OutputField	<i>method</i>), 181	get_Extension() (PMML44Super.RegressionTable	
		<i>method</i>), 214	
get_Extension() (PMML44Super.PairCounts	<i>method</i>), 191	get_Extension() (PMML44Super.RegressorValues	
		<i>method</i>), 215	
get_Extension() (PMML44Super.Parameter	<i>method</i>), 192	get_Extension() (PMML44Super.Residuals	
		<i>method</i>), 217	
get_Extension() (PMML44Super.ParameterList	<i>method</i>), 194	get_Extension() (PMML44Super.ResidualSquareCoefficients	
		<i>method</i>), 216	
get_Extension() (PMML44Super.ParamMatrix	<i>method</i>), 192	get_Extension() (PMML44Super.ResultField	
		<i>method</i>), 217	
get_Extension() (PMML44Super.ParentValue	<i>method</i>), 195	get_Extension() (PMML44Super.RMSprop	
		<i>method</i>), 206	
get_Extension() (PMML44Super.Partition	<i>method</i>), 195	get_Extension() (PMML44Super.ROC	<i>method</i>),
		207	
get_Extension() (PMML44Super.PartitionFieldStats		get_Extension() (PMML44Super.ROCGraph	

`method`), 208
`get_Extension()` (*PMML44Super.RuleSelectionMethod*
`method`), 219
`get_Extension()` (*PMML44Super.RuleSet* `method`),
219
`get_Extension()` (*PMML44Super.RuleSetModel*
`method`), 221
`get_Extension()` (*PMML44Super.Scorecard*
`method`), 224
`get_Extension()` (*PMML44Super.ScoreDistribution*
`method`), 223
`get_Extension()` (*PMML44Super.SeasonalComponent*
`method`), 225
`get_Extension()` (*PMML44Super.SeasonalFactor*
`method`), 226
`get_Extension()` (*PMML44Super.Segment* `method`),
229
`get_Extension()` (*PMML44Super.Segmentation*
`method`), 231
`get_Extension()` (*PMML44Super.Sequence*
`method`), 232
`get_Extension()` (*PMML44Super.SequenceModel*
`method`), 233
`get_Extension()` (*PMML44Super.SequenceReference*
`method`), 235
`get_Extension()` (*PMML44Super.SequenceRule*
`method`), 236
`get_Extension()` (*PMML44Super.SetPredicate*
`method`), 237
`get_Extension()` (*PMML44Super.SetReference*
`method`), 238
`get_Extension()` (*PMML44Super.SGD* `method`),
222
`get_Extension()` (*PMML44Super.SigmoidKernelType*
`method`), 239
`get_Extension()` (*PMML44Super.simpleMatching*
`method`), 303
`get_Extension()` (*PMML44Super.SimplePredicate*
`method`), 240
`get_Extension()` (*PMML44Super.SimpleRule*
`method`), 241
`get_Extension()` (*PMML44Super.SimpleSetPredicate*
`method`), 242
`get_Extension()` (*PMML44Super.squaredEuclidean*
`method`), 304
`get_Extension()` (*PMML44Super.StateSpaceModel*
`method`), 243
`get_Extension()` (*PMML44Super.StateVector*
`method`), 244
`get_Extension()` (*PMML44Super.SupportVector*
`method`), 245
`get_Extension()` (*PMML44Super.SupportVectorMachine*
`method`), 246
`get_Extension()` (*PMML44Super.SupportVectorMachineModel*
`method`), 247
`get_Extension()` (*PMML44Super.SupportVectors*
`method`), 249
`get_Extension()` (*PMML44Super.TableLocator*
`method`), 250
`get_Extension()` (*PMML44Super.tanimoto*
`method`), 304
`get_Extension()` (*PMML44Super.Target* `method`),
251
`get_Extension()` (*PMML44Super.Targets* `method`),
256
`get_Extension()` (*PMML44Super.TargetValue*
`method`), 252
`get_Extension()` (*PMML44Super.TargetValueCount*
`method`), 253
`get_Extension()` (*PMML44Super.TargetValueCounts*
`method`), 253
`get_Extension()` (*PMML44Super.TargetValueStat*
`method`), 254
`get_Extension()` (*PMML44Super.TargetValueStats*
`method`), 255
`get_Extension()` (*PMML44Super.Taxonomy*
`method`), 257
`get_Extension()` (*PMML44Super.TestDistributions*
`method`), 257
`get_Extension()` (*PMML44Super.TextCorpus*
`method`), 259
`get_Extension()` (*PMML44Super.TextDictionary*
`method`), 259
`get_Extension()` (*PMML44Super.TextDocument*
`method`), 260
`get_Extension()` (*PMML44Super.TextIndex*
`method`), 261
`get_Extension()` (*PMML44Super.TextIndexNormalization*
`method`), 263
`get_Extension()` (*PMML44Super.TextModel*
`method`), 264
`get_Extension()` (*PMML44Super.TextModelNormalization*
`method`), 266
`get_Extension()` (*PMML44Super.TextModelSimilarity*
`method`), 267
`get_Extension()` (*PMML44Super.Time* `method`),
269
`get_Extension()` (*PMML44Super.TimeSeriesModel*
`method`), 273
`get_Extension()` (*PMML44Super.Timestamp*
`method`), 276
`get_Extension()` (*PMML44Super.TrainingInstances*
`method`), 276
`get_Extension()` (*PMML44Super.TrainingParameters*
`method`), 277
`get_Extension()` (*PMML44Super.TransformationDictionary*
`method`), 279
`get_Extension()` (*PMML44Super.TransitionMatrix*

`method`), 280
`get_Extension()` (*PMML44Super.TreeModel* `method`), 280
`get_Extension()` (*PMML44Super.TrendCoefficients* `method`), 282
`get_Extension()` (*PMML44Super.TriangularDistributionForBN* `method`), 284
`get_Extension()` (*PMML44Super.True_* `method`), 284
`get_Extension()` (*PMML44Super.UniformDistribution* `method`), 285
`get_Extension()` (*PMML44Super.UniformDistributionForBN* `method`), 286
`get_Extension()` (*PMML44Super.UnivariateStats* `method`), 286
`get_Extension()` (*PMML44Super.Upper* `method`), 288
`get_Extension()` (*PMML44Super.Value* `method`), 289
`get_Extension()` (*PMML44Super.ValueProbability* `method`), 289
`get_Extension()` (*PMML44Super.VariableWeight* `method`), 290
`get_Extension()` (*PMML44Super.Variance* `method`), 291
`get_Extension()` (*PMML44Super.VarianceCoefficients* `method`), 292
`get_Extension()` (*PMML44Super.VectorDictionary* `method`), 293
`get_Extension()` (*PMML44Super.VectorFields* `method`), 294
`get_Extension()` (*PMML44Super.VectorInstance* `method`), 295
`get_Extension()` (*PMML44Super.VerificationField* `method`), 295
`get_Extension()` (*PMML44Super.VerificationFields* `method`), 296
`get_Extension()` (*PMML44Super.XCoordinates* `method`), 297
`get_Extension()` (*PMML44Super.YCoordinates* `method`), 298
`get_FactorList()` (*PMML44Super.GeneralRegressionModel* `method`), 106
`get_False()` (*PMML44Super.Attribute* `method`), 27
`get_False()` (*PMML44Super.CompoundPredicate* `method`), 56
`get_False()` (*PMML44Super.CompoundRule* `method`), 58
`get_False()` (*PMML44Super.Node* `method`), 168
`get_False()` (*PMML44Super.Segment* `method`), 229
`get_False()` (*PMML44Super.SimpleRule* `method`), 241
`get_feature()` (*PMML44Super.OutputField* `method`), 181
`get_feature()` (*PMML44Super.ResultField* `method`), 218
`get_featureMaps()` (*PMML44Super.LayerParameters* `method`), 125
`get_Fields()` (*PMML44Super.Aggregate* `method`), 12
`get_field()` (*PMML44Super.BlockIndicator* `method`), 38
`get_field()` (*PMML44Super.ClusteringField* `method`), 48
`get_field()` (*PMML44Super.Discretize* `method`), 88
`get_field()` (*PMML44Super.DynamicRegressor* `method`), 90
`get_field()` (*PMML44Super.FieldColumnPair* `method`), 95
`get_field()` (*PMML44Super.FieldRef* `method`), 96
`get_field()` (*PMML44Super.FieldValue* `method`), 97
`get_field()` (*PMML44Super.FieldValueCount* `method`), 97
`get_field()` (*PMML44Super.InstanceField* `method`), 114
`get_field()` (*PMML44Super.Item* `method`), 116
`get_field()` (*PMML44Super.KNNInput* `method`), 119
`get_field()` (*PMML44Super.Lag* `method`), 122
`get_field()` (*PMML44Super.NormContinuous* `method`), 171
`get_field()` (*PMML44Super.NormDiscrete* `method`), 172
`get_field()` (*PMML44Super.PartitionFieldStats* `method`), 196
`get_field()` (*PMML44Super.SetPredicate* `method`), 237
`get_field()` (*PMML44Super.SimplePredicate* `method`), 240
`get_field()` (*PMML44Super.SimpleSetPredicate* `method`), 242
`get_field()` (*PMML44Super.Target* `method`), 251
`get_field()` (*PMML44Super.TestDistributions* `method`), 258
`get_field()` (*PMML44Super.TimeSeries* `method`), 272
`get_field()` (*PMML44Super.UnivariateStats* `method`), 287
`get_field()` (*PMML44Super.VariableWeight* `method`), 290
`get_field()` (*PMML44Super.VerificationField* `method`), 295
`get_FieldColumnPair()` (*PMML44Super.ChildParent* `method`), 45
`get_FieldColumnPair()` (*PMML44Super.MapValues* `method`), 138
`get_fieldCount()` (*PMML44Super.ModelVerification* `method`), 152
`get_fieldCount()` (*PMML44Super.TrainingInstances*

[method](#)), 276
[get_fieldName\(\)](#) ([PMML44Super.BayesInput method](#)), 33
[get_fieldName\(\)](#) ([PMML44Super.BayesOutput method](#)), 35
[get_FieldRef\(\)](#) ([PMML44Super.Apply method](#)), 21
[get_FieldRef\(\)](#) ([PMML44Super.Baseline method](#)), 29
[get_FieldRef\(\)](#) ([PMML44Super.ComplexPartialScore method](#)), 55
[get_FieldRef\(\)](#) ([PMML44Super.DefineFunction method](#)), 80
[get_FieldRef\(\)](#) ([PMML44Super.DerivedField method](#)), 83
[get_FieldRef\(\)](#) ([PMML44Super.Lower method](#)), 135
[get_FieldRef\(\)](#) ([PMML44Super.Mean method](#)), 141
[get_FieldRef\(\)](#) ([PMML44Super.OutputField method](#)), 181
[get_FieldRef\(\)](#) ([PMML44Super.PredictorTerm method](#)), 203
[get_FieldRef\(\)](#) ([PMML44Super.TextIndex method](#)), 261
[get_FieldRef\(\)](#) ([PMML44Super.Upper method](#)), 288
[get_FieldRef\(\)](#) ([PMML44Super.Variance method](#)), 291
[get_FieldRef\(\)](#) ([PMML44Super.VectorFields method](#)), 294
[get_FieldValue\(\)](#) ([PMML44Super.COUNT_TABLE_TYPE method](#)), 40
[get_FieldValue\(\)](#) ([PMML44Super.FieldValue method](#)), 96
[get_FieldValueCount\(\)](#) ([PMML44Super.COUNT_TABLE_TYPE method](#)), 40
[get_FieldValueCount\(\)](#) ([PMML44Super.FieldValue method](#)), 96
[get_fieldWeight\(\)](#) ([PMML44Super.ClusteringField method](#)), 48
[get_fieldWeight\(\)](#) ([PMML44Super.KNNInput method](#)), 119
[get_file\(\)](#) ([PMML44Super.TextDocument method](#)), 260
[get_FinalNoise\(\)](#) ([PMML44Super.ThetaRecursionState method](#)), 268
[get_FinalNu\(\)](#) ([PMML44Super.ThetaRecursionState method](#)), 268
[get_FinalOmega\(\)](#) ([PMML44Super.KalmanState method](#)), 120
[get_FinalPredictedNoise\(\)](#) ([PMML44Super.ThetaRecursionState method](#)), 268
[get_FinalStateVector\(\)](#) ([PMML44Super.KalmanState method](#)), 120
[get_FinalTheta\(\)](#) ([PMML44Super.ThetaRecursionState method](#)), 268
[get_flattenAxis\(\)](#) ([PMML44Super.LayerParameters method](#)), 125
[get_framework\(\)](#) ([PMML44Super.TrainingParameters method](#)), 277
[get_from\(\)](#) ([PMML44Super.Con method](#)), 59
[get_fStatistic\(\)](#) ([PMML44Super.MultivariateStat method](#)), 153
[get_fStatistic\(\)](#) ([PMML44Super.PredictiveModelQuality method](#)), 200
[get_function\(\)](#) ([PMML44Super.Aggregate method](#)), 12
[get_function\(\)](#) ([PMML44Super.Apply method](#)), 21
[get_functionName\(\)](#) ([PMML44Super.AnomalyDetectionModel method](#)), 15
[get_functionName\(\)](#) ([PMML44Super.AssociationModel method](#)), 24
[get_functionName\(\)](#) ([PMML44Super.BaselineModel method](#)), 31
[get_functionName\(\)](#) ([PMML44Super.BayesianNetworkModel method](#)), 36
[get_functionName\(\)](#) ([PMML44Super.ClusteringModel method](#)), 49
[get_functionName\(\)](#) ([PMML44Super.DecisionTree method](#)), 76
[get_functionName\(\)](#) ([PMML44Super.DeepNetwork method](#)), 78
[get_functionName\(\)](#) ([PMML44Super.GaussianProcessModel method](#)), 104
[get_functionName\(\)](#) ([PMML44Super.GeneralRegressionModel method](#)), 107
[get_functionName\(\)](#) ([PMML44Super.MiningModel method](#)), 146
[get_functionName\(\)](#) ([PMML44Super.NaiveBayesModel method](#)), 156
[get_functionName\(\)](#) ([PMML44Super.NearestNeighborModel method](#)), 158
[get_functionName\(\)](#) ([PMML44Super.NeuralNetwork method](#)), 164
[get_functionName\(\)](#) ([PMML44Super.Regression](#)

[method](#)), 211
[get_functionName\(\)](#) ([PMML44Super.RegressionModel](#) [method](#)), 213
[get_functionName\(\)](#) ([PMML44Super.RuleSetModel](#) [method](#)), 221
[get_functionName\(\)](#) ([PMML44Super.Scorecard](#) [method](#)), 224
[get_functionName\(\)](#) ([PMML44Super.SequenceModel](#) [method](#)), 234
[get_functionName\(\)](#) ([PMML44Super.SupportVectorMachineModel](#) [method](#)), 248
[get_functionName\(\)](#) ([PMML44Super.TextModel](#) [method](#)), 264
[get_functionName\(\)](#) ([PMML44Super.TimeSeriesModel](#) [method](#)), 274
[get_functionName\(\)](#) ([PMML44Super.TreeModel](#) [method](#)), 281
[get_futureValuesMethod\(\)](#) ([PMML44Super.DynamicRegressor](#) [method](#)), 90
[get_fValue\(\)](#) ([PMML44Super.AnovaRow](#) [method](#)), 17
[get_gamma\(\)](#) ([PMML44Super.AbsoluteExponentialKernel](#) [method](#)), 7
[get_gamma\(\)](#) ([PMML44Super.ARDSquaredExponentialKernel](#) [method](#)), 4
[get_gamma\(\)](#) ([PMML44Super.GeneralizedExponentialKernel](#) [method](#)), 109
[get_gamma\(\)](#) ([PMML44Super.PolynomialKernelType](#) [method](#)), 199
[get_gamma\(\)](#) ([PMML44Super.RadialBasisKernel](#) [method](#)), 209
[get_gamma\(\)](#) ([PMML44Super.RadialBasisKernelType](#) [method](#)), 210
[get_gamma\(\)](#) ([PMML44Super.SigmoidKernelType](#) [method](#)), 239
[get_gamma\(\)](#) ([PMML44Super.Trend_ExpoSmooth](#) [method](#)), 283
[get_gammaConstraint\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 125
[get_gammaInitializer\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 125
[get_gammaRegularizer\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 125
[get_gap\(\)](#) ([PMML44Super.Delimiter](#) [method](#)), 81
[get_GARCH\(\)](#) ([PMML44Super.TimeSeriesModel](#) [method](#)), 273
[get_GARCHPart\(\)](#) ([PMML44Super.GARCH](#) [method](#)), 101
[get_GaussianDistribution\(\)](#) ([PMML44Super.Alternate](#) [method](#)), 12
[get_GaussianDistribution\(\)](#) ([PMML44Super.Baseline](#) [method](#)), 29
[get_GaussianDistribution\(\)](#) ([PMML44Super.TargetValueStat](#) [method](#)), 254
[get_gaussianDropoutRate\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 125
[get_gaussianNoiseStdev\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 125
[get_GaussianProcessModel\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) [method](#)), 14
[get_GaussianProcessModel\(\)](#) ([PMML44Super.PMML](#) [method](#)), 186
[get_GaussianProcessModel\(\)](#) ([PMML44Super.Segment](#) [method](#)), 229
[get_GeneralizedExponentialKernel\(\)](#) ([PMML44Super.GaussianProcessModel](#) [method](#)), 104
[get_generalLUAAlpha\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 125
[get_GeneralRegressionModel\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) [method](#)), 14
[get_GeneralRegressionModel\(\)](#) ([PMML44Super.PMML](#) [method](#)), 186
[get_GeneralRegressionModel\(\)](#) ([PMML44Super.Segment](#) [method](#)), 229
[get_globalTermWeights\(\)](#) ([PMML44Super.TextModelNormalization](#) [method](#)), 266
[get_gp\(\)](#) ([PMML44Super.GARCHPart](#) [method](#)), 102
[get_gq\(\)](#) ([PMML44Super.GARCHPart](#) [method](#)), 102
[get_groupField\(\)](#) ([PMML44Super.Aggregate](#) [method](#)), 12
[get_Header\(\)](#) ([PMML44Super.PMML](#) [method](#)), 186
[get_highValue\(\)](#) ([PMML44Super.MiningField](#) [method](#)), 145
[get_HVector\(\)](#) ([PMML44Super.KalmanState](#) [method](#)), 120
[get_i\(\)](#) ([PMML44Super.Theta](#) [method](#)), 267
[get_id\(\)](#) ([PMML44Super.AssociationRule](#) [method](#)), 26
[get_id\(\)](#) ([PMML44Super.Cluster](#) [method](#)), 47
[get_id\(\)](#) ([PMML44Super.Item](#) [method](#)), 116
[get_id\(\)](#) ([PMML44Super.Itemset](#) [method](#)), 118

[get_id\(\) \(PMML44Super.NeuralInput method\), 160](#)
[get_id\(\) \(PMML44Super.Neuron method\), 167](#)
[get_id\(\) \(PMML44Super.Node method\), 168](#)
[get_id\(\) \(PMML44Super.Segment method\), 229](#)
[get_id\(\) \(PMML44Super.Sequence method\), 232](#)
[get_id\(\) \(PMML44Super.SequenceRule method\), 236](#)
[get_id\(\) \(PMML44Super.SetPredicate method\), 237](#)
[get_id\(\) \(PMML44Super.SimpleRule method\), 241](#)
[get_id\(\) \(PMML44Super.TextDocument method\), 260](#)
[get_id\(\) \(PMML44Super.VectorInstance method\), 295](#)
[get_importance\(\) \(PMML44Super.MiningField method\), 145](#)
[get_importance\(\) \(PMML44Super.MultivariateStat method\), 153](#)
[get_index\(\) \(PMML44Super.TimeValue method\), 275](#)
[get_Indices\(\) \(PMML44Super.INT_SparseArray method\), 112](#)
[get_Indices\(\) \(PMML44Super.REAL_SparseArray method\), 205](#)
[get_inferenceMethod\(\) \(PMML44Super.BayesianNetworkModel method\), 36](#)
[get_inField\(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[get_initialScore\(\) \(PMML44Super.Scorecard method\), 224](#)
[get_InlineTable\(\) \(PMML44Super.ChildParent method\), 45](#)
[get_InlineTable\(\) \(PMML44Super.MapValues method\), 138](#)
[get_InlineTable\(\) \(PMML44Super.ModelVerification method\), 151](#)
[get_InlineTable\(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[get_InlineTable\(\) \(PMML44Super.TrainingInstances method\), 276](#)
[get_inputDimension\(\) \(PMML44Super.LayerParameters method\), 126](#)
[get_inputFieldName\(\) \(PMML44Super.NetworkLayer method\), 159](#)
[get_InstanceField\(\) \(PMML44Super.InstanceFields method\), 115](#)
[get_InstanceFields\(\) \(PMML44Super.TrainingInstances method\), 276](#)
[get_instanceIdVariable\(\) \(PMML44Super.NearestNeighborModel method\), 158](#)
[get_INT_Entries\(\) \(PMML44Super.INT_SparseArray method\), 112](#)
[get_intercept\(\) \(PMML44Super.RegressionTable method\), 214](#)
[get_intercept\(\) \(PMML44Super.StateSpaceModel method\), 243](#)
[get_interpolationMethod\(\) \(PMML44Super.TimeSeries method\), 272](#)
[get_interQuartileRange\(\) \(PMML44Super.NumericInfo method\), 174](#)
[get_Interval\(\) \(PMML44Super.ContStats method\), 63](#)
[get_Interval\(\) \(PMML44Super.DataField method\), 73](#)
[get_Interval\(\) \(PMML44Super.DiscretizeBin method\), 89](#)
[get_Interval\(\) \(PMML44Super.EventValues method\), 91](#)
[get_invalidFreq\(\) \(PMML44Super.Counts method\), 70](#)
[get_invalidValueReplacement\(\) \(PMML44Super.MiningField method\), 145](#)
[get_invalidValueTreatment\(\) \(PMML44Super.Apply method\), 21](#)
[get_invalidValueTreatment\(\) \(PMML44Super.MiningField method\), 145](#)
[get_isCaseSensitive\(\) \(PMML44Super.TextIndex method\), 261](#)
[get_isCaseSensitive\(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[get_isCenterField\(\) \(PMML44Super.ClusteringField method\), 48](#)
[get_isCyclic\(\) \(PMML44Super.DataField method\), 73](#)
[get_isFinalResult\(\) \(PMML44Super.OutputField method\), 181](#)
[get_isIntercept\(\) \(PMML44Super.MultivariateStat method\), 153](#)
[get_isMultiValued\(\) \(PMML44Super.OutputField method\), 181](#)
[get_isRecursive\(\) \(PMML44Super.ChildParent method\), 45](#)
[get_isScorable\(\) \(PMML44Super.AssociationModel method\), 24](#)
[get_isScorable\(\) \(PMML44Super.BaselineModel method\), 31](#)
[get_isScorable\(\) \(PMML44Super.BayesianNetworkModel method\), 36](#)

<code>get_isScorable()</code> (PMML44Super.ClusteringModel method), 49	<code>get_kernelConstraint()</code> (PMML44Super.LayerParameters method), 126
<code>get_isScorable()</code> (PMML44Super.DeepNetwork method), 78	<code>get_kernelInitializer()</code> (PMML44Super.LayerParameters method), 126
<code>get_isScorable()</code> (PMML44Super.GaussianProcessModel method), 104	<code>get_kernelRegularizer()</code> (PMML44Super.LayerParameters method), 126
<code>get_isScorable()</code> (PMML44Super.GeneralRegressionModel method), 107	<code>get_kind()</code> (PMML44Super.ComparisonMeasure method), 53
<code>get_isScorable()</code> (PMML44Super.MiningModel method), 146	<code>get_kind()</code> (PMML44Super.Matrix method), 140
<code>get_isScorable()</code> (PMML44Super.NaiveBayesModel method), 156	<code>get_KNNInput()</code> (PMML44Super.KNNInputs method), 120
<code>get_isScorable()</code> (PMML44Super.NearestNeighborModel method), 158	<code>get_KNNInputs()</code> (PMML44Super.NearestNeighborModel method), 157
<code>get_isScorable()</code> (PMML44Super.NeuralNetwork method), 164	<code>get_KohonenMap()</code> (PMML44Super.Cluster method), 47
<code>get_isScorable()</code> (PMML44Super.RegressionModel method), 213	<code>get_label()</code> (PMML44Super.BaselineStratum method), 32
<code>get_isScorable()</code> (PMML44Super.RuleSetModel method), 221	<code>get_label()</code> (PMML44Super.Parameter method), 192
<code>get_isScorable()</code> (PMML44Super.Scorecard method), 224	<code>get_Lag()</code> (PMML44Super.Apply method), 21
<code>get_isScorable()</code> (PMML44Super.SequenceModel method), 234	<code>get_Lag()</code> (PMML44Super.ComplexPartialScore method), 55
<code>get_isScorable()</code> (PMML44Super.SupportVectorMachineModel method), 248	<code>get_Lag()</code> (PMML44Super.DefineFunction method), 80
<code>get_isScorable()</code> (PMML44Super.TextModel method), 264	<code>get_Lag()</code> (PMML44Super.DerivedField method), 83
<code>get_isScorable()</code> (PMML44Super.TimeSeriesModel method), 274	<code>get_Lag()</code> (PMML44Super.Lower method), 135
<code>get_isScorable()</code> (PMML44Super.TreeModel method), 281	<code>get_Lag()</code> (PMML44Super.Mean method), 141
<code>get_isTransformed()</code> (PMML44Super.TrainingInstances method), 276	<code>get_Lag()</code> (PMML44Super.OutputField method), 181
<code>get_Item()</code> (PMML44Super.AssociationModel method), 24	<code>get_Lag()</code> (PMML44Super.TextIndex method), 261
<code>get_Item()</code> (PMML44Super.SequenceModel method), 233	<code>get_Lag()</code> (PMML44Super.Upper method), 288
<code>get_itemRef()</code> (PMML44Super.ItemRef method), 117	<code>get_Lag()</code> (PMML44Super.Variance method), 291
<code>get_ItemRef()</code> (PMML44Super.Itemset method), 118	<code>get_Lambda()</code> (PMML44Super.AbsoluteExponentialKernel method), 7
<code>get_Itemset()</code> (PMML44Super.AssociationModel method), 24	<code>get_Lambda()</code> (PMML44Super.ARDSquaredExponentialKernel method), 4
<code>get_Itemset()</code> (PMML44Super.SequenceModel method), 233	<code>get_Lambda()</code> (PMML44Super.GeneralizedExponentialKernel method), 109
<code>get_j()</code> (PMML44Super.Theta method), 267	<code>get_lambda()</code> (PMML44Super.RadialBasisKernel method), 209
<code>get_jaccard()</code> (PMML44Super.ComparisonMeasure method), 53	<code>get_LayerBias()</code> (PMML44Super.NetworkLayer method), 159
<code>get_KalmanState()</code> (PMML44Super.MaximumLikelihoodStat method), 140	<code>get_layerId()</code> (PMML44Super.NetworkLayer method), 159
<code>get_kernel()</code> (PMML44Super.LayerParameters method), 126	<code>get_LayerParameters()</code> (PMML44Super.NetworkLayer method), 159
	<code>get_layerType()</code> (PMML44Super.NetworkLayer method), 159
	<code>get_LayerWeights()</code> (PMML44Super.NetworkLayer method), 159

[get_learningRate\(\)](#) (*PMML44Super.Adadelta method*), 8
[get_learningRate\(\)](#) (*PMML44Super.Adagrad method*), 9
[get_learningRate\(\)](#) (*PMML44Super.Adam method*), 10
[get_learningRate\(\)](#) (*PMML44Super.Adamax method*), 11
[get_learningRate\(\)](#) (*PMML44Super.Nadam method*), 155
[get_learningRate\(\)](#) (*PMML44Super.RMSprop method*), 206
[get_learningRate\(\)](#) (*PMML44Super.SGD method*), 222
[get_leftMargin\(\)](#) (*PMML44Super.Interval method*), 116
[get_length\(\)](#) (*PMML44Super.TextDocument method*), 260
[get_length\(\)](#) (*PMML44Super.TimeCycle method*), 270
[get_lengthLimit\(\)](#) (*PMML44Super.AssociationModel method*), 24
[get_Level\(\)](#) (*PMML44Super.ExponentialSmoothing method*), 92
[get_leverage\(\)](#) (*PMML44Super.AssociationRule method*), 26
[get_lift\(\)](#) (*PMML44Super.AssociationRule method*), 26
[get_lift\(\)](#) (*PMML44Super.SequenceRule method*), 236
[get_LiftData\(\)](#) (*PMML44Super.PredictiveModelQuality method*), 200
[get_LiftGraph\(\)](#) (*PMML44Super.ModelLiftGraph method*), 150
[get_LiftGraph\(\)](#) (*PMML44Super.OptimumLiftGraph method*), 178
[get_LiftGraph\(\)](#) (*PMML44Super.RandomLiftGraph method*), 210
[get_LinearKernelType\(\)](#) (*PMML44Super.SupportVectorMachineModel method*), 247
[get_LinearNorm\(\)](#) (*PMML44Super.NormContinuous method*), 171
[get_linkFunction\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 107
[get_linkParameter\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 107
[get_localTermWeights\(\)](#) (*PMML44Super.TextIndex method*), 261
[get_localTermWeights\(\)](#) (*PMML44Super.TextModelNormalization method*), 266
[get_LocalTransformations\(\)](#) (*PMML44Super.AnomalyDetectionModel method*), 14
[get_LocalTransformations\(\)](#) (*PMML44Super.AssociationModel method*), 24
[get_LocalTransformations\(\)](#) (*PMML44Super.BaselineModel method*), 31
[get_LocalTransformations\(\)](#) (*PMML44Super.BayesianNetworkModel method*), 35
[get_LocalTransformations\(\)](#) (*PMML44Super.ClusteringModel method*), 49
[get_LocalTransformations\(\)](#) (*PMML44Super.DecisionTree method*), 75
[get_LocalTransformations\(\)](#) (*PMML44Super.DeepNetwork method*), 78
[get_LocalTransformations\(\)](#) (*PMML44Super.GaussianProcessModel method*), 104
[get_LocalTransformations\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 106
[get_LocalTransformations\(\)](#) (*PMML44Super.MiningModel method*), 146
[get_LocalTransformations\(\)](#) (*PMML44Super.NaiveBayesModel method*), 156
[get_LocalTransformations\(\)](#) (*PMML44Super.NearestNeighborModel method*), 157
[get_LocalTransformations\(\)](#) (*PMML44Super.NeuralNetwork method*), 163
[get_LocalTransformations\(\)](#) (*PMML44Super.Regression method*), 211
[get_LocalTransformations\(\)](#) (*PMML44Super.RegressionModel method*), 213
[get_LocalTransformations\(\)](#) (*PMML44Super.RuleSetModel method*), 221
[get_LocalTransformations\(\)](#) (*PMML44Super.Scorecard method*), 224
[get_LocalTransformations\(\)](#) (*PMML44Super.SequenceModel method*), 233
[get_LocalTransformations\(\)](#) (*PMML44Super.SupportVectorMachineModel method*), 247
[get_LocalTransformations\(\)](#) (*PMML44Super.TextModel method*), 264

`get_LocalTransformations()` (PMML44Super.TimeSeriesModel method), 273
`get_LocalTransformations()` (PMML44Super.TreeModel method), 280
`get_LognormalDistributionForBN()` (PMML44Super.ContinuousDistribution method), 65
`get_loss()` (PMML44Super.Losses method), 134
`get_Losses()` (PMML44Super.TrainingParameters method), 277
`get_Lower()` (PMML44Super.TriangularDistributionForBN method), 284
`get_lower()` (PMML44Super.UniformDistribution method), 285
`get_Lower()` (PMML44Super.UniformDistributionForBN method), 286
`get_lowValue()` (PMML44Super.MiningField method), 145
`get_MA()` (PMML44Super.ARMAPart method), 6
`get_MA()` (PMML44Super.NonseasonalComponent method), 169
`get_MA()` (PMML44Super.SeasonalComponent method), 225
`get_MACoefficients()` (PMML44Super.MA method), 136
`get_MACoefficients()` (PMML44Super.ResidualSquareCoefficients method), 216
`get_MACoefficients()` (PMML44Super.VarianceCoefficients method), 292
`get_magnitude()` (PMML44Super.OutlierEffect method), 179
`get_mapMissingTo()` (PMML44Super.Apply method), 21
`get_mapMissingTo()` (PMML44Super.Discretize method), 88
`get_mapMissingTo()` (PMML44Super.FieldRef method), 96
`get_mapMissingTo()` (PMML44Super.MapValues method), 138
`get_mapMissingTo()` (PMML44Super.NormContinuous method), 171
`get_mapMissingTo()` (PMML44Super.NormDiscrete method), 172
`get_mappedValue()` (PMML44Super.Item method), 116
`get_MapValues()` (PMML44Super.Apply method), 21
`get_MapValues()` (PMML44Super.ComplexPartialScore method), 55
`get_MapValues()` (PMML44Super.DefineFunction method), 80
`get_MapValues()` (PMML44Super.DerivedField method), 83
`get_MapValues()` (PMML44Super.Lower method), 135
`get_MapValues()` (PMML44Super.Mean method), 141
`get_MapValues()` (PMML44Super.OutputField method), 181
`get_MapValues()` (PMML44Super.TextIndex method), 261
`get_MapValues()` (PMML44Super.Upper method), 288
`get_MapValues()` (PMML44Super.Variance method), 291
`get_maskValue()` (PMML44Super.LayerParameters method), 126
`get_MatCell()` (PMML44Super.Matrix method), 139
`get_Matrix()` (PMML44Super.Comparisons method), 54
`get_Matrix()` (PMML44Super.ConfusionMatrix method), 60
`get_Matrix()` (PMML44Super.CorrelationMethods method), 68
`get_Matrix()` (PMML44Super.CorrelationValues method), 69
`get_Matrix()` (PMML44Super.Covariances method), 71
`get_Matrix()` (PMML44Super.DocumentTermMatrix method), 89
`get_Matrix()` (PMML44Super.FinalOmega method), 99
`get_Matrix()` (PMML44Super.MeasurementMatrix method), 142
`get_Matrix()` (PMML44Super.Predictor method), 202
`get_Matrix()` (PMML44Super.TransitionMatrix method), 280
`get_max()` (PMML44Super.Target method), 251
`get_max()` (PMML44Super.Time method), 269
`get_maximum()` (PMML44Super.ComparisonMeasure method), 53
`get_maximum()` (PMML44Super.NumericInfo method), 174
`get_maximumAntConsSeparationTime()` (PMML44Super.Constraints method), 62
`get_maximumItemsetSeparationTime()` (PMML44Super.Constraints method), 62
`get_MaximumLikelihoodStat()` (PMML44Super.ARIMA method), 5
`get_maximumNumberOfAntecedentItems()` (PMML44Super.Constraints method), 62
`get_maximumNumberOfConsequentItems()`

(PMML44Super.Constraints method), 62
 get_maximumNumberOfItems() (PMML44Super.Constraints method), 62
 get_maximumOrder() (PMML44Super.NonseasonalFactor method), 170
 get_maximumOrder() (PMML44Super.SeasonalFactor method), 226
 get_maximumTotalSequenceTime() (PMML44Super.Constraints method), 62
 get_maxLevenshteinDistance() (PMML44Super.TextIndex method), 261
 get_maxLevenshteinDistance() (PMML44Super.TextIndexNormalization method), 263
 get_maxNumberOfItemsPerTA() (PMML44Super.AssociationModel method), 24
 get_maxNumberOfItemsPerTransaction() (PMML44Super.SequenceModel method), 234
 get_maxNumberOfTAsPerTAGroup() (PMML44Super.SequenceModel method), 234
 get_maxTime() (PMML44Super.BaseCumHazardTables method), 28
 get_maxTime() (PMML44Super.BaselineStratum method), 32
 get_maxWins() (PMML44Super.SupportVectorMachineModel method), 53
 get_mean() (PMML44Super.AnyDistribution method), 19
 get_mean() (PMML44Super.GaussianDistribution method), 103
 get_Mean() (PMML44Super.LognormalDistributionForBN method), 134
 get_Mean() (PMML44Super.NormalDistributionForBN method), 173
 get_mean() (PMML44Super.NumericInfo method), 174
 get_mean() (PMML44Super.PoissonDistribution method), 198
 get_mean() (PMML44Super.Time method), 269
 get_Mean() (PMML44Super.TriangularDistributionForBN method), 284
 get_meanAbsoluteError() (PMML44Super.PredictiveModelQuality method), 200
 get_meanError() (PMML44Super.PredictiveModelQuality method), 200
 get_meanOfSquares() (PMML44Super.AnovaRow method), 17
 get_meanSquaredError() (PMML44Super.PredictiveModelQuality method), 200
 get_MeasurementMatrix() (PMML44Super.StateSpaceModel method), 243
 get_median() (PMML44Super.NumericInfo method), 174
 get_mergeLayerConcatOperationAxes() (PMML44Super.LayerParameters method), 126
 get_mergeLayerDotNormalize() (PMML44Super.LayerParameters method), 126
 get_mergeLayerDotOperationAxis() (PMML44Super.LayerParameters method), 126
 get_mergeLayerOp() (PMML44Super.LayerParameters method), 126
 get_method() (PMML44Super.MaximumLikelihoodStat method), 141
 get_metric() (PMML44Super.Metrics method), 143
 get_Metrics() (PMML44Super.TrainingParameters method), 277
 get_mimeType() (PMML44Super.DataField method), 73
 get_min() (PMML44Super.Target method), 251
 get_min() (PMML44Super.Time method), 269
 get_minimum() (PMML44Super.ComparisonMeasure method), 53
 get_minimum() (PMML44Super.NumericInfo method), 175
 get_minimumAntConsSeparationTime() (PMML44Super.Constraints method), 62
 get_minimumConfidence() (PMML44Super.AssociationModel method), 24
 get_minimumConfidence() (PMML44Super.Constraints method), 62
 get_minimumItemsetSeparationTime() (PMML44Super.Constraints method), 62
 get_minimumLift() (PMML44Super.Constraints method), 62
 get_minimumNumberOfAntecedentItems() (PMML44Super.Constraints method), 62
 get_minimumNumberOfConsequentItems() (PMML44Super.Constraints method), 62
 get_minimumNumberOfItems() (PMML44Super.Constraints method), 62
 get_minimumSupport() (PMML44Super.AssociationModel method), 24
 get_minimumSupport() (PMML44Super.Constraints method), 62
 get_minimumTotalSequenceTime() (PMML44Super.Constraints method), 62
 get_MiningBuildTask() (PMML44Super.PMML method), 186

[get_MiningField\(\)](#)
 (PMML44Super.MiningSchema method), [148](#)

[get_MiningModel\(\)](#)
 (PMML44Super.AnomalyDetectionModel method), [14](#)

[get_MiningModel\(\)](#) (PMML44Super.PMML method), [186](#)

[get_MiningModel\(\)](#) (PMML44Super.Segment method), [229](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.AnomalyDetectionModel method), [14](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.AssociationModel method), [24](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.BaselineModel method), [31](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.BayesianNetworkModel method), [35](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.ClusteringModel method), [49](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.DeepNetwork method), [78](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.GaussianProcessModel method), [104](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.GeneralRegressionModel method), [106](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.MiningModel method), [146](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.NaiveBayesModel method), [156](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.NearestNeighborModel method), [157](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.NeuralNetwork method), [163](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.RegressionModel method), [213](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.RuleSetModel method), [221](#)

[get_MiningSchema\(\)](#) (PMML44Super.Scorecard method), [224](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.SequenceModel method), [233](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.SupportVectorMachineModel method), [247](#)

[get_MiningSchema\(\)](#) (PMML44Super.TextModel method), [264](#)

[get_MiningSchema\(\)](#)
 (PMML44Super.TimeSeriesModel method), [273](#)

[get_MiningSchema\(\)](#) (PMML44Super.TreeModel method), [280](#)

[get_minkowski\(\)](#) (PMML44Super.ComparisonMeasure method), [53](#)

[get_missingFreq\(\)](#) (PMML44Super.Counts method), [70](#)

[get_missingThreshold\(\)](#)
 (PMML44Super.Segmentation method), [231](#)

[get_missingValuePenalty\(\)](#)
 (PMML44Super.DecisionTree method), [76](#)

[get_missingValuePenalty\(\)](#)
 (PMML44Super.TreeModel method), [281](#)

[get_missingValueReplacement\(\)](#)
 (PMML44Super.MiningField method), [145](#)

[get_missingValueStrategy\(\)](#)
 (PMML44Super.DecisionTree method), [76](#)

[get_missingValueStrategy\(\)](#)
 (PMML44Super.TreeModel method), [281](#)

[get_missingValueTreatment\(\)](#)
 (PMML44Super.MiningField method), [145](#)

[get_MissingValueWeights\(\)](#)
 (PMML44Super.ClusteringModel method), [49](#)

[get_modalValue\(\)](#) (PMML44Super.DiscrStats method), [85](#)

[get_modelClass\(\)](#) (PMML44Super.ClusteringModel method), [49](#)

[get_modelDF\(\)](#) (PMML44Super.GeneralRegressionModel method), [107](#)

[get_ModelExplanation\(\)](#)
 (PMML44Super.BaselineModel method), [31](#)

[get_ModelExplanation\(\)](#)
 (PMML44Super.BayesianNetworkModel method), [36](#)

[get_ModelExplanation\(\)](#)
 (PMML44Super.ClusteringModel method), [49](#)

[get_ModelExplanation\(\)](#)
 (PMML44Super.DeepNetwork method), [78](#)

[get_ModelExplanation\(\)](#)
 (PMML44Super.GaussianProcessModel method), [104](#)

[get_ModelExplanation\(\)](#)

(PMML44Super.GeneralRegressionModel
 method), 106
 get_ModelExplanation() (PMML44Super.MiningModel method), 146
 get_ModelExplanation() (PMML44Super.NaiveBayesModel method), 156
 get_ModelExplanation() (PMML44Super.NearestNeighborModel method), 158
 get_ModelExplanation() (PMML44Super.NeuralNetwork method), 163
 get_ModelExplanation() (PMML44Super.RegressionModel method), 213
 get_ModelExplanation() (PMML44Super.RuleSetModel method), 221
 get_ModelExplanation() (PMML44Super.Scorecard method), 224
 get_ModelExplanation() (PMML44Super.SupportVectorMachineModel method), 247
 get_ModelExplanation() (PMML44Super.TextModel method), 264
 get_ModelExplanation() (PMML44Super.TimeSeriesModel method), 273
 get_ModelExplanation() (PMML44Super.TreeModel method), 281
 get_ModelLiftGraph() (PMML44Super.LiftData method), 130
 get_modelName() (PMML44Super.AnomalyDetectionModel method), 15
 get_modelName() (PMML44Super.AssociationModel method), 24
 get_modelName() (PMML44Super.BaselineModel method), 31
 get_modelName() (PMML44Super.BayesianNetworkModel method), 36
 get_modelName() (PMML44Super.ClusteringModel method), 49
 get_modelName() (PMML44Super.DecisionTree method), 76
 get_modelName() (PMML44Super.DeepNetwork method), 78
 get_modelName() (PMML44Super.GaussianProcessModel method), 104
 get_modelName() (PMML44Super.GeneralRegressionModel method), 107
 get_modelName() (PMML44Super.MiningModel method), 146
 get_modelName() (PMML44Super.NaiveBayesModel method), 156
 get_modelName() (PMML44Super.NearestNeighborModel method), 158
 get_modelName() (PMML44Super.NeuralNetwork method), 164
 get_modelName() (PMML44Super.Regression method), 211
 get_modelName() (PMML44Super.RegressionModel method), 213
 get_modelName() (PMML44Super.RuleSetModel method), 221
 get_modelName() (PMML44Super.Scorecard method), 224
 get_modelName() (PMML44Super.SequenceModel method), 234
 get_modelName() (PMML44Super.SupportVectorMachineModel method), 248
 get_modelName() (PMML44Super.TextModel method), 265
 get_modelName() (PMML44Super.TimeSeriesModel method), 274
 get_modelName() (PMML44Super.TreeModel method), 281
 get_ModelStats() (PMML44Super.AssociationModel method), 24
 get_ModelStats() (PMML44Super.BaselineModel method), 31
 get_ModelStats() (PMML44Super.BayesianNetworkModel method), 36
 get_ModelStats() (PMML44Super.ClusteringModel method), 49
 get_ModelStats() (PMML44Super.DecisionTree method), 75
 get_ModelStats() (PMML44Super.DeepNetwork method), 78
 get_ModelStats() (PMML44Super.GaussianProcessModel method), 104
 get_ModelStats() (PMML44Super.GeneralRegressionModel method), 106
 get_ModelStats() (PMML44Super.MiningModel method), 146
 get_ModelStats() (PMML44Super.NaiveBayesModel method), 156
 get_ModelStats() (PMML44Super.NearestNeighborModel method), 158
 get_ModelStats() (PMML44Super.NeuralNetwork method), 163
 get_ModelStats() (PMML44Super.Regression method), 211
 get_ModelStats() (PMML44Super.RegressionModel method), 213
 get_ModelStats() (PMML44Super.RuleSetModel method), 221
 get_ModelStats() (PMML44Super.Scorecard method), 224

`method)`, 224
`get_ModelStats()` (*PMML44Super.SequenceModel* `method)`, 233
`get_ModelStats()` (*PMML44Super.SupportVectorMachineModel* `method)`, 247
`get_ModelStats()` (*PMML44Super.TextModel* `method)`, 264
`get_ModelStats()` (*PMML44Super.TimeSeriesModel* `method)`, 273
`get_ModelStats()` (*PMML44Super.TreeModel* `method)`, 281
`get_modelType()` (*PMML44Super.BayesianNetworkModel* `method)`, 36
`get_modelType()` (*PMML44Super.GeneralRegressionModel* `method)`, 107
`get_modelType()` (*PMML44Super.RegressionModel* `method)`, 213
`get_ModelVerification()` (*PMML44Super.AnomalyDetectionModel* `method)`, 15
`get_ModelVerification()` (*PMML44Super.AssociationModel* `method)`, 24
`get_ModelVerification()` (*PMML44Super.BaselineModel* `method)`, 31
`get_ModelVerification()` (*PMML44Super.BayesianNetworkModel* `method)`, 36
`get_ModelVerification()` (*PMML44Super.ClusteringModel* `method)`, 49
`get_ModelVerification()` (*PMML44Super.DeepNetwork* `method)`, 78
`get_ModelVerification()` (*PMML44Super.GaussianProcessModel* `method)`, 104
`get_ModelVerification()` (*PMML44Super.GeneralRegressionModel* `method)`, 106
`get_ModelVerification()` (*PMML44Super.MiningModel* `method)`, 146
`get_ModelVerification()` (*PMML44Super.NaiveBayesModel* `method)`, 156
`get_ModelVerification()` (*PMML44Super.NearestNeighborModel* `method)`, 158
`get_ModelVerification()` (*PMML44Super.NeuralNetwork* `method)`, 163
`get_ModelVerification()` (*PMML44Super.RegressionModel* `method)`, 213
`get_ModelVerification()` (*PMML44Super.RuleSetModel* `method)`, 221
`get_ModelVerification()` (*PMML44Super.Scorecard* `method)`, 224
`get_ModelVerification()` (*PMML44Super.SupportVectorMachineModel* `method)`, 247
`get_ModelVerification()` (*PMML44Super.TextModel* `method)`, 264
`get_ModelVerification()` (*PMML44Super.TimeSeriesModel* `method)`, 273
`get_ModelVerification()` (*PMML44Super.TreeModel* `method)`, 281
`get_modelVersion()` (*PMML44Super.Header* `method)`, 111
`get_momentum()` (*PMML44Super.SGD* `method)`, 222
`get_movingMeanInitializer()` (*PMML44Super.LayerParameters* `method)`, 126
`get_movingVarianceInitializer()` (*PMML44Super.LayerParameters* `method)`, 126
`get_multipleModelMethod()` (*PMML44Super.Segmentation* `method)`, 231
`get_MultivariateStat()` (*PMML44Super.MultivariateStats* `method)`, 154
`get_MultivariateStats()` (*PMML44Super.ModelStats* `method)`, 151
`get_n()` (*PMML44Super.ArrayType* `method)`, 23
`get_n()` (*PMML44Super.INT_SparseArray* `method)`, 112
`get_n()` (*PMML44Super.Lag* `method)`, 122
`get_n()` (*PMML44Super.REAL_SparseArray* `method)`, 205
`get_Nadam()` (*PMML44Super.Optimizers* `method)`, 177
`get_NaiveBayesModel()` (*PMML44Super.AnomalyDetectionModel* `method)`, 15
`get_NaiveBayesModel()` (*PMML44Super.PMML* `method)`, 186
`get_NaiveBayesModel()` (*PMML44Super.Segment* `method)`, 229
`get_name()` (*PMML44Super.Application* `method)`, 20
`get_name()` (*PMML44Super.CategoricalPredictor* `method)`, 41
`get_name()` (*PMML44Super.Characteristic* `method)`, 43
`get_name()` (*PMML44Super.Cluster* `method)`, 47
`get_name()` (*PMML44Super.ContinuousNode* `method)`, 66

`get_name()` (*PMML44Super.DataField* method), 73
`get_name()` (*PMML44Super.DefineFunction* method), 80
`get_name()` (*PMML44Super.DerivedField* method), 83
`get_name()` (*PMML44Super.DiscreteNode* method), 87
`get_name()` (*PMML44Super.Extension* method), 93
`get_name()` (*PMML44Super.MiningField* method), 145
`get_name()` (*PMML44Super.MultivariateStat* method), 153
`get_name()` (*PMML44Super.NumericPredictor* method), 176
`get_name()` (*PMML44Super.OutputField* method), 181
`get_name()` (*PMML44Super.Parameter* method), 192
`get_name()` (*PMML44Super.ParameterField* method), 193
`get_name()` (*PMML44Super.Partition* method), 195
`get_name()` (*PMML44Super.Predictor* method), 202
`get_name()` (*PMML44Super.PredictorTerm* method), 203
`get_name()` (*PMML44Super.ResultField* method), 218
`get_name()` (*PMML44Super.Taxonomy* method), 257
`get_name()` (*PMML44Super.TextDocument* method), 260
`get_nbCols()` (*PMML44Super.Matrix* method), 140
`get_nbCorrect()` (*PMML44Super.RuleSet* method), 219
`get_nbCorrect()` (*PMML44Super.SimpleRule* method), 241
`get_nbRows()` (*PMML44Super.Matrix* method), 140
`get_NearestNeighborModel()` (*PMML44Super.AnomalyDetectionModel* method), 15
`get_NearestNeighborModel()` (*PMML44Super.PMML* method), 187
`get_NearestNeighborModel()` (*PMML44Super.Segment* method), 229
`get_negativeTargetFieldDisplayValue()` (*PMML44Super.ROC* method), 207
`get_negativeTargetFieldValue()` (*PMML44Super.ROC* method), 207
`get_nesterov()` (*PMML44Super.SGD* method), 222
`get_NetworkLayer()` (*PMML44Super.DeepNetwork* method), 78
`get_NeuralInput()` (*PMML44Super.NeuralInputs* method), 161
`get_NeuralInputs()` (*PMML44Super.NeuralNetwork* method), 163
`get_NeuralLayer()` (*PMML44Super.NeuralNetwork* method), 163
`get_NeuralNetwork()` (*PMML44Super.AnomalyDetectionModel* method), 15
`get_NeuralNetwork()` (*PMML44Super.PMML* method), 187
`get_NeuralNetwork()` (*PMML44Super.Segment* method), 229
`get_NeuralOutput()` (*PMML44Super.NeuralOutputs* method), 166
`get_NeuralOutputs()` (*PMML44Super.DeepNetwork* method), 78
`get_NeuralOutputs()` (*PMML44Super.NeuralNetwork* method), 163
`get_Neuron()` (*PMML44Super.NeuralLayer* method), 162
`get_Node()` (*PMML44Super.DecisionTree* method), 76
`get_Node()` (*PMML44Super.Node* method), 168
`get_Node()` (*PMML44Super.TreeModel* method), 281
`get_noiseVariance()` (*PMML44Super.AbsoluteExponentialKernel* method), 7
`get_noiseVariance()` (*PMML44Super.ARDSquaredExponentialKernel* method), 4
`get_noiseVariance()` (*PMML44Super.GeneralizedExponentialKernel* method), 109
`get_noiseVariance()` (*PMML44Super.RadialBasisKernel* method), 209
`get_NonseasonalComponent()` (*PMML44Super.ARIMA* method), 5
`get_NonseasonalFactor()` (*PMML44Super.Denominator* method), 82
`get_NonseasonalFactor()` (*PMML44Super.Numerator* method), 174
`get_norm()` (*PMML44Super.LinearNorm* method), 132
`get_NormalDistributionForBN()` (*PMML44Super.ContinuousDistribution* method), 65
`get_normalizationMethod()` (*PMML44Super.DeepNetwork* method), 78
`get_normalizationMethod()` (*PMML44Super.NetworkLayer* method), 159
`get_normalizationMethod()`

(*PMML44Super.NeuralLayer* method), 162
`get_normalizationMethod()`
(*PMML44Super.NeuralNetwork* method), 164
`get_normalizationMethod()`
(*PMML44Super.Regression* method), 211
`get_normalizationMethod()`
(*PMML44Super.RegressionModel* method), 213
`get_normalizationScheme()`
(*PMML44Super.TestDistributions* method), 258
`get_NormalizedCountTable()`
(*PMML44Super.Baseline* method), 29
`get_NormContinuous()` (*PMML44Super.Apply* method), 21
`get_NormContinuous()`
(*PMML44Super.ComplexPartialScore* method), 55
`get_NormContinuous()`
(*PMML44Super.DefineFunction* method), 80
`get_NormContinuous()`
(*PMML44Super.DerivedField* method), 83
`get_NormContinuous()` (*PMML44Super.Lower* method), 135
`get_NormContinuous()` (*PMML44Super.Mean* method), 141
`get_NormContinuous()`
(*PMML44Super.OutputField* method), 181
`get_NormContinuous()` (*PMML44Super.TextIndex* method), 261
`get_NormContinuous()` (*PMML44Super.Upper* method), 288
`get_NormContinuous()` (*PMML44Super.Variance* method), 291
`get_NormDiscrete()` (*PMML44Super.Apply* method), 21
`get_NormDiscrete()`
(*PMML44Super.ComplexPartialScore* method), 55
`get_NormDiscrete()`
(*PMML44Super.DefineFunction* method), 80
`get_NormDiscrete()` (*PMML44Super.DerivedField* method), 83
`get_NormDiscrete()` (*PMML44Super.Lower* method), 135
`get_NormDiscrete()` (*PMML44Super.Mean* method), 141
`get_NormDiscrete()` (*PMML44Super.OutputField* method), 181
`get_NormDiscrete()` (*PMML44Super.TextIndex* method), 261
`get_NormDiscrete()` (*PMML44Super.Upper* method), 288
`get_NormDiscrete()` (*PMML44Super.Variance* method), 291
`get_noTrueChildStrategy()`
(*PMML44Super.DecisionTree* method), 76
`get_noTrueChildStrategy()`
(*PMML44Super.TreeModel* method), 281
`get_NUM_ARRAY()` (*PMML44Super.ContStats* method), 63
`get_numberOfAttributes()`
(*PMML44Super.SupportVectors* method), 249
`get_numberOfClusters()`
(*PMML44Super.ClusteringModel* method), 49
`get_numberOfCoefficients()`
(*PMML44Super.Coefficients* method), 52
`get_numberOfDocuments()`
(*PMML44Super.TextModel* method), 265
`get_numberOfFields()`
(*PMML44Super.DataDictionary* method), 72
`get_numberOfFields()`
(*PMML44Super.VectorFields* method), 294
`get_numberOfInputs()`
(*PMML44Super.NeuralInputs* method), 161
`get_numberOfItems()`
(*PMML44Super.AssociationModel* method), 24
`get_numberOfItems()` (*PMML44Super.Itemset* method), 118
`get_numberOfItemsets()`
(*PMML44Super.AssociationModel* method), 24
`get_numberOfLayers()`
(*PMML44Super.DeepNetwork* method), 78
`get_numberOfLayers()`
(*PMML44Super.NeuralNetwork* method), 164
`get_numberOfNeighbors()`
(*PMML44Super.NearestNeighborModel* method), 158
`get_numberOfNeurons()`
(*PMML44Super.NeuralLayer* method), 162
`get_numberOfOutputs()`
(*PMML44Super.NeuralOutputs* method), 166
`get_numberOfRules()`
(*PMML44Super.AssociationModel* method), 24
`get_numberOfSets()` (*PMML44Super.Sequence* method), 232
`get_numberOfSets()`
(*PMML44Super.SequenceRule* method), 236

`get_numberOfSupportVectors()` (PMML44Super.SupportVectors method), 249
`get_numberOfTerms()` (PMML44Super.TextModel method), 265
`get_numberOfTransactionGroups()` (PMML44Super.SequenceModel method), 234
`get_numberOfTransactions()` (PMML44Super.AssociationModel method), 24
`get_numberOfTransactions()` (PMML44Super.SequenceModel method), 234
`get_numberOfVectors()` (PMML44Super.VectorDictionary method), 293
`get_Numerator()` (PMML44Super.DynamicRegressor method), 90
`get_NumericInfo()` (PMML44Super.PartitionFieldStats method), 196
`get_NumericInfo()` (PMML44Super.UnivariateStats method), 287
`get_NumericPredictor()` (PMML44Super.RegressionTable method), 214
`get_numOfPredictors()` (PMML44Super.PredictiveModelQuality method), 200
`get_numOfRecords()` (PMML44Super.PredictiveModelQuality method), 200
`get_numOfRecordsWeighted()` (PMML44Super.PredictiveModelQuality method), 200
`get_numTopCategories()` (PMML44Super.OutputField method), 181
`get_occurrence()` (PMML44Super.Sequence method), 232
`get_occurrence()` (PMML44Super.SequenceRule method), 236
`get_offDiagDefault()` (PMML44Super.Matrix method), 140
`get_offset()` (PMML44Super.TimeAnchor method), 270
`get_offsetValue()` (PMML44Super.GeneralRegressionModel method), 107
`get_offsetVariable()` (PMML44Super.GeneralRegressionModel method), 107
`get_operator()` (PMML44Super.SetPredicate method), 237
`get_operator()` (PMML44Super.SimplePredicate method), 240
`get_optimizer()` (PMML44Super.GaussianProcessModel method), 105
`get_Optimizers()` (PMML44Super.TrainingParameters method), 277
`get_OptimumLiftGraph()` (PMML44Super.LiftData method), 130
`get_optype()` (PMML44Super.DataField method), 73
`get_optype()` (PMML44Super.DefineFunction method), 80
`get_optype()` (PMML44Super.DerivedField method), 83
`get_optype()` (PMML44Super.MiningField method), 145
`get_optype()` (PMML44Super.OutputField method), 181
`get_optype()` (PMML44Super.ParameterField method), 193
`get_optype()` (PMML44Super.ResultField method), 218
`get_optype()` (PMML44Super.Target method), 251
`get_orig()` (PMML44Super.LinearNorm method), 132
`get_outField()` (PMML44Super.TextIndexNormalization method), 263
`get_OutlierEffect()` (PMML44Super.ARIMA method), 5
`get_outliers()` (PMML44Super.MiningField method), 145
`get_outliers()` (PMML44Super.NormContinuous method), 171
`get_Output()` (PMML44Super.AnomalyDetectionModel method), 15
`get_Output()` (PMML44Super.AssociationModel method), 24
`get_Output()` (PMML44Super.BaselineModel method), 31
`get_Output()` (PMML44Super.BayesianNetworkModel method), 36
`get_Output()` (PMML44Super.ClusteringModel method), 49
`get_Output()` (PMML44Super.DecisionTree method), 76
`get_Output()` (PMML44Super.DeepNetwork method), 78
`get_Output()` (PMML44Super.GaussianProcessModel method), 104
`get_Output()` (PMML44Super.GeneralRegressionModel method), 107
`get_Output()` (PMML44Super.MiningModel method), 146
`get_Output()` (PMML44Super.NaiveBayesModel method), 146

`method`), 156
`get_Output()` (*PMML44Super.NearestNeighborModel*
`method`), 158
`get_Output()` (*PMML44Super.NeuralNetwork*
`method`), 163
`get_Output()` (*PMML44Super.Regression* `method`),
211
`get_Output()` (*PMML44Super.RegressionModel*
`method`), 213
`get_Output()` (*PMML44Super.RuleSetModel*
`method`), 221
`get_Output()` (*PMML44Super.Scorecard* `method`),
224
`get_Output()` (*PMML44Super.SupportVectorMachineModel*
`method`), 247
`get_Output()` (*PMML44Super.TextModel* `method`),
264
`get_Output()` (*PMML44Super.TimeSeriesModel*
`method`), 273
`get_Output()` (*PMML44Super.TreeModel* `method`),
281
`get_outputColumn()` (*PMML44Super.MapValues*
`method`), 138
`get_outputDimension()`
(*PMML44Super.LayerParameters* `method`),
126
`get_OutputField()` (*PMML44Super.Output*
`method`), 180
`get_outputNeuron()`
(*PMML44Super.NeuralOutput* `method`),
165
`get_p()` (*PMML44Super.ARMAPart* `method`), 6
`get_p()` (*PMML44Super.NonseasonalComponent*
`method`), 169
`get_P()` (*PMML44Super.SeasonalComponent*
`method`), 226
`get_p_parameter()` (*PMML44Super.minkowski*
`method`), 302
`get_pad()` (*PMML44Super.LayerParameters* `method`),
126
`get_paddingDims()`
(*PMML44Super.LayerParameters* `method`),
126
`get_PairCounts()` (*PMML44Super.BayesInput*
`method`), 33
`get_Parameter()` (*PMML44Super.ParameterList*
`method`), 194
`get_ParameterField()`
(*PMML44Super.DefineFunction* `method`),
80
`get_ParameterList()`
(*PMML44Super.AnomalyDetectionModel*
`method`), 15
`get_ParameterList()`
(*PMML44Super.GeneralRegressionModel*
`method`), 107
`get_parameterName()` (*PMML44Super.PCell*
`method`), 183
`get_parameterName()` (*PMML44Super.PPCell*
`method`), 189
`get_ParamMatrix()`
(*PMML44Super.GeneralRegressionModel*
`method`), 107
`get_parent()` (*PMML44Super.ParentValue* `method`),
195
`get_parentField()` (*PMML44Super.ChildParent*
`method`), 45
`get_parentLevelField()`
(*PMML44Super.ChildParent* `method`), 45
`get_ParentValue()`
(*PMML44Super.ContinuousConditionalProbability*
`method`), 64
`get_ParentValue()`
(*PMML44Super.DiscreteConditionalProbability*
`method`), 85
`get_partialScore()` (*PMML44Super.Attribute*
`method`), 27
`get_Partition()` (*PMML44Super.Cluster* `method`),
47
`get_Partition()` (*PMML44Super.Node* `method`),
168
`get_Partition()` (*PMML44Super.TargetValue*
`method`), 252
`get_PartitionFieldStats()`
(*PMML44Super.Partition* `method`), 195
`get_PastVariances()`
(*PMML44Super.VarianceCoefficients* `method`),
292
`get_PCell()` (*PMML44Super.ParamMatrix* `method`),
192
`get_pCol()` (*PMML44Super.PCovCell* `method`), 184
`get_PCovCell()` (*PMML44Super.PCovMatrix*
`method`), 185
`get_PCovMatrix()` (*PMML44Super.GeneralRegressionModel*
`method`), 107
`get_period()` (*PMML44Super.SeasonalComponent*
`method`), 226
`get_period()` (*PMML44Super.Seasonality_ExpoSmooth*
`method`), 228
`get_period()` (*PMML44Super.StateSpaceModel*
`method`), 243
`get_periodDeficit()`
(*PMML44Super.MaximumLikelihoodStat*
`method`), 141
`get_permuteDims()`
(*PMML44Super.LayerParameters* `method`),
126
`get_phase()` (*PMML44Super.Seasonality_ExpoSmooth*

`method`), 228
`get_phi()` (*PMML44Super.Trend_ExpoSmooth method*), 283
`get_pointwiseConstraint()` (*PMML44Super.LayerParameters method*), 126
`get_PoissonDistribution()` (*PMML44Super.Alternate method*), 13
`get_PoissonDistribution()` (*PMML44Super.Baseline method*), 29
`get_PoissonDistribution()` (*PMML44Super.TargetValueStat method*), 254
`get_PolynomialKernelType()` (*PMML44Super.SupportVectorMachineModel method*), 247
`get_poolSize()` (*PMML44Super.LayerParameters method*), 126
`get_positiveTargetFieldDisplayValue()` (*PMML44Super.ROC method*), 207
`get_positiveTargetFieldValue()` (*PMML44Super.ROC method*), 207
`get_PPCell()` (*PMML44Super.PPMatrix method*), 190
`get_PPMatrix()` (*PMML44Super.GeneralRegressionModel method*), 107
`get_precision()` (*PMML44Super.VerificationField method*), 295
`get_predictionMethod()` (*PMML44Super.ARIMA method*), 5
`get_PredictiveModelQuality()` (*PMML44Super.ModelExplanation method*), 149
`get_Predictor()` (*PMML44Super.CovariateList method*), 72
`get_Predictor()` (*PMML44Super.FactorList method*), 94
`get_predictorName()` (*PMML44Super.PPCell method*), 189
`get_PredictorTerm()` (*PMML44Super.RegressionTable method*), 214
`get_priorProbability()` (*PMML44Super.TargetValue method*), 252
`get_probability()` (*PMML44Super.ScoreDistribution method*), 223
`get_probability()` (*PMML44Super.ValueProbability method*), 289
`get_property()` (*PMML44Super.Value method*), 289
`get_pRow()` (*PMML44Super.PCovCell method*), 184
`get_PsiVector()` (*PMML44Super.StateSpaceModel method*), 243
`get_pValue()` (*PMML44Super.AnovaRow method*), 17
`get_pValueAlpha()` (*PMML44Super.MultivariateStat method*), 153
`get_pValueFinal()` (*PMML44Super.MultivariateStat method*), 153
`get_pValueInitial()` (*PMML44Super.MultivariateStat method*), 153
`get_q()` (*PMML44Super.ARMAPart method*), 6
`get_q()` (*PMML44Super.NonseasonalComponent method*), 169
`get_Q()` (*PMML44Super.SeasonalComponent method*), 226
`get_Quantile()` (*PMML44Super.NumericInfo method*), 174
`get_quantileLimit()` (*PMML44Super.Quantile method*), 204
`get_quantileValue()` (*PMML44Super.Quantile method*), 204
`get_r_squared()` (*PMML44Super.PredictiveModelQuality method*), 200
`get_RadialBasisKernel()` (*PMML44Super.GaussianProcessModel method*), 104
`get_RadialBasisKernelType()` (*PMML44Super.SupportVectorMachineModel method*), 247
`get_RandomLiftGraph()` (*PMML44Super.LiftData method*), 130
`get_rank()` (*PMML44Super.OutputField method*), 181
`get_rankBasis()` (*PMML44Super.OutputField method*), 181
`get_rankingQuality()` (*PMML44Super.LiftData method*), 130
`get_rankOrder()` (*PMML44Super.OutputField method*), 181
`get_REAL_Entries()` (*PMML44Super.REAL_SparseArray method*), 205
`get_REAL_SparseArray()` (*PMML44Super.TrendCoefficients method*), 282
`get_REAL_SparseArray()` (*PMML44Super.VectorInstance method*), 295
`get_reasonCode()` (*PMML44Super.Attribute method*), 27
`get_reasonCode()` (*PMML44Super.Characteristic method*), 43
`get_reasonCodeAlgorithm()`

(PMML44Super.Scorecard method), 224
 get_recordCount () (PMML44Super.ModelVerification method), 152
 get_recordCount () (PMML44Super.Node method), 168
 get_recordCount () (PMML44Super.RuleSet method), 220
 get_recordCount () (PMML44Super.ScoreDistribution method), 223
 get_recordCount () (PMML44Super.SimpleRule method), 241
 get_recordCount () (PMML44Super.TrainingInstances method), 276
 get_recursive () (PMML44Super.TextIndexNormalization method), 263
 get_referencePoint () (PMML44Super.Parameter method), 192
 get_regexField () (PMML44Super.TextIndexNormalization method), 263
 get_Regression () (PMML44Super.MiningModel method), 146
 get_Regression () (PMML44Super.Node method), 168
 get_RegressionModel () (PMML44Super.AnomalyDetectionModel method), 15
 get_RegressionModel () (PMML44Super.PMML method), 187
 get_RegressionModel () (PMML44Super.Segment method), 229
 get_RegressionTable () (PMML44Super.Regression method), 211
 get_RegressionTable () (PMML44Super.RegressionModel method), 213
 get_RegressorValues () (PMML44Super.DynamicRegressor method), 90
 get_repeatVectorTimes () (PMML44Super.LayerParameters method), 126
 get_rescaleConstant () (PMML44Super.Target method), 251
 get_rescaleFactor () (PMML44Super.Target method), 251
 get_resetValue () (PMML44Super.TestDistributions method), 258
 get_reshapeTarget () (PMML44Super.LayerParameters method), 126
 get_Residuals () (PMML44Super.MA method), 136
 get_Residuals () (PMML44Super.ResidualSquareCoefficients method), 216
 get_ResidualSquareCoefficients () (PMML44Super.GARCHPart method), 102
 get_ResultField () (PMML44Super.DecisionTree method), 76
 get_ResultField () (PMML44Super.Regression method), 211
 get_rho () (PMML44Super.Adadelta method), 8
 get_rho () (PMML44Super.RMSprop method), 206
 get_rightMargin () (PMML44Super.Interval method), 116
 get_RMSE () (PMML44Super.ARIMA method), 5
 get_RMSE () (PMML44Super.ExponentialSmoothing method), 92
 get_RMSprop () (PMML44Super.Optimizers method), 177
 get_ROC () (PMML44Super.PredictiveModelQuality method), 200
 get_ROCGraph () (PMML44Super.ROC method), 207
 get_rootMeanSquaredError () (PMML44Super.PredictiveModelQuality method), 200
 get_row () (PMML44Super.InlineTable method), 113
 get_row () (PMML44Super.MatCell method), 139
 get_ruleFeature () (PMML44Super.OutputField method), 181
 get_RuleSelectionMethod () (PMML44Super.RuleSet method), 219
 get_RuleSet () (PMML44Super.RuleSetModel method), 221
 get_RuleSetModel () (PMML44Super.AnomalyDetectionModel method), 15
 get_RuleSetModel () (PMML44Super.PMML method), 187
 get_RuleSetModel () (PMML44Super.Segment method), 229
 get_sample () (PMML44Super.COUNT_TABLE_TYPE method), 40
 get_schedule_decay () (PMML44Super.Nadam method), 155
 get_score () (PMML44Super.Node method), 168
 get_score () (PMML44Super.SimpleRule method), 241
 get_Scorecard () (PMML44Super.AnomalyDetectionModel method), 15
 get_Scorecard () (PMML44Super.PMML method), 187
 get_Scorecard () (PMML44Super.Segment method), 229
 get_ScoreDistribution () (PMML44Super.Node method), 168
 get_ScoreDistribution ()

(PMML44Super.RuleSet method), 219
 get_ScoreDistribution() (PMML44Super.SimpleRule method), 241
 get_SeasonalComponent() (PMML44Super.ARIMA method), 5
 get_SeasonalFactor() (PMML44Super.Denominator method), 82
 get_SeasonalFactor() (PMML44Super.Numerator method), 174
 get_Seasonality_ExpoSmooth() (PMML44Super.ExponentialSmoothing method), 92
 get_SeasonalTrendDecomposition() (PMML44Super.TimeSeriesModel method), 273
 get_Segment() (PMML44Super.Segmentation method), 231
 get_Segmentation() (PMML44Super.MiningModel method), 146
 get_segmentId() (PMML44Super.OutputField method), 181
 get_seqId() (PMML44Super.SequenceReference method), 235
 get_Sequence() (PMML44Super.SequenceModel method), 233
 get_SequenceModel() (PMML44Super.AnomalyDetectionModel method), 15
 get_SequenceModel() (PMML44Super.PMML method), 187
 get_SequenceModel() (PMML44Super.Segment method), 229
 get_SequenceReference() (PMML44Super.AntecedentSequence method), 18
 get_SequenceReference() (PMML44Super.ConsequentSequence method), 60
 get_SequenceRule() (PMML44Super.SequenceModel method), 234
 get_setId() (PMML44Super.SetReference method), 238
 get_SetPredicate() (PMML44Super.SequenceModel method), 234
 get_SetReference() (PMML44Super.Sequence method), 232
 get_SGD() (PMML44Super.Optimizers method), 177
 get_SigmoidKernelType() (PMML44Super.SupportVectorMachineModel method), 247
 get_similarityScale() (PMML44Super.ClusteringField method), 48
 get_similarityType() (PMML44Super.TextModelSimilarity method), 267
 get_simpleMatching() (PMML44Super.ComparisonMeasure method), 53
 get_SimplePredicate() (PMML44Super.Attribute method), 27
 get_SimplePredicate() (PMML44Super.CompoundPredicate method), 56
 get_SimplePredicate() (PMML44Super.CompoundRule method), 58
 get_SimplePredicate() (PMML44Super.Node method), 168
 get_SimplePredicate() (PMML44Super.Segment method), 229
 get_SimplePredicate() (PMML44Super.SimpleRule method), 241
 get_SimpleRule() (PMML44Super.CompoundRule method), 58
 get_SimpleRule() (PMML44Super.RuleSet method), 219
 get_SimpleSetPredicate() (PMML44Super.Attribute method), 27
 get_SimpleSetPredicate() (PMML44Super.CompoundPredicate method), 56
 get_SimpleSetPredicate() (PMML44Super.CompoundRule method), 58
 get_SimpleSetPredicate() (PMML44Super.Node method), 168
 get_SimpleSetPredicate() (PMML44Super.Segment method), 229
 get_SimpleSetPredicate() (PMML44Super.SimpleRule method), 241
 get_size() (PMML44Super.Cluster method), 47
 get_size() (PMML44Super.Partition method), 195
 get_slicingAxis() (PMML44Super.LayerParameters method), 126
 get_smoothedValue() (PMML44Super.Level method), 129
 get_smoothedValue() (PMML44Super.Trend_ExpoSmooth method), 283
 get_SpectralAnalysis() (PMML44Super.TimeSeriesModel method), 273
 get_splitCharacteristic() (PMML44Super.DecisionTree method), 76

[get_splitCharacteristic\(\)](#)
 (PMML44Super.TreeModel method), 281
[get_sqlWhere\(\)](#) (PMML44Super.Aggregate
 method), 12
[get_squaredEuclidean\(\)](#)
 (PMML44Super.ComparisonMeasure method),
 53
[get_SSB\(\)](#) (PMML44Super.ClusteringModelQuality
 method), 50
[get_SSE\(\)](#) (PMML44Super.ClusteringModelQuality
 method), 50
[get_standardDeviation\(\)](#)
 (PMML44Super.NumericInfo method), 175
[get_standardDeviation\(\)](#) (PMML44Super.Time
 method), 269
[get_standardError\(\)](#) (PMML44Super.TimeValue
 method), 275
[get_startTime\(\)](#) (PMML44Super.OutlierEffect
 method), 179
[get_startTime\(\)](#) (PMML44Super.TimeSeries
 method), 272
[get_startTimeVariable\(\)](#)
 (PMML44Super.GeneralRegressionModel
 method), 107
[get_StateSpaceModel\(\)](#)
 (PMML44Super.TimeSeriesModel method),
 273
[get_StateVector\(\)](#)
 (PMML44Super.StateSpaceModel method),
 243
[get_statusVariable\(\)](#)
 (PMML44Super.GeneralRegressionModel
 method), 107
[get_stdError\(\)](#) (PMML44Super.MultivariateStat
 method), 153
[get_stepsize\(\)](#) (PMML44Super.TimeAnchor
 method), 270
[get_stride\(\)](#) (PMML44Super.LayerParameters
 method), 126
[get_subjectIDVariable\(\)](#)
 (PMML44Super.GeneralRegressionModel
 method), 107
[get_sumOfSquares\(\)](#) (PMML44Super.AnovaRow
 method), 17
[get_sumSquaredError\(\)](#)
 (PMML44Super.PredictiveModelQuality
 method), 200
[get_sumSquaredRegression\(\)](#)
 (PMML44Super.PredictiveModelQuality
 method), 200
[get_support\(\)](#) (PMML44Super.AssociationRule
 method), 26
[get_support\(\)](#) (PMML44Super.Itemset method), 118
[get_support\(\)](#) (PMML44Super.Sequence method),
 232
[get_support\(\)](#) (PMML44Super.SequenceRule
 method), 236
[get_SupportVector\(\)](#)
 (PMML44Super.SupportVectors method),
 249
[get_SupportVectorMachine\(\)](#)
 (PMML44Super.SupportVectorMachineModel
 method), 247
[get_SupportVectorMachineModel\(\)](#)
 (PMML44Super.AnomalyDetectionModel
 method), 15
[get_SupportVectorMachineModel\(\)](#)
 (PMML44Super.PMML method), 187
[get_SupportVectorMachineModel\(\)](#)
 (PMML44Super.Segment method), 229
[get_SupportVectors\(\)](#)
 (PMML44Super.SupportVectorMachine
 method), 246
[get_svmRepresentation\(\)](#)
 (PMML44Super.SupportVectorMachineModel
 method), 248
[get_TableLocator\(\)](#) (PMML44Super.ChildParent
 method), 45
[get_TableLocator\(\)](#) (PMML44Super.MapValues
 method), 138
[get_TableLocator\(\)](#)
 (PMML44Super.TextIndexNormalization
 method), 263
[get_TableLocator\(\)](#)
 (PMML44Super.TrainingInstances method),
 276
[get_tanimoto\(\)](#) (PMML44Super.ComparisonMeasure
 method), 53
[get_target\(\)](#) (PMML44Super.Anova method), 17
[get_Target\(\)](#) (PMML44Super.Targets method), 256
[get_targetCategory\(\)](#)
 (PMML44Super.MultivariateStats method),
 154
[get_targetCategory\(\)](#) (PMML44Super.PCell
 method), 183
[get_targetCategory\(\)](#) (PMML44Super.PCovCell
 method), 184
[get_targetCategory\(\)](#) (PMML44Super.PPCell
 method), 189
[get_targetCategory\(\)](#)
 (PMML44Super.RegressionTable method),
 214
[get_targetCategory\(\)](#)
 (PMML44Super.SupportVectorMachine
 method), 246
[get_targetField\(\)](#)
 (PMML44Super.DynamicRegressor method),
 90

<code>get_targetField()</code> (PMML44Super.OutputField method), 181	<code>(PMML44Super.TargetValueCounts method), 254</code>
<code>get_targetField()</code> (PMML44Super.PredictiveModelQuality method), 200	<code>get_TargetValueCounts()</code> (PMML44Super.BayesOutput method), 35
<code>get_targetField()</code> (PMML44Super.PsiVector method), 203	<code>get_TargetValueCounts()</code> (PMML44Super.PairCounts method), 191
<code>get_targetFieldDisplayValue()</code> (PMML44Super.LiftData method), 130	<code>get_TargetValueStat()</code> (PMML44Super.TargetValueStats method), 255
<code>get_targetFieldName()</code> (PMML44Super.RegressionModel method), 213	<code>get_TargetValueStats()</code> (PMML44Super.BayesInput method), 33
<code>get_targetFieldValue()</code> (PMML44Super.LiftData method), 130	<code>get_targetVariableName()</code> (PMML44Super.GeneralRegressionModel method), 107
<code>get_targetReferenceCategory()</code> (PMML44Super.GeneralRegressionModel method), 107	<code>get_Taxonomy()</code> (PMML44Super.DataDictionary method), 72
<code>get_Targets()</code> (PMML44Super.BaselineModel method), 31	<code>get_taxonomy()</code> (PMML44Super.DataField method), 74
<code>get_Targets()</code> (PMML44Super.BayesianNetworkModel method), 36	<code>get_Taxonomy()</code> (PMML44Super.TextDictionary method), 259
<code>get_Targets()</code> (PMML44Super.DecisionTree method), 76	<code>get_tCol()</code> (PMML44Super.PCovCell method), 184
<code>get_Targets()</code> (PMML44Super.DeepNetwork method), 78	<code>get_TestDistributions()</code> (PMML44Super.BaselineModel method), 31
<code>get_Targets()</code> (PMML44Super.GaussianProcessModel method), 104	<code>get_testStatistic()</code> (PMML44Super.TestDistributions method), 258
<code>get_Targets()</code> (PMML44Super.GeneralRegressionModel method), 107	<code>get_TextCorpus()</code> (PMML44Super.TextModel method), 264
<code>get_Targets()</code> (PMML44Super.MiningModel method), 146	<code>get_TextDictionary()</code> (PMML44Super.TextModel method), 264
<code>get_Targets()</code> (PMML44Super.NaiveBayesModel method), 156	<code>get_TextDocument()</code> (PMML44Super.TextCorpus method), 259
<code>get_Targets()</code> (PMML44Super.NearestNeighborModel method), 158	<code>get_textField()</code> (PMML44Super.TextIndex method), 261
<code>get_Targets()</code> (PMML44Super.NeuralNetwork method), 163	<code>get_TextIndex()</code> (PMML44Super.Apply method), 21
<code>get_Targets()</code> (PMML44Super.Regression method), 211	<code>get_TextIndex()</code> (PMML44Super.ComplexPartialScore method), 55
<code>get_Targets()</code> (PMML44Super.RegressionModel method), 213	<code>get_TextIndex()</code> (PMML44Super.DefineFunction method), 80
<code>get_Targets()</code> (PMML44Super.RuleSetModel method), 221	<code>get_TextIndex()</code> (PMML44Super.DerivedField method), 83
<code>get_Targets()</code> (PMML44Super.Scorecard method), 224	<code>get_TextIndex()</code> (PMML44Super.Lower method), 135
<code>get_Targets()</code> (PMML44Super.SupportVectorMachineModel method), 247	<code>get_TextIndex()</code> (PMML44Super.Mean method), 142
<code>get_Targets()</code> (PMML44Super.TextModel method), 264	<code>get_TextIndex()</code> (PMML44Super.OutputField method), 181
<code>get_Targets()</code> (PMML44Super.TreeModel method), 281	<code>get_TextIndex()</code> (PMML44Super.TextIndex method), 261
<code>get_TargetValue()</code> (PMML44Super.Target method), 251	<code>get_TextIndex()</code> (PMML44Super.Upper method), 288
<code>get_TargetValueCount()</code>	<code>get_TextIndex()</code> (PMML44Super.Variance

[method](#)), 291
[get_TextIndexNormalization\(\)](#) ([PMML44Super.TextIndex](#) method), 261
[get_TextModel\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) method), 15
[get_TextModel\(\)](#) ([PMML44Super.PMML](#) method), 187
[get_TextModel\(\)](#) ([PMML44Super.Segment](#) method), 229
[get_TextModelNormalization\(\)](#) ([PMML44Super.TextModel](#) method), 264
[get_TextModelSimiliarity\(\)](#) ([PMML44Super.TextModel](#) method), 264
[get_Theta\(\)](#) ([PMML44Super.FinalTheta](#) method), 101
[get_theta\(\)](#) ([PMML44Super.Theta](#) method), 267
[get_ThetaRecursionState\(\)](#) ([PMML44Super.MaximumLikelihoodStat](#) method), 141
[get_threshold\(\)](#) ([PMML44Super.NaiveBayesModel](#) method), 156
[get_threshold\(\)](#) ([PMML44Super.NearestNeighborModel](#) method), 158
[get_threshold\(\)](#) ([PMML44Super.NeuralLayer](#) method), 162
[get_threshold\(\)](#) ([PMML44Super.NeuralNetwork](#) method), 164
[get_threshold\(\)](#) ([PMML44Super.OutputField](#) method), 181
[get_threshold\(\)](#) ([PMML44Super.SupportVectorMachine](#) method), 246
[get_threshold\(\)](#) ([PMML44Super.SupportVectorMachineModel](#) method), 248
[get_Time\(\)](#) ([PMML44Super.AntecedentSequence](#) method), 18
[get_time\(\)](#) ([PMML44Super.BaselineCell](#) method), 30
[get_Time\(\)](#) ([PMML44Super.ConsequentSequence](#) method), 60
[get_Time\(\)](#) ([PMML44Super.Sequence](#) method), 232
[get_Time\(\)](#) ([PMML44Super.SequenceRule](#) method), 236
[get_time\(\)](#) ([PMML44Super.TimeValue](#) method), 275
[get_TimeAnchor\(\)](#) ([PMML44Super.TimeSeries](#) method), 272
[get_TimeCycle\(\)](#) ([PMML44Super.TimeAnchor](#) method), 269
[get_TimeException\(\)](#) ([PMML44Super.TimeAnchor](#) method), 269
[get_TimeSeries\(\)](#) ([PMML44Super.RegressorValues](#) method), 215
[get_TimeSeries\(\)](#) ([PMML44Super.TimeSeriesModel](#) method), 273
[get_TimeSeriesModel\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) method), 15
[get_TimeSeriesModel\(\)](#) ([PMML44Super.PMML](#) method), 187
[get_TimeSeriesModel\(\)](#) ([PMML44Super.Segment](#) method), 229
[get_Timestamp\(\)](#) ([PMML44Super.Header](#) method), 111
[get_Timestamp\(\)](#) ([PMML44Super.TimeValue](#) method), 275
[get_TimeValue\(\)](#) ([PMML44Super.ExponentialSmoothing](#) method), 92
[get_TimeValue\(\)](#) ([PMML44Super.TimeSeries](#) method), 272
[get_tokenize\(\)](#) ([PMML44Super.TextIndex](#) method), 261
[get_tokenize\(\)](#) ([PMML44Super.TextIndexNormalization](#) method), 263
[get_top_k_categories_for_accuracy\(\)](#) ([PMML44Super.Metrics](#) method), 143
[get_totalFreq\(\)](#) ([PMML44Super.Counts](#) method), 70
[get_totalSquaresSum\(\)](#) ([PMML44Super.ContStats](#) method), 63
[get_totalValuesSum\(\)](#) ([PMML44Super.ContStats](#) method), 63
[get_trainingBackend\(\)](#) ([PMML44Super.DerivedField](#) method), 83
[get_TrainingInstances\(\)](#) ([PMML44Super.GaussianProcessModel](#) method), 104
[get_TrainingInstances\(\)](#) ([PMML44Super.NearestNeighborModel](#) method), 158
[get_TrainingParameters\(\)](#) ([PMML44Super.DeepNetwork](#) method), 78
[get_TransferFunctionValues\(\)](#) ([PMML44Super.RegressorValues](#) method), 215
[get_transformation\(\)](#) ([PMML44Super.ARIMA](#) method), 5
[get_transformation\(\)](#) ([PMML44Super.DynamicRegressor](#) method), 90
[get_transformation\(\)](#) ([PMML44Super.ExponentialSmoothing](#) method), 92
[get_TransformationDictionary\(\)](#) ([PMML44Super.PMML](#) method), 187
[get_TransitionMatrix\(\)](#) ([PMML44Super.StateSpaceModel](#) method), 243
[get_TreeModel\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) method), 15

[get_TreeModel\(\)](#) (*PMML44Super.PMML method*), 187
[get_TreeModel\(\)](#) (*PMML44Super.Segment method*), 229
[get_trend\(\)](#) (*PMML44Super.Trend_ExpoSmooth method*), 283
[get_Trend_ExpoSmooth\(\)](#) (*PMML44Super.ExponentialSmoothing method*), 92
[get_TrendCoefficients\(\)](#) (*PMML44Super.RegressorValues method*), 215
[get_trialsValue\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 107
[get_trialsVariable\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 107
[get_TriangularDistributionForBN\(\)](#) (*PMML44Super.ContinuousDistribution method*), 65
[get_tRow\(\)](#) (*PMML44Super.PCovCell method*), 184
[get_True\(\)](#) (*PMML44Super.Attribute method*), 27
[get_True\(\)](#) (*PMML44Super.CompoundPredicate method*), 56
[get_True\(\)](#) (*PMML44Super.CompoundRule method*), 58
[get_True\(\)](#) (*PMML44Super.Node method*), 168
[get_True\(\)](#) (*PMML44Super.Segment method*), 229
[get_True\(\)](#) (*PMML44Super.SimpleRule method*), 241
[get_tValue\(\)](#) (*PMML44Super.MultivariateStat method*), 153
[get_type\(\)](#) (*PMML44Super.AnovaRow method*), 17
[get_type\(\)](#) (*PMML44Super.ArrayType method*), 23
[get_type\(\)](#) (*PMML44Super.OutlierEffect method*), 179
[get_type\(\)](#) (*PMML44Super.PCovMatrix method*), 185
[get_type\(\)](#) (*PMML44Super.Seasonality_ExpoSmooth method*), 228
[get_type\(\)](#) (*PMML44Super.TimeAnchor method*), 270
[get_type\(\)](#) (*PMML44Super.TimeCycle method*), 270
[get_type\(\)](#) (*PMML44Super.TimeException method*), 271
[get_UniformDistribution\(\)](#) (*PMML44Super.Alternate method*), 13
[get_UniformDistribution\(\)](#) (*PMML44Super.Baseline method*), 29
[get_UniformDistribution\(\)](#) (*PMML44Super.TargetValueStat method*), 254
[get_UniformDistributionForBN\(\)](#) (*PMML44Super.ContinuousDistribution method*), 65
[get_unit\(\)](#) (*PMML44Super.Seasonality_ExpoSmooth method*), 228
[get_UnivariateStats\(\)](#) (*PMML44Super.ModelStats method*), 151
[get_Upper\(\)](#) (*PMML44Super.TriangularDistributionForBN method*), 284
[get_upper\(\)](#) (*PMML44Super.UniformDistribution method*), 285
[get_Upper\(\)](#) (*PMML44Super.UniformDistributionForBN method*), 286
[get_upsamplingSize\(\)](#) (*PMML44Super.LayerParameters method*), 126
[get_usage\(\)](#) (*PMML44Super.TimeSeries method*), 272
[get_usageType\(\)](#) (*PMML44Super.MiningField method*), 145
[get_useReasonCodes\(\)](#) (*PMML44Super.Scorecard method*), 224
[get_value\(\)](#) (*PMML44Super.BaselineStratum method*), 32
[get_value\(\)](#) (*PMML44Super.CategoricalPredictor method*), 41
[get_value\(\)](#) (*PMML44Super.Category method*), 42
[get_value\(\)](#) (*PMML44Super.Coefficient method*), 51
[get_Value\(\)](#) (*PMML44Super.DataField method*), 73
[get_value\(\)](#) (*PMML44Super.Decision method*), 75
[get_Value\(\)](#) (*PMML44Super.DerivedField method*), 83
[get_Value\(\)](#) (*PMML44Super.EventValues method*), 91
[get_value\(\)](#) (*PMML44Super.Extension method*), 93
[get_value\(\)](#) (*PMML44Super.FieldValue method*), 97
[get_value\(\)](#) (*PMML44Super.FieldValueCount method*), 97
[get_value\(\)](#) (*PMML44Super.Item method*), 116
[get_value\(\)](#) (*PMML44Super.NormDiscrete method*), 172
[get_value\(\)](#) (*PMML44Super.OutputField method*), 181
[get_value\(\)](#) (*PMML44Super.PairCounts method*), 191
[get_value\(\)](#) (*PMML44Super.Parameter method*), 192
[get_value\(\)](#) (*PMML44Super.ParentValue method*), 195
[get_value\(\)](#) (*PMML44Super.PCovCell method*), 184
[get_value\(\)](#) (*PMML44Super.PPCell method*), 189
[get_value\(\)](#) (*PMML44Super.ResultField method*), 218
[get_value\(\)](#) (*PMML44Super.ScoreDistribution method*), 223
[get_value\(\)](#) (*PMML44Super.SimplePredicate*

`method`), 240
`get_value()` (*PMML44Super.TargetValue method*), 252
`get_value()` (*PMML44Super.TargetValueCount method*), 253
`get_value()` (*PMML44Super.TargetValueStat method*), 254
`get_value()` (*PMML44Super.TimeValue method*), 275
`get_value()` (*PMML44Super.Value method*), 289
`get_value()` (*PMML44Super.ValueProbability method*), 289
`get_valueOf_()` (*PMML44Super.Annotation method*), 13
`get_valueOf_()` (*PMML44Super.ArrayType method*), 23
`get_valueOf_()` (*PMML44Super.Constant method*), 61
`get_valueOf_()` (*PMML44Super.LayerBias method*), 123
`get_valueOf_()` (*PMML44Super.LayerWeights method*), 128
`get_valueOf_()` (*PMML44Super.MatCell method*), 139
`get_valueOf_()` (*PMML44Super.Timestamp method*), 276
`get_ValueProbability()` (*PMML44Super.DiscreteConditionalProbability method*), 85
`get_ValueProbability()` (*PMML44Super.DiscreteNode method*), 86
`get_VariableWeight()` (*PMML44Super.Segment method*), 229
`get_variance()` (*PMML44Super.AnyDistribution method*), 19
`get_variance()` (*PMML44Super.GaussianDistribution method*), 103
`get_Variance()` (*PMML44Super.LognormalDistributionForBN method*), 134
`get_Variance()` (*PMML44Super.NormalDistributionForBN method*), 173
`get_variance()` (*PMML44Super.PsiVector method*), 203
`get_variance()` (*PMML44Super.StateSpaceModel method*), 244
`get_VarianceCoefficients()` (*PMML44Super.GARCHPart method*), 102
`get_VectorDictionary()` (*PMML44Super.SupportVectorMachineModel method*), 248
`get_VectorFields()` (*PMML44Super.VectorDictionary method*), 293
`get_vectorId()` (*PMML44Super.SupportVector*

`method`), 245
`get_VectorInstance()` (*PMML44Super.VectorDictionary method*), 293
`get_VerificationField()` (*PMML44Super.VerificationFields method*), 296
`get_VerificationFields()` (*PMML44Super.ModelVerification method*), 151
`get_version()` (*PMML44Super.Application method*), 20
`get_version()` (*PMML44Super.PMML method*), 187
`get_weight()` (*PMML44Super.Con method*), 59
`get_weight()` (*PMML44Super.Item method*), 116
`get_weight()` (*PMML44Super.Segment method*), 229
`get_weight()` (*PMML44Super.SimpleRule method*), 241
`get_weighted()` (*PMML44Super.PartitionFieldStats method*), 196
`get_weighted()` (*PMML44Super.UnivariateStats method*), 287
`get_weightField()` (*PMML44Super.TestDistributions method*), 258
`get_weightsFlattenAxis()` (*PMML44Super.LayerWeights method*), 128
`get_weightsShape()` (*PMML44Super.LayerWeights method*), 128
`get_width()` (*PMML44Super.NeuralLayer method*), 162
`get_width()` (*PMML44Super.NeuralNetwork method*), 164
`get_width()` (*PMML44Super.Neuron method*), 167
`get_windowSize()` (*PMML44Super.TestDistributions method*), 258
`get_wordSeparatorCharacterRE()` (*PMML44Super.TextIndex method*), 261
`get_wordSeparatorCharacterRE()` (*PMML44Super.TextIndexNormalization method*), 263
`get_XCoordinates()` (*PMML44Super.LiftGraph method*), 131
`get_XCoordinates()` (*PMML44Super.ROCGraph method*), 208
`get_YCoordinates()` (*PMML44Super.LiftGraph method*), 131
`get_YCoordinates()` (*PMML44Super.ROCGraph method*), 208
`get_zeroThreshold()` (*PMML44Super.VerificationField method*), 295

H

- `hasContent_()` (*PMML44Super.AbsoluteExponentialKernel method*), 7
- `hasContent_()` (*PMML44Super.Adadelta method*), 8
- `hasContent_()` (*PMML44Super.Adagrad method*), 9
- `hasContent_()` (*PMML44Super.Adam method*), 10
- `hasContent_()` (*PMML44Super.Adamax method*), 11
- `hasContent_()` (*PMML44Super.Aggregate method*), 12
- `hasContent_()` (*PMML44Super.Alternate method*), 13
- `hasContent_()` (*PMML44Super.Annotation method*), 13
- `hasContent_()` (*PMML44Super.AnomalyDetectionModel method*), 15
- `hasContent_()` (*PMML44Super.Anova method*), 17
- `hasContent_()` (*PMML44Super.AnovaRow method*), 17
- `hasContent_()` (*PMML44Super.AntecedentSequence method*), 18
- `hasContent_()` (*PMML44Super.AnyDistribution method*), 19
- `hasContent_()` (*PMML44Super.Application method*), 20
- `hasContent_()` (*PMML44Super.Apply method*), 21
- `hasContent_()` (*PMML44Super.AR method*), 3
- `hasContent_()` (*PMML44Super.ARDSquaredExponentialKernel method*), 4
- `hasContent_()` (*PMML44Super.ARIMA method*), 5
- `hasContent_()` (*PMML44Super.ARMAPart method*), 6
- `hasContent_()` (*PMML44Super.ArrayType method*), 23
- `hasContent_()` (*PMML44Super.AssociationModel method*), 24
- `hasContent_()` (*PMML44Super.AssociationRule method*), 26
- `hasContent_()` (*PMML44Super.Attribute method*), 27
- `hasContent_()` (*PMML44Super.BaseCumHazardTables method*), 28
- `hasContent_()` (*PMML44Super.Baseline method*), 29
- `hasContent_()` (*PMML44Super.BaselineCell method*), 30
- `hasContent_()` (*PMML44Super.BaselineModel method*), 31
- `hasContent_()` (*PMML44Super.BaselineStratum method*), 32
- `hasContent_()` (*PMML44Super.BayesianNetworkModel method*), 36
- `hasContent_()` (*PMML44Super.BayesianNetworkNodes method*), 37
- `hasContent_()` (*PMML44Super.BayesInput method*), 33
- `hasContent_()` (*PMML44Super.BayesInputs method*), 34
- `hasContent_()` (*PMML44Super.BayesOutput method*), 35
- `hasContent_()` (*PMML44Super.binarySimilarity method*), 299
- `hasContent_()` (*PMML44Super.BlockIndicator method*), 38
- `hasContent_()` (*PMML44Super.BoundaryValueMeans method*), 39
- `hasContent_()` (*PMML44Super.BoundaryValues method*), 39
- `hasContent_()` (*PMML44Super.CategoricalPredictor method*), 41
- `hasContent_()` (*PMML44Super.Categories method*), 42
- `hasContent_()` (*PMML44Super.Category method*), 42
- `hasContent_()` (*PMML44Super.Characteristic method*), 43
- `hasContent_()` (*PMML44Super.Characteristics method*), 44
- `hasContent_()` (*PMML44Super.chebychev method*), 299
- `hasContent_()` (*PMML44Super.ChildParent method*), 45
- `hasContent_()` (*PMML44Super.cityBlock method*), 300
- `hasContent_()` (*PMML44Super.ClassLabels method*), 46
- `hasContent_()` (*PMML44Super.Cluster method*), 47
- `hasContent_()` (*PMML44Super.ClusteringField method*), 48
- `hasContent_()` (*PMML44Super.ClusteringModel method*), 49
- `hasContent_()` (*PMML44Super.ClusteringModelQuality method*), 50
- `hasContent_()` (*PMML44Super.Coefficient method*), 51
- `hasContent_()` (*PMML44Super.Coefficients method*), 52
- `hasContent_()` (*PMML44Super.ComparisonMeasure method*), 53
- `hasContent_()` (*PMML44Super.Comparisons method*), 54
- `hasContent_()` (*PMML44Super.ComplexPartialScore method*), 55
- `hasContent_()` (*PMML44Super.CompoundPredicate method*), 56
- `hasContent_()` (*PMML44Super.CompoundRule method*), 58
- `hasContent_()` (*PMML44Super.Con method*), 59
- `hasContent_()` (*PMML44Super.ConfusionMatrix method*), 59

`method`), 60
`hasContent_()` (*PMML44Super.ConsequentSequence method*), 60
`hasContent_()` (*PMML44Super.Constant method*), 61
`hasContent_()` (*PMML44Super.Constraints method*), 62
`hasContent_()` (*PMML44Super.ContinuousConditionalProbability method*), 64
`hasContent_()` (*PMML44Super.ContinuousDistribution method*), 65
`hasContent_()` (*PMML44Super.ContinuousNode method*), 66
`hasContent_()` (*PMML44Super.ContStats method*), 63
`hasContent_()` (*PMML44Super.CorrelationFields method*), 67
`hasContent_()` (*PMML44Super.CorrelationMethods method*), 68
`hasContent_()` (*PMML44Super.Correlations method*), 69
`hasContent_()` (*PMML44Super.CorrelationValues method*), 69
`hasContent_()` (*PMML44Super.COUNT_TABLE_TYPE method*), 40
`hasContent_()` (*PMML44Super.Counts method*), 70
`hasContent_()` (*PMML44Super.Covariances method*), 71
`hasContent_()` (*PMML44Super.CovariateList method*), 72
`hasContent_()` (*PMML44Super.DataDictionary method*), 72
`hasContent_()` (*PMML44Super.DataField method*), 74
`hasContent_()` (*PMML44Super.Decision method*), 75
`hasContent_()` (*PMML44Super.Decisions method*), 77
`hasContent_()` (*PMML44Super.DecisionTree method*), 76
`hasContent_()` (*PMML44Super.DeepNetwork method*), 78
`hasContent_()` (*PMML44Super.DefineFunction method*), 80
`hasContent_()` (*PMML44Super.Delimiter method*), 81
`hasContent_()` (*PMML44Super.Denominator method*), 82
`hasContent_()` (*PMML44Super.DerivedField method*), 83
`hasContent_()` (*PMML44Super.DiscreteConditionalProbability method*), 86
`hasContent_()` (*PMML44Super.DiscreteNode method*), 87
`hasContent_()` (*PMML44Super.Discretize method*), 88
`hasContent_()` (*PMML44Super.DiscretizeBin method*), 89
`hasContent_()` (*PMML44Super.DiscrStats method*), 85
`hasContent_()` (*PMML44Super.DocumentTermMatrix method*), 89
`hasContent_()` (*PMML44Super.DynamicRegressor method*), 90
`hasContent_()` (*PMML44Super.euclidean method*), 301
`hasContent_()` (*PMML44Super.EventValues method*), 91
`hasContent_()` (*PMML44Super.ExponentialSmoothing method*), 92
`hasContent_()` (*PMML44Super.Extension method*), 93
`hasContent_()` (*PMML44Super.FactorList method*), 94
`hasContent_()` (*PMML44Super.False_ method*), 94
`hasContent_()` (*PMML44Super.FieldColumnPair method*), 95
`hasContent_()` (*PMML44Super.FieldRef method*), 96
`hasContent_()` (*PMML44Super.FieldValue method*), 97
`hasContent_()` (*PMML44Super.FieldValueCount method*), 98
`hasContent_()` (*PMML44Super.FinalNoise method*), 98
`hasContent_()` (*PMML44Super.FinalNu method*), 99
`hasContent_()` (*PMML44Super.FinalOmega method*), 99
`hasContent_()` (*PMML44Super.FinalPredictedNoise method*), 100
`hasContent_()` (*PMML44Super.FinalStateVector method*), 100
`hasContent_()` (*PMML44Super.FinalTheta method*), 101
`hasContent_()` (*PMML44Super.GARCH method*), 101
`hasContent_()` (*PMML44Super.GARCHPart method*), 102
`hasContent_()` (*PMML44Super.GaussianDistribution method*), 103
`hasContent_()` (*PMML44Super.GaussianProcessModel method*), 105
`hasContent_()` (*PMML44Super.GeneralizedExponentialKernel method*), 109
`hasContent_()` (*PMML44Super.GeneralRegressionModel method*), 107
`hasContent_()` (*PMML44Super.Header method*), 111
`hasContent_()` (*PMML44Super.HVector method*),

- 110
- `hasContent_()` (*PMML44Super.Indices method*), 113
- `hasContent_()` (*PMML44Super.InlineTable method*), 113
- `hasContent_()` (*PMML44Super.InstanceField method*), 114
- `hasContent_()` (*PMML44Super.InstanceFields method*), 115
- `hasContent_()` (*PMML44Super.INT_Entries method*), 112
- `hasContent_()` (*PMML44Super.INT_SparseArray method*), 112
- `hasContent_()` (*PMML44Super.Interval method*), 116
- `hasContent_()` (*PMML44Super.Item method*), 116
- `hasContent_()` (*PMML44Super.ItemRef method*), 117
- `hasContent_()` (*PMML44Super.Itemset method*), 118
- `hasContent_()` (*PMML44Super.jaccard method*), 301
- `hasContent_()` (*PMML44Super.KalmanState method*), 120
- `hasContent_()` (*PMML44Super.KNNInput method*), 119
- `hasContent_()` (*PMML44Super.KNNInputs method*), 120
- `hasContent_()` (*PMML44Super.KohonenMap method*), 121
- `hasContent_()` (*PMML44Super.Lag method*), 122
- `hasContent_()` (*PMML44Super.Lambda method*), 123
- `hasContent_()` (*PMML44Super.LayerBias method*), 123
- `hasContent_()` (*PMML44Super.LayerParameters method*), 126
- `hasContent_()` (*PMML44Super.LayerWeights method*), 128
- `hasContent_()` (*PMML44Super.Level method*), 129
- `hasContent_()` (*PMML44Super.LiftData method*), 130
- `hasContent_()` (*PMML44Super.LiftGraph method*), 131
- `hasContent_()` (*PMML44Super.LinearKernelType method*), 132
- `hasContent_()` (*PMML44Super.LinearNorm method*), 132
- `hasContent_()` (*PMML44Super.LocalTransformations method*), 133
- `hasContent_()` (*PMML44Super.LognormalDistributionForBN method*), 134
- `hasContent_()` (*PMML44Super.Losses method*), 134
- `hasContent_()` (*PMML44Super.Lower method*), 135
- `hasContent_()` (*PMML44Super.MA method*), 136
- `hasContent_()` (*PMML44Super.MACoefficients method*), 137
- `hasContent_()` (*PMML44Super.MapValues method*), 138
- `hasContent_()` (*PMML44Super.MatCell method*), 139
- `hasContent_()` (*PMML44Super.Matrix method*), 140
- `hasContent_()` (*PMML44Super.MaximumLikelihoodStat method*), 141
- `hasContent_()` (*PMML44Super.Mean method*), 142
- `hasContent_()` (*PMML44Super.MeasurementMatrix method*), 142
- `hasContent_()` (*PMML44Super.Metrics method*), 143
- `hasContent_()` (*PMML44Super.MiningBuildTask method*), 144
- `hasContent_()` (*PMML44Super.MiningField method*), 145
- `hasContent_()` (*PMML44Super.MiningModel method*), 146
- `hasContent_()` (*PMML44Super.MiningSchema method*), 148
- `hasContent_()` (*PMML44Super.minkowski method*), 302
- `hasContent_()` (*PMML44Super.MissingValueWeights method*), 148
- `hasContent_()` (*PMML44Super.ModelExplanation method*), 149
- `hasContent_()` (*PMML44Super.ModelLiftGraph method*), 150
- `hasContent_()` (*PMML44Super.ModelStats method*), 151
- `hasContent_()` (*PMML44Super.ModelVerification method*), 152
- `hasContent_()` (*PMML44Super.MultivariateStat method*), 153
- `hasContent_()` (*PMML44Super.MultivariateStats method*), 154
- `hasContent_()` (*PMML44Super.Nadam method*), 155
- `hasContent_()` (*PMML44Super.NaiveBayesModel method*), 156
- `hasContent_()` (*PMML44Super.NearestNeighborModel method*), 158
- `hasContent_()` (*PMML44Super.NetworkLayer method*), 159
- `hasContent_()` (*PMML44Super.NeuralInput method*), 160
- `hasContent_()` (*PMML44Super.NeuralInputs method*), 161
- `hasContent_()` (*PMML44Super.NeuralLayer method*), 162
- `hasContent_()` (*PMML44Super.NeuralNetwork method*), 164
- `hasContent_()` (*PMML44Super.NeuralOutput method*), 165

<code>hasContent_()</code> (PMML44Super.NeuralOutputs method), 166	<code>hasContent_()</code> (PMML44Super.PoissonDistribution method), 198
<code>hasContent_()</code> (PMML44Super.Neuron method), 167	<code>hasContent_()</code> (PMML44Super.PolynomialKernelType method), 199
<code>hasContent_()</code> (PMML44Super.Node method), 168	<code>hasContent_()</code> (PMML44Super.PPCell method), 189
<code>hasContent_()</code> (PMML44Super.NonseasonalComponent method), 169	<code>hasContent_()</code> (PMML44Super.PPMatrix method), 190
<code>hasContent_()</code> (PMML44Super.NonseasonalFactor method), 170	<code>hasContent_()</code> (PMML44Super.PredictiveModelQuality method), 200
<code>hasContent_()</code> (PMML44Super.NormalDistributionForBN method), 173	<code>hasContent_()</code> (PMML44Super.Predictor method), 202
<code>hasContent_()</code> (PMML44Super.NormContinuous method), 171	<code>hasContent_()</code> (PMML44Super.PredictorTerm method), 203
<code>hasContent_()</code> (PMML44Super.NormDiscrete method), 172	<code>hasContent_()</code> (PMML44Super.PsiVector method), 203
<code>hasContent_()</code> (PMML44Super.Numerator method), 174	<code>hasContent_()</code> (PMML44Super.Quantile method), 204
<code>hasContent_()</code> (PMML44Super.NumericInfo method), 175	<code>hasContent_()</code> (PMML44Super.RadialBasisKernel method), 209
<code>hasContent_()</code> (PMML44Super.NumericPredictor method), 176	<code>hasContent_()</code> (PMML44Super.RadialBasisKernelType method), 210
<code>hasContent_()</code> (PMML44Super.Optimizers method), 177	<code>hasContent_()</code> (PMML44Super.RandomLiftGraph method), 210
<code>hasContent_()</code> (PMML44Super.OptimumLiftGraph method), 178	<code>hasContent_()</code> (PMML44Super.REAL_Entries method), 205
<code>hasContent_()</code> (PMML44Super.OutlierEffect method), 179	<code>hasContent_()</code> (PMML44Super.REAL_SparseArray method), 205
<code>hasContent_()</code> (PMML44Super.Output method), 180	<code>hasContent_()</code> (PMML44Super.Regression method), 211
<code>hasContent_()</code> (PMML44Super.OutputField method), 181	<code>hasContent_()</code> (PMML44Super.RegressionModel method), 213
<code>hasContent_()</code> (PMML44Super.PairCounts method), 191	<code>hasContent_()</code> (PMML44Super.RegressionTable method), 214
<code>hasContent_()</code> (PMML44Super.Parameter method), 192	<code>hasContent_()</code> (PMML44Super.RegressorValues method), 215
<code>hasContent_()</code> (PMML44Super.ParameterField method), 193	<code>hasContent_()</code> (PMML44Super.Residuals method), 217
<code>hasContent_()</code> (PMML44Super.ParameterList method), 194	<code>hasContent_()</code> (PMML44Super.ResidualSquareCoefficients method), 216
<code>hasContent_()</code> (PMML44Super.ParamMatrix method), 192	<code>hasContent_()</code> (PMML44Super.ResultField method), 218
<code>hasContent_()</code> (PMML44Super.ParentValue method), 195	<code>hasContent_()</code> (PMML44Super.RMSprop method), 206
<code>hasContent_()</code> (PMML44Super.Partition method), 195	<code>hasContent_()</code> (PMML44Super.ROC method), 207
<code>hasContent_()</code> (PMML44Super.PartitionFieldStats method), 196	<code>hasContent_()</code> (PMML44Super.ROCGraph method), 208
<code>hasContent_()</code> (PMML44Super.PastVariances method), 197	<code>hasContent_()</code> (PMML44Super.row method), 302
<code>hasContent_()</code> (PMML44Super.PCell method), 183	<code>hasContent_()</code> (PMML44Super.RuleSelectionMethod method), 219
<code>hasContent_()</code> (PMML44Super.PCovCell method), 184	<code>hasContent_()</code> (PMML44Super.RuleSet method), 220
<code>hasContent_()</code> (PMML44Super.PCovMatrix method), 185	<code>hasContent_()</code> (PMML44Super.RuleSetModel method), 221
<code>hasContent_()</code> (PMML44Super.PMML method), 187	<code>hasContent_()</code> (PMML44Super.Scorecard method),

- 224
- `hasContent_()` (*PMML44Super.ScoreDistribution method*), 223
- `hasContent_()` (*PMML44Super.SeasonalComponent method*), 226
- `hasContent_()` (*PMML44Super.SeasonalFactor method*), 226
- `hasContent_()` (*PMML44Super.Seasonality_ExpoSmooth method*), 228
- `hasContent_()` (*PMML44Super.SeasonalTrendDecomposition method*), 227
- `hasContent_()` (*PMML44Super.Segment method*), 230
- `hasContent_()` (*PMML44Super.Segmentation method*), 231
- `hasContent_()` (*PMML44Super.Sequence method*), 232
- `hasContent_()` (*PMML44Super.SequenceModel method*), 234
- `hasContent_()` (*PMML44Super.SequenceReference method*), 235
- `hasContent_()` (*PMML44Super.SequenceRule method*), 236
- `hasContent_()` (*PMML44Super.SetPredicate method*), 237
- `hasContent_()` (*PMML44Super.SetReference method*), 238
- `hasContent_()` (*PMML44Super.SGD method*), 222
- `hasContent_()` (*PMML44Super.SigmoidKernelType method*), 239
- `hasContent_()` (*PMML44Super.simpleMatching method*), 303
- `hasContent_()` (*PMML44Super.SimplePredicate method*), 240
- `hasContent_()` (*PMML44Super.SimpleRule method*), 241
- `hasContent_()` (*PMML44Super.SimpleSetPredicate method*), 242
- `hasContent_()` (*PMML44Super.SpectralAnalysis method*), 243
- `hasContent_()` (*PMML44Super.squaredEuclidean method*), 304
- `hasContent_()` (*PMML44Super.StateSpaceModel method*), 244
- `hasContent_()` (*PMML44Super.StateVector method*), 244
- `hasContent_()` (*PMML44Super.SupportVector method*), 245
- `hasContent_()` (*PMML44Super.SupportVectorMachine method*), 246
- `hasContent_()` (*PMML44Super.SupportVectorMachineModel method*), 248
- `hasContent_()` (*PMML44Super.SupportVectors method*), 249
- `hasContent_()` (*PMML44Super.TableLocator method*), 250
- `hasContent_()` (*PMML44Super.tanimoto method*), 304
- `hasContent_()` (*PMML44Super.Target method*), 251
- `hasContent_()` (*PMML44Super.Targets method*), 256
- `hasContent_()` (*PMML44Super.TargetValue method*), 252
- `hasContent_()` (*PMML44Super.TargetValueCount method*), 253
- `hasContent_()` (*PMML44Super.TargetValueCounts method*), 254
- `hasContent_()` (*PMML44Super.TargetValueStat method*), 254
- `hasContent_()` (*PMML44Super.TargetValueStats method*), 255
- `hasContent_()` (*PMML44Super.Taxonomy method*), 257
- `hasContent_()` (*PMML44Super.TestDistributions method*), 258
- `hasContent_()` (*PMML44Super.TextCorpus method*), 259
- `hasContent_()` (*PMML44Super.TextDictionary method*), 259
- `hasContent_()` (*PMML44Super.TextDocument method*), 260
- `hasContent_()` (*PMML44Super.TextIndex method*), 262
- `hasContent_()` (*PMML44Super.TextIndexNormalization method*), 263
- `hasContent_()` (*PMML44Super.TextModel method*), 265
- `hasContent_()` (*PMML44Super.TextModelNormalization method*), 266
- `hasContent_()` (*PMML44Super.TextModelSimiliarity method*), 267
- `hasContent_()` (*PMML44Super.Theta method*), 267
- `hasContent_()` (*PMML44Super.ThetaRecursionState method*), 268
- `hasContent_()` (*PMML44Super.Time method*), 269
- `hasContent_()` (*PMML44Super.TimeAnchor method*), 270
- `hasContent_()` (*PMML44Super.TimeCycle method*), 271
- `hasContent_()` (*PMML44Super.TimeException method*), 271
- `hasContent_()` (*PMML44Super.TimeSeries method*), 272
- `hasContent_()` (*PMML44Super.TimeSeriesModel method*), 274
- `hasContent_()` (*PMML44Super.Timestamp method*), 276
- `hasContent_()` (*PMML44Super.TimeValue method*),

- 275
- `hasContent_()` (*PMML44Super.TrainingInstances* method), 276
- `hasContent_()` (*PMML44Super.TrainingParameters* method), 277
- `hasContent_()` (*PMML44Super.TransferFunctionValues* method), 278
- `hasContent_()` (*PMML44Super.TransformationDictionary* method), 279
- `hasContent_()` (*PMML44Super.TransitionMatrix* method), 280
- `hasContent_()` (*PMML44Super.TreeModel* method), 281
- `hasContent_()` (*PMML44Super.Trend_ExpoSmooth* method), 283
- `hasContent_()` (*PMML44Super.TrendCoefficients* method), 282
- `hasContent_()` (*PMML44Super.TriangularDistributionForBN* method), 284
- `hasContent_()` (*PMML44Super.True_* method), 284
- `hasContent_()` (*PMML44Super.UniformDistribution* method), 285
- `hasContent_()` (*PMML44Super.UniformDistributionForBN* method), 286
- `hasContent_()` (*PMML44Super.UnivariateStats* method), 287
- `hasContent_()` (*PMML44Super.Upper* method), 288
- `hasContent_()` (*PMML44Super.Value* method), 289
- `hasContent_()` (*PMML44Super.ValueProbability* method), 289
- `hasContent_()` (*PMML44Super.VariableWeight* method), 290
- `hasContent_()` (*PMML44Super.Variance* method), 291
- `hasContent_()` (*PMML44Super.VarianceCoefficients* method), 292
- `hasContent_()` (*PMML44Super.VectorDictionary* method), 293
- `hasContent_()` (*PMML44Super.VectorFields* method), 294
- `hasContent_()` (*PMML44Super.VectorInstance* method), 295
- `hasContent_()` (*PMML44Super.VerificationField* method), 295
- `hasContent_()` (*PMML44Super.VerificationFields* method), 296
- `hasContent_()` (*PMML44Super.XCoordinates* method), 297
- `hasContent_()` (*PMML44Super.YCoordinates* method), 298
- `Header` (class in *PMML44Super*), 110
- `HVector` (class in *PMML44Super*), 110
- `Indices` (class in *PMML44Super*), 112
- `InlineTable` (class in *PMML44Super*), 113
- `insert_Adadelta_at()` (*PMML44Super.Optimizers* method), 177
- `insert_Adagrad_at()` (*PMML44Super.Optimizers* method), 177
- `insert_Adam_at()` (*PMML44Super.Optimizers* method), 177
- `insert_Adamax_at()` (*PMML44Super.Optimizers* method), 177
- `insert_Aggregate_at()` (*PMML44Super.Apply* method), 21
- `insert_Annotation_at()` (*PMML44Super.Header* method), 111
- `insert_AnomalyDetectionModel_at()` (*PMML44Super.PMML* method), 187
- `insert_AnovaRow_at()` (*PMML44Super.Anova* method), 17
- `insert_anytypeobjs_()` (*PMML44Super.Extension* method), 93
- `insert_anytypeobjs_()` (*PMML44Super.row* method), 302
- `insert_Apply_at()` (*PMML44Super.Apply* method), 21
- `insert_Array_at()` (*PMML44Super.DiscrStats* method), 85
- `insert_Array_at()` (*PMML44Super.Matrix* method), 140
- `insert_Array_at()` (*PMML44Super.PartitionFieldStats* method), 196
- `insert_AssociationModel_at()` (*PMML44Super.PMML* method), 187
- `insert_AssociationRule_at()` (*PMML44Super.AssociationModel* method), 24
- `insert_Attribute_at()` (*PMML44Super.Characteristic* method), 43
- `insert_BaselineCell_at()` (*PMML44Super.BaseCumHazardTables* method), 28
- `insert_BaselineCell_at()` (*PMML44Super.BaselineStratum* method), 32
- `insert_BaselineModel_at()` (*PMML44Super.PMML* method), 187
- `insert_BaselineStratum_at()` (*PMML44Super.BaseCumHazardTables* method), 28
- `insert_BayesianNetworkModel_at()` (*PMML44Super.PMML* method), 187
- `insert_BayesInput_at()` (*PMML44Super.BayesInputs* method), 34

`insert_BlockIndicator_at()`
 (*PMML44Super.Lag method*), 122
`insert_CategoricalPredictor_at()`
 (*PMML44Super.RegressionTable method*), 214
`insert_CategoricalPredictor_at()`
 (*PMML44Super.VectorFields method*), 294
`insert_Category_at()`
 (*PMML44Super.Categories method*), 42
`insert_Characteristic_at()`
 (*PMML44Super.Characteristics method*), 44
`insert_ChildParent_at()`
 (*PMML44Super.Taxonomy method*), 257
`insert_Cluster_at()`
 (*PMML44Super.ClusteringModel method*), 49
`insert_ClusteringField_at()`
 (*PMML44Super.ClusteringModel method*), 49
`insert_ClusteringModel_at()`
 (*PMML44Super.PMML method*), 187
`insert_ClusteringModelQuality_at()`
 (*PMML44Super.ModelExplanation method*), 149
`insert_Coefficient_at()`
 (*PMML44Super.Coefficients method*), 52
`insert_CompoundPredicate_at()`
 (*PMML44Super.CompoundPredicate method*), 56
`insert_CompoundRule_at()`
 (*PMML44Super.CompoundRule method*), 58
`insert_CompoundRule_at()`
 (*PMML44Super.RuleSet method*), 220
`insert_Con_at()` (*PMML44Super.Neuron method*), 167
`insert_Constant_at()` (*PMML44Super.Apply method*), 21
`insert_ContinuousConditionalProbability_at()`
 (*PMML44Super.ContinuousNode method*), 66
`insert_ContinuousDistribution_at()`
 (*PMML44Super.ContinuousConditionalProbability method*), 64
`insert_ContinuousDistribution_at()`
 (*PMML44Super.ContinuousNode method*), 66
`insert_ContinuousNode_at()`
 (*PMML44Super.BayesianNetworkNodes method*), 37
`insert_DataField_at()`
 (*PMML44Super.DataDictionary method*), 72
`insert_Decision_at()` (*PMML44Super.Decisions method*), 77
`insert_DecisionTree_at()`
 (*PMML44Super.MiningModel method*), 146
`insert_DeepNetwork_at()`
 (*PMML44Super.PMML method*), 187
`insert_DefineFunction_at()`
 (*PMML44Super.TransformationDictionary method*), 279
`insert_Delimiter_at()`
 (*PMML44Super.Sequence method*), 232
`insert_DerivedField_at()`
 (*PMML44Super.ContinuousNode method*), 66
`insert_DerivedField_at()`
 (*PMML44Super.DiscreteNode method*), 87
`insert_DerivedField_at()`
 (*PMML44Super.LocalTransformations method*), 133
`insert_DerivedField_at()`
 (*PMML44Super.TransformationDictionary method*), 279
`insert_DiscreteConditionalProbability_at()`
 (*PMML44Super.DiscreteNode method*), 87
`insert_DiscreteNode_at()`
 (*PMML44Super.BayesianNetworkNodes method*), 37
`insert_Discretize_at()` (*PMML44Super.Apply method*), 21
`insert_DiscretizeBin_at()`
 (*PMML44Super.Discretize method*), 88
`insert_DynamicRegressor_at()`
 (*PMML44Super.ARIMA method*), 5
`insert_DynamicRegressor_at()`
 (*PMML44Super.StateSpaceModel method*), 244
`insert_Extension_at()`
 (*PMML44Super.AbsoluteExponentialKernel method*), 7
`insert_Extension_at()`
 (*PMML44Super.Adadelata method*), 8
`insert_Extension_at()` (*PMML44Super.Adagrad method*), 9
`insert_Extension_at()` (*PMML44Super.Adam method*), 10
`insert_Extension_at()` (*PMML44Super.Adamax method*), 11
`insert_Extension_at()`
 (*PMML44Super.Aggregate method*), 12
`insert_Extension_at()`
 (*PMML44Super.Annotation method*), 13
`insert_Extension_at()`
 (*PMML44Super.AnomalyDetectionModel method*), 15
`insert_Extension_at()` (*PMML44Super.Anova method*), 17

<code>insert_Extension_at()</code> (<i>PMML44Super.AnovaRow method</i>), 17	<code>insert_Extension_at()</code> (<i>PMML44Super.BlockIndicator method</i>), 38
<code>insert_Extension_at()</code> (<i>PMML44Super.AntecedentSequence method</i>), 18	<code>insert_Extension_at()</code> (<i>PMML44Super.BoundaryValueMeans method</i>), 39
<code>insert_Extension_at()</code> (<i>PMML44Super.AnyDistribution method</i>), 19	<code>insert_Extension_at()</code> (<i>PMML44Super.BoundaryValues method</i>), 39
<code>insert_Extension_at()</code> (<i>PMML44Super.Application method</i>), 20	<code>insert_Extension_at()</code> (<i>PMML44Super.CategoricalPredictor method</i>), 41
<code>insert_Extension_at()</code> (<i>PMML44Super.Apply method</i>), 21	<code>insert_Extension_at()</code> (<i>PMML44Super.Categories method</i>), 42
<code>insert_Extension_at()</code> (<i>PMML44Super.AR method</i>), 3	<code>insert_Extension_at()</code> (<i>PMML44Super.Category method</i>), 42
<code>insert_Extension_at()</code> (<i>PMML44Super.ARDSquaredExponentialKernel method</i>), 4	<code>insert_Extension_at()</code> (<i>PMML44Super.Characteristic method</i>), 43
<code>insert_Extension_at()</code> (<i>PMML44Super.ARIMA method</i>), 5	<code>insert_Extension_at()</code> (<i>PMML44Super.Characteristics method</i>), 44
<code>insert_Extension_at()</code> (<i>PMML44Super.ARMAPart method</i>), 6	<code>insert_Extension_at()</code> (<i>PMML44Super.chebychev method</i>), 299
<code>insert_Extension_at()</code> (<i>PMML44Super.AssociationModel method</i>), 24	<code>insert_Extension_at()</code> (<i>PMML44Super.ChildParent method</i>), 45
<code>insert_Extension_at()</code> (<i>PMML44Super.AssociationRule method</i>), 26	<code>insert_Extension_at()</code> (<i>PMML44Super.cityBlock method</i>), 300
<code>insert_Extension_at()</code> (<i>PMML44Super.Attribute method</i>), 27	<code>insert_Extension_at()</code> (<i>PMML44Super.ClassLabels method</i>), 46
<code>insert_Extension_at()</code> (<i>PMML44Super.BaseCumHazardTables method</i>), 28	<code>insert_Extension_at()</code> (<i>PMML44Super.Cluster method</i>), 47
<code>insert_Extension_at()</code> (<i>PMML44Super.BaselineCell method</i>), 30	<code>insert_Extension_at()</code> (<i>PMML44Super.ClusteringField method</i>), 48
<code>insert_Extension_at()</code> (<i>PMML44Super.BaselineModel method</i>), 31	<code>insert_Extension_at()</code> (<i>PMML44Super.ClusteringModel method</i>), 49
<code>insert_Extension_at()</code> (<i>PMML44Super.BaselineStratum method</i>), 32	<code>insert_Extension_at()</code> (<i>PMML44Super.ClusteringModelQuality method</i>), 50
<code>insert_Extension_at()</code> (<i>PMML44Super.BayesianNetworkModel method</i>), 36	<code>insert_Extension_at()</code> (<i>PMML44Super.Coefficient method</i>), 51
<code>insert_Extension_at()</code> (<i>PMML44Super.BayesianNetworkNodes method</i>), 37	<code>insert_Extension_at()</code> (<i>PMML44Super.Coefficients method</i>), 52
<code>insert_Extension_at()</code> (<i>PMML44Super.BayesInput method</i>), 33	<code>insert_Extension_at()</code> (<i>PMML44Super.ComparisonMeasure method</i>), 53
<code>insert_Extension_at()</code> (<i>PMML44Super.BayesInputs method</i>), 34	<code>insert_Extension_at()</code> (<i>PMML44Super.Comparisons method</i>), 54
<code>insert_Extension_at()</code> (<i>PMML44Super.BayesOutput method</i>), 35	<code>insert_Extension_at()</code> (<i>PMML44Super.ComplexPartialScore method</i>), 55
<code>insert_Extension_at()</code> (<i>PMML44Super.binarySimilarity method</i>), 299	<code>insert_Extension_at()</code>

(PMML44Super.CompoundPredicate method),
 57
 insert_Extension_at() (PMML44Super.CompoundRule method),
 58
 insert_Extension_at() (PMML44Super.Con
 method), 59
 insert_Extension_at() (PMML44Super.ConfusionMatrix method),
 60
 insert_Extension_at() (PMML44Super.ConsequentSequence method),
 60
 insert_Extension_at() (PMML44Super.Constraints method), 62
 insert_Extension_at() (PMML44Super.ContinuousConditionalProbability
 method), 64
 insert_Extension_at() (PMML44Super.ContinuousDistribution
 method), 65
 insert_Extension_at() (PMML44Super.ContinuousNode method),
 66
 insert_Extension_at() (PMML44Super.ContStats method), 63
 insert_Extension_at() (PMML44Super.CorrelationFields method), 67
 insert_Extension_at() (PMML44Super.CorrelationMethods method),
 68
 insert_Extension_at() (PMML44Super.Correlations method), 69
 insert_Extension_at() (PMML44Super.CorrelationValues method), 69
 insert_Extension_at() (PMML44Super.COUNT_TABLE_TYPE
 method), 40
 insert_Extension_at() (PMML44Super.Counts
 method), 70
 insert_Extension_at() (PMML44Super.Covariances method), 71
 insert_Extension_at() (PMML44Super.CovariateList method), 72
 insert_Extension_at() (PMML44Super.DataDictionary method),
 72
 insert_Extension_at() (PMML44Super.DataField method), 74
 insert_Extension_at() (PMML44Super.Decision
 method), 75
 insert_Extension_at() (PMML44Super.Decisions method), 77
 insert_Extension_at() (PMML44Super.DecisionTree method), 76
 insert_Extension_at() (PMML44Super.DeepNetwork method),
 78
 insert_Extension_at() (PMML44Super.DefineFunction method),
 80
 insert_Extension_at() (PMML44Super.Delimiter method), 81
 insert_Extension_at() (PMML44Super.Denominator method), 82
 insert_Extension_at() (PMML44Super.DerivedField method), 83
 insert_Extension_at() (PMML44Super.DiscreteConditionalProbability
 method), 86
 insert_Extension_at() (PMML44Super.DiscreteNode method), 87
 insert_Extension_at() (PMML44Super.Discretize method), 88
 insert_Extension_at() (PMML44Super.DiscretizeBin method), 89
 insert_Extension_at() (PMML44Super.DiscrStats method), 85
 insert_Extension_at() (PMML44Super.DocumentTermMatrix
 method), 89
 insert_Extension_at() (PMML44Super.DynamicRegressor method),
 90
 insert_Extension_at() (PMML44Super.euclidean method), 301
 insert_Extension_at() (PMML44Super.EventValues method), 91
 insert_Extension_at() (PMML44Super.FactorList method), 94
 insert_Extension_at() (PMML44Super.False_
 method), 94
 insert_Extension_at() (PMML44Super.FieldColumnPair method),
 95
 insert_Extension_at() (PMML44Super.FieldRef
 method), 96
 insert_Extension_at() (PMML44Super.FieldValue method), 97
 insert_Extension_at() (PMML44Super.FieldValueCount method),
 98
 insert_Extension_at() (PMML44Super.GARCH
 method), 101
 insert_Extension_at() (PMML44Super.GARCHPart method), 102
 insert_Extension_at() (PMML44Super.GaussianDistribution

`method)`, 103
`insert_Extension_at()`
 (`PMML44Super.GaussianProcessModel`
 `method)`, 105
`insert_Extension_at()`
 (`PMML44Super.GeneralizedExponentialKernel`
 `method)`, 109
`insert_Extension_at()`
 (`PMML44Super.GeneralRegressionModel`
 `method)`, 107
`insert_Extension_at()` (`PMML44Super.Header`
 `method)`, 111
`insert_Extension_at()`
 (`PMML44Super.InlineTable` `method)`, 113
`insert_Extension_at()`
 (`PMML44Super.InstanceField` `method)`, 114
`insert_Extension_at()`
 (`PMML44Super.InstanceFields` `method)`,
 115
`insert_Extension_at()` (`PMML44Super.Interval`
 `method)`, 116
`insert_Extension_at()` (`PMML44Super.Item`
 `method)`, 117
`insert_Extension_at()` (`PMML44Super.ItemRef`
 `method)`, 117
`insert_Extension_at()` (`PMML44Super.Itemset`
 `method)`, 118
`insert_Extension_at()` (`PMML44Super.jaccard`
 `method)`, 301
`insert_Extension_at()`
 (`PMML44Super.KNNInput` `method)`, 119
`insert_Extension_at()`
 (`PMML44Super.KNNInputs` `method)`, 120
`insert_Extension_at()`
 (`PMML44Super.KohonenMap` `method)`, 121
`insert_Extension_at()` (`PMML44Super.Lag`
 `method)`, 122
`insert_Extension_at()` (`PMML44Super.Lambda`
 `method)`, 123
`insert_Extension_at()`
 (`PMML44Super.LayerBias` `method)`, 124
`insert_Extension_at()`
 (`PMML44Super.LayerParameters` `method)`,
 126
`insert_Extension_at()`
 (`PMML44Super.LayerWeights` `method)`, 129
`insert_Extension_at()` (`PMML44Super.LiftData`
 `method)`, 130
`insert_Extension_at()`
 (`PMML44Super.LiftGraph` `method)`, 131
`insert_Extension_at()`
 (`PMML44Super.LinearKernelType` `method)`,
 132
`insert_Extension_at()`
 (`PMML44Super.LinearNorm` `method)`, 132
`insert_Extension_at()`
 (`PMML44Super.LocalTransformations`
 `method)`, 133
`insert_Extension_at()`
 (`PMML44Super.LognormalDistributionForBN`
 `method)`, 134
`insert_Extension_at()` (`PMML44Super.Losses`
 `method)`, 135
`insert_Extension_at()` (`PMML44Super.Lower`
 `method)`, 135
`insert_Extension_at()` (`PMML44Super.MA`
 `method)`, 136
`insert_Extension_at()`
 (`PMML44Super.MACoefficients` `method)`,
 137
`insert_Extension_at()`
 (`PMML44Super.MapValues` `method)`, 138
`insert_Extension_at()` (`PMML44Super.Mean`
 `method)`, 142
`insert_Extension_at()`
 (`PMML44Super.MeasurementMatrix` `method)`,
 142
`insert_Extension_at()` (`PMML44Super.Metrics`
 `method)`, 143
`insert_Extension_at()`
 (`PMML44Super.MiningBuildTask` `method)`,
 144
`insert_Extension_at()`
 (`PMML44Super.MiningField` `method)`, 145
`insert_Extension_at()`
 (`PMML44Super.MiningModel` `method)`, 147
`insert_Extension_at()`
 (`PMML44Super.MiningSchema` `method)`,
 148
`insert_Extension_at()`
 (`PMML44Super.minkowski` `method)`, 302
`insert_Extension_at()`
 (`PMML44Super.MissingValueWeights` `method)`,
 148
`insert_Extension_at()`
 (`PMML44Super.ModelExplanation` `method)`,
 149
`insert_Extension_at()`
 (`PMML44Super.ModelLiftGraph` `method)`,
 150
`insert_Extension_at()`
 (`PMML44Super.ModelStats` `method)`, 151
`insert_Extension_at()`
 (`PMML44Super.ModelVerification` `method)`,
 152
`insert_Extension_at()`
 (`PMML44Super.MultivariateStat` `method)`,
 153

[insert_Extension_at\(\)](#)
 (PMML44Super.MultivariateStats method), [154](#)
[insert_Extension_at\(\)](#) (PMML44Super.Nadam method), [155](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NaiveBayesModel method), [156](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NearestNeighborModel method), [158](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NetworkLayer method), [160](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NeuralInput method), [160](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NeuralInputs method), [161](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NeuralLayer method), [162](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NeuralNetwork method), [164](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NeuralOutput method), [165](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NeuralOutputs method), [166](#)
[insert_Extension_at\(\)](#) (PMML44Super.Neuron method), [167](#)
[insert_Extension_at\(\)](#) (PMML44Super.Node method), [168](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NonseasonalComponent method), [169](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NonseasonalFactor method), [170](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NormalDistributionForBN method), [173](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NormContinuous method), [171](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NormDiscrete method), [172](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.Numerator method), [174](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NumericInfo method), [175](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.NumericPredictor method), [176](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.Optimizers method), [177](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.OptimumLiftGraph method), [178](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.OutlierEffect method), [179](#)
[insert_Extension_at\(\)](#) (PMML44Super.Output method), [180](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.OutputField method), [181](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PairCounts method), [191](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.Parameter method), [192](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.ParameterList method), [194](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.ParamMatrix method), [192](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.ParentValue method), [195](#)
[insert_Extension_at\(\)](#) (PMML44Super.Partition method), [195](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PartitionFieldStats method), [196](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PastVariances method), [197](#)
[insert_Extension_at\(\)](#) (PMML44Super.PCell method), [183](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PCovCell method), [184](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PCovMatrix method), [185](#)
[insert_Extension_at\(\)](#) (PMML44Super.PMML method), [187](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PoissonDistribution method), [198](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PolynomialKernelType method), [199](#)
[insert_Extension_at\(\)](#) (PMML44Super.PPCell method), [189](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PPMatrix method), [190](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.PredictiveModelQuality method), [200](#)
[insert_Extension_at\(\)](#)
 (PMML44Super.Predictor method), [202](#)

`insert_Extension_at()` (*PMML44Super.PredictorTerm method*), 203
`insert_Extension_at()` (*PMML44Super.PsiVector method*), 204
`insert_Extension_at()` (*PMML44Super.Quantile method*), 204
`insert_Extension_at()` (*PMML44Super.RadialBasisKernel method*), 209
`insert_Extension_at()` (*PMML44Super.RadialBasisKernelType method*), 210
`insert_Extension_at()` (*PMML44Super.RandomLiftGraph method*), 210
`insert_Extension_at()` (*PMML44Super.Regression method*), 211
`insert_Extension_at()` (*PMML44Super.RegressionModel method*), 213
`insert_Extension_at()` (*PMML44Super.RegressionTable method*), 214
`insert_Extension_at()` (*PMML44Super.RegressorValues method*), 215
`insert_Extension_at()` (*PMML44Super.Residuals method*), 217
`insert_Extension_at()` (*PMML44Super.ResidualSquareCoefficients method*), 216
`insert_Extension_at()` (*PMML44Super.ResultField method*), 218
`insert_Extension_at()` (*PMML44Super.RMSprop method*), 206
`insert_Extension_at()` (*PMML44Super.ROC method*), 207
`insert_Extension_at()` (*PMML44Super.ROCGraph method*), 208
`insert_Extension_at()` (*PMML44Super.RuleSelectionMethod method*), 219
`insert_Extension_at()` (*PMML44Super.RuleSet method*), 220
`insert_Extension_at()` (*PMML44Super.RuleSetModel method*), 221
`insert_Extension_at()` (*PMML44Super.Scorecard method*), 224
`insert_Extension_at()` (*PMML44Super.ScoreDistribution method*), 223
`insert_Extension_at()` (*PMML44Super.SeasonalComponent method*), 226
`insert_Extension_at()` (*PMML44Super.SeasonalFactor method*), 227
`insert_Extension_at()` (*PMML44Super.Segment method*), 230
`insert_Extension_at()` (*PMML44Super.Segmentation method*), 231
`insert_Extension_at()` (*PMML44Super.Sequence method*), 232
`insert_Extension_at()` (*PMML44Super.SequenceModel method*), 234
`insert_Extension_at()` (*PMML44Super.SequenceReference method*), 235
`insert_Extension_at()` (*PMML44Super.SequenceRule method*), 236
`insert_Extension_at()` (*PMML44Super.SetPredicate method*), 237
`insert_Extension_at()` (*PMML44Super.SetReference method*), 238
`insert_Extension_at()` (*PMML44Super.SGD method*), 222
`insert_Extension_at()` (*PMML44Super.SigmoidKernelType method*), 239
`insert_Extension_at()` (*PMML44Super.simpleMatching method*), 303
`insert_Extension_at()` (*PMML44Super.SimplePredicate method*), 240
`insert_Extension_at()` (*PMML44Super.SimpleRule method*), 241
`insert_Extension_at()` (*PMML44Super.SimpleSetPredicate method*), 242
`insert_Extension_at()` (*PMML44Super.squaredEuclidean method*), 304
`insert_Extension_at()` (*PMML44Super.StateSpaceModel method*), 244
`insert_Extension_at()` (*PMML44Super.StateVector method*), 244
`insert_Extension_at()` (*PMML44Super.SupportVector method*), 245
`insert_Extension_at()` (*PMML44Super.SupportVectorMachine method*), 246

`insert_Extension_at()` (*PMML44Super.SupportVectorMachineModel method*), 248
`insert_Extension_at()` (*PMML44Super.SupportVectors method*), 249
`insert_Extension_at()` (*PMML44Super.TableLocator method*), 250
`insert_Extension_at()` (*PMML44Super.tanimoto method*), 304
`insert_Extension_at()` (*PMML44Super.Target method*), 251
`insert_Extension_at()` (*PMML44Super.Targets method*), 256
`insert_Extension_at()` (*PMML44Super.TargetValue method*), 252
`insert_Extension_at()` (*PMML44Super.TargetValueCount method*), 253
`insert_Extension_at()` (*PMML44Super.TargetValueCounts method*), 254
`insert_Extension_at()` (*PMML44Super.TargetValueStat method*), 254
`insert_Extension_at()` (*PMML44Super.TargetValueStats method*), 255
`insert_Extension_at()` (*PMML44Super.Taxonomy method*), 257
`insert_Extension_at()` (*PMML44Super.TestDistributions method*), 258
`insert_Extension_at()` (*PMML44Super.TextCorpus method*), 259
`insert_Extension_at()` (*PMML44Super.TextDictionary method*), 259
`insert_Extension_at()` (*PMML44Super.TextDocument method*), 260
`insert_Extension_at()` (*PMML44Super.TextIndex method*), 262
`insert_Extension_at()` (*PMML44Super.TextIndexNormalization method*), 263
`insert_Extension_at()` (*PMML44Super.TextModel method*), 265
`insert_Extension_at()` (*PMML44Super.TextModelNormalization method*), 266
`insert_Extension_at()` (*PMML44Super.TextModelSimiliarity method*), 267
`insert_Extension_at()` (*PMML44Super.Time method*), 269
`insert_Extension_at()` (*PMML44Super.TimeSeriesModel method*), 274
`insert_Extension_at()` (*PMML44Super.Timestamp method*), 276
`insert_Extension_at()` (*PMML44Super.TrainingInstances method*), 277
`insert_Extension_at()` (*PMML44Super.TrainingParameters method*), 278
`insert_Extension_at()` (*PMML44Super.TransformationDictionary method*), 279
`insert_Extension_at()` (*PMML44Super.TransitionMatrix method*), 280
`insert_Extension_at()` (*PMML44Super.TreeModel method*), 281
`insert_Extension_at()` (*PMML44Super.TrendCoefficients method*), 282
`insert_Extension_at()` (*PMML44Super.TriangularDistributionForBN method*), 284
`insert_Extension_at()` (*PMML44Super.True_ method*), 284
`insert_Extension_at()` (*PMML44Super.UniformDistribution method*), 285
`insert_Extension_at()` (*PMML44Super.UniformDistributionForBN method*), 286
`insert_Extension_at()` (*PMML44Super.UnivariateStats method*), 287
`insert_Extension_at()` (*PMML44Super.Upper method*), 288
`insert_Extension_at()` (*PMML44Super.Value method*), 289
`insert_Extension_at()` (*PMML44Super.ValueProbability method*), 289
`insert_Extension_at()` (*PMML44Super.VariableWeight method*), 290
`insert_Extension_at()` (*PMML44Super.Variance method*), 291
`insert_Extension_at()` (*PMML44Super.VarianceCoefficients method*), 292
`insert_Extension_at()`

(PMML44Super.VectorDictionary method), 293
 insert_Extension_at() (PMML44Super.VectorFields method), 294
 insert_Extension_at() (PMML44Super.VectorInstance method), 295
 insert_Extension_at() (PMML44Super.VerificationField method), 296
 insert_Extension_at() (PMML44Super.VerificationFields method), 296
 insert_Extension_at() (PMML44Super.XCoordinates method), 297
 insert_Extension_at() (PMML44Super.YCoordinates method), 298
 insert_False_at() (PMML44Super.CompoundPredicate method), 57
 insert_FieldColumnPair_at() (PMML44Super.ChildParent method), 45
 insert_FieldColumnPair_at() (PMML44Super.MapValues method), 138
 insert_FieldRef_at() (PMML44Super.Apply method), 21
 insert_FieldRef_at() (PMML44Super.Baseline method), 29
 insert_FieldRef_at() (PMML44Super.PredictorTerm method), 203
 insert_FieldRef_at() (PMML44Super.VectorFields method), 294
 insert_FieldValue_at() (PMML44Super.COUNT_TABLE_TYPE method), 40
 insert_FieldValue_at() (PMML44Super.FieldValue method), 97
 insert_FieldValueCount_at() (PMML44Super.COUNT_TABLE_TYPE method), 40
 insert_FieldValueCount_at() (PMML44Super.FieldValue method), 97
 insert_GaussianProcessModel_at() (PMML44Super.PMML method), 187
 insert_GeneralRegressionModel_at() (PMML44Super.PMML method), 187
 insert_InstanceField_at() (PMML44Super.InstanceFields method), 115
 insert_Interval_at() (PMML44Super.ContStats method), 63
 insert_Interval_at() (PMML44Super.DataField method), 74
 insert_Interval_at() (PMML44Super.EventValues method), 91
 insert_Item_at() (PMML44Super.AssociationModel method), 24
 insert_Item_at() (PMML44Super.SequenceModel method), 234
 insert_ItemRef_at() (PMML44Super.Itemset method), 118
 insert_Itemset_at() (PMML44Super.AssociationModel method), 24
 insert_Itemset_at() (PMML44Super.SequenceModel method), 234
 insert_KNNInput_at() (PMML44Super.KNNInputs method), 120
 insert_Lag_at() (PMML44Super.Apply method), 21
 insert_Lambda_at() (PMML44Super.AbsoluteExponentialKernel method), 7
 insert_Lambda_at() (PMML44Super.ARDSquaredExponentialKernel method), 4
 insert_Lambda_at() (PMML44Super.GeneralizedExponentialKernel method), 109
 insert_LiftData_at() (PMML44Super.PredictiveModelQuality method), 200
 insert_LinearNorm_at() (PMML44Super.NormContinuous method), 171
 insert_MapValues_at() (PMML44Super.Apply method), 21
 insert_MatCell_at() (PMML44Super.Matrix method), 140
 insert_MiningField_at() (PMML44Super.MiningSchema method), 148
 insert_MiningModel_at() (PMML44Super.PMML method), 187
 insert_MultivariateStat_at() (PMML44Super.MultivariateStats method), 154
 insert_MultivariateStats_at() (PMML44Super.ModelStats method), 151
 insert_Nadam_at() (PMML44Super.Optimizers method), 177
 insert_NaiveBayesModel_at() (PMML44Super.PMML method), 187
 insert_NearestNeighborModel_at() (PMML44Super.PMML method), 187
 insert_NetworkLayer_at()

(PMML44Super.DeepNetwork method), 79
 insert_NeuralInput_at() (PMML44Super.NeuralInputs method), 161
 insert_NeuralLayer_at() (PMML44Super.NeuralNetwork method), 164
 insert_NeuralNetwork_at() (PMML44Super.PMML method), 187
 insert_NeuralOutput_at() (PMML44Super.NeuralOutputs method), 166
 insert_Neuron_at() (PMML44Super.NeuralLayer method), 162
 insert_Node_at() (PMML44Super.Node method), 168
 insert_NormContinuous_at() (PMML44Super.Apply method), 21
 insert_NormDiscrete_at() (PMML44Super.Apply method), 21
 insert_NUM_ARRAY_at() (PMML44Super.ContStats method), 63
 insert_NumericPredictor_at() (PMML44Super.RegressionTable method), 214
 insert_OutlierEffect_at() (PMML44Super.ARIMA method), 5
 insert_OutputField_at() (PMML44Super.Output method), 180
 insert_PairCounts_at() (PMML44Super.BayesInput method), 33
 insert_Parameter_at() (PMML44Super.ParameterList method), 194
 insert_ParameterField_at() (PMML44Super.DefineFunction method), 80
 insert_ParentValue_at() (PMML44Super.ContinuousConditionalProbability method), 64
 insert_ParentValue_at() (PMML44Super.DiscreteConditionalProbability method), 86
 insert_PartitionFieldStats_at() (PMML44Super.Partition method), 196
 insert_PCell_at() (PMML44Super.ParamMatrix method), 192
 insert_PCovCell_at() (PMML44Super.PCovMatrix method), 185
 insert_PPCell_at() (PMML44Super.PPMatrix method), 190
 insert_PredictiveModelQuality_at() (PMML44Super.ModelExplanation method), 149
 insert_Predictor_at() (PMML44Super.CovariateList method), 72
 insert_Predictor_at() (PMML44Super.FactorList method), 94
 insert_PredictorTerm_at() (PMML44Super.RegressionTable method), 215
 insert_Quantile_at() (PMML44Super.NumericInfo method), 175
 insert_Regression_at() (PMML44Super.MiningModel method), 147
 insert_RegressionModel_at() (PMML44Super.PMML method), 187
 insert_RegressionTable_at() (PMML44Super.Regression method), 211
 insert_RegressionTable_at() (PMML44Super.RegressionModel method), 213
 insert_ResultField_at() (PMML44Super.DecisionTree method), 76
 insert_ResultField_at() (PMML44Super.Regression method), 211
 insert_RMSprop_at() (PMML44Super.Optimizers method), 177
 insert_row_at() (PMML44Super.InlineTable method), 113
 insert_RuleSelectionMethod_at() (PMML44Super.RuleSet method), 220
 insert_RuleSetModel_at() (PMML44Super.PMML method), 187
 insert_Scorecard_at() (PMML44Super.PMML method), 187
 insert_ScoreDistribution_at() (PMML44Super.Node method), 168
 insert_ScoreDistribution_at() (PMML44Super.RuleSet method), 220
 insert_ScoreDistribution_at() (PMML44Super.SimpleRule method), 241
 insert_Segment_at() (PMML44Super.Segmentation method), 231
 insert_Sequence_at() (PMML44Super.SequenceModel method), 234
 insert_SequenceModel_at() (PMML44Super.PMML method), 187
 insert_SequenceRule_at() (PMML44Super.SequenceModel method), 234
 insert_SetPredicate_at() (PMML44Super.SequenceModel method), 234
 insert_SetReference_at() (PMML44Super.Sequence method), 232
 insert_SGD_at() (PMML44Super.Optimizers

`method`), 177
`insert_SimplePredicate_at()`
 (*PMML44Super.CompoundPredicate method*), 57
`insert_SimpleRule_at()`
 (*PMML44Super.CompoundRule method*), 58
`insert_SimpleRule_at()`
 (*PMML44Super.RuleSet method*), 220
`insert_SimpleSetPredicate_at()`
 (*PMML44Super.CompoundPredicate method*), 57
`insert_SupportVector_at()`
 (*PMML44Super.SupportVectors method*), 249
`insert_SupportVectorMachine_at()`
 (*PMML44Super.SupportVectorMachineModel method*), 248
`insert_SupportVectorMachineModel_at()`
 (*PMML44Super.PMML method*), 187
`insert_Target_at()` (*PMML44Super.Targets method*), 256
`insert_TargetValue_at()`
 (*PMML44Super.Target method*), 251
`insert_TargetValueCount_at()`
 (*PMML44Super.TargetValueCounts method*), 254
`insert_TargetValueStat_at()`
 (*PMML44Super.TargetValueStats method*), 255
`insert_Taxonomy_at()`
 (*PMML44Super.DataDictionary method*), 73
`insert_TextDocument_at()`
 (*PMML44Super.TextCorpus method*), 259
`insert_TextIndex_at()` (*PMML44Super.Apply method*), 22
`insert_TextIndexNormalization_at()`
 (*PMML44Super.TextIndex method*), 262
`insert_TextModel_at()` (*PMML44Super.PMML method*), 187
`insert_Theta_at()` (*PMML44Super.FinalTheta method*), 101
`insert_TimeCycle_at()`
 (*PMML44Super.TimeAnchor method*), 270
`insert_TimeException_at()`
 (*PMML44Super.TimeAnchor method*), 270
`insert_TimeSeries_at()`
 (*PMML44Super.TimeSeriesModel method*), 274
`insert_TimeSeriesModel_at()`
 (*PMML44Super.PMML method*), 187
`insert_TimeValue_at()`
 (*PMML44Super.ExponentialSmoothing method*), 92
`insert_TimeValue_at()`
 (*PMML44Super.TimeSeries method*), 272
`insert_TreeModel_at()` (*PMML44Super.PMML method*), 187
`insert_True_at()` (*PMML44Super.CompoundPredicate method*), 57
`insert_UnivariateStats_at()`
 (*PMML44Super.ModelStats method*), 151
`insert_Value_at()` (*PMML44Super.DataField method*), 74
`insert_Value_at()` (*PMML44Super.DerivedField method*), 83
`insert_Value_at()` (*PMML44Super.EventValues method*), 91
`insert_ValueProbability_at()`
 (*PMML44Super.DiscreteConditionalProbability method*), 86
`insert_ValueProbability_at()`
 (*PMML44Super.DiscreteNode method*), 87
`insert_VectorInstance_at()`
 (*PMML44Super.VectorDictionary method*), 293
`insert_VerificationField_at()`
 (*PMML44Super.VerificationFields method*), 296
`InstanceField` (*class in PMML44Super*), 114
`InstanceFields` (*class in PMML44Super*), 114
`INT_Entries` (*class in PMML44Super*), 111
`INT_SparseArray` (*class in PMML44Super*), 112
`Interval` (*class in PMML44Super*), 115
`Item` (*class in PMML44Super*), 116
`ItemRef` (*class in PMML44Super*), 117
`Itemset` (*class in PMML44Super*), 118
J
`jaccard` (*class in PMML44Super*), 301
K
`KalmanState` (*class in PMML44Super*), 120
`KNNInput` (*class in PMML44Super*), 118
`KNNInputs` (*class in PMML44Super*), 119
`KohonenMap` (*class in PMML44Super*), 121
L
`Lag` (*class in PMML44Super*), 121
`Lambda` (*class in PMML44Super*), 122
`LayerBias` (*class in PMML44Super*), 123
`LayerParameters` (*class in PMML44Super*), 124
`LayerWeights` (*class in PMML44Super*), 128
`Level` (*class in PMML44Super*), 129
`LiftData` (*class in PMML44Super*), 129
`LiftGraph` (*class in PMML44Super*), 130
`LinearKernelType` (*class in PMML44Super*), 131

LinearNorm (class in *PMML44Super*), 132
 LocalTransformations (class in *PMML44Super*), 133
 LognormalDistributionForBN (class in *PMML44Super*), 133
 Losses (class in *PMML44Super*), 134
 Lower (class in *PMML44Super*), 135

M

MA (class in *PMML44Super*), 136
 MACoefficients (class in *PMML44Super*), 137
 MapValues (class in *PMML44Super*), 137
 MatCell (class in *PMML44Super*), 138
 Matrix (class in *PMML44Super*), 139
 MaximumLikelihoodStat (class in *PMML44Super*), 140
 Mean (class in *PMML44Super*), 141
 MeasurementMatrix (class in *PMML44Super*), 142
 metadata (module), 313
 Metrics (class in *PMML44Super*), 143
 MiningBuildTask (class in *PMML44Super*), 143
 MiningField (class in *PMML44Super*), 144
 MiningModel (class in *PMML44Super*), 146
 MiningSchema (class in *PMML44Super*), 147
 minkowski (class in *PMML44Super*), 301
 MissingValueWeights (class in *PMML44Super*), 148
 ModelExplanation (class in *PMML44Super*), 149
 ModelLiftGraph (class in *PMML44Super*), 150
 ModelStats (class in *PMML44Super*), 150
 ModelVerification (class in *PMML44Super*), 151
 MultivariateStat (class in *PMML44Super*), 152
 MultivariateStats (class in *PMML44Super*), 154

N

Nadam (class in *PMML44Super*), 154
 NaiveBayesModel (class in *PMML44Super*), 155
 NearestNeighborModel (class in *PMML44Super*), 157
 NetworkLayer (class in *PMML44Super*), 159
 NeuralInput (class in *PMML44Super*), 160
 NeuralInputs (class in *PMML44Super*), 161
 NeuralLayer (class in *PMML44Super*), 161
 NeuralNetwork (class in *PMML44Super*), 163
 NeuralOutput (class in *PMML44Super*), 165
 NeuralOutputs (class in *PMML44Super*), 165
 Neuron (class in *PMML44Super*), 166
 Node (class in *PMML44Super*), 167
 NonseasonalComponent (class in *PMML44Super*), 169
 NonseasonalFactor (class in *PMML44Super*), 170
 NormalDistributionForBN (class in *PMML44Super*), 172
 NormContinuous (class in *PMML44Super*), 171

NormDiscrete (class in *PMML44Super*), 172
 Numerator (class in *PMML44Super*), 173
 NumericInfo (class in *PMML44Super*), 174
 NumericPredictor (class in *PMML44Super*), 175

O

Optimizers (class in *PMML44Super*), 176
 OptimumLiftGraph (class in *PMML44Super*), 178
 OutlierEffect (class in *PMML44Super*), 178
 Output (class in *PMML44Super*), 179
 OutputField (class in *PMML44Super*), 180

P

PairCounts (class in *PMML44Super*), 190
 Parameter (class in *PMML44Super*), 192
 ParameterField (class in *PMML44Super*), 193
 ParameterList (class in *PMML44Super*), 193
 ParamMatrix (class in *PMML44Super*), 191
 ParentValue (class in *PMML44Super*), 194
 Partition (class in *PMML44Super*), 195
 PartitionFieldStats (class in *PMML44Super*), 196
 PastVariances (class in *PMML44Super*), 197
 PCell (class in *PMML44Super*), 182
 PCovCell (class in *PMML44Super*), 183
 PCovMatrix (class in *PMML44Super*), 184
 PMML (class in *PMML44Super*), 185
 PMML44Super (module), 3
 PoissonDistribution (class in *PMML44Super*), 197
 PolynomialKernelType (class in *PMML44Super*), 198
 PPCell (class in *PMML44Super*), 189
 PPMatrix (class in *PMML44Super*), 190
 PredictiveModelQuality (class in *PMML44Super*), 199
 Predictor (class in *PMML44Super*), 201
 PredictorTerm (class in *PMML44Super*), 202
 PsiVector (class in *PMML44Super*), 203

Q

Quantile (class in *PMML44Super*), 204

R

RadialBasisKernel (class in *PMML44Super*), 208
 RadialBasisKernelType (class in *PMML44Super*), 209
 RandomLiftGraph (class in *PMML44Super*), 210
 REAL_Entries (class in *PMML44Super*), 205
 REAL_SparseArray (class in *PMML44Super*), 205
 Regression (class in *PMML44Super*), 211
 RegressionModel (class in *PMML44Super*), 212
 RegressionTable (class in *PMML44Super*), 214

RegressorValues (class in PMML44Super), 215
replace_Adadelta_at ()
 (PMML44Super.Optimizers method), 177
replace_Adagrad_at ()
 (PMML44Super.Optimizers method), 177
replace_Adam_at () (PMML44Super.Optimizers
 method), 177
replace_Adamax_at () (PMML44Super.Optimizers
 method), 177
replace_Aggregate_at () (PMML44Super.Apply
 method), 22
replace_Annotation_at ()
 (PMML44Super.Header method), 111
replace_AnomalyDetectionModel_at ()
 (PMML44Super.PMML method), 187
replace_AnovaRow_at () (PMML44Super.Anova
 method), 17
replace_Apply_at () (PMML44Super.Apply
 method), 22
replace_Array_at () (PMML44Super.DiscrStats
 method), 85
replace_Array_at () (PMML44Super.Matrix
 method), 140
replace_Array_at ()
 (PMML44Super.PartitionFieldStats method),
 196
replace_AssociationModel_at ()
 (PMML44Super.PMML method), 187
replace_AssociationRule_at ()
 (PMML44Super.AssociationModel method), 24
replace_Attribute_at ()
 (PMML44Super.Characteristic method),
 43
replace_BaselineCell_at ()
 (PMML44Super.BaseCumHazardTables
 method), 28
replace_BaselineCell_at ()
 (PMML44Super.BaselineStratum method),
 32
replace_BaselineModel_at ()
 (PMML44Super.PMML method), 187
replace_BaselineStratum_at ()
 (PMML44Super.BaseCumHazardTables
 method), 28
replace_BayesianNetworkModel_at ()
 (PMML44Super.PMML method), 188
replace_BayesInput_at ()
 (PMML44Super.BayesInputs method), 34
replace_BlockIndicator_at ()
 (PMML44Super.Lag method), 122
replace_CategoricalPredictor_at ()
 (PMML44Super.RegressionTable method),
 215
replace_CategoricalPredictor_at ()
 (PMML44Super.VectorFields method), 294
replace_Category_at ()
 (PMML44Super.Categories method), 42
replace_Characteristic_at ()
 (PMML44Super.Characteristics method),
 44
replace_ChildParent_at ()
 (PMML44Super.Taxonomy method), 257
replace_Cluster_at ()
 (PMML44Super.ClusteringModel method),
 49
replace_ClusteringField_at ()
 (PMML44Super.ClusteringModel method),
 49
replace_ClusteringModel_at ()
 (PMML44Super.PMML method), 188
replace_ClusteringModelQuality_at ()
 (PMML44Super.ModelExplanation method),
 149
replace_Coefficient_at ()
 (PMML44Super.Coefficients method), 52
replace_CompoundPredicate_at ()
 (PMML44Super.CompoundPredicate method),
 57
replace_CompoundRule_at ()
 (PMML44Super.CompoundRule method),
 58
replace_CompoundRule_at ()
 (PMML44Super.RuleSet method), 220
replace_Con_at () (PMML44Super.Neuron
 method), 167
replace_Constant_at () (PMML44Super.Apply
 method), 22
replace_ContinuousConditionalProbability_at ()
 (PMML44Super.ContinuousNode method), 66
replace_ContinuousDistribution_at ()
 (PMML44Super.ContinuousConditionalProbability
 method), 64
replace_ContinuousDistribution_at ()
 (PMML44Super.ContinuousNode method), 66
replace_ContinuousNode_at ()
 (PMML44Super.BayesianNetworkNodes
 method), 37
replace_DataField_at ()
 (PMML44Super.DataDictionary method),
 73
replace_Decision_at ()
 (PMML44Super.Decisions method), 77
replace_DecisionTree_at ()
 (PMML44Super.MiningModel method), 147
replace_DeepNetwork_at ()
 (PMML44Super.PMML method), 188
replace_DefineFunction_at ()
 (PMML44Super.TransformationDictionary

`method)`, 279
`replace_Delimiter_at()`
 (*PMML44Super.Sequence method*), 232
`replace_DerivedField_at()`
 (*PMML44Super.ContinuousNode method*), 67
`replace_DerivedField_at()`
 (*PMML44Super.DiscreteNode method*), 87
`replace_DerivedField_at()`
 (*PMML44Super.LocalTransformations method*), 133
`replace_DerivedField_at()`
 (*PMML44Super.TransformationDictionary method*), 279
`replace_DiscreteConditionalProbability_at()`
 (*PMML44Super.DiscreteNode method*), 87
`replace_DiscreteNode_at()`
 (*PMML44Super.BayesianNetworkNodes method*), 37
`replace_Discretize_at()` (*PMML44Super.Apply method*), 22
`replace_DiscretizeBin_at()`
 (*PMML44Super.Discretize method*), 88
`replace_DynamicRegressor_at()`
 (*PMML44Super.ARIMA method*), 5
`replace_DynamicRegressor_at()`
 (*PMML44Super.StateSpaceModel method*), 244
`replace_Extension_at()`
 (*PMML44Super.AbsoluteExponentialKernel method*), 7
`replace_Extension_at()`
 (*PMML44Super.Adadelta method*), 8
`replace_Extension_at()`
 (*PMML44Super.Adagrad method*), 9
`replace_Extension_at()` (*PMML44Super.Adam method*), 10
`replace_Extension_at()`
 (*PMML44Super.Adamax method*), 11
`replace_Extension_at()`
 (*PMML44Super.Aggregate method*), 12
`replace_Extension_at()`
 (*PMML44Super.Annotation method*), 13
`replace_Extension_at()`
 (*PMML44Super.AnomalyDetectionModel method*), 15
`replace_Extension_at()` (*PMML44Super.Anova method*), 17
`replace_Extension_at()`
 (*PMML44Super.AnovaRow method*), 18
`replace_Extension_at()`
 (*PMML44Super.AntecedentSequence method*), 18
`replace_Extension_at()`
 (*PMML44Super.AnyDistribution method*), 19
`replace_Extension_at()`
 (*PMML44Super.Application method*), 20
`replace_Extension_at()` (*PMML44Super.Apply method*), 22
`replace_Extension_at()` (*PMML44Super.AR method*), 3
`replace_Extension_at()`
 (*PMML44Super.ARDSquaredExponentialKernel method*), 4
`replace_Extension_at()`
 (*PMML44Super.ARIMA method*), 5
`replace_Extension_at()`
 (*PMML44Super.ARMAPart method*), 6
`replace_Extension_at()`
 (*PMML44Super.AssociationModel method*), 24
`replace_Extension_at()`
 (*PMML44Super.AssociationRule method*), 26
`replace_Extension_at()`
 (*PMML44Super.Attribute method*), 27
`replace_Extension_at()`
 (*PMML44Super.BaseCumHazardTables method*), 28
`replace_Extension_at()`
 (*PMML44Super.BaselineCell method*), 30
`replace_Extension_at()`
 (*PMML44Super.BaselineModel method*), 31
`replace_Extension_at()`
 (*PMML44Super.BaselineStratum method*), 32
`replace_Extension_at()`
 (*PMML44Super.BayesianNetworkModel method*), 36
`replace_Extension_at()`
 (*PMML44Super.BayesianNetworkNodes method*), 37
`replace_Extension_at()`
 (*PMML44Super.BayesInput method*), 33
`replace_Extension_at()`
 (*PMML44Super.BayesInputs method*), 34
`replace_Extension_at()`
 (*PMML44Super.BayesOutput method*), 35
`replace_Extension_at()`
 (*PMML44Super.binarySimilarity method*), 299
`replace_Extension_at()`
 (*PMML44Super.BlockIndicator method*), 38
`replace_Extension_at()`
 (*PMML44Super.BoundaryValueMeans method*), 39

```

replace_Extension_at ()
    (PMML44Super.BoundaryValues method), 39
replace_Extension_at ()
    (PMML44Super.CategoricalPredictor method), 41
replace_Extension_at ()
    (PMML44Super.Categories method), 42
replace_Extension_at ()
    (PMML44Super.Category method), 42
replace_Extension_at ()
    (PMML44Super.Characteristic method), 43
replace_Extension_at ()
    (PMML44Super.Characteristics method), 44
replace_Extension_at ()
    (PMML44Super.chebychev method), 299
replace_Extension_at ()
    (PMML44Super.ChildParent method), 45
replace_Extension_at ()
    (PMML44Super.cityBlock method), 300
replace_Extension_at ()
    (PMML44Super.ClassLabels method), 46
replace_Extension_at () (PMML44Super.Cluster
    method), 47
replace_Extension_at ()
    (PMML44Super.ClusteringField method), 48
replace_Extension_at ()
    (PMML44Super.ClusteringModel method), 49
replace_Extension_at ()
    (PMML44Super.ClusteringModelQuality
    method), 50
replace_Extension_at ()
    (PMML44Super.Coefficient method), 51
replace_Extension_at ()
    (PMML44Super.Coefficients method), 52
replace_Extension_at ()
    (PMML44Super.ComparisonMeasure method), 53
replace_Extension_at ()
    (PMML44Super.Comparisons method), 54
replace_Extension_at ()
    (PMML44Super.ComplexPartialScore method), 55
replace_Extension_at ()
    (PMML44Super.CompoundPredicate method), 57
replace_Extension_at ()
    (PMML44Super.CompoundRule method), 58
replace_Extension_at () (PMML44Super.Con
    method), 59
replace_Extension_at ()
    (PMML44Super.ConfusionMatrix method), 60
replace_Extension_at ()
    (PMML44Super.ConsequentSequence method), 60
replace_Extension_at ()
    (PMML44Super.Constraints method), 62
replace_Extension_at ()
    (PMML44Super.ContinuousConditionalProbability
    method), 64
replace_Extension_at ()
    (PMML44Super.ContinuousDistribution
    method), 65
replace_Extension_at ()
    (PMML44Super.ContinuousNode method), 67
replace_Extension_at ()
    (PMML44Super.ContStats method), 63
replace_Extension_at ()
    (PMML44Super.CorrelationFields method), 67
replace_Extension_at ()
    (PMML44Super.CorrelationMethods method), 68
replace_Extension_at ()
    (PMML44Super.Correlations method), 69
replace_Extension_at ()
    (PMML44Super.CorrelationValues method), 69
replace_Extension_at ()
    (PMML44Super.COUNT_TABLE_TYPE
    method), 40
replace_Extension_at () (PMML44Super.Counts
    method), 70
replace_Extension_at ()
    (PMML44Super.Covariances method), 71
replace_Extension_at ()
    (PMML44Super.CovariateList method), 72
replace_Extension_at ()
    (PMML44Super.DataDictionary method), 73
replace_Extension_at ()
    (PMML44Super.DataField method), 74
replace_Extension_at ()
    (PMML44Super.Decision method), 75
replace_Extension_at ()
    (PMML44Super.Decisions method), 77
replace_Extension_at ()
    (PMML44Super.DecisionTree method), 76
replace_Extension_at ()
    (PMML44Super.DeepNetwork method), 79
replace_Extension_at ()
    (PMML44Super.DefineFunction method),

```

80

`replace_Extension_at()`
 (*PMML44Super.Delimiter method*), 81

`replace_Extension_at()`
 (*PMML44Super.Denominator method*), 82

`replace_Extension_at()`
 (*PMML44Super.DerivedField method*), 83

`replace_Extension_at()`
 (*PMML44Super.DiscreteConditionalProbability method*), 86

`replace_Extension_at()`
 (*PMML44Super.DiscreteNode method*), 87

`replace_Extension_at()`
 (*PMML44Super.Discretize method*), 88

`replace_Extension_at()`
 (*PMML44Super.DiscretizeBin method*), 89

`replace_Extension_at()`
 (*PMML44Super.DiscrStats method*), 85

`replace_Extension_at()`
 (*PMML44Super.DocumentTermMatrix method*), 89

`replace_Extension_at()`
 (*PMML44Super.DynamicRegressor method*), 90

`replace_Extension_at()`
 (*PMML44Super.euclidean method*), 301

`replace_Extension_at()`
 (*PMML44Super.EventValues method*), 91

`replace_Extension_at()`
 (*PMML44Super.FactorList method*), 94

`replace_Extension_at()` (*PMML44Super.False_ method*), 94

`replace_Extension_at()`
 (*PMML44Super.FieldColumnPair method*), 95

`replace_Extension_at()`
 (*PMML44Super.FieldRef method*), 96

`replace_Extension_at()`
 (*PMML44Super.FieldValue method*), 97

`replace_Extension_at()`
 (*PMML44Super.FieldValueCount method*), 98

`replace_Extension_at()`
 (*PMML44Super.GARCH method*), 102

`replace_Extension_at()`
 (*PMML44Super.GARCHPart method*), 102

`replace_Extension_at()`
 (*PMML44Super.GaussianDistribution method*), 103

`replace_Extension_at()`
 (*PMML44Super.GaussianProcessModel method*), 105

`replace_Extension_at()`
 (*PMML44Super.GeneralizedExponentialKernel method*), 109

`replace_Extension_at()`
 (*PMML44Super.GeneralRegressionModel method*), 107

`replace_Extension_at()`
 (*PMML44Super.Header method*), 111

`replace_Extension_at()`
 (*PMML44Super.InlineTable method*), 113

`replace_Extension_at()`
 (*PMML44Super.InstanceField method*), 114

`replace_Extension_at()`
 (*PMML44Super.InstanceFields method*), 115

`replace_Extension_at()`
 (*PMML44Super.Interval method*), 116

`replace_Extension_at()` (*PMML44Super.Item method*), 117

`replace_Extension_at()`
 (*PMML44Super.ItemRef method*), 117

`replace_Extension_at()` (*PMML44Super.Itemset method*), 118

`replace_Extension_at()`
 (*PMML44Super.jaccard method*), 301

`replace_Extension_at()`
 (*PMML44Super.KNNInput method*), 119

`replace_Extension_at()`
 (*PMML44Super.KNNInputs method*), 120

`replace_Extension_at()`
 (*PMML44Super.KohonenMap method*), 121

`replace_Extension_at()` (*PMML44Super.Lag method*), 122

`replace_Extension_at()`
 (*PMML44Super.Lambda method*), 123

`replace_Extension_at()`
 (*PMML44Super.LayerBias method*), 124

`replace_Extension_at()`
 (*PMML44Super.LayerParameters method*), 126

`replace_Extension_at()`
 (*PMML44Super.LayerWeights method*), 129

`replace_Extension_at()`
 (*PMML44Super.LiftData method*), 130

`replace_Extension_at()`
 (*PMML44Super.LiftGraph method*), 131

`replace_Extension_at()`
 (*PMML44Super.LinearKernelType method*), 132

`replace_Extension_at()`
 (*PMML44Super.LinearNorm method*), 132

`replace_Extension_at()`
 (*PMML44Super.LocalTransformations method*), 133

`replace_Extension_at()`
 (*PMML44Super.LognormalDistributionForBN method*), 133

`method)`, 134
`replace_Extension_at()` (`PMML44Super.Losses` `method`), 135
`replace_Extension_at()` (`PMML44Super.Lower` `method`), 135
`replace_Extension_at()` (`PMML44Super.MA` `method`), 136
`replace_Extension_at()` (`PMML44Super.MACoefficients` `method`), 137
`replace_Extension_at()` (`PMML44Super.MapValues` `method`), 138
`replace_Extension_at()` (`PMML44Super.Mean` `method`), 142
`replace_Extension_at()` (`PMML44Super.MeasurementMatrix` `method`), 143
`replace_Extension_at()` (`PMML44Super.Metrics` `method`), 143
`replace_Extension_at()` (`PMML44Super.MiningBuildTask` `method`), 144
`replace_Extension_at()` (`PMML44Super.MiningField` `method`), 145
`replace_Extension_at()` (`PMML44Super.MiningModel` `method`), 147
`replace_Extension_at()` (`PMML44Super.MiningSchema` `method`), 148
`replace_Extension_at()` (`PMML44Super.minkowski` `method`), 302
`replace_Extension_at()` (`PMML44Super.MissingValueWeights` `method`), 148
`replace_Extension_at()` (`PMML44Super.ModelExplanation` `method`), 149
`replace_Extension_at()` (`PMML44Super.ModelLiftGraph` `method`), 150
`replace_Extension_at()` (`PMML44Super.ModelStats` `method`), 151
`replace_Extension_at()` (`PMML44Super.ModelVerification` `method`), 152
`replace_Extension_at()` (`PMML44Super.MultivariateStat` `method`), 153
`replace_Extension_at()` (`PMML44Super.MultivariateStats` `method`), 154
`replace_Extension_at()` (`PMML44Super.Nadam` `method`), 155
`replace_Extension_at()` (`PMML44Super.NaiveBayesModel` `method`), 156
`replace_Extension_at()` (`PMML44Super.NearestNeighborModel` `method`), 158
`replace_Extension_at()` (`PMML44Super.NetworkLayer` `method`), 160
`replace_Extension_at()` (`PMML44Super.NeuralInput` `method`), 160
`replace_Extension_at()` (`PMML44Super.NeuralInputs` `method`), 161
`replace_Extension_at()` (`PMML44Super.NeuralLayer` `method`), 162
`replace_Extension_at()` (`PMML44Super.NeuralNetwork` `method`), 164
`replace_Extension_at()` (`PMML44Super.NeuralOutput` `method`), 165
`replace_Extension_at()` (`PMML44Super.NeuralOutputs` `method`), 166
`replace_Extension_at()` (`PMML44Super.Neuron` `method`), 167
`replace_Extension_at()` (`PMML44Super.Node` `method`), 168
`replace_Extension_at()` (`PMML44Super.NonseasonalComponent` `method`), 170
`replace_Extension_at()` (`PMML44Super.NonseasonalFactor` `method`), 170
`replace_Extension_at()` (`PMML44Super.NormalDistributionForBN` `method`), 173
`replace_Extension_at()` (`PMML44Super.NormContinuous` `method`), 171
`replace_Extension_at()` (`PMML44Super.NormDiscrete` `method`), 172
`replace_Extension_at()` (`PMML44Super.Numerator` `method`), 174
`replace_Extension_at()` (`PMML44Super.NumericInfo` `method`), 175
`replace_Extension_at()` (`PMML44Super.NumericPredictor` `method`), 176
`replace_Extension_at()` (`PMML44Super.Optimizers` `method`), 177
`replace_Extension_at()` (`PMML44Super.OptimumLiftGraph` `method`), 178

`replace_Extension_at()`
 (*PMML44Super.OutlierEffect method*), 179
`replace_Extension_at()` (*PMML44Super.Output method*), 180
`replace_Extension_at()`
 (*PMML44Super.OutputField method*), 181
`replace_Extension_at()`
 (*PMML44Super.PairCounts method*), 191
`replace_Extension_at()`
 (*PMML44Super.Parameter method*), 192
`replace_Extension_at()`
 (*PMML44Super.ParameterList method*), 194
`replace_Extension_at()`
 (*PMML44Super.ParamMatrix method*), 192
`replace_Extension_at()`
 (*PMML44Super.ParentValue method*), 195
`replace_Extension_at()`
 (*PMML44Super.Partition method*), 196
`replace_Extension_at()`
 (*PMML44Super.PartitionFieldStats method*), 197
`replace_Extension_at()`
 (*PMML44Super.PastVariances method*), 197
`replace_Extension_at()` (*PMML44Super.PCell method*), 183
`replace_Extension_at()`
 (*PMML44Super.PCovCell method*), 184
`replace_Extension_at()`
 (*PMML44Super.PCovMatrix method*), 185
`replace_Extension_at()` (*PMML44Super.PMML method*), 188
`replace_Extension_at()`
 (*PMML44Super.PoissonDistribution method*), 198
`replace_Extension_at()`
 (*PMML44Super.PolynomialKernelType method*), 199
`replace_Extension_at()` (*PMML44Super.PPCell method*), 189
`replace_Extension_at()`
 (*PMML44Super.PPMatrix method*), 190
`replace_Extension_at()`
 (*PMML44Super.PredictiveModelQuality method*), 200
`replace_Extension_at()`
 (*PMML44Super.Predictor method*), 202
`replace_Extension_at()`
 (*PMML44Super.PredictorTerm method*), 203
`replace_Extension_at()`
 (*PMML44Super.PsiVector method*), 204
`replace_Extension_at()`
 (*PMML44Super.Quantile method*), 204
`replace_Extension_at()`
 (*PMML44Super.RadialBasisKernel method*), 209
`replace_Extension_at()`
 (*PMML44Super.RadialBasisKernelType method*), 210
`replace_Extension_at()`
 (*PMML44Super.RandomLiftGraph method*), 210
`replace_Extension_at()`
 (*PMML44Super.Regression method*), 211
`replace_Extension_at()`
 (*PMML44Super.RegressionModel method*), 213
`replace_Extension_at()`
 (*PMML44Super.RegressionTable method*), 215
`replace_Extension_at()`
 (*PMML44Super.RegressorValues method*), 215
`replace_Extension_at()`
 (*PMML44Super.Residuals method*), 217
`replace_Extension_at()`
 (*PMML44Super.ResidualSquareCoefficients method*), 216
`replace_Extension_at()`
 (*PMML44Super.ResultField method*), 218
`replace_Extension_at()`
 (*PMML44Super.RMSprop method*), 206
`replace_Extension_at()` (*PMML44Super.ROC method*), 207
`replace_Extension_at()`
 (*PMML44Super.ROCGraph method*), 208
`replace_Extension_at()`
 (*PMML44Super.RuleSelectionMethod method*), 219
`replace_Extension_at()`
 (*PMML44Super.RuleSet method*), 220
`replace_Extension_at()`
 (*PMML44Super.RuleSetModel method*), 221
`replace_Extension_at()`
 (*PMML44Super.Scorecard method*), 224
`replace_Extension_at()`
 (*PMML44Super.ScoreDistribution method*), 223
`replace_Extension_at()`
 (*PMML44Super.SeasonalComponent method*), 226
`replace_Extension_at()`
 (*PMML44Super.SeasonalFactor method*), 227
`replace_Extension_at()`

[\(PMML44Super.Segment method\), 230](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.Segmentation method\), 231](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.Sequence method\), 232](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SequenceModel method\), 234](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SequenceReference method\), 235](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SequenceRule method\), 236](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SetPredicate method\), 237](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SetReference method\), 238](#)
[replace_Extension_at \(\)](#) [\(PMML44Super.SGD method\), 222](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SigmoidKernelType method\), 239](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.simpleMatching method\), 303](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SimplePredicate method\), 240](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SimpleRule method\), 241](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SimpleSetPredicate method\), 242](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.squaredEuclidean method\), 304](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.StateSpaceModel method\), 244](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.StateVector method\), 245](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SupportVector method\), 245](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SupportVectorMachine method\), 246](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SupportVectorMachineModel method\), 248](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.SupportVectors method\), 249](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TableLocator method\), 250](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.tanimoto method\), 304](#)
[replace_Extension_at \(\)](#) [\(PMML44Super.Target method\), 251](#)
[replace_Extension_at \(\)](#) [\(PMML44Super.Targets method\), 256](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TargetValue method\), 252](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TargetValueCount method\), 253](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TargetValueCounts method\), 254](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TargetValueStat method\), 254](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TargetValueStats method\), 255](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.Taxonomy method\), 257](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TestDistributions method\), 258](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextCorpus method\), 259](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextDictionary method\), 259](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextDocument method\), 260](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextIndex method\), 262](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextIndexNormalization method\), 263](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextModel method\), 265](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextModelNormalization method\), 266](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TextModelSimiliarity method\), 267](#)
[replace_Extension_at \(\)](#) [\(PMML44Super.Time method\), 269](#)
[replace_Extension_at \(\)](#)
 [\(PMML44Super.TimeSeriesModel method\), 274](#)
[replace_Extension_at \(\)](#)

(PMML44Super.Timestamp method), 276
 replace_Extension_at ()
 (PMML44Super.TrainingInstances method), 277
 replace_Extension_at ()
 (PMML44Super.TrainingParameters method), 278
 replace_Extension_at ()
 (PMML44Super.TransformationDictionary method), 279
 replace_Extension_at ()
 (PMML44Super.TransitionMatrix method), 280
 replace_Extension_at ()
 (PMML44Super.TreeModel method), 281
 replace_Extension_at ()
 (PMML44Super.TrendCoefficients method), 282
 replace_Extension_at ()
 (PMML44Super.TriangularDistributionForBN method), 284
 replace_Extension_at () (PMML44Super.True_ method), 284
 replace_Extension_at ()
 (PMML44Super.UniformDistribution method), 285
 replace_Extension_at ()
 (PMML44Super.UniformDistributionForBN method), 286
 replace_Extension_at ()
 (PMML44Super.UnivariateStats method), 287
 replace_Extension_at () (PMML44Super.Upper method), 288
 replace_Extension_at () (PMML44Super.Value method), 289
 replace_Extension_at ()
 (PMML44Super.ValueProbability method), 289
 replace_Extension_at ()
 (PMML44Super.VariableWeight method), 290
 replace_Extension_at ()
 (PMML44Super.Variance method), 291
 replace_Extension_at ()
 (PMML44Super.VarianceCoefficients method), 292
 replace_Extension_at ()
 (PMML44Super.VectorDictionary method), 293
 replace_Extension_at ()
 (PMML44Super.VectorFields method), 294
 replace_Extension_at ()
 (PMML44Super.VectorInstance method), 295
 replace_Extension_at ()
 (PMML44Super.VerificationField method), 296
 replace_Extension_at ()
 (PMML44Super.VerificationFields method), 296
 replace_Extension_at ()
 (PMML44Super.XCoordinates method), 297
 replace_Extension_at ()
 (PMML44Super.YCoordinates method), 298
 replace_False_at ()
 (PMML44Super.CompoundPredicate method), 57
 replace_FieldColumnPair_at ()
 (PMML44Super.ChildParent method), 45
 replace_FieldColumnPair_at ()
 (PMML44Super.MapValues method), 138
 replace_FieldRef_at () (PMML44Super.Apply method), 22
 replace_FieldRef_at () (PMML44Super.Baseline method), 29
 replace_FieldRef_at ()
 (PMML44Super.PredictorTerm method), 203
 replace_FieldRef_at ()
 (PMML44Super.VectorFields method), 294
 replace_FieldValue_at ()
 (PMML44Super.COUNT_TABLE_TYPE method), 40
 replace_FieldValue_at ()
 (PMML44Super.FieldValue method), 97
 replace_FieldValueCount_at ()
 (PMML44Super.COUNT_TABLE_TYPE method), 40
 replace_FieldValueCount_at ()
 (PMML44Super.FieldValue method), 97
 replace_GaussianProcessModel_at ()
 (PMML44Super.PMML method), 188
 replace_GeneralRegressionModel_at ()
 (PMML44Super.PMML method), 188
 replace_InstanceField_at ()
 (PMML44Super.InstanceFields method), 115
 replace_Interval_at ()
 (PMML44Super.ContStats method), 63
 replace_Interval_at ()
 (PMML44Super.DataField method), 74
 replace_Interval_at ()
 (PMML44Super.EventValues method), 91
 replace_Item_at ()
 (PMML44Super.AssociationModel method), 24
 replace_Item_at ()

<code>(PMML44Super.SequenceModel method),</code>	<code>(PMML44Super.NeuralNetwork method),</code>
234	164
<code>replace_ItemRef_at() (PMML44Super.Itemset</code>	<code>replace_NeuralNetwork_at()</code>
<code>method), 118</code>	<code>(PMML44Super.PMML method), 188</code>
<code>replace_Itemset_at()</code>	<code>replace_NeuralOutput_at()</code>
<code>(PMML44Super.AssociationModel method), 24</code>	<code>(PMML44Super.NeuralOutputs method),</code>
<code>replace_Itemset_at()</code>	166
<code>(PMML44Super.SequenceModel method),</code>	<code>replace_Neuron_at()</code>
234	<code>(PMML44Super.NeuralLayer method), 162</code>
<code>replace_KNNInput_at()</code>	<code>replace_Node_at() (PMML44Super.Node method),</code>
<code>(PMML44Super.KNNInputs method), 120</code>	168
<code>replace_Lag_at() (PMML44Super.Apply method),</code>	<code>replace_NormContinuous_at()</code>
22	<code>(PMML44Super.Apply method), 22</code>
<code>replace_Lambda_at()</code>	<code>replace_NormDiscrete_at()</code>
<code>(PMML44Super.AbsoluteExponentialKernel</code>	<code>(PMML44Super.Apply method), 22</code>
<code>method), 7</code>	<code>replace_NUM_ARRAY_at()</code>
<code>replace_Lambda_at()</code>	<code>(PMML44Super.ContStats method), 63</code>
<code>(PMML44Super.ARDSquaredExponentialKernel</code>	<code>replace_NumericPredictor_at()</code>
<code>method), 4</code>	<code>(PMML44Super.RegressionTable method),</code>
<code>replace_Lambda_at()</code>	215
<code>(PMML44Super.GeneralizedExponentialKernel</code>	<code>replace_OutlierEffect_at()</code>
<code>method), 109</code>	<code>(PMML44Super.ARIMA method), 5</code>
<code>replace_LiftData_at()</code>	<code>replace_OutputField_at()</code>
<code>(PMML44Super.PredictiveModelQuality</code>	<code>(PMML44Super.Output method), 180</code>
<code>method), 200</code>	<code>replace_PairCounts_at()</code>
<code>replace_LinearNorm_at()</code>	<code>(PMML44Super.BayesInput method), 33</code>
<code>(PMML44Super.NormContinuous method),</code>	<code>replace_Parameter_at()</code>
171	<code>(PMML44Super.ParameterList method),</code>
<code>replace_MapValues_at() (PMML44Super.Apply</code>	194
<code>method), 22</code>	<code>replace_ParameterField_at()</code>
<code>replace_MatCell_at() (PMML44Super.Matrix</code>	<code>(PMML44Super.DefineFunction method),</code>
<code>method), 140</code>	80
<code>replace_MiningField_at()</code>	<code>replace_ParentValue_at()</code>
<code>(PMML44Super.MiningSchema method),</code>	<code>(PMML44Super.ContinuousConditionalProbability</code>
148	<code>method), 65</code>
<code>replace_MiningModel_at()</code>	<code>replace_ParentValue_at()</code>
<code>(PMML44Super.PMML method), 188</code>	<code>(PMML44Super.DiscreteConditionalProbability</code>
<code>replace_MultivariateStat_at()</code>	<code>method), 86</code>
<code>(PMML44Super.MultivariateStats method),</code>	<code>replace_PartitionFieldStats_at()</code>
154	<code>(PMML44Super.Partition method), 196</code>
<code>replace_MultivariateStats_at()</code>	<code>replace_PCell_at() (PMML44Super.ParamMatrix</code>
<code>(PMML44Super.ModelStats method), 151</code>	<code>method), 192</code>
<code>replace_Nadam_at() (PMML44Super.Optimizers</code>	<code>replace_PCovCell_at()</code>
<code>method), 177</code>	<code>(PMML44Super.PCovMatrix method), 185</code>
<code>replace_NaiveBayesModel_at()</code>	<code>replace_PPCell_at() (PMML44Super.PPMatrix</code>
<code>(PMML44Super.PMML method), 188</code>	<code>method), 190</code>
<code>replace_NearestNeighborModel_at()</code>	<code>replace_PredictiveModelQuality_at()</code>
<code>(PMML44Super.PMML method), 188</code>	<code>(PMML44Super.ModelExplanation method),</code>
<code>replace_NetworkLayer_at()</code>	149
<code>(PMML44Super.DeepNetwork method),</code>	<code>replace_Predictor_at()</code>
79	<code>(PMML44Super.CovariateList method), 72</code>
<code>replace_NeuralInput_at()</code>	<code>replace_Predictor_at()</code>
<code>(PMML44Super.NeuralInputs method), 161</code>	<code>(PMML44Super.FactorList method), 94</code>
<code>replace_NeuralLayer_at()</code>	<code>replace_PredictorTerm_at()</code>

(PMML44Super.RegressionTable method), 215
 replace_Quantile_at()
 (PMML44Super.NumericInfo method), 175
 replace_Regression_at()
 (PMML44Super.MiningModel method), 147
 replace_RegressionModel_at()
 (PMML44Super.PMML method), 188
 replace_RegressionTable_at()
 (PMML44Super.Regression method), 212
 replace_RegressionTable_at()
 (PMML44Super.RegressionModel method), 213
 replace_ResultField_at()
 (PMML44Super.DecisionTree method), 76
 replace_ResultField_at()
 (PMML44Super.Regression method), 212
 replace_RMSprop_at()
 (PMML44Super.Optimizers method), 177
 replace_row_at() (PMML44Super.InlineTable method), 113
 replace_RuleSelectionMethod_at()
 (PMML44Super.RuleSet method), 220
 replace_RuleSetModel_at()
 (PMML44Super.PMML method), 188
 replace_Scorecard_at() (PMML44Super.PMML method), 188
 replace_ScoreDistribution_at()
 (PMML44Super.Node method), 168
 replace_ScoreDistribution_at()
 (PMML44Super.RuleSet method), 220
 replace_ScoreDistribution_at()
 (PMML44Super.SimpleRule method), 241
 replace_Segment_at()
 (PMML44Super.Segmentation method), 231
 replace_Sequence_at()
 (PMML44Super.SequenceModel method), 234
 replace_SequenceModel_at()
 (PMML44Super.PMML method), 188
 replace_SequenceRule_at()
 (PMML44Super.SequenceModel method), 234
 replace_SetPredicate_at()
 (PMML44Super.SequenceModel method), 234
 replace_SetReference_at()
 (PMML44Super.Sequence method), 232
 replace_SGD_at() (PMML44Super.Optimizers method), 177
 replace_SimplePredicate_at()
 (PMML44Super.CompoundPredicate method), 57
 replace_SimpleRule_at()

(PMML44Super.CompoundRule method), 58
 replace_SimpleRule_at()
 (PMML44Super.RuleSet method), 220
 replace_SimpleSetPredicate_at()
 (PMML44Super.CompoundPredicate method), 57
 replace_SupportVector_at()
 (PMML44Super.SupportVectors method), 249
 replace_SupportVectorMachine_at()
 (PMML44Super.SupportVectorMachineModel method), 248
 replace_SupportVectorMachineModel_at()
 (PMML44Super.PMML method), 188
 replace_Target_at() (PMML44Super.Targets method), 256
 replace_TargetValue_at()
 (PMML44Super.Target method), 251
 replace_TargetValueCount_at()
 (PMML44Super.TargetValueCounts method), 254
 replace_TargetValueStat_at()
 (PMML44Super.TargetValueStats method), 255
 replace_Taxonomy_at()
 (PMML44Super.DataDictionary method), 73
 replace_TextDocument_at()
 (PMML44Super.TextCorpus method), 259
 replace_TextIndex_at() (PMML44Super.Apply method), 22
 replace_TextIndexNormalization_at()
 (PMML44Super.TextIndex method), 262
 replace_TextModel_at() (PMML44Super.PMML method), 188
 replace_Theta_at() (PMML44Super.FinalTheta method), 101
 replace_TimeCycle_at()
 (PMML44Super.TimeAnchor method), 270
 replace_TimeException_at()
 (PMML44Super.TimeAnchor method), 270
 replace_TimeSeries_at()
 (PMML44Super.TimeSeriesModel method), 274
 replace_TimeSeriesModel_at()
 (PMML44Super.PMML method), 188
 replace_TimeValue_at()
 (PMML44Super.ExponentialSmoothing method), 92
 replace_TimeValue_at()
 (PMML44Super.TimeSeries method), 272
 replace_TreeModel_at() (PMML44Super.PMML method), 188

`replace_True_at()` (*PMML44Super.CompoundPredicate method*), 57
`replace_UnivariateStats_at()` (*PMML44Super.ModelStats method*), 151
`replace_Value_at()` (*PMML44Super.DataField method*), 74
`replace_Value_at()` (*PMML44Super.DerivedField method*), 83
`replace_Value_at()` (*PMML44Super.EventValues method*), 91
`replace_ValueProbability_at()` (*PMML44Super.DiscreteConditionalProbability method*), 86
`replace_ValueProbability_at()` (*PMML44Super.DiscreteNode method*), 87
`replace_VectorInstance_at()` (*PMML44Super.VectorDictionary method*), 293
`replace_VerificationField_at()` (*PMML44Super.VerificationFields method*), 296
Residuals (class in *PMML44Super*), 216
ResidualSquareCoefficients (class in *PMML44Super*), 216
ResultField (class in *PMML44Super*), 217
RMSprop (class in *PMML44Super*), 206
ROC (class in *PMML44Super*), 207
ROCGraph (class in *PMML44Super*), 208
row (class in *PMML44Super*), 302
RuleSelectionMethod (class in *PMML44Super*), 218
RuleSet (class in *PMML44Super*), 219
RuleSetModel (class in *PMML44Super*), 220
S
Scorecard (class in *PMML44Super*), 223
ScoreDistribution (class in *PMML44Super*), 222
SeasonalComponent (class in *PMML44Super*), 225
SeasonalFactor (class in *PMML44Super*), 226
Seasonality_ExpoSmooth (class in *PMML44Super*), 227
SeasonalTrendDecomposition (class in *PMML44Super*), 227
Segment (class in *PMML44Super*), 228
Segmentation (class in *PMML44Super*), 230
Sequence (class in *PMML44Super*), 231
SequenceModel (class in *PMML44Super*), 233
SequenceReference (class in *PMML44Super*), 235
SequenceRule (class in *PMML44Super*), 236
`set_AbsoluteExponentialKernel()` (*PMML44Super.GaussianProcessModel method*), 105
`set_absoluteValue()` (*PMML44Super.Coefficients method*), 52
`set_activationFunction()` (*PMML44Super.LayerParameters method*), 126
`set_activationFunction()` (*PMML44Super.NeuralLayer method*), 162
`set_activationFunction()` (*PMML44Super.NeuralNetwork method*), 164
`set_activityRegularizerL1()` (*PMML44Super.LayerParameters method*), 126
`set_activityRegularizerL2()` (*PMML44Super.LayerParameters method*), 126
`set_Adadelta()` (*PMML44Super.Optimizers method*), 177
`set_Adagrad()` (*PMML44Super.Optimizers method*), 177
`set_Adam()` (*PMML44Super.Optimizers method*), 177
`set_Adamax()` (*PMML44Super.Optimizers method*), 177
`set_adj_r_squared()` (*PMML44Super.PredictiveModelQuality method*), 201
`set_affinity()` (*PMML44Super.AssociationRule method*), 26
`set_Aggregate()` (*PMML44Super.Apply method*), 22
`set_Aggregate()` (*PMML44Super.ComplexPartialScore method*), 55
`set_Aggregate()` (*PMML44Super.DefineFunction method*), 80
`set_Aggregate()` (*PMML44Super.DerivedField method*), 83
`set_aggregate()` (*PMML44Super.Lag method*), 122
`set_Aggregate()` (*PMML44Super.Lower method*), 136
`set_Aggregate()` (*PMML44Super.Mean method*), 142
`set_Aggregate()` (*PMML44Super.OutputField method*), 181
`set_Aggregate()` (*PMML44Super.TextIndex method*), 262
`set_Aggregate()` (*PMML44Super.Upper method*), 288
`set_Aggregate()` (*PMML44Super.Variance method*), 291
`set_AIC()` (*PMML44Super.PredictiveModelQuality method*), 200
`set_AICc()` (*PMML44Super.PredictiveModelQuality method*), 200
`set_algorithm()` (*PMML44Super.OutputField*

[method](#)), [182](#)
[set_algorithmName\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) [method](#)), [16](#)
[set_algorithmName\(\)](#) ([PMML44Super.AssociationModel](#) [method](#)), [25](#)
[set_algorithmName\(\)](#) ([PMML44Super.BaselineModel](#) [method](#)), [31](#)
[set_algorithmName\(\)](#) ([PMML44Super.BayesianNetworkModel](#) [method](#)), [36](#)
[set_algorithmName\(\)](#) ([PMML44Super.ClusteringModel](#) [method](#)), [50](#)
[set_algorithmName\(\)](#) ([PMML44Super.DecisionTree](#) [method](#)), [76](#)
[set_algorithmName\(\)](#) ([PMML44Super.DeepNetwork](#) [method](#)), [79](#)
[set_algorithmName\(\)](#) ([PMML44Super.GaussianProcessModel](#) [method](#)), [105](#)
[set_algorithmName\(\)](#) ([PMML44Super.GeneralRegressionModel](#) [method](#)), [108](#)
[set_algorithmName\(\)](#) ([PMML44Super.MiningModel](#) [method](#)), [147](#)
[set_algorithmName\(\)](#) ([PMML44Super.NaiveBayesModel](#) [method](#)), [157](#)
[set_algorithmName\(\)](#) ([PMML44Super.NearestNeighborModel](#) [method](#)), [158](#)
[set_algorithmName\(\)](#) ([PMML44Super.NeuralNetwork](#) [method](#)), [164](#)
[set_algorithmName\(\)](#) ([PMML44Super.Regression](#) [method](#)), [212](#)
[set_algorithmName\(\)](#) ([PMML44Super.RegressionModel](#) [method](#)), [213](#)
[set_algorithmName\(\)](#) ([PMML44Super.RuleSetModel](#) [method](#)), [221](#)
[set_algorithmName\(\)](#) ([PMML44Super.Scorecard](#) [method](#)), [225](#)
[set_algorithmName\(\)](#) ([PMML44Super.SequenceModel](#) [method](#)), [234](#)
[set_algorithmName\(\)](#) ([PMML44Super.SupportVectorMachineModel](#) [method](#)), [248](#)
[set_algorithmName\(\)](#) ([PMML44Super.TextModel](#) [method](#)), [265](#)
[set_algorithmName\(\)](#) ([PMML44Super.TimeSeriesModel](#) [method](#)), [274](#)
[set_algorithmName\(\)](#) ([PMML44Super.TreeModel](#) [method](#)), [281](#)
[set_algorithmType\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) [method](#)), [16](#)
[set_alpha\(\)](#) ([PMML44Super.Level](#) [method](#)), [129](#)
[set_alphaDropoutRate\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), [126](#)
[set_alphaDropoutSeed\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), [126](#)
[set_Alternate\(\)](#) ([PMML44Super.TestDistributions](#) [method](#)), [258](#)
[set_alternateTargetCategory\(\)](#) ([PMML44Super.SupportVectorMachine](#) [method](#)), [246](#)
[set_altitude\(\)](#) ([PMML44Super.NeuralLayer](#) [method](#)), [162](#)
[set_altitude\(\)](#) ([PMML44Super.NeuralNetwork](#) [method](#)), [164](#)
[set_altitude\(\)](#) ([PMML44Super.Neuron](#) [method](#)), [167](#)
[set_Annotation\(\)](#) ([PMML44Super.Header](#) [method](#)), [111](#)
[set_AnomalyDetectionModel\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) [method](#)), [15](#)
[set_AnomalyDetectionModel\(\)](#) ([PMML44Super.PMML](#) [method](#)), [188](#)
[set_AnomalyDetectionModel\(\)](#) ([PMML44Super.Segment](#) [method](#)), [230](#)
[set_Anova\(\)](#) ([PMML44Super.UnivariateStats](#) [method](#)), [287](#)
[set_AnovaRow\(\)](#) ([PMML44Super.Anova](#) [method](#)), [17](#)
[set_antecedent\(\)](#) ([PMML44Super.AssociationRule](#) [method](#)), [26](#)
[set_AntecedentSequence\(\)](#) ([PMML44Super.SequenceRule](#) [method](#)), [236](#)
[set_AnyDistribution\(\)](#) ([PMML44Super.Alternate](#) [method](#)), [13](#)
[set_AnyDistribution\(\)](#) ([PMML44Super.Baseline](#) [method](#)), [29](#)
[set_AnyDistribution\(\)](#) ([PMML44Super.TargetValueStat](#) [method](#)), [254](#)
[set_anytypeobjs_\(\)](#) ([PMML44Super.Extension](#) [method](#)), [93](#)
[set_anytypeobjs_\(\)](#) ([PMML44Super.row](#) [method](#)),

302
set_Application() (PMML44Super.Header method), 111
set_Apply() (PMML44Super.Apply method), 22
set_Apply() (PMML44Super.ComplexPartialScore method), 55
set_Apply() (PMML44Super.DefineFunction method), 80
set_Apply() (PMML44Super.DerivedField method), 83
set_Apply() (PMML44Super.Lower method), 136
set_Apply() (PMML44Super.Mean method), 142
set_Apply() (PMML44Super.OutputField method), 181
set_Apply() (PMML44Super.TextIndex method), 262
set_Apply() (PMML44Super.Upper method), 288
set_Apply() (PMML44Super.Variance method), 291
set_AR() (PMML44Super.ARMAPart method), 6
set_AR() (PMML44Super.NonseasonalComponent method), 170
set_AR() (PMML44Super.SeasonalComponent method), 226
set_architectureName() (PMML44Super.DerivedField method), 84
set_architectureName() (PMML44Super.TrainingParameters method), 278
set_ARDSquaredExponentialKernel() (PMML44Super.GaussianProcessModel method), 105
set_ARIMA() (PMML44Super.TimeSeriesModel method), 274
set_ARMAPart() (PMML44Super.GARCH method), 102
set_Array() (PMML44Super.AR method), 3
set_Array() (PMML44Super.BoundaryValueMeans method), 39
set_Array() (PMML44Super.BoundaryValues method), 39
set_Array() (PMML44Super.ClassLabels method), 46
set_Array() (PMML44Super.Cluster method), 47
set_Array() (PMML44Super.CorrelationFields method), 67
set_Array() (PMML44Super.DiscrStats method), 85
set_Array() (PMML44Super.FinalNoise method), 98
set_Array() (PMML44Super.FinalNu method), 99
set_Array() (PMML44Super.FinalPredictedNoise method), 100
set_Array() (PMML44Super.FinalStateVector method), 100
set_Array() (PMML44Super.HVector method), 110
set_Array() (PMML44Super.Lambda method), 123
set_Array() (PMML44Super.MACoefficients method), 137
set_Array() (PMML44Super.Matrix method), 140
set_Array() (PMML44Super.MissingValueWeights method), 148
set_Array() (PMML44Super.NonseasonalFactor method), 170
set_Array() (PMML44Super.PartitionFieldStats method), 197
set_Array() (PMML44Super.PastVariances method), 197
set_Array() (PMML44Super.PsiVector method), 204
set_Array() (PMML44Super.Residuals method), 217
set_Array() (PMML44Super.SeasonalFactor method), 227
set_Array() (PMML44Super.Seasonality_ExpoSmooth method), 228
set_Array() (PMML44Super.SetPredicate method), 238
set_Array() (PMML44Super.SimpleSetPredicate method), 242
set_Array() (PMML44Super.StateVector method), 245
set_Array() (PMML44Super.TextDictionary method), 259
set_Array() (PMML44Super.TimeCycle method), 271
set_Array() (PMML44Super.TimeException method), 271
set_Array() (PMML44Super.TransferFunctionValues method), 278
set_Array() (PMML44Super.Trend_ExpoSmooth method), 283
set_Array() (PMML44Super.VectorInstance method), 295
set_Array() (PMML44Super.XCoordinates method), 297
set_Array() (PMML44Super.YCoordinates method), 298
set_AssociationModel() (PMML44Super.AnomalyDetectionModel method), 15
set_AssociationModel() (PMML44Super.PMML method), 188
set_AssociationModel() (PMML44Super.Segment method), 230
set_AssociationRule() (PMML44Super.AssociationModel method), 25
set_Attribute() (PMML44Super.Characteristic method), 43
set_avgNumberOfItemsPerTA() (PMML44Super.AssociationModel method), 25
set_avgNumberOfItemsPerTransaction() (PMML44Super.SequenceModel method), 235
set_avgNumberOfTAsPerTAGroup()

(PMML44Super.SequenceModel method), 235
 set_BaseCumHazardTables() (PMML44Super.GeneralRegressionModel method), 107
 set_Baseline() (PMML44Super.TestDistributions method), 258
 set_BaselineCell() (PMML44Super.BaseCumHazardTables method), 28
 set_BaselineCell() (PMML44Super.BaselineStratum method), 32
 set_baselineMethod() (PMML44Super.Scorecard method), 225
 set_BaselineModel() (PMML44Super.AnomalyDetectionModel method), 15
 set_BaselineModel() (PMML44Super.PMML method), 188
 set_BaselineModel() (PMML44Super.Segment method), 230
 set_baselineScore() (PMML44Super.Characteristic method), 43
 set_baselineScore() (PMML44Super.Scorecard method), 225
 set_baselineStrataVariable() (PMML44Super.GeneralRegressionModel method), 108
 set_BaselineStratum() (PMML44Super.BaseCumHazardTables method), 28
 set_batchNormalizationAxis() (PMML44Super.LayerParameters method), 126
 set_batchNormalizationCenter() (PMML44Super.LayerParameters method), 126
 set_batchNormalizationEpsilon() (PMML44Super.LayerParameters method), 126
 set_batchNormalizationMomentum() (PMML44Super.LayerParameters method), 126
 set_batchNormalizationScale() (PMML44Super.LayerParameters method), 127
 set_batchSize() (PMML44Super.LayerParameters method), 127
 set_BayesianNetworkModel() (PMML44Super.AnomalyDetectionModel method), 15
 set_BayesianNetworkModel() (PMML44Super.PMML method), 188
 set_BayesianNetworkModel() (PMML44Super.Segment method), 230
 set_BayesianNetworkNodes() (PMML44Super.BayesianNetworkModel method), 36
 set_BayesInput() (PMML44Super.BayesInputs method), 34
 set_BayesInputs() (PMML44Super.NaiveBayesModel method), 156
 set_BayesOutput() (PMML44Super.NaiveBayesModel method), 156
 set_bestFit() (PMML44Super.TimeSeriesModel method), 274
 set_beta() (PMML44Super.PCell method), 183
 set_beta_1() (PMML44Super.Adam method), 10
 set_beta_1() (PMML44Super.Adamax method), 11
 set_beta_1() (PMML44Super.Nadam method), 155
 set_beta_2() (PMML44Super.Adam method), 10
 set_beta_2() (PMML44Super.Adamax method), 11
 set_beta_2() (PMML44Super.Nadam method), 155
 set_betaConstraint() (PMML44Super.LayerParameters method), 127
 set_betaInitializer() (PMML44Super.LayerParameters method), 127
 set_betaRegularizer() (PMML44Super.LayerParameters method), 127
 set_bias() (PMML44Super.Neuron method), 167
 set_biasConstraint() (PMML44Super.LayerParameters method), 127
 set_biasFlattenAxis() (PMML44Super.LayerBias method), 124
 set_biasInitializer() (PMML44Super.LayerParameters method), 127
 set_biasRegularizer() (PMML44Super.LayerParameters method), 127
 set_biasShape() (PMML44Super.LayerBias method), 124
 set_BIC() (PMML44Super.PredictiveModelQuality method), 201
 set_binarySimilarity() (PMML44Super.ComparisonMeasure method), 53
 set_binValue() (PMML44Super.DiscretizeBin method), 89
 set_BlockIndicator() (PMML44Super.Lag

`method`), 122
`set_booleanOperator()` (*PMML44Super.CompoundPredicate method*), 57
`set_booleanOperator()` (*PMML44Super.SimpleSetPredicate method*), 242
`set_BoundaryValueMeans()` (*PMML44Super.LiftGraph method*), 131
`set_BoundaryValues()` (*PMML44Super.LiftGraph method*), 131
`set_BoundaryValues()` (*PMML44Super.ROCGraph method*), 208
`set_businessProblem()` (*PMML44Super.Decisions method*), 77
`set_c00_parameter()` (*PMML44Super.binarySimilarity method*), 299
`set_c01_parameter()` (*PMML44Super.binarySimilarity method*), 299
`set_c10_parameter()` (*PMML44Super.binarySimilarity method*), 299
`set_c11_parameter()` (*PMML44Super.binarySimilarity method*), 299
`set_cardinality()` (*PMML44Super.Counts method*), 70
`set_castInteger()` (*PMML44Super.Target method*), 251
`set_CategoricalPredictor()` (*PMML44Super.RegressionTable method*), 215
`set_CategoricalPredictor()` (*PMML44Super.VectorFields method*), 294
`set_categoricalScoringMethod()` (*PMML44Super.NearestNeighborModel method*), 158
`set_Categories()` (*PMML44Super.Predictor method*), 202
`set_Category()` (*PMML44Super.Categories method*), 42
`set_category()` (*PMML44Super.Item method*), 117
`set_category()` (*PMML44Super.MultivariateStat method*), 153
`set_Characteristic()` (*PMML44Super.Characteristics method*), 44
`set_Characteristics()` (*PMML44Super.Scorecard method*), 224
`set_chebychev()` (*PMML44Super.ComparisonMeasure method*), 53
`set_childField()` (*PMML44Super.ChildParent method*), 45
`set_ChildParent()` (*PMML44Super.Taxonomy method*), 257
`set_chiSquareValue()` (*PMML44Super.MultivariateStat method*), 153
`set_cityBlock()` (*PMML44Super.ComparisonMeasure method*), 53
`set_classificationMethod()` (*PMML44Super.SupportVectorMachineModel method*), 248
`set_ClassLabels()` (*PMML44Super.ConfusionMatrix method*), 60
`set_clipnorm()` (*PMML44Super.Optimizers method*), 177
`set_clipvalue()` (*PMML44Super.Optimizers method*), 177
`set_closure()` (*PMML44Super.Interval method*), 116
`set_Cluster()` (*PMML44Super.ClusteringModel method*), 49
`set_ClusteringField()` (*PMML44Super.ClusteringModel method*), 49
`set_ClusteringModel()` (*PMML44Super.AnomalyDetectionModel method*), 15
`set_ClusteringModel()` (*PMML44Super.PMML method*), 188
`set_ClusteringModel()` (*PMML44Super.Segment method*), 230
`set_ClusteringModelQuality()` (*PMML44Super.ModelExplanation method*), 149
`set_coef0()` (*PMML44Super.PolynomialKernelType method*), 199
`set_coef0()` (*PMML44Super.SigmoidKernelType method*), 239
`set_coefficient()` (*PMML44Super.CategoricalPredictor method*), 41
`set_Coefficient()` (*PMML44Super.Coefficients method*), 52
`set_coefficient()` (*PMML44Super.NumericPredictor method*), 176
`set_coefficient()` (*PMML44Super.PredictorTerm method*), 203
`set_Coefficients()` (*PMML44Super.SupportVectorMachine method*), 246
`set_col()` (*PMML44Super.MatCell method*), 139
`set_column()` (*PMML44Super.FieldColumnPair*

`method`), 95
`set_column()` (*PMML44Super.InstanceField method*), 114
`set_column()` (*PMML44Super.VerificationField method*), 296
`set_compareFunction()` (*PMML44Super.ClusteringField method*), 48
`set_compareFunction()` (*PMML44Super.ComparisonMeasure method*), 53
`set_compareFunction()` (*PMML44Super.KNNInput method*), 119
`set_ComparisonMeasure()` (*PMML44Super.ClusteringModel method*), 49
`set_ComparisonMeasure()` (*PMML44Super.NearestNeighborModel method*), 158
`set_Comparisons()` (*PMML44Super.ClusteringField method*), 48
`set_ComplexPartialScore()` (*PMML44Super.Attribute method*), 27
`set_CompoundPredicate()` (*PMML44Super.Attribute method*), 27
`set_CompoundPredicate()` (*PMML44Super.CompoundPredicate method*), 57
`set_CompoundPredicate()` (*PMML44Super.CompoundRule method*), 58
`set_CompoundPredicate()` (*PMML44Super.Node method*), 168
`set_CompoundPredicate()` (*PMML44Super.Segment method*), 230
`set_CompoundPredicate()` (*PMML44Super.SimpleRule method*), 241
`set_CompoundRule()` (*PMML44Super.CompoundRule method*), 58
`set_CompoundRule()` (*PMML44Super.RuleSet method*), 220
`set_Con()` (*PMML44Super.Neuron method*), 167
`set_confidence()` (*PMML44Super.AssociationRule method*), 26
`set_confidence()` (*PMML44Super.ScoreDistribution method*), 223
`set_confidence()` (*PMML44Super.SequenceRule method*), 237
`set_confidence()` (*PMML44Super.SimpleRule method*), 241
`set_confidenceLevel()` (*PMML44Super.MultivariateStat method*), 153
`set_confidenceLowerBound()` (*PMML44Super.MultivariateStat method*), 153
`set_confidenceUpperBound()` (*PMML44Super.MultivariateStat method*), 153
`set_ConfusionMatrix()` (*PMML44Super.PredictiveModelQuality method*), 201
`set_connectionLayerId()` (*PMML44Super.NetworkLayer method*), 160
`set_consequent()` (*PMML44Super.AssociationRule method*), 26
`set_ConsequentSequence()` (*PMML44Super.SequenceRule method*), 236
`set_Constant()` (*PMML44Super.Apply method*), 22
`set_constant()` (*PMML44Super.ARMAPart method*), 7
`set_Constant()` (*PMML44Super.ComplexPartialScore method*), 55
`set_Constant()` (*PMML44Super.DefineFunction method*), 80
`set_Constant()` (*PMML44Super.DerivedField method*), 83
`set_constant()` (*PMML44Super.GARCHPart method*), 102
`set_Constant()` (*PMML44Super.Lower method*), 136
`set_Constant()` (*PMML44Super.Mean method*), 142
`set_Constant()` (*PMML44Super.OutputField method*), 181
`set_Constant()` (*PMML44Super.TextIndex method*), 262
`set_Constant()` (*PMML44Super.Upper method*), 288
`set_Constant()` (*PMML44Super.Variance method*), 291
`set_constantTerm()` (*PMML44Super.ARIMA method*), 6
`set_Constraints()` (*PMML44Super.SequenceModel method*), 234
`set_ContinuousConditionalProbability()` (*PMML44Super.ContinuousNode method*), 67
`set_ContinuousDistribution()` (*PMML44Super.ContinuousConditionalProbability method*), 65
`set_ContinuousDistribution()` (*PMML44Super.ContinuousNode method*), 67
`set_ContinuousNode()`

(PMML44Super.BayesianNetworkNodes
 method), 37
 set_continuousScoringMethod()
 (PMML44Super.NearestNeighborModel
 method), 158
 set_contrastMatrixType()
 (PMML44Super.Predictor method), 202
 set_ContStats() (PMML44Super.UnivariateStats
 method), 287
 set_coord1() (PMML44Super.KohonenMap
 method), 121
 set_coord2() (PMML44Super.KohonenMap
 method), 121
 set_coord3() (PMML44Super.KohonenMap
 method), 121
 set_copyright() (PMML44Super.Header method),
 111
 set_CorrelationFields()
 (PMML44Super.Correlations method), 69
 set_CorrelationMethods()
 (PMML44Super.Correlations method), 69
 set_Correlations()
 (PMML44Super.ModelExplanation method),
 149
 set_CorrelationValues()
 (PMML44Super.Correlations method), 70
 set_count() (PMML44Super.ContinuousConditionalProbability
 method), 65
 set_count() (PMML44Super.ContinuousNode
 method), 67
 set_count() (PMML44Super.DiscreteConditionalProbability
 method), 86
 set_count() (PMML44Super.DiscreteNode method),
 87
 set_count() (PMML44Super.FieldValueCount
 method), 98
 set_count() (PMML44Super.TargetValueCount
 method), 253
 set_count() (PMML44Super.TimeException
 method), 271
 set_countHits() (PMML44Super.TextIndex
 method), 262
 set_Counts() (PMML44Super.PartitionFieldStats
 method), 197
 set_Counts() (PMML44Super.UnivariateStats
 method), 287
 set_CountTable() (PMML44Super.Baseline
 method), 29
 set_Covariances() (PMML44Super.Cluster
 method), 47
 set_CovariateList()
 (PMML44Super.GeneralRegressionModel
 method), 107
 set_criterion() (PMML44Super.RuleSelectionMethod
 method), 219
 set_croppingDims()
 (PMML44Super.LayerParameters method),
 127
 set_cumHazard() (PMML44Super.BaselineCell
 method), 30
 set_cumulativeLink()
 (PMML44Super.GeneralRegressionModel
 method), 108
 set_d() (PMML44Super.NonseasonalComponent
 method), 170
 set_D() (PMML44Super.SeasonalComponent
 method), 226
 set_d00_parameter()
 (PMML44Super.binarySimilarity method),
 299
 set_d01_parameter()
 (PMML44Super.binarySimilarity method),
 299
 set_d10_parameter()
 (PMML44Super.binarySimilarity method),
 299
 set_d11_parameter()
 (PMML44Super.binarySimilarity method),
 299
 set_dampingCoefficient()
 (PMML44Super.OutlierEffect method), 179
 set_DataDictionary() (PMML44Super.PMML
 method), 188
 set_DataField() (PMML44Super.DataDictionary
 method), 73
 set_dataName() (PMML44Super.ClusteringModelQuality
 method), 51
 set_dataName() (PMML44Super.PredictiveModelQuality
 method), 201
 set_dataset() (PMML44Super.TrainingParameters
 method), 278
 set_datasetName() (PMML44Super.DerivedField
 method), 84
 set_dataType() (PMML44Super.Constant method),
 61
 set_dataType() (PMML44Super.DataField
 method), 74
 set_dataType() (PMML44Super.DefineFunction
 method), 81
 set_dataType() (PMML44Super.DerivedField
 method), 84
 set_dataType() (PMML44Super.Discretize
 method), 88
 set_dataType() (PMML44Super.MapValues
 method), 138
 set_dataType() (PMML44Super.OutputField
 method), 182
 set_dataType() (PMML44Super.ParameterField

[method](#)), 193
[set_dataType\(\)](#) ([PMML44Super.ResultField](#) [method](#)), 218
[set_dataUsage\(\)](#) ([PMML44Super.PredictiveModelQuality](#) [method](#)), 201
[set_decayRate\(\)](#) ([PMML44Super.Adadelta](#) [method](#)), 8
[set_decayRate\(\)](#) ([PMML44Super.Adagrad](#) [method](#)), 9
[set_decayRate\(\)](#) ([PMML44Super.Adam](#) [method](#)), 10
[set_decayRate\(\)](#) ([PMML44Super.Adamax](#) [method](#)), 11
[set_decayRate\(\)](#) ([PMML44Super.RMSprop](#) [method](#)), 206
[set_decayRate\(\)](#) ([PMML44Super.SGD](#) [method](#)), 222
[set_Decision\(\)](#) ([PMML44Super.Decisions](#) [method](#)), 77
[set_Decisions\(\)](#) ([PMML44Super.OutputField](#) [method](#)), 181
[set_DecisionTree\(\)](#) ([PMML44Super.MiningModel](#) [method](#)), 147
[set_DecisionTree\(\)](#) ([PMML44Super.Node](#) [method](#)), 168
[set_DeepNetwork\(\)](#) ([PMML44Super.AnomalyDetectionModel](#) [method](#)), 15
[set_DeepNetwork\(\)](#) ([PMML44Super.PMML](#) [method](#)), 188
[set_DeepNetwork\(\)](#) ([PMML44Super.Segment](#) [method](#)), 230
[set_defaultChild\(\)](#) ([PMML44Super.Node](#) [method](#)), 169
[set_defaultConfidence\(\)](#) ([PMML44Super.RuleSet](#) [method](#)), 220
[set_defaultScore\(\)](#) ([PMML44Super.RuleSet](#) [method](#)), 220
[set_defaultValue\(\)](#) ([PMML44Super.Apply](#) [method](#)), 22
[set_defaultValue\(\)](#) ([PMML44Super.Discretize](#) [method](#)), 88
[set_defaultValue\(\)](#) ([PMML44Super.INT_SparseArray](#) [method](#)), 112
[set_defaultValue\(\)](#) ([PMML44Super.MapValues](#) [method](#)), 138
[set_defaultValue\(\)](#) ([PMML44Super.REAL_SparseArray](#) [method](#)), 206
[set_defaultValue\(\)](#) ([PMML44Super.TargetValue](#) [method](#)), 252
[set_DefineFunction\(\)](#) ([PMML44Super.TransformationDictionary](#) [method](#)), 279
[set_degree\(\)](#) ([PMML44Super.GeneralizedExponentialKernel](#) [method](#)), 109
[set_degree\(\)](#) ([PMML44Super.PolynomialKernelType](#) [method](#)), 199
[set_degreesOfFreedom\(\)](#) ([PMML44Super.AnovaRow](#) [method](#)), 18
[set_degreesOfFreedom\(\)](#) ([PMML44Super.PredictiveModelQuality](#) [method](#)), 201
[set_delay\(\)](#) ([PMML44Super.DynamicRegressor](#) [method](#)), 90
[set_delimiter\(\)](#) ([PMML44Super.Delimiter](#) [method](#)), 81
[set_Delimiter\(\)](#) ([PMML44Super.Sequence](#) [method](#)), 232
[set_Delimiter\(\)](#) ([PMML44Super.SequenceRule](#) [method](#)), 236
[set_delta\(\)](#) ([PMML44Super.Seasonality_ExpoSmooth](#) [method](#)), 228
[set_Denominator\(\)](#) ([PMML44Super.DynamicRegressor](#) [method](#)), 90
[set_depthMultiplier\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 127
[set_depthwiseConstraint\(\)](#) ([PMML44Super.LayerParameters](#) [method](#)), 127
[set_DerivedField\(\)](#) ([PMML44Super.BayesInput](#) [method](#)), 33
[set_DerivedField\(\)](#) ([PMML44Super.ContinuousNode](#) [method](#)), 67
[set_DerivedField\(\)](#) ([PMML44Super.DiscreteNode](#) [method](#)), 87
[set_DerivedField\(\)](#) ([PMML44Super.LocalTransformations](#) [method](#)), 133
[set_DerivedField\(\)](#) ([PMML44Super.NeuralInput](#) [method](#)), 161
[set_DerivedField\(\)](#) ([PMML44Super.NeuralOutput](#) [method](#)), 165
[set_DerivedField\(\)](#) ([PMML44Super.TransformationDictionary](#) [method](#)), 279
[set_description\(\)](#) ([PMML44Super.AbsoluteExponentialKernel](#) [method](#)), 8
[set_description\(\)](#) ([PMML44Super.ARDSquaredExponentialKernel](#) [method](#)), 4
[set_description\(\)](#) ([PMML44Super.Decision](#)

`method`), 75
`set_description()` (*PMML44Super.Decisions method*), 77
`set_description()` (*PMML44Super.GeneralizedExponentialKernel method*), 110
`set_description()` (*PMML44Super.Header method*), 111
`set_description()` (*PMML44Super.LinearKernelType method*), 132
`set_description()` (*PMML44Super.PolynomialKernelType method*), 199
`set_description()` (*PMML44Super.RadialBasisKernel method*), 209
`set_description()` (*PMML44Super.RadialBasisKernelType method*), 210
`set_description()` (*PMML44Super.SigmoidKernelType method*), 239
`set_df()` (*PMML44Super.MultivariateStat method*), 153
`set_df()` (*PMML44Super.PCell method*), 183
`set_diagDefault()` (*PMML44Super.Matrix method*), 140
`set_difference()` (*PMML44Super.NonseasonalFactor method*), 170
`set_difference()` (*PMML44Super.SeasonalFactor method*), 227
`set_dilationRate()` (*PMML44Super.LayerParameters method*), 127
`set_DiscreteConditionalProbability()` (*PMML44Super.DiscreteNode method*), 87
`set_DiscreteNode()` (*PMML44Super.BayesianNetworkNodes method*), 37
`set_Discretize()` (*PMML44Super.Apply method*), 22
`set_Discretize()` (*PMML44Super.ComplexPartialScore method*), 55
`set_Discretize()` (*PMML44Super.DefineFunction method*), 80
`set_Discretize()` (*PMML44Super.DerivedField method*), 84
`set_Discretize()` (*PMML44Super.Lower method*), 136
`set_Discretize()` (*PMML44Super.Mean method*), 142
`set_Discretize()` (*PMML44Super.OutputField method*), 181
`set_Discretize()` (*PMML44Super.TextIndex method*), 262
`set_Discretize()` (*PMML44Super.Upper method*), 288
`set_Discretize()` (*PMML44Super.Variance method*), 291
`set_DiscretizeBin()` (*PMML44Super.Discretize method*), 88
`set_DiscrStats()` (*PMML44Super.UnivariateStats method*), 287
`set_displayName()` (*PMML44Super.DataField method*), 74
`set_displayName()` (*PMML44Super.DerivedField method*), 84
`set_displayName()` (*PMML44Super.OutputField method*), 182
`set_displayName()` (*PMML44Super.ParameterField method*), 193
`set_displayName()` (*PMML44Super.ResultField method*), 218
`set_displayName()` (*PMML44Super.TimeAnchor method*), 270
`set_displayName()` (*PMML44Super.TimeCycle method*), 271
`set_displayValue()` (*PMML44Super.Decision method*), 75
`set_displayValue()` (*PMML44Super.TargetValue method*), 252
`set_displayValue()` (*PMML44Super.Value method*), 289
`set_distParameter()` (*PMML44Super.GeneralRegressionModel method*), 108
`set_distribution()` (*PMML44Super.GeneralRegressionModel method*), 108
`set_documentNormalization()` (*PMML44Super.TextModelNormalization method*), 266
`set_DocumentTermMatrix()` (*PMML44Super.TextModel method*), 265
`set_dropoutNoiseShape()` (*PMML44Super.LayerParameters method*), 127
`set_dropoutRate()` (*PMML44Super.LayerParameters method*), 127
`set_dropoutSeed()` (*PMML44Super.LayerParameters method*), 127
`set_DynamicRegressor()` (*PMML44Super.ARIMA method*), 5
`set_DynamicRegressor()`

(PMML44Super.StateSpaceModel method), 244
 set_endTime() (PMML44Super.TimeSeries method), 272
 set_endTimeVariable() (PMML44Super.GeneralRegressionModel method), 108
 set_epsilon() (PMML44Super.Adadelta method), 8
 set_epsilon() (PMML44Super.Adagrad method), 9
 set_epsilon() (PMML44Super.Adam method), 10
 set_epsilon() (PMML44Super.Adamax method), 11
 set_epsilon() (PMML44Super.Nadam method), 155
 set_epsilon() (PMML44Super.RMSprop method), 206
 set_euclidean() (PMML44Super.ComparisonMeasure method), 53
 set_EventValues() (PMML44Super.GeneralRegressionModel method), 107
 set_exponent() (PMML44Super.MultivariateStat method), 153
 set_exponent() (PMML44Super.NumericPredictor method), 176
 set_ExponentialSmoothing() (PMML44Super.TimeSeriesModel method), 274
 set_extender() (PMML44Super.Extension method), 93
 set_Extension() (PMML44Super.AbsoluteExponentialKernel method), 8
 set_Extension() (PMML44Super.Adadelta method), 8
 set_Extension() (PMML44Super.Adagrad method), 9
 set_Extension() (PMML44Super.Adam method), 10
 set_Extension() (PMML44Super.Adamax method), 11
 set_Extension() (PMML44Super.Aggregate method), 12
 set_Extension() (PMML44Super.Annotation method), 13
 set_Extension() (PMML44Super.AnomalyDetectionModel method), 15
 set_Extension() (PMML44Super.Anova method), 17
 set_Extension() (PMML44Super.AnovaRow method), 18
 set_Extension() (PMML44Super.AntecedentSequence method), 18
 set_Extension() (PMML44Super.AnyDistribution method), 19
 set_Extension() (PMML44Super.Application method), 20
 set_Extension() (PMML44Super.Apply method), 22
 set_Extension() (PMML44Super.AR method), 3
 set_Extension() (PMML44Super.ARDSquaredExponentialKernel method), 4
 set_Extension() (PMML44Super.ARIMA method), 5
 set_Extension() (PMML44Super.ARMAPart method), 7
 set_Extension() (PMML44Super.AssociationModel method), 25
 set_Extension() (PMML44Super.AssociationRule method), 26
 set_Extension() (PMML44Super.Attribute method), 27
 set_Extension() (PMML44Super.BaseCumHazardTables method), 28
 set_Extension() (PMML44Super.BaselineCell method), 30
 set_Extension() (PMML44Super.BaselineModel method), 31
 set_Extension() (PMML44Super.BaselineStratum method), 32
 set_Extension() (PMML44Super.BayesianNetworkModel method), 36
 set_Extension() (PMML44Super.BayesianNetworkNodes method), 37
 set_Extension() (PMML44Super.BayesInput method), 34
 set_Extension() (PMML44Super.BayesInputs method), 34
 set_Extension() (PMML44Super.BayesOutput method), 35
 set_Extension() (PMML44Super.binarySimilarity method), 299
 set_Extension() (PMML44Super.BlockIndicator method), 38
 set_Extension() (PMML44Super.BoundaryValueMeans method), 39
 set_Extension() (PMML44Super.BoundaryValues method), 39
 set_Extension() (PMML44Super.CategoricalPredictor method), 41
 set_Extension() (PMML44Super.Categories method), 42
 set_Extension() (PMML44Super.Category method), 42
 set_Extension() (PMML44Super.Characteristic method), 43
 set_Extension() (PMML44Super.Characteristics method), 44
 set_Extension() (PMML44Super.chebychev method), 299
 set_Extension() (PMML44Super.ChildParent method), 299

<i>method</i>), 45		<i>set_Extension()</i> (PMML44Super.Covariances <i>method</i>), 71	
<i>set_Extension()</i> (PMML44Super.cityBlock <i>method</i>), 300		<i>set_Extension()</i> (PMML44Super.CovariateList <i>method</i>), 72	
<i>set_Extension()</i> (PMML44Super.ClassLabels <i>method</i>), 46		<i>set_Extension()</i> (PMML44Super.DataDictionary <i>method</i>), 73	
<i>set_Extension()</i> (PMML44Super.Cluster <i>method</i>), 47		<i>set_Extension()</i> (PMML44Super.DataField <i>method</i>), 74	
<i>set_Extension()</i> (PMML44Super.ClusteringField <i>method</i>), 48		<i>set_Extension()</i> (PMML44Super.Decision <i>method</i>), 75	
<i>set_Extension()</i> (PMML44Super.ClusteringModel <i>method</i>), 49		<i>set_Extension()</i> (PMML44Super.Decisions <i>method</i>), 77	
<i>set_Extension()</i> (PMML44Super.ClusteringModelQuality <i>method</i>), 51		<i>set_Extension()</i> (PMML44Super.DecisionTree <i>method</i>), 76	
<i>set_Extension()</i> (PMML44Super.Coefficient <i>method</i>), 51		<i>set_Extension()</i> (PMML44Super.DeepNetwork <i>method</i>), 79	
<i>set_Extension()</i> (PMML44Super.Coefficients <i>method</i>), 52		<i>set_Extension()</i> (PMML44Super.DefineFunction <i>method</i>), 80	
<i>set_Extension()</i> (PMML44Super.ComparisonMeasure <i>method</i>), 53		<i>set_Extension()</i> (PMML44Super.Delimiter <i>method</i>), 81	
<i>set_Extension()</i> (PMML44Super.Comparisons <i>method</i>), 54		<i>set_Extension()</i> (PMML44Super.Denominator <i>method</i>), 82	
<i>set_Extension()</i> (PMML44Super.ComplexPartialScore <i>method</i>), 55		<i>set_Extension()</i> (PMML44Super.DerivedField <i>method</i>), 84	
<i>set_Extension()</i> (PMML44Super.CompoundPredicate <i>method</i>), 57		<i>set_Extension()</i> (PMML44Super.DiscreteConditionalProbability <i>method</i>), 86	
<i>set_Extension()</i> (PMML44Super.CompoundRule <i>method</i>), 58		<i>set_Extension()</i> (PMML44Super.DiscreteNode <i>method</i>), 87	
<i>set_Extension()</i> (PMML44Super.Con <i>method</i>), 59		<i>set_Extension()</i> (PMML44Super.Discretize <i>method</i>), 88	
<i>set_Extension()</i> (PMML44Super.ConfusionMatrix <i>method</i>), 60		<i>set_Extension()</i> (PMML44Super.DiscretizeBin <i>method</i>), 89	
<i>set_Extension()</i> (PMML44Super.ConsequentSequence <i>method</i>), 60		<i>set_Extension()</i> (PMML44Super.DiscrStats <i>method</i>), 85	
<i>set_Extension()</i> (PMML44Super.Constraints <i>method</i>), 62		<i>set_Extension()</i> (PMML44Super.DocumentTermMatrix <i>method</i>), 89	
<i>set_Extension()</i> (PMML44Super.ContinuousConditionalProbability <i>method</i>), 65		<i>set_Extension()</i> (PMML44Super.DynamicRegressor <i>method</i>), 90	
<i>set_Extension()</i> (PMML44Super.ContinuousDistribution <i>method</i>), 65		<i>set_Extension()</i> (PMML44Super.euclidean <i>method</i>), 301	
<i>set_Extension()</i> (PMML44Super.ContinuousNode <i>method</i>), 67		<i>set_Extension()</i> (PMML44Super.EventValues <i>method</i>), 91	
<i>set_Extension()</i> (PMML44Super.ContStats <i>method</i>), 64		<i>set_Extension()</i> (PMML44Super.FactorList <i>method</i>), 94	
<i>set_Extension()</i> (PMML44Super.CorrelationFields <i>method</i>), 67		<i>set_Extension()</i> (PMML44Super.False_ <i>method</i>), 94	
<i>set_Extension()</i> (PMML44Super.CorrelationMethods <i>method</i>), 68		<i>set_Extension()</i> (PMML44Super.FieldColumnPair <i>method</i>), 95	
<i>set_Extension()</i> (PMML44Super.Correlations <i>method</i>), 70		<i>set_Extension()</i> (PMML44Super.FieldRef <i>method</i>), 96	
<i>set_Extension()</i> (PMML44Super.CorrelationValues <i>method</i>), 69		<i>set_Extension()</i> (PMML44Super.FieldValue <i>method</i>), 97	
<i>set_Extension()</i> (PMML44Super.COUNT_TABLE_TYPE <i>method</i>), 40		<i>set_Extension()</i> (PMML44Super.FieldValueCount <i>method</i>), 98	
<i>set_Extension()</i> (PMML44Super.Counts <i>method</i>), 70			

`set_Extension()` (*PMML44Super.GARCH method*), [102](#)
`set_Extension()` (*PMML44Super.GARCHPart method*), [102](#)
`set_Extension()` (*PMML44Super.GaussianDistribution method*), [103](#)
`set_Extension()` (*PMML44Super.GaussianProcessModel method*), [105](#)
`set_Extension()` (*PMML44Super.GeneralizedExponentialKernel method*), [109](#)
`set_Extension()` (*PMML44Super.GeneralRegressionModel method*), [107](#)
`set_Extension()` (*PMML44Super.Header method*), [111](#)
`set_Extension()` (*PMML44Super.InlineTable method*), [114](#)
`set_Extension()` (*PMML44Super.InstanceField method*), [114](#)
`set_Extension()` (*PMML44Super.InstanceFields method*), [115](#)
`set_Extension()` (*PMML44Super.Interval method*), [116](#)
`set_Extension()` (*PMML44Super.Item method*), [117](#)
`set_Extension()` (*PMML44Super.ItemRef method*), [117](#)
`set_Extension()` (*PMML44Super.Itemset method*), [118](#)
`set_Extension()` (*PMML44Super.jaccard method*), [301](#)
`set_Extension()` (*PMML44Super.KNNInput method*), [119](#)
`set_Extension()` (*PMML44Super.KNNInputs method*), [120](#)
`set_Extension()` (*PMML44Super.KohonenMap method*), [121](#)
`set_Extension()` (*PMML44Super.Lag method*), [122](#)
`set_Extension()` (*PMML44Super.Lambda method*), [123](#)
`set_Extension()` (*PMML44Super.LayerBias method*), [124](#)
`set_Extension()` (*PMML44Super.LayerParameters method*), [126](#)
`set_Extension()` (*PMML44Super.LayerWeights method*), [129](#)
`set_Extension()` (*PMML44Super.LiftData method*), [130](#)
`set_Extension()` (*PMML44Super.LiftGraph method*), [131](#)
`set_Extension()` (*PMML44Super.LinearKernelType method*), [132](#)
`set_Extension()` (*PMML44Super.LinearNorm method*), [132](#)
`set_Extension()` (*PMML44Super.LocalTransformations method*), [133](#)
`set_Extension()` (*PMML44Super.LognormalDistributionForBN method*), [134](#)
`set_Extension()` (*PMML44Super.Losses method*), [135](#)
`set_Extension()` (*PMML44Super.Lower method*), [136](#)
`set_Extension()` (*PMML44Super.MA method*), [136](#)
`set_Extension()` (*PMML44Super.MACoefficients method*), [137](#)
`set_Extension()` (*PMML44Super.MapValues method*), [138](#)
`set_Extension()` (*PMML44Super.Mean method*), [142](#)
`set_Extension()` (*PMML44Super.MeasurementMatrix method*), [143](#)
`set_Extension()` (*PMML44Super.Metrics method*), [143](#)
`set_Extension()` (*PMML44Super.MiningBuildTask method*), [144](#)
`set_Extension()` (*PMML44Super.MiningField method*), [145](#)
`set_Extension()` (*PMML44Super.MiningModel method*), [147](#)
`set_Extension()` (*PMML44Super.MiningSchema method*), [148](#)
`set_Extension()` (*PMML44Super.minkowski method*), [302](#)
`set_Extension()` (*PMML44Super.MissingValueWeights method*), [148](#)
`set_Extension()` (*PMML44Super.ModelExplanation method*), [149](#)
`set_Extension()` (*PMML44Super.ModelLiftGraph method*), [150](#)
`set_Extension()` (*PMML44Super.ModelStats method*), [151](#)
`set_Extension()` (*PMML44Super.ModelVerification method*), [152](#)
`set_Extension()` (*PMML44Super.MultivariateStat method*), [153](#)
`set_Extension()` (*PMML44Super.MultivariateStats method*), [154](#)
`set_Extension()` (*PMML44Super.Nadam method*), [155](#)
`set_Extension()` (*PMML44Super.NaiveBayesModel method*), [156](#)
`set_Extension()` (*PMML44Super.NearestNeighborModel method*), [158](#)
`set_Extension()` (*PMML44Super.NetworkLayer method*), [160](#)
`set_Extension()` (*PMML44Super.NeuralInput method*), [161](#)
`set_Extension()` (*PMML44Super.NeuralInputs method*), [161](#)

<code>set_Extension()</code> (PMML44Super.NeuralLayer method), 162	<code>set_Extension()</code> (PMML44Super.PCell method), 183
<code>set_Extension()</code> (PMML44Super.NeuralNetwork method), 164	<code>set_Extension()</code> (PMML44Super.PCovCell method), 184
<code>set_Extension()</code> (PMML44Super.NeuralOutput method), 165	<code>set_Extension()</code> (PMML44Super.PCovMatrix method), 185
<code>set_Extension()</code> (PMML44Super.NeuralOutputs method), 166	<code>set_Extension()</code> (PMML44Super.PMML method), 188
<code>set_Extension()</code> (PMML44Super.Neuron method), 167	<code>set_Extension()</code> (PMML44Super.PoissonDistribution method), 198
<code>set_Extension()</code> (PMML44Super.Node method), 168	<code>set_Extension()</code> (PMML44Super.PolynomialKernelType method), 199
<code>set_Extension()</code> (PMML44Super.NonseasonalComponent method), 170	<code>set_Extension()</code> (PMML44Super.PPCell method), 189
<code>set_Extension()</code> (PMML44Super.NonseasonalFactor method), 170	<code>set_Extension()</code> (PMML44Super.PPMatrix method), 190
<code>set_Extension()</code> (PMML44Super.NormalDistributionForBN method), 173	<code>set_Extension()</code> (PMML44Super.PredictiveModelQuality method), 201
<code>set_Extension()</code> (PMML44Super.NormContinuous method), 171	<code>set_Extension()</code> (PMML44Super.Predictor method), 202
<code>set_Extension()</code> (PMML44Super.NormDiscrete method), 172	<code>set_Extension()</code> (PMML44Super.PredictorTerm method), 203
<code>set_Extension()</code> (PMML44Super.Numerator method), 174	<code>set_Extension()</code> (PMML44Super.PsiVector method), 204
<code>set_Extension()</code> (PMML44Super.NumericInfo method), 175	<code>set_Extension()</code> (PMML44Super.Quantile method), 204
<code>set_Extension()</code> (PMML44Super.NumericPredictor method), 176	<code>set_Extension()</code> (PMML44Super.RadialBasisKernel method), 209
<code>set_Extension()</code> (PMML44Super.Optimizers method), 177	<code>set_Extension()</code> (PMML44Super.RadialBasisKernelType method), 210
<code>set_Extension()</code> (PMML44Super.OptimumLiftGraph method), 178	<code>set_Extension()</code> (PMML44Super.RandomLiftGraph method), 210
<code>set_Extension()</code> (PMML44Super.OutlierEffect method), 179	<code>set_Extension()</code> (PMML44Super.Regression method), 212
<code>set_Extension()</code> (PMML44Super.Output method), 180	<code>set_Extension()</code> (PMML44Super.RegressionModel method), 213
<code>set_Extension()</code> (PMML44Super.OutputField method), 181	<code>set_Extension()</code> (PMML44Super.RegressionTable method), 215
<code>set_Extension()</code> (PMML44Super.PairCounts method), 191	<code>set_Extension()</code> (PMML44Super.RegressorValues method), 216
<code>set_Extension()</code> (PMML44Super.Parameter method), 192	<code>set_Extension()</code> (PMML44Super.Residuals method), 217
<code>set_Extension()</code> (PMML44Super.ParameterList method), 194	<code>set_Extension()</code> (PMML44Super.ResidualSquareCoefficients method), 216
<code>set_Extension()</code> (PMML44Super.ParamMatrix method), 192	<code>set_Extension()</code> (PMML44Super.ResultField method), 218
<code>set_Extension()</code> (PMML44Super.ParentValue method), 195	<code>set_Extension()</code> (PMML44Super.RMSprop method), 206
<code>set_Extension()</code> (PMML44Super.Partition method), 196	<code>set_Extension()</code> (PMML44Super.ROC method), 207
<code>set_Extension()</code> (PMML44Super.PartitionFieldStats method), 197	<code>set_Extension()</code> (PMML44Super.ROCGraph method), 208
<code>set_Extension()</code> (PMML44Super.PastVariances method), 197	<code>set_Extension()</code> (PMML44Super.RuleSelectionMethod method), 219

<code>set_Extension()</code> (<i>PMML44Super.RuleSet</i> method), 220	<code>set_Extension()</code> (<i>PMML44Super.TableLocator</i> method), 250
<code>set_Extension()</code> (<i>PMML44Super.RuleSetModel</i> method), 221	<code>set_Extension()</code> (<i>PMML44Super.tanimoto</i> method), 304
<code>set_Extension()</code> (<i>PMML44Super.Scorecard</i> method), 224	<code>set_Extension()</code> (<i>PMML44Super.Target</i> method), 251
<code>set_Extension()</code> (<i>PMML44Super.ScoreDistribution</i> method), 223	<code>set_Extension()</code> (<i>PMML44Super.Targets</i> method), 256
<code>set_Extension()</code> (<i>PMML44Super.SeasonalComponents</i> method), 226	<code>set_Extension()</code> (<i>PMML44Super.TargetValue</i> method), 252
<code>set_Extension()</code> (<i>PMML44Super.SeasonalFactor</i> method), 227	<code>set_Extension()</code> (<i>PMML44Super.TargetValueCount</i> method), 253
<code>set_Extension()</code> (<i>PMML44Super.Segment</i> method), 230	<code>set_Extension()</code> (<i>PMML44Super.TargetValueCounts</i> method), 254
<code>set_Extension()</code> (<i>PMML44Super.Segmentation</i> method), 231	<code>set_Extension()</code> (<i>PMML44Super.TargetValueStat</i> method), 255
<code>set_Extension()</code> (<i>PMML44Super.Sequence</i> method), 232	<code>set_Extension()</code> (<i>PMML44Super.TargetValueStats</i> method), 255
<code>set_Extension()</code> (<i>PMML44Super.SequenceModel</i> method), 234	<code>set_Extension()</code> (<i>PMML44Super.Taxonomy</i> method), 257
<code>set_Extension()</code> (<i>PMML44Super.SequenceReference</i> method), 235	<code>set_Extension()</code> (<i>PMML44Super.TestDistributions</i> method), 258
<code>set_Extension()</code> (<i>PMML44Super.SequenceRule</i> method), 237	<code>set_Extension()</code> (<i>PMML44Super.TextCorpus</i> method), 259
<code>set_Extension()</code> (<i>PMML44Super.SetPredicate</i> method), 238	<code>set_Extension()</code> (<i>PMML44Super.TextDictionary</i> method), 259
<code>set_Extension()</code> (<i>PMML44Super.SetReference</i> method), 238	<code>set_Extension()</code> (<i>PMML44Super.TextDocument</i> method), 260
<code>set_Extension()</code> (<i>PMML44Super.SGD</i> method), 222	<code>set_Extension()</code> (<i>PMML44Super.TextIndex</i> method), 262
<code>set_Extension()</code> (<i>PMML44Super.SigmoidKernelType</i> method), 239	<code>set_Extension()</code> (<i>PMML44Super.TextIndexNormalization</i> method), 263
<code>set_Extension()</code> (<i>PMML44Super.simpleMatching</i> method), 303	<code>set_Extension()</code> (<i>PMML44Super.TextModel</i> method), 265
<code>set_Extension()</code> (<i>PMML44Super.SimplePredicate</i> method), 240	<code>set_Extension()</code> (<i>PMML44Super.TextModelNormalization</i> method), 266
<code>set_Extension()</code> (<i>PMML44Super.SimpleRule</i> method), 241	<code>set_Extension()</code> (<i>PMML44Super.TextModelSimiliarity</i> method), 267
<code>set_Extension()</code> (<i>PMML44Super.SimpleSetPredicate</i> method), 242	<code>set_Extension()</code> (<i>PMML44Super.Time</i> method), 269
<code>set_Extension()</code> (<i>PMML44Super.squaredEuclidean</i> method), 304	<code>set_Extension()</code> (<i>PMML44Super.TimeSeriesModel</i> method), 274
<code>set_Extension()</code> (<i>PMML44Super.StateSpaceModel</i> method), 244	<code>set_Extension()</code> (<i>PMML44Super.Timestamp</i> method), 276
<code>set_Extension()</code> (<i>PMML44Super.StateVector</i> method), 245	<code>set_Extension()</code> (<i>PMML44Super.TrainingInstances</i> method), 277
<code>set_Extension()</code> (<i>PMML44Super.SupportVector</i> method), 245	<code>set_Extension()</code> (<i>PMML44Super.TrainingParameters</i> method), 278
<code>set_Extension()</code> (<i>PMML44Super.SupportVectorMachine</i> method), 246	<code>set_Extension()</code> (<i>PMML44Super.TransformationDictionary</i> method), 279
<code>set_Extension()</code> (<i>PMML44Super.SupportVectorMachineModel</i> method), 248	<code>set_Extension()</code> (<i>PMML44Super.TransitionMatrix</i> method), 280
<code>set_Extension()</code> (<i>PMML44Super.SupportVectors</i> method), 249	<code>set_Extension()</code> (<i>PMML44Super.TreeModel</i> method), 281

set_Extension() (PMML44Super.TrendCoefficients method), 282
 set_Extension() (PMML44Super.TriangularDistributionForBN method), 284
 set_Extension() (PMML44Super.True_ method), 284
 set_Extension() (PMML44Super.UniformDistribution method), 285
 set_Extension() (PMML44Super.UniformDistributionForBN method), 286
 set_Extension() (PMML44Super.UnivariateStats method), 287
 set_Extension() (PMML44Super.Upper method), 288
 set_Extension() (PMML44Super.Value method), 289
 set_Extension() (PMML44Super.ValueProbability method), 289
 set_Extension() (PMML44Super.VariableWeight method), 290
 set_Extension() (PMML44Super.Variance method), 291
 set_Extension() (PMML44Super.VarianceCoefficients method), 292
 set_Extension() (PMML44Super.VectorDictionary method), 293
 set_Extension() (PMML44Super.VectorFields method), 294
 set_Extension() (PMML44Super.VectorInstance method), 295
 set_Extension() (PMML44Super.VerificationField method), 296
 set_Extension() (PMML44Super.VerificationFields method), 296
 set_Extension() (PMML44Super.XCoordinates method), 297
 set_Extension() (PMML44Super.YCoordinates method), 298
 set_FactorList() (PMML44Super.GeneralRegressionModel method), 107
 set_False() (PMML44Super.Attribute method), 27
 set_False() (PMML44Super.CompoundPredicate method), 57
 set_False() (PMML44Super.CompoundRule method), 58
 set_False() (PMML44Super.Node method), 168
 set_False() (PMML44Super.Segment method), 230
 set_False() (PMML44Super.SimpleRule method), 241
 set_feature() (PMML44Super.OutputField method), 182
 set_feature() (PMML44Super.ResultField method), 218
 set_featureMaps() (PMML44Super.LayerParameters method), 127
 set_field() (PMML44Super.Aggregate method), 12
 set_field() (PMML44Super.BlockIndicator method), 38
 set_field() (PMML44Super.ClusteringField method), 48
 set_field() (PMML44Super.Discretize method), 88
 set_field() (PMML44Super.DynamicRegressor method), 90
 set_field() (PMML44Super.FieldColumnPair method), 95
 set_field() (PMML44Super.FieldRef method), 96
 set_field() (PMML44Super.FieldValue method), 97
 set_field() (PMML44Super.FieldValueCount method), 98
 set_field() (PMML44Super.InstanceField method), 114
 set_field() (PMML44Super.Item method), 117
 set_field() (PMML44Super.KNNInput method), 119
 set_field() (PMML44Super.Lag method), 122
 set_field() (PMML44Super.NormContinuous method), 171
 set_field() (PMML44Super.NormDiscrete method), 172
 set_field() (PMML44Super.PartitionFieldStats method), 197
 set_field() (PMML44Super.SetPredicate method), 238
 set_field() (PMML44Super.SimplePredicate method), 240
 set_field() (PMML44Super.SimpleSetPredicate method), 242
 set_field() (PMML44Super.Target method), 251
 set_field() (PMML44Super.TestDistributions method), 258
 set_field() (PMML44Super.TimeSeries method), 272
 set_field() (PMML44Super.UnivariateStats method), 287
 set_field() (PMML44Super.VariableWeight method), 290
 set_field() (PMML44Super.VerificationField method), 296
 set_FieldColumnPair() (PMML44Super.ChildParent method), 45
 set_FieldColumnPair() (PMML44Super.MapValues method), 138
 set_fieldCount() (PMML44Super.ModelVerification method), 152
 set_fieldCount() (PMML44Super.TrainingInstances method), 277
 set_fieldName() (PMML44Super.BayesInput method), 33

[set_fieldName\(\)](#) (PMML44Super.BayesOutput method), 35
[set_FieldRef\(\)](#) (PMML44Super.Apply method), 22
[set_FieldRef\(\)](#) (PMML44Super.Baseline method), 29
[set_FieldRef\(\)](#) (PMML44Super.ComplexPartialScore method), 55
[set_FieldRef\(\)](#) (PMML44Super.DefineFunction method), 80
[set_FieldRef\(\)](#) (PMML44Super.DerivedField method), 84
[set_FieldRef\(\)](#) (PMML44Super.Lower method), 136
[set_FieldRef\(\)](#) (PMML44Super.Mean method), 142
[set_FieldRef\(\)](#) (PMML44Super.OutputField method), 181
[set_FieldRef\(\)](#) (PMML44Super.PredictorTerm method), 203
[set_FieldRef\(\)](#) (PMML44Super.TextIndex method), 262
[set_FieldRef\(\)](#) (PMML44Super.Upper method), 288
[set_FieldRef\(\)](#) (PMML44Super.Variance method), 291
[set_FieldRef\(\)](#) (PMML44Super.VectorFields method), 294
[set_FieldValue\(\)](#) (PMML44Super.COUNT_TABLE_TYPE method), 40
[set_FieldValue\(\)](#) (PMML44Super.FieldValue method), 97
[set_FieldValueCount\(\)](#) (PMML44Super.COUNT_TABLE_TYPE method), 40
[set_FieldValueCount\(\)](#) (PMML44Super.FieldValue method), 97
[set_fieldWeight\(\)](#) (PMML44Super.ClusteringField method), 48
[set_fieldWeight\(\)](#) (PMML44Super.KNNInput method), 119
[set_file\(\)](#) (PMML44Super.TextDocument method), 260
[set_FinalNoise\(\)](#) (PMML44Super.ThetaRecursionState method), 268
[set_FinalNu\(\)](#) (PMML44Super.ThetaRecursionState method), 268
[set_FinalOmega\(\)](#) (PMML44Super.KalmanState method), 121
[set_FinalPredictedNoise\(\)](#) (PMML44Super.ThetaRecursionState method), 268
[set_FinalStateVector\(\)](#) (PMML44Super.KalmanState method), 121
[set_FinalTheta\(\)](#) (PMML44Super.ThetaRecursionState method), 268
[set_flattenAxis\(\)](#) (PMML44Super.LayerParameters method), 127
[set_framework\(\)](#) (PMML44Super.TrainingParameters method), 278
[set_from\(\)](#) (PMML44Super.Con method), 59
[set_fStatistic\(\)](#) (PMML44Super.MultivariateStat method), 153
[set_fStatistic\(\)](#) (PMML44Super.PredictiveModelQuality method), 201
[set_function\(\)](#) (PMML44Super.Aggregate method), 12
[set_function\(\)](#) (PMML44Super.Apply method), 22
[set_functionName\(\)](#) (PMML44Super.AnomalyDetectionModel method), 16
[set_functionName\(\)](#) (PMML44Super.AssociationModel method), 25
[set_functionName\(\)](#) (PMML44Super.BaselineModel method), 31
[set_functionName\(\)](#) (PMML44Super.BayesianNetworkModel method), 36
[set_functionName\(\)](#) (PMML44Super.ClusteringModel method), 50
[set_functionName\(\)](#) (PMML44Super.DecisionTree method), 76
[set_functionName\(\)](#) (PMML44Super.DeepNetwork method), 79
[set_functionName\(\)](#) (PMML44Super.GaussianProcessModel method), 105
[set_functionName\(\)](#) (PMML44Super.GeneralRegressionModel method), 108
[set_functionName\(\)](#) (PMML44Super.MiningModel method), 147
[set_functionName\(\)](#) (PMML44Super.NaiveBayesModel method), 157
[set_functionName\(\)](#) (PMML44Super.NearestNeighborModel method), 158
[set_functionName\(\)](#) (PMML44Super.NeuralNetwork method), 164
[set_functionName\(\)](#) (PMML44Super.Regression method), 212
[set_functionName\(\)](#) (PMML44Super.RegressionModel method),

[213](#)
 set_functionName() (PMML44Super.RuleSetModel method), [221](#)
 set_functionName() (PMML44Super.Scorecard method), [225](#)
 set_functionName() (PMML44Super.SequenceModel method), [235](#)
 set_functionName() (PMML44Super.SupportVectorMachineModel method), [248](#)
 set_functionName() (PMML44Super.TextModel method), [265](#)
 set_functionName() (PMML44Super.TimeSeriesModel method), [274](#)
 set_functionName() (PMML44Super.TreeModel method), [281](#)
 set_futureValuesMethod() (PMML44Super.DynamicRegressor method), [90](#)
 set_fValue() (PMML44Super.AnovaRow method), [18](#)
 set_gamma() (PMML44Super.AbsoluteExponentialKernel method), [8](#)
 set_gamma() (PMML44Super.ARDSquaredExponentialKernel method), [4](#)
 set_gamma() (PMML44Super.GeneralizedExponentialKernel method), [110](#)
 set_gamma() (PMML44Super.PolynomialKernelType method), [199](#)
 set_gamma() (PMML44Super.RadialBasisKernel method), [209](#)
 set_gamma() (PMML44Super.RadialBasisKernelType method), [210](#)
 set_gamma() (PMML44Super.SigmoidKernelType method), [239](#)
 set_gamma() (PMML44Super.Trend_ExpoSmooth method), [283](#)
 set_gammaConstraint() (PMML44Super.LayerParameters method), [127](#)
 set_gammaInitializer() (PMML44Super.LayerParameters method), [127](#)
 set_gammaRegularizer() (PMML44Super.LayerParameters method), [127](#)
 set_gap() (PMML44Super.Delimiter method), [81](#)
 set_GARCH() (PMML44Super.TimeSeriesModel method), [274](#)
 set_GARCHPart() (PMML44Super.GARCH method), [102](#)
 set_GaussianDistribution() (PMML44Super.Alternate method), [13](#)
 set_GaussianDistribution() (PMML44Super.Baseline method), [29](#)
 set_GaussianDistribution() (PMML44Super.TargetValueStat method), [255](#)
 set_gaussianDropoutRate() (PMML44Super.LayerParameters method), [127](#)
 set_gaussianNoiseStddev() (PMML44Super.LayerParameters method), [127](#)
 set_GaussianProcessModel() (PMML44Super.AnomalyDetectionModel method), [15](#)
 set_GaussianProcessModel() (PMML44Super.PMML method), [188](#)
 set_GaussianProcessModel() (PMML44Super.Segment method), [230](#)
 set_GeneralizedExponentialKernel() (PMML44Super.GaussianProcessModel method), [105](#)
 set_generalLUAlpha() (PMML44Super.LayerParameters method), [127](#)
 set_GeneralRegressionModel() (PMML44Super.AnomalyDetectionModel method), [15](#)
 set_GeneralRegressionModel() (PMML44Super.PMML method), [188](#)
 set_GeneralRegressionModel() (PMML44Super.Segment method), [230](#)
 set_globalTermWeights() (PMML44Super.TextModelNormalization method), [266](#)
 set_gp() (PMML44Super.GARCHPart method), [102](#)
 set_gq() (PMML44Super.GARCHPart method), [102](#)
 set_groupField() (PMML44Super.Aggregate method), [12](#)
 set_Header() (PMML44Super.PMML method), [188](#)
 set_highValue() (PMML44Super.MiningField method), [145](#)
 set_HVector() (PMML44Super.KalmanState method), [121](#)
 set_i() (PMML44Super.Theta method), [267](#)
 set_id() (PMML44Super.AssociationRule method), [26](#)
 set_id() (PMML44Super.Cluster method), [47](#)
 set_id() (PMML44Super.Item method), [117](#)
 set_id() (PMML44Super.Itemset method), [118](#)
 set_id() (PMML44Super.NeuralInput method), [161](#)
 set_id() (PMML44Super.Neuron method), [167](#)
 set_id() (PMML44Super.Node method), [169](#)

- [set_id\(\) \(PMML44Super.Segment method\), 230](#)
[set_id\(\) \(PMML44Super.Sequence method\), 232](#)
[set_id\(\) \(PMML44Super.SequenceRule method\), 237](#)
[set_id\(\) \(PMML44Super.SetPredicate method\), 238](#)
[set_id\(\) \(PMML44Super.SimpleRule method\), 241](#)
[set_id\(\) \(PMML44Super.TextDocument method\), 260](#)
[set_id\(\) \(PMML44Super.VectorInstance method\), 295](#)
[set_importance\(\) \(PMML44Super.MiningField method\), 145](#)
[set_importance\(\) \(PMML44Super.MultivariateStat method\), 153](#)
[set_index\(\) \(PMML44Super.TimeValue method\), 275](#)
[set_Indices\(\) \(PMML44Super.INT_SparseArray method\), 112](#)
[set_Indices\(\) \(PMML44Super.REAL_SparseArray method\), 205](#)
[set_inferenceMethod\(\) \(PMML44Super.BayesianNetworkModel method\), 36](#)
[set_inField\(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[set_initialScore\(\) \(PMML44Super.Scorecard method\), 225](#)
[set_InlineTable\(\) \(PMML44Super.ChildParent method\), 45](#)
[set_InlineTable\(\) \(PMML44Super.MapValues method\), 138](#)
[set_InlineTable\(\) \(PMML44Super.ModelVerification method\), 152](#)
[set_InlineTable\(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[set_InlineTable\(\) \(PMML44Super.TrainingInstances method\), 277](#)
[set_inputDimension\(\) \(PMML44Super.LayerParameters method\), 127](#)
[set_inputFieldName\(\) \(PMML44Super.NetworkLayer method\), 160](#)
[set_InstanceField\(\) \(PMML44Super.InstanceFields method\), 115](#)
[set_InstanceFields\(\) \(PMML44Super.TrainingInstances method\), 277](#)
[set_instanceIdVariable\(\) \(PMML44Super.NearestNeighborModel method\), 158](#)
[set_INT_Entries\(\) \(PMML44Super.INT_SparseArray method\), 112](#)
[set_intercept\(\) \(PMML44Super.RegressionTable method\), 215](#)
[set_intercept\(\) \(PMML44Super.StateSpaceModel method\), 244](#)
[set_interpolationMethod\(\) \(PMML44Super.TimeSeries method\), 272](#)
[set_interQuartileRange\(\) \(PMML44Super.NumericInfo method\), 175](#)
[set_Interval\(\) \(PMML44Super.ContStats method\), 64](#)
[set_Interval\(\) \(PMML44Super.DataField method\), 74](#)
[set_Interval\(\) \(PMML44Super.DiscretizeBin method\), 89](#)
[set_Interval\(\) \(PMML44Super.EventValues method\), 91](#)
[set_invalidFreq\(\) \(PMML44Super.Counts method\), 70](#)
[set_invalidValueReplacement\(\) \(PMML44Super.MiningField method\), 145](#)
[set_invalidValueTreatment\(\) \(PMML44Super.Apply method\), 22](#)
[set_invalidValueTreatment\(\) \(PMML44Super.MiningField method\), 145](#)
[set_isCaseSensitive\(\) \(PMML44Super.TextIndex method\), 262](#)
[set_isCaseSensitive\(\) \(PMML44Super.TextIndexNormalization method\), 263](#)
[set_isCenterField\(\) \(PMML44Super.ClusteringField method\), 48](#)
[set_isCyclic\(\) \(PMML44Super.DataField method\), 74](#)
[set_isFinalResult\(\) \(PMML44Super.OutputField method\), 182](#)
[set_isIntercept\(\) \(PMML44Super.MultivariateStat method\), 153](#)
[set_isMultiValued\(\) \(PMML44Super.OutputField method\), 182](#)
[set_isRecursive\(\) \(PMML44Super.ChildParent method\), 45](#)
[set_isScorable\(\) \(PMML44Super.AssociationModel method\), 25](#)
[set_isScorable\(\) \(PMML44Super.BaselineModel method\), 31](#)
[set_isScorable\(\) \(PMML44Super.BayesianNetworkModel method\), 36](#)
[set_isScorable\(\) \(PMML44Super.ClusteringModel method\), 50](#)
[set_isScorable\(\) \(PMML44Super.DeepNetwork](#)

method), 79
 set_isScorable() (PMML44Super.GaussianProcessModel method), 105
 set_isScorable() (PMML44Super.GeneralRegressionModel method), 108
 set_isScorable() (PMML44Super.MiningModel method), 147
 set_isScorable() (PMML44Super.NaiveBayesModel method), 157
 set_isScorable() (PMML44Super.NearestNeighborModel method), 158
 set_isScorable() (PMML44Super.NeuralNetworkModel method), 164
 set_isScorable() (PMML44Super.RegressionModel method), 213
 set_isScorable() (PMML44Super.RuleSetModel method), 221
 set_isScorable() (PMML44Super.Scorecard method), 225
 set_isScorable() (PMML44Super.SequenceModel method), 235
 set_isScorable() (PMML44Super.SupportVectorMachineModel method), 248
 set_isScorable() (PMML44Super.TextModel method), 265
 set_isScorable() (PMML44Super.TimeSeriesModel method), 274
 set_isScorable() (PMML44Super.TreeModel method), 281
 set_isTransformed() (PMML44Super.TrainingInstances method), 277
 set_Item() (PMML44Super.AssociationModel method), 25
 set_Item() (PMML44Super.SequenceModel method), 234
 set_itemRef() (PMML44Super.ItemRef method), 117
 set_ItemRef() (PMML44Super.Itemset method), 118
 set_Itemset() (PMML44Super.AssociationModel method), 25
 set_Itemset() (PMML44Super.SequenceModel method), 234
 set_j() (PMML44Super.Theta method), 267
 set_jaccard() (PMML44Super.ComparisonMeasure method), 53
 set_KalmanState() (PMML44Super.MaximumLikelihoodStat method), 141
 set_kernel() (PMML44Super.LayerParameters method), 127
 set_kernelConstraint() (PMML44Super.LayerParameters method), 127
 set_kernelInitializer() (PMML44Super.LayerParameters method), 127
 set_kernelRegularizer() (PMML44Super.LayerParameters method), 127
 set_kind() (PMML44Super.ComparisonMeasure method), 53
 set_kind() (PMML44Super.Matrix method), 140
 set_KNNInput() (PMML44Super.KNNInputs method), 120
 set_KNNInputs() (PMML44Super.NearestNeighborModel method), 158
 set_KohonenMap() (PMML44Super.Cluster method), 47
 set_label() (PMML44Super.BaselineStratum method), 32
 set_label() (PMML44Super.Parameter method), 192
 set_Lag() (PMML44Super.Apply method), 22
 set_Lag() (PMML44Super.ComplexPartialScore method), 55
 set_Lag() (PMML44Super.DefineFunction method), 80
 set_Lag() (PMML44Super.DerivedField method), 84
 set_Lag() (PMML44Super.Lower method), 136
 set_Lag() (PMML44Super.Mean method), 142
 set_Lag() (PMML44Super.OutputField method), 181
 set_Lag() (PMML44Super.TextIndex method), 262
 set_Lag() (PMML44Super.Upper method), 288
 set_Lag() (PMML44Super.Variance method), 291
 set_Lambda() (PMML44Super.AbsoluteExponentialKernel method), 8
 set_Lambda() (PMML44Super.ARDSquaredExponentialKernel method), 4
 set_Lambda() (PMML44Super.GeneralizedExponentialKernel method), 109
 set_lambda() (PMML44Super.RadialBasisKernel method), 209
 set_LayerBias() (PMML44Super.NetworkLayer method), 160
 set_layerId() (PMML44Super.NetworkLayer method), 160
 set_LayerParameters() (PMML44Super.NetworkLayer method), 160
 set_layerType() (PMML44Super.NetworkLayer method), 160
 set_LayerWeights() (PMML44Super.NetworkLayer method), 160
 set_learningRate() (PMML44Super.Adadelata method), 8
 set_learningRate() (PMML44Super.Adagrad

`method`), 9
`set_learningRate()` (PMML44Super.Adam `method`), 10
`set_learningRate()` (PMML44Super.Adamax `method`), 11
`set_learningRate()` (PMML44Super.Nadam `method`), 155
`set_learningRate()` (PMML44Super.RMSprop `method`), 206
`set_learningRate()` (PMML44Super.SGD `method`), 222
`set_leftMargin()` (PMML44Super.Interval `method`), 116
`set_length()` (PMML44Super.TextDocument `method`), 260
`set_length()` (PMML44Super.TimeCycle `method`), 271
`set_lengthLimit()` (PMML44Super.AssociationModel `method`), 25
`set_Level()` (PMML44Super.ExponentialSmoothing `method`), 92
`set_leverage()` (PMML44Super.AssociationRule `method`), 26
`set_lift()` (PMML44Super.AssociationRule `method`), 26
`set_lift()` (PMML44Super.SequenceRule `method`), 237
`set_LiftData()` (PMML44Super.PredictiveModelQuality `method`), 201
`set_LiftGraph()` (PMML44Super.ModelLiftGraph `method`), 150
`set_LiftGraph()` (PMML44Super.OptimumLiftGraph `method`), 178
`set_LiftGraph()` (PMML44Super.RandomLiftGraph `method`), 210
`set_LinearKernelType()` (PMML44Super.SupportVectorMachineModel `method`), 248
`set_LinearNorm()` (PMML44Super.NormContinuous `method`), 171
`set_linkFunction()` (PMML44Super.GeneralRegressionModel `method`), 108
`set_linkParameter()` (PMML44Super.GeneralRegressionModel `method`), 108
`set_localTermWeights()` (PMML44Super.TextIndex `method`), 262
`set_localTermWeights()` (PMML44Super.TextModelNormalization `method`), 266
`set_LocalTransformations()` (PMML44Super.AnomalyDetectionModel `method`), 15
`set_LocalTransformations()` (PMML44Super.AssociationModel `method`), 25
`set_LocalTransformations()` (PMML44Super.BaselineModel `method`), 31
`set_LocalTransformations()` (PMML44Super.BayesianNetworkModel `method`), 36
`set_LocalTransformations()` (PMML44Super.ClusteringModel `method`), 49
`set_LocalTransformations()` (PMML44Super.DecisionTree `method`), 76
`set_LocalTransformations()` (PMML44Super.DeepNetwork `method`), 79
`set_LocalTransformations()` (PMML44Super.GaussianProcessModel `method`), 105
`set_LocalTransformations()` (PMML44Super.GeneralRegressionModel `method`), 108
`set_LocalTransformations()` (PMML44Super.MiningModel `method`), 147
`set_LocalTransformations()` (PMML44Super.NaiveBayesModel `method`), 156
`set_LocalTransformations()` (PMML44Super.NearestNeighborModel `method`), 158
`set_LocalTransformations()` (PMML44Super.NeuralNetwork `method`), 164
`set_LocalTransformations()` (PMML44Super.Regression `method`), 212
`set_LocalTransformations()` (PMML44Super.RegressionModel `method`), 213
`set_LocalTransformations()` (PMML44Super.RuleSetModel `method`), 221
`set_LocalTransformations()` (PMML44Super.Scorecard `method`), 224
`set_LocalTransformations()` (PMML44Super.SequenceModel `method`), 234
`set_LocalTransformations()` (PMML44Super.SupportVectorMachineModel `method`), 248
`set_LocalTransformations()` (PMML44Super.TextModel `method`), 265
`set_LocalTransformations()` (PMML44Super.TimeSeriesModel `method`), 274

`set_LocalTransformations()` (PMML44Super.TreeModel method), 281

`set_LognormalDistributionForBN()` (PMML44Super.ContinuousDistribution method), 65

`set_loss()` (PMML44Super.Losses method), 135

`set_Losses()` (PMML44Super.TrainingParameters method), 278

`set_Lower()` (PMML44Super.TriangularDistributionForBN method), 284

`set_lower()` (PMML44Super.UniformDistribution method), 285

`set_Lower()` (PMML44Super.UniformDistributionForBN method), 286

`set_lowValue()` (PMML44Super.MiningField method), 145

`set_MA()` (PMML44Super.ARMAPart method), 7

`set_MA()` (PMML44Super.NonseasonalComponent method), 170

`set_MA()` (PMML44Super.SeasonalComponent method), 226

`set_MACoefficients()` (PMML44Super.MA method), 136

`set_MACoefficients()` (PMML44Super.ResidualSquareCoefficients method), 216

`set_MACoefficients()` (PMML44Super.VarianceCoefficients method), 292

`set_magnitude()` (PMML44Super.OutlierEffect method), 179

`set_mapMissingTo()` (PMML44Super.Apply method), 22

`set_mapMissingTo()` (PMML44Super.Discretize method), 88

`set_mapMissingTo()` (PMML44Super.FieldRef method), 96

`set_mapMissingTo()` (PMML44Super.MapValues method), 138

`set_mapMissingTo()` (PMML44Super.NormContinuous method), 171

`set_mapMissingTo()` (PMML44Super.NormDiscrete method), 172

`set_mappedValue()` (PMML44Super.Item method), 117

`set_MapValues()` (PMML44Super.Apply method), 22

`set_MapValues()` (PMML44Super.ComplexPartialScore method), 56

`set_MapValues()` (PMML44Super.DefineFunction method), 80

`set_MapValues()` (PMML44Super.DerivedField method), 84

`set_MapValues()` (PMML44Super.Lower method), 136

`set_MapValues()` (PMML44Super.Mean method), 142

`set_MapValues()` (PMML44Super.OutputField method), 182

`set_MapValues()` (PMML44Super.TextIndex method), 262

`set_MapValues()` (PMML44Super.Upper method), 288

`set_MapValues()` (PMML44Super.Variance method), 291

`set_maskValue()` (PMML44Super.LayerParameters method), 127

`set_MatCell()` (PMML44Super.Matrix method), 140

`set_Matrix()` (PMML44Super.Comparisons method), 54

`set_Matrix()` (PMML44Super.ConfusionMatrix method), 60

`set_Matrix()` (PMML44Super.CorrelationMethods method), 68

`set_Matrix()` (PMML44Super.CorrelationValues method), 69

`set_Matrix()` (PMML44Super.Covariances method), 71

`set_Matrix()` (PMML44Super.DocumentTermMatrix method), 89

`set_Matrix()` (PMML44Super.FinalOmega method), 99

`set_Matrix()` (PMML44Super.MeasurementMatrix method), 143

`set_Matrix()` (PMML44Super.Predictor method), 202

`set_Matrix()` (PMML44Super.TransitionMatrix method), 280

`set_max()` (PMML44Super.Target method), 251

`set_max()` (PMML44Super.Time method), 269

`set_maximum()` (PMML44Super.ComparisonMeasure method), 53

`set_maximum()` (PMML44Super.NumericInfo method), 175

`set_maximumAntConsSeparationTime()` (PMML44Super.Constraints method), 62

`set_maximumItemsetSeparationTime()` (PMML44Super.Constraints method), 62

`set_MaximumLikelihoodStat()` (PMML44Super.ARIMA method), 6

`set_maximumNumberOfAntecedentItems()` (PMML44Super.Constraints method), 62

`set_maximumNumberOfConsequentItems()` (PMML44Super.Constraints method), 62

`set_maximumNumberOfItems()` (PMML44Super.Constraints method), 62

`set_maximumOrder()` (*PMML44Super.NonseasonalFactor* method), 171
`set_maximumOrder()` (*PMML44Super.SeasonalFactor* method), 227
`set_maximumTotalSequenceTime()` (*PMML44Super.Constraints* method), 62
`set_maxLevenshteinDistance()` (*PMML44Super.TextIndex* method), 262
`set_maxLevenshteinDistance()` (*PMML44Super.TextIndexNormalization* method), 263
`set_maxNumberOfItemsPerTA()` (*PMML44Super.AssociationModel* method), 25
`set_maxNumberOfItemsPerTransaction()` (*PMML44Super.SequenceModel* method), 235
`set_maxNumberOfTAsPerTAGroup()` (*PMML44Super.SequenceModel* method), 235
`set_maxTime()` (*PMML44Super.BaseCumHazardTables* method), 28
`set_maxTime()` (*PMML44Super.BaselineStratum* method), 32
`set_maxWins()` (*PMML44Super.SupportVectorMachineModel* method), 54
`set_mean()` (*PMML44Super.AnyDistribution* method), 19
`set_mean()` (*PMML44Super.GaussianDistribution* method), 103
`set_Mean()` (*PMML44Super.LognormalDistributionForBN* method), 134
`set_Mean()` (*PMML44Super.NormalDistributionForBN* method), 173
`set_mean()` (*PMML44Super.NumericInfo* method), 175
`set_mean()` (*PMML44Super.PoissonDistribution* method), 198
`set_mean()` (*PMML44Super.Time* method), 269
`set_Mean()` (*PMML44Super.TriangularDistributionForBN* method), 284
`set_meanAbsoluteError()` (*PMML44Super.PredictiveModelQuality* method), 201
`set_meanError()` (*PMML44Super.PredictiveModelQuality* method), 201
`set_meanOfSquares()` (*PMML44Super.AnovaRow* method), 18
`set_meanSquaredError()` (*PMML44Super.PredictiveModelQuality* method), 201
`set_MeasurementMatrix()` (*PMML44Super.StateSpaceModel* method), 244
`set_median()` (*PMML44Super.NumericInfo* method), 175
`set_mergeLayerConcatOperationAxes()` (*PMML44Super.LayerParameters* method), 127
`set_mergeLayerDotNormalize()` (*PMML44Super.LayerParameters* method), 127
`set_mergeLayerDotOperationAxis()` (*PMML44Super.LayerParameters* method), 127
`set_mergeLayerOp()` (*PMML44Super.LayerParameters* method), 127
`set_method()` (*PMML44Super.MaximumLikelihoodStat* method), 141
`set_metric()` (*PMML44Super.Metrics* method), 143
`set_Metrics()` (*PMML44Super.TrainingParameters* method), 278
`set_mimeType()` (*PMML44Super.DataField* method), 74
`set_min()` (*PMML44Super.Target* method), 251
`set_min()` (*PMML44Super.Time* method), 269
`set_minimum()` (*PMML44Super.ComparisonMeasure* method), 175
`set_minimum()` (*PMML44Super.NumericInfo* method), 175
`set_minimumAntConsSeparationTime()` (*PMML44Super.Constraints* method), 62
`set_minimumConfidence()` (*PMML44Super.AssociationModel* method), 25
`set_minimumConfidence()` (*PMML44Super.Constraints* method), 62
`set_minimumItemsetSeparationTime()` (*PMML44Super.Constraints* method), 62
`set_minimumLift()` (*PMML44Super.Constraints* method), 62
`set_minimumNumberOfAntecedentItems()` (*PMML44Super.Constraints* method), 62
`set_minimumNumberOfConsequentItems()` (*PMML44Super.Constraints* method), 63
`set_minimumNumberOfItems()` (*PMML44Super.Constraints* method), 63
`set_minimumSupport()` (*PMML44Super.AssociationModel* method), 25
`set_minimumSupport()` (*PMML44Super.Constraints* method), 63
`set_minimumTotalSequenceTime()` (*PMML44Super.Constraints* method), 63
`set_MiningBuildTask()` (*PMML44Super.PMML* method), 188
`set_MiningField()` (*PMML44Super.MiningSchema* method), 148

`set_MiningModel()`
 (*PMML44Super.AnomalyDetectionModel*
 method), 15
`set_MiningModel()` (*PMML44Super.PMML*
 method), 188
`set_MiningModel()` (*PMML44Super.Segment*
 method), 230
`set_MiningSchema()`
 (*PMML44Super.AnomalyDetectionModel*
 method), 15
`set_MiningSchema()`
 (*PMML44Super.AssociationModel* *method*), 25
`set_MiningSchema()`
 (*PMML44Super.BaselineModel* *method*),
 31
`set_MiningSchema()`
 (*PMML44Super.BayesianNetworkModel*
 method), 36
`set_MiningSchema()`
 (*PMML44Super.ClusteringModel* *method*),
 49
`set_MiningSchema()`
 (*PMML44Super.DeepNetwork* *method*),
 79
`set_MiningSchema()`
 (*PMML44Super.GaussianProcessModel*
 method), 105
`set_MiningSchema()`
 (*PMML44Super.GeneralRegressionModel*
 method), 108
`set_MiningSchema()`
 (*PMML44Super.MiningModel* *method*), 147
`set_MiningSchema()`
 (*PMML44Super.NaiveBayesModel* *method*),
 156
`set_MiningSchema()`
 (*PMML44Super.NearestNeighborModel*
 method), 158
`set_MiningSchema()`
 (*PMML44Super.NeuralNetwork* *method*),
 164
`set_MiningSchema()`
 (*PMML44Super.RegressionModel* *method*),
 213
`set_MiningSchema()`
 (*PMML44Super.RuleSetModel* *method*),
 221
`set_MiningSchema()` (*PMML44Super.Scorecard*
 method), 224
`set_MiningSchema()`
 (*PMML44Super.SequenceModel* *method*),
 234
`set_MiningSchema()`
 (*PMML44Super.SupportVectorMachineModel*
 method), 248
`set_MiningSchema()` (*PMML44Super.TextModel*
 method), 265
`set_MiningSchema()`
 (*PMML44Super.TimeSeriesModel* *method*),
 274
`set_MiningSchema()` (*PMML44Super.TreeModel*
 method), 281
`set_minkowski()` (*PMML44Super.ComparisonMeasure*
 method), 54
`set_missingFreq()` (*PMML44Super.Counts*
 method), 70
`set_missingThreshold()`
 (*PMML44Super.Segmentation* *method*), 231
`set_missingValuePenalty()`
 (*PMML44Super.DecisionTree* *method*), 76
`set_missingValuePenalty()`
 (*PMML44Super.TreeModel* *method*), 281
`set_missingValueReplacement()`
 (*PMML44Super.MiningField* *method*), 145
`set_missingValueStrategy()`
 (*PMML44Super.DecisionTree* *method*), 76
`set_missingValueStrategy()`
 (*PMML44Super.TreeModel* *method*), 281
`set_missingValueTreatment()`
 (*PMML44Super.MiningField* *method*), 145
`set_MissingValueWeights()`
 (*PMML44Super.ClusteringModel* *method*),
 49
`set_modalValue()` (*PMML44Super.DiscrStats*
 method), 85
`set_modelClass()` (*PMML44Super.ClusteringModel*
 method), 50
`set_modelDF()` (*PMML44Super.GeneralRegressionModel*
 method), 108
`set_ModelExplanation()`
 (*PMML44Super.BaselineModel* *method*),
 31
`set_ModelExplanation()`
 (*PMML44Super.BayesianNetworkModel*
 method), 36
`set_ModelExplanation()`
 (*PMML44Super.ClusteringModel* *method*),
 50
`set_ModelExplanation()`
 (*PMML44Super.DeepNetwork* *method*),
 79
`set_ModelExplanation()`
 (*PMML44Super.GaussianProcessModel*
 method), 105
`set_ModelExplanation()`
 (*PMML44Super.GeneralRegressionModel*
 method), 108
`set_ModelExplanation()`

<code>(PMML44Super.MiningModel method), 147</code>	<code>set_modelName()</code> <i>(PMML44Super.NeuralNetwork method), 164</i>
<code>set_ModelExplanation()</code>	<code>set_modelName()</code> <i>(PMML44Super.Regression method), 212</i>
<code>(PMML44Super.NaiveBayesModel method), 156</code>	<code>set_modelName()</code> <i>(PMML44Super.RegressionModel method), 213</i>
<code>set_ModelExplanation()</code>	<code>set_modelName()</code> <i>(PMML44Super.RuleSetModel method), 221</i>
<code>(PMML44Super.NearestNeighborModel method), 158</code>	<code>set_modelName()</code> <i>(PMML44Super.Scorecard method), 225</i>
<code>set_ModelExplanation()</code>	<code>set_modelName()</code> <i>(PMML44Super.SequenceModel method), 235</i>
<code>(PMML44Super.NeuralNetwork method), 164</code>	<code>set_modelName()</code> <i>(PMML44Super.SupportVectorMachineModel method), 248</i>
<code>set_ModelExplanation()</code>	<code>set_modelName()</code> <i>(PMML44Super.TextModel method), 265</i>
<code>(PMML44Super.RegressionModel method), 213</code>	<code>set_modelName()</code> <i>(PMML44Super.TimeSeriesModel method), 274</i>
<code>set_ModelExplanation()</code>	<code>set_modelName()</code> <i>(PMML44Super.TreeModel method), 281</i>
<code>(PMML44Super.RuleSetModel method), 221</code>	<code>set_ModelStats()</code> <i>(PMML44Super.AssociationModel method), 25</i>
<code>set_ModelExplanation()</code>	<code>set_ModelStats()</code> <i>(PMML44Super.BaselineModel method), 31</i>
<code>(PMML44Super.Scorecard method), 224</code>	<code>set_ModelStats()</code> <i>(PMML44Super.BayesianNetworkModel method), 36</i>
<code>set_ModelExplanation()</code>	<code>set_ModelStats()</code> <i>(PMML44Super.ClusteringModel method), 50</i>
<code>(PMML44Super.SupportVectorMachineModel method), 248</code>	<code>set_ModelStats()</code> <i>(PMML44Super.DecisionTree method), 76</i>
<code>set_ModelExplanation()</code>	<code>set_ModelStats()</code> <i>(PMML44Super.DeepNetwork method), 79</i>
<code>(PMML44Super.TextModel method), 265</code>	<code>set_ModelStats()</code> <i>(PMML44Super.GaussianProcessModel method), 105</i>
<code>set_ModelExplanation()</code>	<code>set_ModelStats()</code> <i>(PMML44Super.GeneralRegressionModel method), 108</i>
<code>(PMML44Super.TimeSeriesModel method), 274</code>	<code>set_ModelStats()</code> <i>(PMML44Super.MiningModel method), 147</i>
<code>set_ModelExplanation()</code>	<code>set_ModelStats()</code> <i>(PMML44Super.NaiveBayesModel method), 156</i>
<code>(PMML44Super.TreeModel method), 281</code>	<code>set_ModelStats()</code> <i>(PMML44Super.NearestNeighborModel method), 158</i>
<code>set_ModelLiftGraph()</code> <i>(PMML44Super.LiftData method), 130</i>	<code>set_ModelStats()</code> <i>(PMML44Super.NeuralNetwork method), 164</i>
<code>set_modelName()</code> <i>(PMML44Super.AnomalyDetectionModel method), 16</i>	<code>set_ModelStats()</code> <i>(PMML44Super.Regression method), 212</i>
<code>set_modelName()</code> <i>(PMML44Super.AssociationModel method), 25</i>	<code>set_ModelStats()</code> <i>(PMML44Super.RegressionModel method), 213</i>
<code>set_modelName()</code> <i>(PMML44Super.BaselineModel method), 31</i>	<code>set_ModelStats()</code> <i>(PMML44Super.RuleSetModel method), 221</i>
<code>set_modelName()</code> <i>(PMML44Super.BayesianNetworkModel method), 36</i>	<code>set_ModelStats()</code> <i>(PMML44Super.Scorecard method), 225</i>
<code>set_modelName()</code> <i>(PMML44Super.ClusteringModel method), 50</i>	<code>set_ModelStats()</code> <i>(PMML44Super.SequenceModel method), 234</i>
<code>set_modelName()</code> <i>(PMML44Super.DecisionTree method), 76</i>	
<code>set_modelName()</code> <i>(PMML44Super.DeepNetwork method), 79</i>	
<code>set_modelName()</code> <i>(PMML44Super.GaussianProcessModel method), 105</i>	
<code>set_modelName()</code> <i>(PMML44Super.GeneralRegressionModel method), 108</i>	
<code>set_modelName()</code> <i>(PMML44Super.MiningModel method), 147</i>	
<code>set_modelName()</code> <i>(PMML44Super.NaiveBayesModel method), 157</i>	
<code>set_modelName()</code> <i>(PMML44Super.NearestNeighborModel method), 158</i>	

[set_ModelStats\(\)](#) (*PMML44Super.SupportVectorMachineModel* method), 248
[set_ModelStats\(\)](#) (*PMML44Super.TextModel* method), 265
[set_ModelStats\(\)](#) (*PMML44Super.TimeSeriesModel* method), 274
[set_ModelStats\(\)](#) (*PMML44Super.TreeModel* method), 281
[set_modelType\(\)](#) (*PMML44Super.BayesianNetworkModel* method), 36
[set_modelType\(\)](#) (*PMML44Super.GeneralRegressionModel* method), 108
[set_modelType\(\)](#) (*PMML44Super.RegressionModel* method), 213
[set_ModelVerification\(\)](#) (*PMML44Super.AnomalyDetectionModel* method), 15
[set_ModelVerification\(\)](#) (*PMML44Super.AssociationModel* method), 25
[set_ModelVerification\(\)](#) (*PMML44Super.BaselineModel* method), 31
[set_ModelVerification\(\)](#) (*PMML44Super.BayesianNetworkModel* method), 36
[set_ModelVerification\(\)](#) (*PMML44Super.ClusteringModel* method), 50
[set_ModelVerification\(\)](#) (*PMML44Super.DeepNetwork* method), 79
[set_ModelVerification\(\)](#) (*PMML44Super.GaussianProcessModel* method), 105
[set_ModelVerification\(\)](#) (*PMML44Super.GeneralRegressionModel* method), 108
[set_ModelVerification\(\)](#) (*PMML44Super.MiningModel* method), 147
[set_ModelVerification\(\)](#) (*PMML44Super.NaiveBayesModel* method), 156
[set_ModelVerification\(\)](#) (*PMML44Super.NearestNeighborModel* method), 158
[set_ModelVerification\(\)](#) (*PMML44Super.NeuralNetwork* method), 164
[set_ModelVerification\(\)](#) (*PMML44Super.RegressionModel* method), 213
[set_ModelVerification\(\)](#) (*PMML44Super.RuleSetModel* method), 221
[set_ModelVerification\(\)](#) (*PMML44Super.Scorecard* method), 225
[set_ModelVerification\(\)](#) (*PMML44Super.SupportVectorMachineModel* method), 248
[set_ModelVerification\(\)](#) (*PMML44Super.TextModel* method), 265
[set_ModelVerification\(\)](#) (*PMML44Super.TimeSeriesModel* method), 274
[set_ModelVerification\(\)](#) (*PMML44Super.TreeModel* method), 281
[set_modelVersion\(\)](#) (*PMML44Super.Header* method), 111
[set_momentum\(\)](#) (*PMML44Super.SGD* method), 222
[set_movingMeanInitializer\(\)](#) (*PMML44Super.LayerParameters* method), 127
[set_movingVarianceInitializer\(\)](#) (*PMML44Super.LayerParameters* method), 127
[set_multipleModelMethod\(\)](#) (*PMML44Super.Segmentation* method), 231
[set_MultivariateStat\(\)](#) (*PMML44Super.MultivariateStats* method), 154
[set_MultivariateStats\(\)](#) (*PMML44Super.ModelStats* method), 151
[set_n\(\)](#) (*PMML44Super.ArrayType* method), 23
[set_n\(\)](#) (*PMML44Super.INT_SparseArray* method), 112
[set_n\(\)](#) (*PMML44Super.Lag* method), 122
[set_n\(\)](#) (*PMML44Super.REAL_SparseArray* method), 206
[set_Nadam\(\)](#) (*PMML44Super.Optimizers* method), 177
[set_NaiveBayesModel\(\)](#) (*PMML44Super.AnomalyDetectionModel* method), 15
[set_NaiveBayesModel\(\)](#) (*PMML44Super.PMML* method), 188
[set_NaiveBayesModel\(\)](#) (*PMML44Super.Segment* method), 230
[set_name\(\)](#) (*PMML44Super.Application* method), 20
[set_name\(\)](#) (*PMML44Super.CategoricalPredictor* method), 41
[set_name\(\)](#) (*PMML44Super.Characteristic* method), 43
[set_name\(\)](#) (*PMML44Super.Cluster* method), 47
[set_name\(\)](#) (*PMML44Super.ContinuousNode* method), 67
[set_name\(\)](#) (*PMML44Super.DataField* method), 74
[set_name\(\)](#) (*PMML44Super.DefineFunction* method), 81

`set_name()` (*PMML44Super.DerivedField method*), 84
`set_name()` (*PMML44Super.DiscreteNode method*), 87
`set_name()` (*PMML44Super.Extension method*), 93
`set_name()` (*PMML44Super.MiningField method*), 145
`set_name()` (*PMML44Super.MultivariateStat method*), 153
`set_name()` (*PMML44Super.NumericPredictor method*), 176
`set_name()` (*PMML44Super.OutputField method*), 182
`set_name()` (*PMML44Super.Parameter method*), 193
`set_name()` (*PMML44Super.ParameterField method*), 193
`set_name()` (*PMML44Super.Partition method*), 196
`set_name()` (*PMML44Super.Predictor method*), 202
`set_name()` (*PMML44Super.PredictorTerm method*), 203
`set_name()` (*PMML44Super.ResultField method*), 218
`set_name()` (*PMML44Super.Taxonomy method*), 257
`set_name()` (*PMML44Super.TextDocument method*), 260
`set_nbCols()` (*PMML44Super.Matrix method*), 140
`set_nbCorrect()` (*PMML44Super.RuleSet method*), 220
`set_nbCorrect()` (*PMML44Super.SimpleRule method*), 241
`set_nbRows()` (*PMML44Super.Matrix method*), 140
`set_NearestNeighborModel()` (*PMML44Super.AnomalyDetectionModel method*), 15
`set_NearestNeighborModel()` (*PMML44Super.PMML method*), 188
`set_NearestNeighborModel()` (*PMML44Super.Segment method*), 230
`set_negativeTargetFieldDisplayValue()` (*PMML44Super.ROC method*), 207
`set_negativeTargetFieldValue()` (*PMML44Super.ROC method*), 207
`set_nesterov()` (*PMML44Super.SGD method*), 222
`set_NetworkLayer()` (*PMML44Super.DeepNetwork method*), 79
`set_NeuralInput()` (*PMML44Super.NeuralInputs method*), 161
`set_NeuralInputs()` (*PMML44Super.NeuralNetwork method*), 164
`set_NeuralLayer()` (*PMML44Super.NeuralNetwork method*), 164
`set_NeuralNetwork()` (*PMML44Super.AnomalyDetectionModel method*), 16
`set_NeuralNetwork()` (*PMML44Super.PMML method*), 188
`set_NeuralNetwork()` (*PMML44Super.Segment method*), 230
`set_NeuralOutput()` (*PMML44Super.NeuralOutputs method*), 166
`set_NeuralOutputs()` (*PMML44Super.DeepNetwork method*), 79
`set_NeuralOutputs()` (*PMML44Super.NeuralNetwork method*), 164
`set_Neuron()` (*PMML44Super.NeuralLayer method*), 162
`set_Node()` (*PMML44Super.DecisionTree method*), 76
`set_Node()` (*PMML44Super.Node method*), 168
`set_Node()` (*PMML44Super.TreeModel method*), 281
`set_noiseVariance()` (*PMML44Super.AbsoluteExponentialKernel method*), 8
`set_noiseVariance()` (*PMML44Super.ARDSquaredExponentialKernel method*), 4
`set_noiseVariance()` (*PMML44Super.GeneralizedExponentialKernel method*), 110
`set_noiseVariance()` (*PMML44Super.RadialBasisKernel method*), 209
`set_NonseasonalComponent()` (*PMML44Super.ARIMA method*), 6
`set_NonseasonalFactor()` (*PMML44Super.Denominator method*), 82
`set_NonseasonalFactor()` (*PMML44Super.Numerator method*), 174
`set_norm()` (*PMML44Super.LinearNorm method*), 132
`set_NormalDistributionForBN()` (*PMML44Super.ContinuousDistribution method*), 65
`set_normalizationMethod()` (*PMML44Super.DeepNetwork method*), 79
`set_normalizationMethod()` (*PMML44Super.NetworkLayer method*), 160
`set_normalizationMethod()` (*PMML44Super.NeuralLayer method*), 162
`set_normalizationMethod()` (*PMML44Super.NeuralNetwork method*),

164
 set_normalizationMethod()
 (PMML44Super.Regression method), 212
 set_normalizationMethod()
 (PMML44Super.RegressionModel method),
 214
 set_normalizationScheme()
 (PMML44Super.TestDistributions method),
 258
 set_NormalizedCountTable()
 (PMML44Super.Baseline method), 29
 set_NormContinuous() (PMML44Super.Apply
 method), 22
 set_NormContinuous()
 (PMML44Super.ComplexPartialScore method),
 56
 set_NormContinuous()
 (PMML44Super.DefineFunction method),
 80
 set_NormContinuous()
 (PMML44Super.DerivedField method), 84
 set_NormContinuous() (PMML44Super.Lower
 method), 136
 set_NormContinuous() (PMML44Super.Mean
 method), 142
 set_NormContinuous()
 (PMML44Super.OutputField method), 182
 set_NormContinuous() (PMML44Super.TextIndex
 method), 262
 set_NormContinuous() (PMML44Super.Upper
 method), 288
 set_NormContinuous() (PMML44Super.Variance
 method), 291
 set_NormDiscrete() (PMML44Super.Apply
 method), 22
 set_NormDiscrete()
 (PMML44Super.ComplexPartialScore method),
 56
 set_NormDiscrete()
 (PMML44Super.DefineFunction method),
 81
 set_NormDiscrete() (PMML44Super.DerivedField
 method), 84
 set_NormDiscrete() (PMML44Super.Lower
 method), 136
 set_NormDiscrete() (PMML44Super.Mean
 method), 142
 set_NormDiscrete() (PMML44Super.OutputField
 method), 182
 set_NormDiscrete() (PMML44Super.TextIndex
 method), 262
 set_NormDiscrete() (PMML44Super.Upper
 method), 288
 set_NormDiscrete() (PMML44Super.Variance
 method), 291
 set_noTrueChildStrategy()
 (PMML44Super.DecisionTree method), 76
 set_noTrueChildStrategy()
 (PMML44Super.TreeModel method), 281
 set_NUM_ARRAY() (PMML44Super.ContStats
 method), 64
 set_numberOfAttributes()
 (PMML44Super.SupportVectors method),
 249
 set_numberOfClusters()
 (PMML44Super.ClusteringModel method),
 50
 set_numberOfCoefficients()
 (PMML44Super.Coefficients method), 52
 set_numberOfDocuments()
 (PMML44Super.TextModel method), 265
 set_numberOfFields()
 (PMML44Super.DataDictionary method),
 73
 set_numberOfFields()
 (PMML44Super.VectorFields method), 294
 set_numberOfInputs()
 (PMML44Super.NeuralInputs method), 161
 set_numberOfItems()
 (PMML44Super.AssociationModel method), 25
 set_numberOfItems() (PMML44Super.Itemset
 method), 118
 set_numberOfItemsets()
 (PMML44Super.AssociationModel method), 25
 set_numberOfLayers()
 (PMML44Super.DeepNetwork method),
 79
 set_numberOfLayers()
 (PMML44Super.NeuralNetwork method),
 164
 set_numberOfNeighbors()
 (PMML44Super.NearestNeighborModel
 method), 159
 set_numberOfNeurons()
 (PMML44Super.NeuralLayer method), 162
 set_numberOfOutputs()
 (PMML44Super.NeuralOutputs method),
 166
 set_numberOfRules()
 (PMML44Super.AssociationModel method), 25
 set_numberOfSets() (PMML44Super.Sequence
 method), 232
 set_numberOfSets()
 (PMML44Super.SequenceRule method),
 237
 set_numberOfSupportVectors()
 (PMML44Super.SupportVectors method),
 249

`set_numberOfTerms()` (*PMML44Super.TextModel method*), 265
`set_numberOfTransactionGroups()` (*PMML44Super.SequenceModel method*), 235
`set_numberOfTransactions()` (*PMML44Super.AssociationModel method*), 25
`set_numberOfTransactions()` (*PMML44Super.SequenceModel method*), 235
`set_numberOfVectors()` (*PMML44Super.VectorDictionary method*), 293
`set_Numerator()` (*PMML44Super.DynamicRegressor method*), 90
`set_NumericInfo()` (*PMML44Super.PartitionFieldStats method*), 197
`set_NumericInfo()` (*PMML44Super.UnivariateStats method*), 287
`set_NumericPredictor()` (*PMML44Super.RegressionTable method*), 215
`set_numOfPredictors()` (*PMML44Super.PredictiveModelQuality method*), 201
`set_numOfRecords()` (*PMML44Super.PredictiveModelQuality method*), 201
`set_numOfRecordsWeighted()` (*PMML44Super.PredictiveModelQuality method*), 201
`set_numTopCategories()` (*PMML44Super.OutputField method*), 182
`set_occurrence()` (*PMML44Super.Sequence method*), 232
`set_occurrence()` (*PMML44Super.SequenceRule method*), 237
`set_offDiagDefault()` (*PMML44Super.Matrix method*), 140
`set_offset()` (*PMML44Super.TimeAnchor method*), 270
`set_offsetValue()` (*PMML44Super.GeneralRegressionModel method*), 108
`set_offsetVariable()` (*PMML44Super.GeneralRegressionModel method*), 108
`set_operator()` (*PMML44Super.SetPredicate method*), 238
`set_operator()` (*PMML44Super.SimplePredicate method*), 240
`set_optimizer()` (*PMML44Super.GaussianProcessModel method*), 105
`set_Optimizers()` (*PMML44Super.TrainingParameters method*), 278
`set_OptimumLiftGraph()` (*PMML44Super.LiftData method*), 130
`set_optype()` (*PMML44Super.DataField method*), 74
`set_optype()` (*PMML44Super.DefineFunction method*), 81
`set_optype()` (*PMML44Super.DerivedField method*), 84
`set_optype()` (*PMML44Super.MiningField method*), 145
`set_optype()` (*PMML44Super.OutputField method*), 182
`set_optype()` (*PMML44Super.ParameterField method*), 193
`set_optype()` (*PMML44Super.ResultField method*), 218
`set_optype()` (*PMML44Super.Target method*), 251
`set_orig()` (*PMML44Super.LinearNorm method*), 132
`set_outField()` (*PMML44Super.TextIndexNormalization method*), 263
`set_OutlierEffect()` (*PMML44Super.ARIMA method*), 6
`set_outliers()` (*PMML44Super.MiningField method*), 145
`set_outliers()` (*PMML44Super.NormContinuous method*), 171
`set_Output()` (*PMML44Super.AnomalyDetectionModel method*), 16
`set_Output()` (*PMML44Super.AssociationModel method*), 25
`set_Output()` (*PMML44Super.BaselineModel method*), 31
`set_Output()` (*PMML44Super.BayesianNetworkModel method*), 36
`set_Output()` (*PMML44Super.ClusteringModel method*), 50
`set_Output()` (*PMML44Super.DecisionTree method*), 76
`set_Output()` (*PMML44Super.DeepNetwork method*), 79
`set_Output()` (*PMML44Super.GaussianProcessModel method*), 105
`set_Output()` (*PMML44Super.GeneralRegressionModel method*), 108
`set_Output()` (*PMML44Super.MiningModel method*), 147
`set_Output()` (*PMML44Super.NaiveBayesModel method*), 156
`set_Output()` (*PMML44Super.NearestNeighborModel method*), 158

`set_Output()` (PMML44Super.NeuralNetwork method), 164
`set_Output()` (PMML44Super.Regression method), 212
`set_Output()` (PMML44Super.RegressionModel method), 213
`set_Output()` (PMML44Super.RuleSetModel method), 221
`set_Output()` (PMML44Super.Scorecard method), 225
`set_Output()` (PMML44Super.SupportVectorMachineModel method), 248
`set_Output()` (PMML44Super.TextModel method), 265
`set_Output()` (PMML44Super.TimeSeriesModel method), 274
`set_Output()` (PMML44Super.TreeModel method), 281
`set_outputColumn()` (PMML44Super.MapValues method), 138
`set_outputDimension()` (PMML44Super.LayerParameters method), 127
`set_OutputField()` (PMML44Super.Output method), 180
`set_outputNeuron()` (PMML44Super.NeuralOutput method), 165
`set_p()` (PMML44Super.ARMAPart method), 7
`set_p()` (PMML44Super.NonseasonalComponent method), 170
`set_P()` (PMML44Super.SeasonalComponent method), 226
`set_p_parameter()` (PMML44Super.minkowski method), 302
`set_pad()` (PMML44Super.LayerParameters method), 128
`set_paddingDims()` (PMML44Super.LayerParameters method), 128
`set_PairCounts()` (PMML44Super.BayesInput method), 33
`set_Parameter()` (PMML44Super.ParameterList method), 194
`set_ParameterField()` (PMML44Super.DefineFunction method), 81
`set_ParameterList()` (PMML44Super.AnomalyDetectionModel method), 16
`set_ParameterList()` (PMML44Super.GeneralRegressionModel method), 108
`set_parameterName()` (PMML44Super.PCell method), 183
`set_parameterName()` (PMML44Super.PPCell method), 189
`set_ParamMatrix()` (PMML44Super.GeneralRegressionModel method), 108
`set_parent()` (PMML44Super.ParentValue method), 195
`set_parentField()` (PMML44Super.ChildParent method), 45
`set_parentLevelField()` (PMML44Super.ChildParent method), 45
`set_ParentValue()` (PMML44Super.ContinuousConditionalProbability method), 65
`set_ParentValue()` (PMML44Super.DiscreteConditionalProbability method), 86
`set_partialScore()` (PMML44Super.Attribute method), 27
`set_Partition()` (PMML44Super.Cluster method), 47
`set_Partition()` (PMML44Super.Node method), 169
`set_Partition()` (PMML44Super.TargetValue method), 252
`set_PartitionFieldStats()` (PMML44Super.Partition method), 196
`set_PastVariances()` (PMML44Super.VarianceCoefficients method), 292
`set_PCell()` (PMML44Super.ParamMatrix method), 192
`set_pCol()` (PMML44Super.PCovCell method), 184
`set_PCovCell()` (PMML44Super.PCovMatrix method), 185
`set_PCovMatrix()` (PMML44Super.GeneralRegressionModel method), 108
`set_period()` (PMML44Super.SeasonalComponent method), 226
`set_period()` (PMML44Super.Seasonality_ExpoSmooth method), 228
`set_period()` (PMML44Super.StateSpaceModel method), 244
`set_periodDeficit()` (PMML44Super.MaximumLikelihoodStat method), 141
`set_permuteDims()` (PMML44Super.LayerParameters method), 128
`set_phase()` (PMML44Super.Seasonality_ExpoSmooth method), 228
`set_phi()` (PMML44Super.Trend_ExpoSmooth method), 283

[set_pointwiseConstraint\(\)](#) (PMML44Super.LayerParameters method), 128
[set_PoissonDistribution\(\)](#) (PMML44Super.Alternate method), 13
[set_PoissonDistribution\(\)](#) (PMML44Super.Baseline method), 29
[set_PoissonDistribution\(\)](#) (PMML44Super.TargetValueStat method), 255
[set_PolynomialKernelType\(\)](#) (PMML44Super.SupportVectorMachineModel method), 248
[set_poolSize\(\)](#) (PMML44Super.LayerParameters method), 128
[set_positiveTargetFieldDisplayValue\(\)](#) (PMML44Super.ROC method), 207
[set_positiveTargetFieldValue\(\)](#) (PMML44Super.ROC method), 207
[set_PPCell\(\)](#) (PMML44Super.PPMatrix method), 190
[set_PPMatrix\(\)](#) (PMML44Super.GeneralRegressionModel method), 108
[set_precision\(\)](#) (PMML44Super.VerificationField method), 296
[set_predictionMethod\(\)](#) (PMML44Super.ARIMA method), 6
[set_PredictiveModelQuality\(\)](#) (PMML44Super.ModelExplanation method), 149
[set_Predictor\(\)](#) (PMML44Super.CovariateList method), 72
[set_Predictor\(\)](#) (PMML44Super.FactorList method), 94
[set_predictorName\(\)](#) (PMML44Super.PPCell method), 189
[set_PredictorTerm\(\)](#) (PMML44Super.RegressionTable method), 215
[set_priorProbability\(\)](#) (PMML44Super.TargetValue method), 252
[set_probability\(\)](#) (PMML44Super.ScoreDistribution method), 223
[set_probability\(\)](#) (PMML44Super.ValueProbability method), 290
[set_property\(\)](#) (PMML44Super.Value method), 289
[set_pRow\(\)](#) (PMML44Super.PCovCell method), 184
[set_PsiVector\(\)](#) (PMML44Super.StateSpaceModel method), 244
[set_pValue\(\)](#) (PMML44Super.AnovaRow method), 18
[set_pValueAlpha\(\)](#) (PMML44Super.MultivariateStat method), 153
[set_pValueFinal\(\)](#) (PMML44Super.MultivariateStat method), 153
[set_pValueInitial\(\)](#) (PMML44Super.MultivariateStat method), 153
[set_q\(\)](#) (PMML44Super.ARMAPart method), 7
[set_q\(\)](#) (PMML44Super.NonseasonalComponent method), 170
[set_Q\(\)](#) (PMML44Super.SeasonalComponent method), 226
[set_Quantile\(\)](#) (PMML44Super.NumericInfo method), 175
[set_quantileLimit\(\)](#) (PMML44Super.Quantile method), 204
[set_quantileValue\(\)](#) (PMML44Super.Quantile method), 204
[set_r_squared\(\)](#) (PMML44Super.PredictiveModelQuality method), 201
[set_RadialBasisKernel\(\)](#) (PMML44Super.GaussianProcessModel method), 105
[set_RadialBasisKernelType\(\)](#) (PMML44Super.SupportVectorMachineModel method), 248
[set_RandomLiftGraph\(\)](#) (PMML44Super.LiftData method), 130
[set_rank\(\)](#) (PMML44Super.OutputField method), 182
[set_rankBasis\(\)](#) (PMML44Super.OutputField method), 182
[set_rankingQuality\(\)](#) (PMML44Super.LiftData method), 130
[set_rankOrder\(\)](#) (PMML44Super.OutputField method), 182
[set_REAL_Entries\(\)](#) (PMML44Super.REAL_SparseArray method), 206
[set_REAL_SparseArray\(\)](#) (PMML44Super.TrendCoefficients method), 282
[set_REAL_SparseArray\(\)](#) (PMML44Super.VectorInstance method), 295
[set_reasonCode\(\)](#) (PMML44Super.Attribute method), 27
[set_reasonCode\(\)](#) (PMML44Super.Characteristic method), 43
[set_reasonCodeAlgorithm\(\)](#) (PMML44Super.Scorecard method), 225
[set_recordCount\(\)](#) (PMML44Super.ModelVerification method),

- 152
- `set_recordCount()` (PMML44Super.Node method), 169
- `set_recordCount()` (PMML44Super.RuleSet method), 220
- `set_recordCount()` (PMML44Super.ScoreDistribution method), 223
- `set_recordCount()` (PMML44Super.SimpleRule method), 241
- `set_recordCount()` (PMML44Super.TrainingInstances method), 277
- `set_recursive()` (PMML44Super.TextIndexNormalization method), 263
- `set_referencePoint()` (PMML44Super.Parameter method), 193
- `set_regexField()` (PMML44Super.TextIndexNormalization method), 263
- `set_Regression()` (PMML44Super.MiningModel method), 147
- `set_Regression()` (PMML44Super.Node method), 169
- `set_RegressionModel()` (PMML44Super.AnomalyDetectionModel method), 16
- `set_RegressionModel()` (PMML44Super.PMML method), 188
- `set_RegressionModel()` (PMML44Super.Segment method), 230
- `set_RegressionTable()` (PMML44Super.Reggression method), 212
- `set_RegressionTable()` (PMML44Super.ReggressionModel method), 213
- `set_RegressorValues()` (PMML44Super.DynamicRegressor method), 90
- `set_repeatVectorTimes()` (PMML44Super.LayerParameters method), 128
- `set_rescaleConstant()` (PMML44Super.Target method), 251
- `set_rescaleFactor()` (PMML44Super.Target method), 251
- `set_resetValue()` (PMML44Super.TestDistributions method), 258
- `set_reshapeTarget()` (PMML44Super.LayerParameters method), 128
- `set_Residuals()` (PMML44Super.MA method), 136
- `set_Residuals()` (PMML44Super.ResidualSquareCoefficients method), 216
- `set_ResidualSquareCoefficients()` (PMML44Super.GARCHPart method), 102
- `set_ResultField()` (PMML44Super.DecisionTree method), 76
- `set_ResultField()` (PMML44Super.Reggression method), 212
- `set_rho()` (PMML44Super.Adadelata method), 8
- `set_rho()` (PMML44Super.RMSprop method), 207
- `set_rightMargin()` (PMML44Super.Interval method), 116
- `set_RMSE()` (PMML44Super.ARIMA method), 6
- `set_RMSE()` (PMML44Super.ExponentialSmoothing method), 92
- `set_RMSprop()` (PMML44Super.Optimizers method), 177
- `set_ROC()` (PMML44Super.PredictiveModelQuality method), 201
- `set_ROCGraph()` (PMML44Super.ROC method), 207
- `set_rootMeanSquaredError()` (PMML44Super.PredictiveModelQuality method), 201
- `set_row()` (PMML44Super.InlineTable method), 114
- `set_row()` (PMML44Super.MatCell method), 139
- `set_ruleFeature()` (PMML44Super.OutputField method), 182
- `set_RuleSelectionMethod()` (PMML44Super.RuleSet method), 220
- `set_RuleSet()` (PMML44Super.RuleSetModel method), 221
- `set_RuleSetModel()` (PMML44Super.AnomalyDetectionModel method), 16
- `set_RuleSetModel()` (PMML44Super.PMML method), 188
- `set_RuleSetModel()` (PMML44Super.Segment method), 230
- `set_sample()` (PMML44Super.COUNT_TABLE_TYPE method), 40
- `set_schedule_decay()` (PMML44Super.Nadam method), 155
- `set_score()` (PMML44Super.Node method), 169
- `set_score()` (PMML44Super.SimpleRule method), 241
- `set_Scorecard()` (PMML44Super.AnomalyDetectionModel method), 16
- `set_Scorecard()` (PMML44Super.PMML method), 189
- `set_Scorecard()` (PMML44Super.Segment method), 230
- `set_ScoreDistribution()` (PMML44Super.Node method), 169
- `set_ScoreDistribution()` (PMML44Super.RuleSet method), 220
- `set_ScoreDistribution()` (PMML44Super.SimpleRule method), 241

set_SeasonalComponent() (PMML44Super.ARIMA method), 6
 set_SeasonalFactor() (PMML44Super.Denominator method), 82
 set_SeasonalFactor() (PMML44Super.Numerator method), 174
 set_Seasonality_ExpoSmooth() (PMML44Super.ExponentialSmoothing method), 92
 set_SeasonalTrendDecomposition() (PMML44Super.TimeSeriesModel method), 274
 set_Segment() (PMML44Super.Segmentation method), 231
 set_Segmentation() (PMML44Super.MiningModel method), 147
 set_segmentId() (PMML44Super.OutputField method), 182
 set_seqId() (PMML44Super.SequenceReference method), 235
 set_Sequence() (PMML44Super.SequenceModel method), 234
 set_SequenceModel() (PMML44Super.AnomalyDetectionModel method), 16
 set_SequenceModel() (PMML44Super.PMML method), 189
 set_SequenceModel() (PMML44Super.Segment method), 230
 set_SequenceReference() (PMML44Super.AntecedentSequence method), 18
 set_SequenceReference() (PMML44Super.ConsequentSequence method), 60
 set_SequenceRule() (PMML44Super.SequenceModel method), 234
 set_setId() (PMML44Super.SetReference method), 238
 set_SetPredicate() (PMML44Super.SequenceModel method), 234
 set_SetReference() (PMML44Super.Sequence method), 232
 set_SGD() (PMML44Super.Optimizers method), 177
 set_SigmoidKernelType() (PMML44Super.SupportVectorMachineModel method), 248
 set_similarityScale() (PMML44Super.ClusteringField method), 48
 set_similarityType() (PMML44Super.TextModelSimiliarity method), 267
 set_simpleMatching() (PMML44Super.ComparisonMeasure method), 54
 set_SimplePredicate() (PMML44Super.Attribute method), 27
 set_SimplePredicate() (PMML44Super.CompoundPredicate method), 57
 set_SimplePredicate() (PMML44Super.CompoundRule method), 58
 set_SimplePredicate() (PMML44Super.Node method), 169
 set_SimplePredicate() (PMML44Super.Segment method), 230
 set_SimplePredicate() (PMML44Super.SimpleRule method), 241
 set_SimpleRule() (PMML44Super.CompoundRule method), 58
 set_SimpleRule() (PMML44Super.RuleSet method), 220
 set_SimpleSetPredicate() (PMML44Super.Attribute method), 27
 set_SimpleSetPredicate() (PMML44Super.CompoundPredicate method), 57
 set_SimpleSetPredicate() (PMML44Super.CompoundRule method), 58
 set_SimpleSetPredicate() (PMML44Super.Node method), 169
 set_SimpleSetPredicate() (PMML44Super.Segment method), 230
 set_SimpleSetPredicate() (PMML44Super.SimpleRule method), 241
 set_size() (PMML44Super.Cluster method), 47
 set_size() (PMML44Super.Partition method), 196
 set_slicingAxis() (PMML44Super.LayerParameters method), 128
 set_smoothedValue() (PMML44Super.Level method), 129
 set_smoothedValue() (PMML44Super.Trend_ExpoSmooth method), 283
 set_SpectralAnalysis() (PMML44Super.TimeSeriesModel method), 274
 set_splitCharacteristic() (PMML44Super.DecisionTree method), 76
 set_splitCharacteristic() (PMML44Super.TreeModel method), 281
 set_sqlWhere() (PMML44Super.Aggregate

`method)`, 12
`set_squaredEuclidean()`
 (*PMML44Super.ComparisonMeasure method*),
 54
`set_SSB()` (*PMML44Super.ClusteringModelQuality*
 method), 51
`set_SSE()` (*PMML44Super.ClusteringModelQuality*
 method), 51
`set_standardDeviation()`
 (*PMML44Super.NumericInfo method*), 175
`set_standardDeviation()` (*PMML44Super.Time*
 method), 269
`set_standardError()` (*PMML44Super.TimeValue*
 method), 275
`set_startTime()` (*PMML44Super.OutlierEffect*
 method), 179
`set_startTime()` (*PMML44Super.TimeSeries*
 method), 272
`set_startTimeVariable()`
 (*PMML44Super.GeneralRegressionModel*
 method), 108
`set_StateSpaceModel()`
 (*PMML44Super.TimeSeriesModel method*),
 274
`set_StateVector()`
 (*PMML44Super.StateSpaceModel method*),
 244
`set_statusVariable()`
 (*PMML44Super.GeneralRegressionModel*
 method), 108
`set_stdError()` (*PMML44Super.MultivariateStat*
 method), 153
`set_stepsize()` (*PMML44Super.TimeAnchor*
 method), 270
`set_stride()` (*PMML44Super.LayerParameters*
 method), 128
`set_subjectIDVariable()`
 (*PMML44Super.GeneralRegressionModel*
 method), 108
`set_sumOfSquares()` (*PMML44Super.AnovaRow*
 method), 18
`set_sumSquaredError()`
 (*PMML44Super.PredictiveModelQuality*
 method), 201
`set_sumSquaredRegression()`
 (*PMML44Super.PredictiveModelQuality*
 method), 201
`set_support()` (*PMML44Super.AssociationRule*
 method), 26
`set_support()` (*PMML44Super.Itemset method*), 118
`set_support()` (*PMML44Super.Sequence method*),
 232
`set_support()` (*PMML44Super.SequenceRule*
 method), 237
`set_SupportVector()`
 (*PMML44Super.SupportVectors method*),
 249
`set_SupportVectorMachine()`
 (*PMML44Super.SupportVectorMachineModel*
 method), 248
`set_SupportVectorMachineModel()`
 (*PMML44Super.AnomalyDetectionModel*
 method), 16
`set_SupportVectorMachineModel()`
 (*PMML44Super.PMML method*), 189
`set_SupportVectorMachineModel()`
 (*PMML44Super.Segment method*), 230
`set_SupportVectors()`
 (*PMML44Super.SupportVectorMachine*
 method), 246
`set_svmRepresentation()`
 (*PMML44Super.SupportVectorMachineModel*
 method), 248
`set_TableLocator()` (*PMML44Super.ChildParent*
 method), 45
`set_TableLocator()` (*PMML44Super.MapValues*
 method), 138
`set_TableLocator()`
 (*PMML44Super.TextIndexNormalization*
 method), 263
`set_TableLocator()`
 (*PMML44Super.TrainingInstances method*),
 277
`set_tanimoto()` (*PMML44Super.ComparisonMeasure*
 method), 54
`set_target()` (*PMML44Super.Anova method*), 17
`set_Target()` (*PMML44Super.Targets method*), 256
`set_targetCategory()`
 (*PMML44Super.MultivariateStats method*),
 154
`set_targetCategory()` (*PMML44Super.PCell*
 method), 183
`set_targetCategory()` (*PMML44Super.PCovCell*
 method), 184
`set_targetCategory()` (*PMML44Super.PPCell*
 method), 190
`set_targetCategory()`
 (*PMML44Super.RegressionTable method*),
 215
`set_targetCategory()`
 (*PMML44Super.SupportVectorMachine*
 method), 246
`set_targetField()`
 (*PMML44Super.DynamicRegressor method*),
 90
`set_targetField()` (*PMML44Super.OutputField*
 method), 182
`set_targetField()`

(PMML44Super.PredictiveModelQuality method), 201
 set_targetField() (PMML44Super.PsiVector method), 204
 set_targetFieldDisplayValue() (PMML44Super.LiftData method), 130
 set_targetFieldName() (PMML44Super.RegressionModel method), 214
 set_targetFieldValue() (PMML44Super.LiftData method), 130
 set_targetReferenceCategory() (PMML44Super.GeneralRegressionModel method), 108
 set_Targets() (PMML44Super.BaselineModel method), 31
 set_Targets() (PMML44Super.BayesianNetworkModel method), 36
 set_Targets() (PMML44Super.DecisionTree method), 76
 set_Targets() (PMML44Super.DeepNetwork method), 79
 set_Targets() (PMML44Super.GaussianProcessModel method), 105
 set_Targets() (PMML44Super.GeneralRegressionModel method), 108
 set_Targets() (PMML44Super.MiningModel method), 147
 set_Targets() (PMML44Super.NaiveBayesModel method), 156
 set_Targets() (PMML44Super.NearestNeighborModel method), 158
 set_Targets() (PMML44Super.NeuralNetwork method), 164
 set_Targets() (PMML44Super.Regression method), 212
 set_Targets() (PMML44Super.RegressionModel method), 213
 set_Targets() (PMML44Super.RuleSetModel method), 221
 set_Targets() (PMML44Super.Scorecard method), 225
 set_Targets() (PMML44Super.SupportVectorMachineModel method), 248
 set_Targets() (PMML44Super.TextModel method), 265
 set_Targets() (PMML44Super.TreeModel method), 281
 set_TargetValue() (PMML44Super.Target method), 251
 set_TargetValueCount() (PMML44Super.TargetValueCounts method), 254
 set_TargetValueCounts() (PMML44Super.BayesOutput method), 35
 set_TargetValueCounts() (PMML44Super.PairCounts method), 191
 set_TargetValueStat() (PMML44Super.TargetValueStats method), 255
 set_TargetValueStats() (PMML44Super.BayesInput method), 33
 set_targetVariableName() (PMML44Super.GeneralRegressionModel method), 108
 set_Taxonomy() (PMML44Super.DataDictionary method), 73
 set_taxonomy() (PMML44Super.DataField method), 74
 set_Taxonomy() (PMML44Super.TextDictionary method), 259
 set_tCol() (PMML44Super.PCovCell method), 184
 set_TestDistributions() (PMML44Super.BaselineModel method), 31
 set_testStatistic() (PMML44Super.TestDistributions method), 258
 set_TextCorpus() (PMML44Super.TextModel method), 265
 set_TextDictionary() (PMML44Super.TextModel method), 265
 set_TextDocument() (PMML44Super.TextCorpus method), 259
 set_textField() (PMML44Super.TextIndex method), 262
 set_TextIndex() (PMML44Super.Apply method), 22
 set_TextIndex() (PMML44Super.ComplexPartialScore method), 56
 set_TextIndex() (PMML44Super.DefineFunction method), 81
 set_TextIndex() (PMML44Super.DerivedField method), 84
 set_TextIndex() (PMML44Super.Lower method), 136
 set_TextIndex() (PMML44Super.Mean method), 142
 set_TextIndex() (PMML44Super.OutputField method), 182
 set_TextIndex() (PMML44Super.TextIndex method), 262
 set_TextIndex() (PMML44Super.Upper method), 288
 set_TextIndex() (PMML44Super.Variance method), 291
 set_TextIndexNormalization() (PMML44Super.TextIndex method), 262

`set_TextModel()` (*PMML44Super.AnomalyDetectionModel method*), 16
`set_TextModel()` (*PMML44Super.PMML method*), 189
`set_TextModel()` (*PMML44Super.Segment method*), 230
`set_TextModelNormalization()` (*PMML44Super.TextModel method*), 265
`set_TextModelSimiliarity()` (*PMML44Super.TextModel method*), 265
`set_Theta()` (*PMML44Super.FinalTheta method*), 101
`set_theta()` (*PMML44Super.Theta method*), 267
`set_ThetaRecursionState()` (*PMML44Super.MaximumLikelihoodStat method*), 141
`set_threshold()` (*PMML44Super.NaiveBayesModel method*), 157
`set_threshold()` (*PMML44Super.NearestNeighborModel method*), 159
`set_threshold()` (*PMML44Super.NeuralLayer method*), 162
`set_threshold()` (*PMML44Super.NeuralNetwork method*), 164
`set_threshold()` (*PMML44Super.OutputField method*), 182
`set_threshold()` (*PMML44Super.SupportVectorMachine method*), 246
`set_threshold()` (*PMML44Super.SupportVectorMachineModel method*), 248
`set_Time()` (*PMML44Super.AntecedentSequence method*), 19
`set_time()` (*PMML44Super.BaselineCell method*), 30
`set_Time()` (*PMML44Super.ConsequentSequence method*), 60
`set_Time()` (*PMML44Super.Sequence method*), 232
`set_Time()` (*PMML44Super.SequenceRule method*), 237
`set_time()` (*PMML44Super.TimeValue method*), 275
`set_TimeAnchor()` (*PMML44Super.TimeSeries method*), 272
`set_TimeCycle()` (*PMML44Super.TimeAnchor method*), 270
`set_TimeException()` (*PMML44Super.TimeAnchor method*), 270
`set_TimeSeries()` (*PMML44Super.RegressorValues method*), 216
`set_TimeSeries()` (*PMML44Super.TimeSeriesModel method*), 274
`set_TimeSeriesModel()` (*PMML44Super.AnomalyDetectionModel method*), 16
`set_TimeSeriesModel()` (*PMML44Super.PMML method*), 189
`set_TimeSeriesModel()` (*PMML44Super.Segment method*), 230
`set_TimeStamp()` (*PMML44Super.Header method*), 111
`set_TimeStamp()` (*PMML44Super.TimeValue method*), 275
`set_TimeValue()` (*PMML44Super.ExponentialSmoothing method*), 92
`set_TimeValue()` (*PMML44Super.TimeSeries method*), 272
`set_tokenize()` (*PMML44Super.TextIndex method*), 262
`set_tokenize()` (*PMML44Super.TextIndexNormalization method*), 263
`set_top_k_categories_for_accuracy()` (*PMML44Super.Metrics method*), 143
`set_totalFreq()` (*PMML44Super.Counts method*), 70
`set_totalSquaresSum()` (*PMML44Super.ContStats method*), 64
`set_totalValuesSum()` (*PMML44Super.ContStats method*), 64
`set_trainingBackend()` (*PMML44Super.DerivedField method*), 84
`set_TrainingInstances()` (*PMML44Super.GaussianProcessModel method*), 105
`set_TrainingInstances()` (*PMML44Super.NearestNeighborModel method*), 158
`set_TrainingParameters()` (*PMML44Super.DeepNetwork method*), 79
`set_TransferFunctionValues()` (*PMML44Super.RegressorValues method*), 216
`set_transformation()` (*PMML44Super.ARIMA method*), 6
`set_transformation()` (*PMML44Super.DynamicRegressor method*), 91
`set_transformation()` (*PMML44Super.ExponentialSmoothing method*), 92
`set_TransformationDictionary()` (*PMML44Super.PMML method*), 189
`set_TransitionMatrix()` (*PMML44Super.StateSpaceModel method*), 244
`set_TreeModel()` (*PMML44Super.AnomalyDetectionModel method*), 16
`set_TreeModel()` (*PMML44Super.PMML method*), 189
`set_TreeModel()` (*PMML44Super.Segment method*), 230

230

`set_trend()` (PMML44Super.Trend_ExpoSmooth method), 283

`set_Trend_ExpoSmooth()` (PMML44Super.ExponentialSmoothing method), 92

`set_TrendCoefficients()` (PMML44Super.RegressorValues method), 216

`set_trialsValue()` (PMML44Super.GeneralRegressionModel method), 108

`set_trialsVariable()` (PMML44Super.GeneralRegressionModel method), 108

`set_TriangularDistributionForBN()` (PMML44Super.ContinuousDistribution method), 66

`set_tRow()` (PMML44Super.PCovCell method), 184

`set_True()` (PMML44Super.Attribute method), 27

`set_True()` (PMML44Super.CompoundPredicate method), 57

`set_True()` (PMML44Super.CompoundRule method), 58

`set_True()` (PMML44Super.Node method), 169

`set_True()` (PMML44Super.Segment method), 230

`set_True()` (PMML44Super.SimpleRule method), 241

`set_tValue()` (PMML44Super.MultivariateStat method), 153

`set_type()` (PMML44Super.AnovaRow method), 18

`set_type()` (PMML44Super.ArrayType method), 23

`set_type()` (PMML44Super.OutlierEffect method), 179

`set_type()` (PMML44Super.PCovMatrix method), 185

`set_type()` (PMML44Super.Seasonality_ExpoSmooth method), 228

`set_type()` (PMML44Super.TimeAnchor method), 270

`set_type()` (PMML44Super.TimeCycle method), 271

`set_type()` (PMML44Super.TimeException method), 271

`set_UniformDistribution()` (PMML44Super.Alternate method), 13

`set_UniformDistribution()` (PMML44Super.Baseline method), 29

`set_UniformDistribution()` (PMML44Super.TargetValueStat method), 255

`set_UniformDistributionForBN()` (PMML44Super.ContinuousDistribution method), 66

`set_unit()` (PMML44Super.Seasonality_ExpoSmooth method), 228

`set_UnivariateStats()` (PMML44Super.ModelStats method), 151

`set_Upper()` (PMML44Super.TriangularDistributionForBN method), 284

`set_upper()` (PMML44Super.UniformDistribution method), 285

`set_Upper()` (PMML44Super.UniformDistributionForBN method), 286

`set_upsamplingSize()` (PMML44Super.LayerParameters method), 128

`set_usage()` (PMML44Super.TimeSeries method), 272

`set_usageType()` (PMML44Super.MiningField method), 145

`set_useReasonCodes()` (PMML44Super.Scorecard method), 225

`set_value()` (PMML44Super.BaselineStratum method), 32

`set_value()` (PMML44Super.CategoricalPredictor method), 41

`set_value()` (PMML44Super.Category method), 43

`set_value()` (PMML44Super.Coefficient method), 51

`set_Value()` (PMML44Super.DataField method), 74

`set_value()` (PMML44Super.Decision method), 75

`set_Value()` (PMML44Super.DerivedField method), 84

`set_Value()` (PMML44Super.EventValues method), 91

`set_value()` (PMML44Super.Extension method), 93

`set_value()` (PMML44Super.FieldValue method), 97

`set_value()` (PMML44Super.FieldValueCount method), 98

`set_value()` (PMML44Super.Item method), 117

`set_value()` (PMML44Super.NormDiscrete method), 172

`set_value()` (PMML44Super.OutputField method), 182

`set_value()` (PMML44Super.PairCounts method), 191

`set_value()` (PMML44Super.Parameter method), 193

`set_value()` (PMML44Super.ParentValue method), 195

`set_value()` (PMML44Super.PCovCell method), 184

`set_value()` (PMML44Super.PPCell method), 190

`set_value()` (PMML44Super.ResultField method), 218

`set_value()` (PMML44Super.ScoreDistribution method), 223

`set_value()` (PMML44Super.SimplePredicate method), 240

`set_value()` (PMML44Super.TargetValue method), 252

`set_value()` (*PMML44Super.TargetValueCount method*), 253
`set_value()` (*PMML44Super.TargetValueStat method*), 255
`set_value()` (*PMML44Super.TimeValue method*), 275
`set_value()` (*PMML44Super.Value method*), 289
`set_value()` (*PMML44Super.ValueProbability method*), 290
`set_valueOf_()` (*PMML44Super.Annotation method*), 13
`set_valueOf_()` (*PMML44Super.ArrayType method*), 23
`set_valueOf_()` (*PMML44Super.Constant method*), 61
`set_valueOf_()` (*PMML44Super.LayerBias method*), 124
`set_valueOf_()` (*PMML44Super.LayerWeights method*), 129
`set_valueOf_()` (*PMML44Super.MatCell method*), 139
`set_valueOf_()` (*PMML44Super.Timestamp method*), 276
`set_ValueProbability()` (*PMML44Super.DiscreteConditionalProbability method*), 86
`set_ValueProbability()` (*PMML44Super.DiscreteNode method*), 87
`set_VariableWeight()` (*PMML44Super.Segment method*), 230
`set_variance()` (*PMML44Super.AnyDistribution method*), 19
`set_variance()` (*PMML44Super.GaussianDistribution method*), 103
`set_Variance()` (*PMML44Super.LognormalDistributionForBN method*), 134
`set_Variance()` (*PMML44Super.NormalDistributionForBN method*), 173
`set_variance()` (*PMML44Super.PsiVector method*), 204
`set_variance()` (*PMML44Super.StateSpaceModel method*), 244
`set_VarianceCoefficients()` (*PMML44Super.GARCHPart method*), 102
`set_VectorDictionary()` (*PMML44Super.SupportVectorMachineModel method*), 248
`set_VectorFields()` (*PMML44Super.VectorDictionary method*), 293
`set_vectorId()` (*PMML44Super.SupportVector method*), 245
`set_VectorInstance()` (*PMML44Super.VectorDictionary method*), 293
`set_VerificationField()` (*PMML44Super.VerificationFields method*), 296
`set_VerificationFields()` (*PMML44Super.ModelVerification method*), 152
`set_version()` (*PMML44Super.Application method*), 20
`set_version()` (*PMML44Super.PMML method*), 189
`set_weight()` (*PMML44Super.Con method*), 59
`set_weight()` (*PMML44Super.Item method*), 117
`set_weight()` (*PMML44Super.Segment method*), 230
`set_weight()` (*PMML44Super.SimpleRule method*), 241
`set_weighted()` (*PMML44Super.PartitionFieldStats method*), 197
`set_weighted()` (*PMML44Super.UnivariateStats method*), 287
`set_weightField()` (*PMML44Super.TestDistributions method*), 258
`set_weightsFlattenAxis()` (*PMML44Super.LayerWeights method*), 129
`set_weightsShape()` (*PMML44Super.LayerWeights method*), 129
`set_width()` (*PMML44Super.NeuralLayer method*), 162
`set_width()` (*PMML44Super.NeuralNetwork method*), 164
`set_width()` (*PMML44Super.Neuron method*), 167
`set_windowSize()` (*PMML44Super.TestDistributions method*), 258
`set_wordSeparatorCharacterRE()` (*PMML44Super.TextIndex method*), 262
`set_wordSeparatorCharacterRE()` (*PMML44Super.TextIndexNormalization method*), 263
`set_XCoordinates()` (*PMML44Super.LiftGraph method*), 131
`set_XCoordinates()` (*PMML44Super.ROCGraph method*), 208
`set_YCoordinates()` (*PMML44Super.LiftGraph method*), 131
`set_YCoordinates()` (*PMML44Super.ROCGraph method*), 208
`set_zeroThreshold()` (*PMML44Super.VerificationField method*), 296
`SetPredicate` (class in *PMML44Super*), 237
`SetReference` (class in *PMML44Super*), 238
`SGD` (class in *PMML44Super*), 222
`SigmoidKernelType` (class in *PMML44Super*), 238
`simpleMatching` (class in *PMML44Super*), 303

[SimplePredicate \(class in PMML44Super\), 239](#)
[SimpleRule \(class in PMML44Super\), 240](#)
[SimpleSetPredicate \(class in PMML44Super\), 242](#)
[SpectralAnalysis \(class in PMML44Super\), 242](#)
[squaredEuclidean \(class in PMML44Super\), 303](#)
[StateSpaceModel \(class in PMML44Super\), 243](#)
[StateVector \(class in PMML44Super\), 244](#)
[subclass \(PMML44Super.AbsoluteExponentialKernel attribute\), 8](#)
[subclass \(PMML44Super.Adadelata attribute\), 9](#)
[subclass \(PMML44Super.Adagrad attribute\), 9](#)
[subclass \(PMML44Super.Adam attribute\), 10](#)
[subclass \(PMML44Super.Adamax attribute\), 11](#)
[subclass \(PMML44Super.Aggregate attribute\), 12](#)
[subclass \(PMML44Super.Alternate attribute\), 13](#)
[subclass \(PMML44Super.Annotation attribute\), 13](#)
[subclass \(PMML44Super.AnomalyDetectionModel attribute\), 16](#)
[subclass \(PMML44Super.Anova attribute\), 17](#)
[subclass \(PMML44Super.AnovaRow attribute\), 18](#)
[subclass \(PMML44Super.AntecedentSequence attribute\), 19](#)
[subclass \(PMML44Super.AnyDistribution attribute\), 19](#)
[subclass \(PMML44Super.Application attribute\), 20](#)
[subclass \(PMML44Super.Apply attribute\), 22](#)
[subclass \(PMML44Super.AR attribute\), 3](#)
[subclass \(PMML44Super.ARDSquaredExponentialKernel attribute\), 4](#)
[subclass \(PMML44Super.ARIMA attribute\), 6](#)
[subclass \(PMML44Super.ARMAPart attribute\), 7](#)
[subclass \(PMML44Super.ArrayType attribute\), 23](#)
[subclass \(PMML44Super.AssociationModel attribute\), 25](#)
[subclass \(PMML44Super.AssociationRule attribute\), 26](#)
[subclass \(PMML44Super.Attribute attribute\), 27](#)
[subclass \(PMML44Super.BaseCumHazardTables attribute\), 28](#)
[subclass \(PMML44Super.Baseline attribute\), 29](#)
[subclass \(PMML44Super.BaselineCell attribute\), 30](#)
[subclass \(PMML44Super.BaselineModel attribute\), 31](#)
[subclass \(PMML44Super.BaselineStratum attribute\), 32](#)
[subclass \(PMML44Super.BayesianNetworkModel attribute\), 36](#)
[subclass \(PMML44Super.BayesianNetworkNodes attribute\), 37](#)
[subclass \(PMML44Super.BayesInput attribute\), 33](#)
[subclass \(PMML44Super.BayesInputs attribute\), 34](#)
[subclass \(PMML44Super.BayesOutput attribute\), 35](#)
[subclass \(PMML44Super.binarySimilarity attribute\), 299](#)
[subclass \(PMML44Super.BlockIndicator attribute\), 38](#)
[subclass \(PMML44Super.BoundaryValueMeans attribute\), 39](#)
[subclass \(PMML44Super.BoundaryValues attribute\), 39](#)
[subclass \(PMML44Super.CategoricalPredictor attribute\), 41](#)
[subclass \(PMML44Super.Categories attribute\), 42](#)
[subclass \(PMML44Super.Category attribute\), 43](#)
[subclass \(PMML44Super.Characteristic attribute\), 43](#)
[subclass \(PMML44Super.Characteristics attribute\), 44](#)
[subclass \(PMML44Super.chebychev attribute\), 299](#)
[subclass \(PMML44Super.ChildParent attribute\), 45](#)
[subclass \(PMML44Super.cityBlock attribute\), 300](#)
[subclass \(PMML44Super.ClassLabels attribute\), 46](#)
[subclass \(PMML44Super.Cluster attribute\), 47](#)
[subclass \(PMML44Super.ClusteringField attribute\), 48](#)
[subclass \(PMML44Super.ClusteringModel attribute\), 50](#)
[subclass \(PMML44Super.ClusteringModelQuality attribute\), 51](#)
[subclass \(PMML44Super.Coefficient attribute\), 51](#)
[subclass \(PMML44Super.Coefficients attribute\), 52](#)
[subclass \(PMML44Super.ComparisonMeasure attribute\), 54](#)
[subclass \(PMML44Super.Comparisons attribute\), 54](#)
[subclass \(PMML44Super.ComplexPartialScore attribute\), 56](#)
[subclass \(PMML44Super.CompoundPredicate attribute\), 57](#)
[subclass \(PMML44Super.CompoundRule attribute\), 58](#)
[subclass \(PMML44Super.Con attribute\), 59](#)
[subclass \(PMML44Super.ConfusionMatrix attribute\), 60](#)
[subclass \(PMML44Super.ConsequentSequence attribute\), 60](#)
[subclass \(PMML44Super.Constant attribute\), 61](#)
[subclass \(PMML44Super.Constraints attribute\), 63](#)
[subclass \(PMML44Super.ContinuousConditionalProbability attribute\), 65](#)
[subclass \(PMML44Super.ContinuousDistribution attribute\), 66](#)
[subclass \(PMML44Super.ContinuousNode attribute\), 67](#)
[subclass \(PMML44Super.ContStats attribute\), 64](#)
[subclass \(PMML44Super.CorrelationFields attribute\), 67](#)
[subclass \(PMML44Super.CorrelationMethods attribute\), 68](#)
[subclass \(PMML44Super.Correlations attribute\), 70](#)

subclass (PMML44Super.CorrelationValues attribute), 69
 subclass (PMML44Super.COUNT_TABLE_TYPE attribute), 40
 subclass (PMML44Super.Counts attribute), 70
 subclass (PMML44Super.Covariances attribute), 71
 subclass (PMML44Super.CovariateList attribute), 72
 subclass (PMML44Super.DataDictionary attribute), 73
 subclass (PMML44Super.DataField attribute), 74
 subclass (PMML44Super.Decision attribute), 75
 subclass (PMML44Super.Decisions attribute), 77
 subclass (PMML44Super.DecisionTree attribute), 76
 subclass (PMML44Super.DeepNetwork attribute), 79
 subclass (PMML44Super.DefineFunction attribute), 81
 subclass (PMML44Super.Delimiter attribute), 81
 subclass (PMML44Super.Denominator attribute), 82
 subclass (PMML44Super.DerivedField attribute), 84
 subclass (PMML44Super.DiscreteConditionalProbability attribute), 86
 subclass (PMML44Super.DiscreteNode attribute), 87
 subclass (PMML44Super.Discretize attribute), 88
 subclass (PMML44Super.DiscretizeBin attribute), 89
 subclass (PMML44Super.DiscrStats attribute), 85
 subclass (PMML44Super.DocumentTermMatrix attribute), 89
 subclass (PMML44Super.DynamicRegressor attribute), 91
 subclass (PMML44Super.euclidean attribute), 301
 subclass (PMML44Super.EventValues attribute), 91
 subclass (PMML44Super.ExponentialSmoothing attribute), 92
 subclass (PMML44Super.Extension attribute), 93
 subclass (PMML44Super.FactorList attribute), 94
 subclass (PMML44Super.False_ attribute), 94
 subclass (PMML44Super.FieldColumnPair attribute), 95
 subclass (PMML44Super.FieldRef attribute), 96
 subclass (PMML44Super.FieldValue attribute), 97
 subclass (PMML44Super.FieldValueCount attribute), 98
 subclass (PMML44Super.FinalNoise attribute), 98
 subclass (PMML44Super.FinalNu attribute), 99
 subclass (PMML44Super.FinalOmega attribute), 99
 subclass (PMML44Super.FinalPredictedNoise attribute), 100
 subclass (PMML44Super.FinalStateVector attribute), 100
 subclass (PMML44Super.FinalTheta attribute), 101
 subclass (PMML44Super.GARCH attribute), 102
 subclass (PMML44Super.GARCHPart attribute), 103
 subclass (PMML44Super.GaussianDistribution attribute), 103
 subclass (PMML44Super.GaussianProcessModel attribute), 105
 subclass (PMML44Super.GeneralizedExponentialKernel attribute), 110
 subclass (PMML44Super.GeneralRegressionModel attribute), 108
 subclass (PMML44Super.Header attribute), 111
 subclass (PMML44Super.HVector attribute), 110
 subclass (PMML44Super.Indices attribute), 113
 subclass (PMML44Super.InlineTable attribute), 114
 subclass (PMML44Super.InstanceField attribute), 114
 subclass (PMML44Super.InstanceFields attribute), 115
 subclass (PMML44Super.INT_Entries attribute), 112
 subclass (PMML44Super.INT_SparseArray attribute), 112
 subclass (PMML44Super.Interval attribute), 116
 subclass (PMML44Super.Item attribute), 117
 subclass (PMML44Super.ItemRef attribute), 117
 subclass (PMML44Super.Itemset attribute), 118
 subclass (PMML44Super.jaccard attribute), 301
 subclass (PMML44Super.KalmanState attribute), 121
 subclass (PMML44Super.KNNInput attribute), 119
 subclass (PMML44Super.KNNInputs attribute), 120
 subclass (PMML44Super.KohonenMap attribute), 121
 subclass (PMML44Super.Lag attribute), 122
 subclass (PMML44Super.Lambda attribute), 123
 subclass (PMML44Super.LayerBias attribute), 124
 subclass (PMML44Super.LayerParameters attribute), 128
 subclass (PMML44Super.LayerWeights attribute), 129
 subclass (PMML44Super.Level attribute), 129
 subclass (PMML44Super.LiftData attribute), 130
 subclass (PMML44Super.LiftGraph attribute), 131
 subclass (PMML44Super.LinearKernelType attribute), 132
 subclass (PMML44Super.LinearNorm attribute), 132
 subclass (PMML44Super.LocalTransformations attribute), 133
 subclass (PMML44Super.LognormalDistributionForBN attribute), 134
 subclass (PMML44Super.Losses attribute), 135
 subclass (PMML44Super.Lower attribute), 136
 subclass (PMML44Super.MA attribute), 137
 subclass (PMML44Super.MACoefficients attribute), 137
 subclass (PMML44Super.MapValues attribute), 138
 subclass (PMML44Super.MatCell attribute), 139
 subclass (PMML44Super.Matrix attribute), 140
 subclass (PMML44Super.MaximumLikelihoodStat attribute), 141
 subclass (PMML44Super.Mean attribute), 142

subclass (*PMML44Super.MeasurementMatrix attribute*), 143
 subclass (*PMML44Super.Metrics attribute*), 143
 subclass (*PMML44Super.MiningBuildTask attribute*), 144
 subclass (*PMML44Super.MiningField attribute*), 145
 subclass (*PMML44Super.MiningModel attribute*), 147
 subclass (*PMML44Super.MiningSchema attribute*), 148
 subclass (*PMML44Super.minkowski attribute*), 302
 subclass (*PMML44Super.MissingValueWeights attribute*), 148
 subclass (*PMML44Super.ModelExplanation attribute*), 149
 subclass (*PMML44Super.ModelLiftGraph attribute*), 150
 subclass (*PMML44Super.ModelStats attribute*), 151
 subclass (*PMML44Super.ModelVerification attribute*), 152
 subclass (*PMML44Super.MultivariateStat attribute*), 153
 subclass (*PMML44Super.MultivariateStats attribute*), 154
 subclass (*PMML44Super.Nadam attribute*), 155
 subclass (*PMML44Super.NaiveBayesModel attribute*), 157
 subclass (*PMML44Super.NearestNeighborModel attribute*), 159
 subclass (*PMML44Super.NetworkLayer attribute*), 160
 subclass (*PMML44Super.NeuralInput attribute*), 161
 subclass (*PMML44Super.NeuralInputs attribute*), 161
 subclass (*PMML44Super.NeuralLayer attribute*), 162
 subclass (*PMML44Super.NeuralNetwork attribute*), 165
 subclass (*PMML44Super.NeuralOutput attribute*), 165
 subclass (*PMML44Super.NeuralOutputs attribute*), 166
 subclass (*PMML44Super.Neuron attribute*), 167
 subclass (*PMML44Super.Node attribute*), 169
 subclass (*PMML44Super.NonseasonalComponent attribute*), 170
 subclass (*PMML44Super.NonseasonalFactor attribute*), 171
 subclass (*PMML44Super.NormalDistributionForBN attribute*), 173
 subclass (*PMML44Super.NormContinuous attribute*), 171
 subclass (*PMML44Super.NormDiscrete attribute*), 172
 subclass (*PMML44Super.Numerator attribute*), 174
 subclass (*PMML44Super.NumericInfo attribute*), 175
 subclass (*PMML44Super.NumericPredictor attribute*), 176
 subclass (*PMML44Super.Optimizers attribute*), 178
 subclass (*PMML44Super.OptimumLiftGraph attribute*), 178
 subclass (*PMML44Super.OutlierEffect attribute*), 179
 subclass (*PMML44Super.Output attribute*), 180
 subclass (*PMML44Super.OutputField attribute*), 182
 subclass (*PMML44Super.PairCounts attribute*), 191
 subclass (*PMML44Super.Parameter attribute*), 193
 subclass (*PMML44Super.ParameterField attribute*), 193
 subclass (*PMML44Super.ParameterList attribute*), 194
 subclass (*PMML44Super.ParamMatrix attribute*), 192
 subclass (*PMML44Super.ParentValue attribute*), 195
 subclass (*PMML44Super.Partition attribute*), 196
 subclass (*PMML44Super.PartitionFieldStats attribute*), 197
 subclass (*PMML44Super.PastVariances attribute*), 197
 subclass (*PMML44Super.PCell attribute*), 183
 subclass (*PMML44Super.PCovCell attribute*), 184
 subclass (*PMML44Super.PCovMatrix attribute*), 185
 subclass (*PMML44Super.PMML attribute*), 189
 subclass (*PMML44Super.PoissonDistribution attribute*), 198
 subclass (*PMML44Super.PolynomialKernelType attribute*), 199
 subclass (*PMML44Super.PPCell attribute*), 190
 subclass (*PMML44Super.PPMatrix attribute*), 190
 subclass (*PMML44Super.PredictiveModelQuality attribute*), 201
 subclass (*PMML44Super.Predictor attribute*), 202
 subclass (*PMML44Super.PredictorTerm attribute*), 203
 subclass (*PMML44Super.PsiVector attribute*), 204
 subclass (*PMML44Super.Quantile attribute*), 204
 subclass (*PMML44Super.RadialBasisKernel attribute*), 209
 subclass (*PMML44Super.RadialBasisKernelType attribute*), 210
 subclass (*PMML44Super.RandomLiftGraph attribute*), 211
 subclass (*PMML44Super.REAL_Entries attribute*), 205
 subclass (*PMML44Super.REAL_SparseArray attribute*), 206
 subclass (*PMML44Super.Regression attribute*), 212
 subclass (*PMML44Super.RegressionModel attribute*), 214
 subclass (*PMML44Super.RegressionTable attribute*), 215
 subclass (*PMML44Super.RegressorValues attribute*), 216

- subclass (*PMML44Super.Residuals* attribute), 217
- subclass (*PMML44Super.ResidualSquareCoefficients* attribute), 216
- subclass (*PMML44Super.ResultField* attribute), 218
- subclass (*PMML44Super.RMSprop* attribute), 207
- subclass (*PMML44Super.ROC* attribute), 207
- subclass (*PMML44Super.ROCGraph* attribute), 208
- subclass (*PMML44Super.row* attribute), 302
- subclass (*PMML44Super.RuleSelectionMethod* attribute), 219
- subclass (*PMML44Super.RuleSet* attribute), 220
- subclass (*PMML44Super.RuleSetModel* attribute), 221
- subclass (*PMML44Super.Scorecard* attribute), 225
- subclass (*PMML44Super.ScoreDistribution* attribute), 223
- subclass (*PMML44Super.SeasonalComponent* attribute), 226
- subclass (*PMML44Super.SeasonalFactor* attribute), 227
- subclass (*PMML44Super.Seasonality_ExpoSmooth* attribute), 228
- subclass (*PMML44Super.SeasonalTrendDecomposition* attribute), 227
- subclass (*PMML44Super.Segment* attribute), 230
- subclass (*PMML44Super.Segmentation* attribute), 231
- subclass (*PMML44Super.Sequence* attribute), 232
- subclass (*PMML44Super.SequenceModel* attribute), 235
- subclass (*PMML44Super.SequenceReference* attribute), 236
- subclass (*PMML44Super.SequenceRule* attribute), 237
- subclass (*PMML44Super.SetPredicate* attribute), 238
- subclass (*PMML44Super.SetReference* attribute), 238
- subclass (*PMML44Super.SGD* attribute), 222
- subclass (*PMML44Super.SigmoidKernelType* attribute), 239
- subclass (*PMML44Super.simpleMatching* attribute), 303
- subclass (*PMML44Super.SimplePredicate* attribute), 240
- subclass (*PMML44Super.SimpleRule* attribute), 241
- subclass (*PMML44Super.SimpleSetPredicate* attribute), 242
- subclass (*PMML44Super.SpectralAnalysis* attribute), 243
- subclass (*PMML44Super.squaredEuclidean* attribute), 304
- subclass (*PMML44Super.StateSpaceModel* attribute), 244
- subclass (*PMML44Super.StateVector* attribute), 245
- subclass (*PMML44Super.SupportVector* attribute), 245
- subclass (*PMML44Super.SupportVectorMachine* attribute), 246
- subclass (*PMML44Super.SupportVectorMachineModel* attribute), 249
- subclass (*PMML44Super.SupportVectors* attribute), 249
- subclass (*PMML44Super.TableLocator* attribute), 250
- subclass (*PMML44Super.tanimoto* attribute), 304
- subclass (*PMML44Super.Target* attribute), 251
- subclass (*PMML44Super.Targets* attribute), 256
- subclass (*PMML44Super.TargetValue* attribute), 252
- subclass (*PMML44Super.TargetValueCount* attribute), 253
- subclass (*PMML44Super.TargetValueCounts* attribute), 254
- subclass (*PMML44Super.TargetValueStat* attribute), 255
- subclass (*PMML44Super.TargetValueStats* attribute), 255
- subclass (*PMML44Super.Taxonomy* attribute), 257
- subclass (*PMML44Super.TestDistributions* attribute), 258
- subclass (*PMML44Super.TextCorpus* attribute), 259
- subclass (*PMML44Super.TextDictionary* attribute), 260
- subclass (*PMML44Super.TextDocument* attribute), 260
- subclass (*PMML44Super.TextIndex* attribute), 262
- subclass (*PMML44Super.TextIndexNormalization* attribute), 263
- subclass (*PMML44Super.TextModel* attribute), 265
- subclass (*PMML44Super.TextModelNormalization* attribute), 266
- subclass (*PMML44Super.TextModelSimiliarity* attribute), 267
- subclass (*PMML44Super.Theta* attribute), 267
- subclass (*PMML44Super.ThetaRecursionState* attribute), 268
- subclass (*PMML44Super.Time* attribute), 269
- subclass (*PMML44Super.TimeAnchor* attribute), 270
- subclass (*PMML44Super.TimeCycle* attribute), 271
- subclass (*PMML44Super.TimeException* attribute), 271
- subclass (*PMML44Super.TimeSeries* attribute), 272
- subclass (*PMML44Super.TimeSeriesModel* attribute), 274
- subclass (*PMML44Super.Timestamp* attribute), 276
- subclass (*PMML44Super.TimeValue* attribute), 275
- subclass (*PMML44Super.TrainingInstances* attribute), 277
- subclass (*PMML44Super.TrainingParameters* attribute), 278
- subclass (*PMML44Super.TransferFunctionValues* attribute), 278

subclass (*PMML44Super.TransformationDictionary attribute*), 279
 subclass (*PMML44Super.TransitionMatrix attribute*), 280
 subclass (*PMML44Super.TreeModel attribute*), 281
 subclass (*PMML44Super.Trend_ExpoSmooth attribute*), 283
 subclass (*PMML44Super.TrendCoefficients attribute*), 282
 subclass (*PMML44Super.TriangularDistributionForBN attribute*), 284
 subclass (*PMML44Super.True_ attribute*), 284
 subclass (*PMML44Super.UniformDistribution attribute*), 285
 subclass (*PMML44Super.UniformDistributionForBN attribute*), 286
 subclass (*PMML44Super.UnivariateStats attribute*), 287
 subclass (*PMML44Super.Upper attribute*), 288
 subclass (*PMML44Super.Value attribute*), 289
 subclass (*PMML44Super.ValueProbability attribute*), 290
 subclass (*PMML44Super.VariableWeight attribute*), 290
 subclass (*PMML44Super.Variance attribute*), 291
 subclass (*PMML44Super.VarianceCoefficients attribute*), 292
 subclass (*PMML44Super.VectorDictionary attribute*), 293
 subclass (*PMML44Super.VectorFields attribute*), 294
 subclass (*PMML44Super.VectorInstance attribute*), 295
 subclass (*PMML44Super.VerificationField attribute*), 296
 subclass (*PMML44Super.VerificationFields attribute*), 296
 subclass (*PMML44Super.XCoordinates attribute*), 297
 subclass (*PMML44Super.YCoordinates attribute*), 298
 superclass (*PMML44Super.AbsoluteExponentialKernel attribute*), 8
 superclass (*PMML44Super.Adadelata attribute*), 9
 superclass (*PMML44Super.Adagrad attribute*), 9
 superclass (*PMML44Super.Adam attribute*), 10
 superclass (*PMML44Super.Adamax attribute*), 11
 superclass (*PMML44Super.Aggregate attribute*), 12
 superclass (*PMML44Super.Alternate attribute*), 13
 superclass (*PMML44Super.Annotation attribute*), 13
 superclass (*PMML44Super.AnomalyDetectionModel attribute*), 16
 superclass (*PMML44Super.Anova attribute*), 17
 superclass (*PMML44Super.AnovaRow attribute*), 18
 superclass (*PMML44Super.AntecedentSequence attribute*), 19
 superclass (*PMML44Super.AnyDistribution attribute*), 19
 superclass (*PMML44Super.Application attribute*), 20
 superclass (*PMML44Super.Apply attribute*), 22
 superclass (*PMML44Super.AR attribute*), 3
 superclass (*PMML44Super.ARDSquaredExponentialKernel attribute*), 4
 superclass (*PMML44Super.ARIMA attribute*), 6
 superclass (*PMML44Super.ARMAPart attribute*), 7
 superclass (*PMML44Super.ArrayType attribute*), 23
 superclass (*PMML44Super.AssociationModel attribute*), 25
 superclass (*PMML44Super.AssociationRule attribute*), 26
 superclass (*PMML44Super.Attribute attribute*), 27
 superclass (*PMML44Super.BaseCumHazardTables attribute*), 28
 superclass (*PMML44Super.Baseline attribute*), 29
 superclass (*PMML44Super.BaselineCell attribute*), 30
 superclass (*PMML44Super.BaselineModel attribute*), 31
 superclass (*PMML44Super.BaselineStratum attribute*), 32
 superclass (*PMML44Super.BayesianNetworkModel attribute*), 36
 superclass (*PMML44Super.BayesianNetworkNodes attribute*), 37
 superclass (*PMML44Super.BayesInput attribute*), 33
 superclass (*PMML44Super.BayesInputs attribute*), 34
 superclass (*PMML44Super.BayesOutput attribute*), 35
 superclass (*PMML44Super.binarySimilarity attribute*), 299
 superclass (*PMML44Super.BlockIndicator attribute*), 38
 superclass (*PMML44Super.BoundaryValueMeans attribute*), 39
 superclass (*PMML44Super.BoundaryValues attribute*), 39
 superclass (*PMML44Super.CategoricalPredictor attribute*), 41
 superclass (*PMML44Super.Categories attribute*), 42
 superclass (*PMML44Super.Category attribute*), 43
 superclass (*PMML44Super.Characteristic attribute*), 43
 superclass (*PMML44Super.Characteristics attribute*), 44
 superclass (*PMML44Super.chebychev attribute*), 300
 superclass (*PMML44Super.ChildParent attribute*), 45
 superclass (*PMML44Super.cityBlock attribute*), 300

[superclass \(PMML44Super.ClassLabels attribute\), 46](#)
[superclass \(PMML44Super.Cluster attribute\), 47](#)
[superclass \(PMML44Super.ClusteringField attribute\), 48](#)
[superclass \(PMML44Super.ClusteringModel attribute\), 50](#)
[superclass \(PMML44Super.ClusteringModelQuality attribute\), 51](#)
[superclass \(PMML44Super.Coefficient attribute\), 51](#)
[superclass \(PMML44Super.Coefficients attribute\), 52](#)
[superclass \(PMML44Super.ComparisonMeasure attribute\), 54](#)
[superclass \(PMML44Super.Comparisons attribute\), 54](#)
[superclass \(PMML44Super.ComplexPartialScore attribute\), 56](#)
[superclass \(PMML44Super.CompoundPredicate attribute\), 57](#)
[superclass \(PMML44Super.CompoundRule attribute\), 58](#)
[superclass \(PMML44Super.Con attribute\), 59](#)
[superclass \(PMML44Super.ConfusionMatrix attribute\), 60](#)
[superclass \(PMML44Super.ConsequentSequence attribute\), 60](#)
[superclass \(PMML44Super.Constant attribute\), 61](#)
[superclass \(PMML44Super.Constraints attribute\), 63](#)
[superclass \(PMML44Super.ContinuousConditionalProbability attribute\), 65](#)
[superclass \(PMML44Super.ContinuousDistribution attribute\), 66](#)
[superclass \(PMML44Super.ContinuousNode attribute\), 67](#)
[superclass \(PMML44Super.ContStats attribute\), 64](#)
[superclass \(PMML44Super.CorrelationFields attribute\), 67](#)
[superclass \(PMML44Super.CorrelationMethods attribute\), 68](#)
[superclass \(PMML44Super.Correlations attribute\), 70](#)
[superclass \(PMML44Super.CorrelationValues attribute\), 69](#)
[superclass \(PMML44Super.COUNT_TABLE_TYPE attribute\), 40](#)
[superclass \(PMML44Super.Counts attribute\), 70](#)
[superclass \(PMML44Super.Covariances attribute\), 71](#)
[superclass \(PMML44Super.CovariateList attribute\), 72](#)
[superclass \(PMML44Super.DataDictionary attribute\), 73](#)
[superclass \(PMML44Super.DataField attribute\), 74](#)
[superclass \(PMML44Super.Decision attribute\), 75](#)
[superclass \(PMML44Super.Decisions attribute\), 77](#)
[superclass \(PMML44Super.DecisionTree attribute\), 76](#)
[superclass \(PMML44Super.DeepNetwork attribute\), 79](#)
[superclass \(PMML44Super.DefineFunction attribute\), 81](#)
[superclass \(PMML44Super.Delimiter attribute\), 81](#)
[superclass \(PMML44Super.Denominator attribute\), 82](#)
[superclass \(PMML44Super.DerivedField attribute\), 84](#)
[superclass \(PMML44Super.DiscreteConditionalProbability attribute\), 86](#)
[superclass \(PMML44Super.DiscreteNode attribute\), 87](#)
[superclass \(PMML44Super.Discretize attribute\), 88](#)
[superclass \(PMML44Super.DiscretizeBin attribute\), 89](#)
[superclass \(PMML44Super.DiscrStats attribute\), 85](#)
[superclass \(PMML44Super.DocumentTermMatrix attribute\), 89](#)
[superclass \(PMML44Super.DynamicRegressor attribute\), 91](#)
[superclass \(PMML44Super.euclidean attribute\), 301](#)
[superclass \(PMML44Super.EventValues attribute\), 91](#)
[superclass \(PMML44Super.ExponentialSmoothing attribute\), 92](#)
[superclass \(PMML44Super.Extension attribute\), 93](#)
[superclass \(PMML44Super.FactorList attribute\), 94](#)
[superclass \(PMML44Super.False_ attribute\), 94](#)
[superclass \(PMML44Super.FieldColumnPair attribute\), 95](#)
[superclass \(PMML44Super.FieldRef attribute\), 96](#)
[superclass \(PMML44Super.FieldValue attribute\), 97](#)
[superclass \(PMML44Super.FieldValueCount attribute\), 98](#)
[superclass \(PMML44Super.FinalNoise attribute\), 98](#)
[superclass \(PMML44Super.FinalNu attribute\), 99](#)
[superclass \(PMML44Super.FinalOmega attribute\), 99](#)
[superclass \(PMML44Super.FinalPredictedNoise attribute\), 100](#)
[superclass \(PMML44Super.FinalStateVector attribute\), 100](#)
[superclass \(PMML44Super.FinalTheta attribute\), 101](#)
[superclass \(PMML44Super.GARCH attribute\), 102](#)
[superclass \(PMML44Super.GARCHPart attribute\), 103](#)
[superclass \(PMML44Super.GaussianDistribution attribute\), 103](#)
[superclass \(PMML44Super.GaussianProcessModel](#)

- attribute*), 105
- superclass (*PMML44Super.GeneralizedExponentialKernel attribute*), 110
- superclass (*PMML44Super.GeneralRegressionModel attribute*), 108
- superclass (*PMML44Super.Header attribute*), 111
- superclass (*PMML44Super.HVector attribute*), 110
- superclass (*PMML44Super.Indices attribute*), 113
- superclass (*PMML44Super.InlineTable attribute*), 114
- superclass (*PMML44Super.InstanceField attribute*), 114
- superclass (*PMML44Super.InstanceFields attribute*), 115
- superclass (*PMML44Super.INT_Entries attribute*), 112
- superclass (*PMML44Super.INT_SparseArray attribute*), 112
- superclass (*PMML44Super.Interval attribute*), 116
- superclass (*PMML44Super.Item attribute*), 117
- superclass (*PMML44Super.ItemRef attribute*), 117
- superclass (*PMML44Super.Itemset attribute*), 118
- superclass (*PMML44Super.jaccard attribute*), 301
- superclass (*PMML44Super.KalmanState attribute*), 121
- superclass (*PMML44Super.KNNInput attribute*), 119
- superclass (*PMML44Super.KNNInputs attribute*), 120
- superclass (*PMML44Super.KohonenMap attribute*), 121
- superclass (*PMML44Super.Lag attribute*), 122
- superclass (*PMML44Super.Lambda attribute*), 123
- superclass (*PMML44Super.LayerBias attribute*), 124
- superclass (*PMML44Super.LayerParameters attribute*), 128
- superclass (*PMML44Super.LayerWeights attribute*), 129
- superclass (*PMML44Super.Level attribute*), 129
- superclass (*PMML44Super.LiftData attribute*), 130
- superclass (*PMML44Super.LiftGraph attribute*), 131
- superclass (*PMML44Super.LinearKernelType attribute*), 132
- superclass (*PMML44Super.LinearNorm attribute*), 132
- superclass (*PMML44Super.LocalTransformations attribute*), 133
- superclass (*PMML44Super.LognormalDistributionForBN attribute*), 134
- superclass (*PMML44Super.Losses attribute*), 135
- superclass (*PMML44Super.Lower attribute*), 136
- superclass (*PMML44Super.MA attribute*), 137
- superclass (*PMML44Super.MACoefficients attribute*), 137
- superclass (*PMML44Super.MapValues attribute*), 138
- superclass (*PMML44Super.MatCell attribute*), 139
- superclass (*PMML44Super.Matrix attribute*), 140
- superclass (*PMML44Super.MaximumLikelihoodStat attribute*), 141
- superclass (*PMML44Super.Mean attribute*), 142
- superclass (*PMML44Super.MeasurementMatrix attribute*), 143
- superclass (*PMML44Super.Metrics attribute*), 143
- superclass (*PMML44Super.MiningBuildTask attribute*), 144
- superclass (*PMML44Super.MiningField attribute*), 145
- superclass (*PMML44Super.MiningModel attribute*), 147
- superclass (*PMML44Super.MiningSchema attribute*), 148
- superclass (*PMML44Super.minkowski attribute*), 302
- superclass (*PMML44Super.MissingValueWeights attribute*), 149
- superclass (*PMML44Super.ModelExplanation attribute*), 149
- superclass (*PMML44Super.ModelLiftGraph attribute*), 150
- superclass (*PMML44Super.ModelStats attribute*), 151
- superclass (*PMML44Super.ModelVerification attribute*), 152
- superclass (*PMML44Super.MultivariateStat attribute*), 153
- superclass (*PMML44Super.MultivariateStats attribute*), 154
- superclass (*PMML44Super.Nadam attribute*), 155
- superclass (*PMML44Super.NaiveBayesModel attribute*), 157
- superclass (*PMML44Super.NearestNeighborModel attribute*), 159
- superclass (*PMML44Super.NetworkLayer attribute*), 160
- superclass (*PMML44Super.NeuralInput attribute*), 161
- superclass (*PMML44Super.NeuralInputs attribute*), 161
- superclass (*PMML44Super.NeuralLayer attribute*), 163
- superclass (*PMML44Super.NeuralNetwork attribute*), 165
- superclass (*PMML44Super.NeuralOutput attribute*), 165
- superclass (*PMML44Super.NeuralOutputs attribute*), 166
- superclass (*PMML44Super.Neuron attribute*), 167
- superclass (*PMML44Super.Node attribute*), 169
- superclass (*PMML44Super.NonseasonalComponent*

- [attribute\), 170](#)
- [superclass \(PMML44Super.NonseasonalFactor attribute\), 171](#)
- [superclass \(PMML44Super.NormalDistributionForBN attribute\), 173](#)
- [superclass \(PMML44Super.NormContinuous attribute\), 172](#)
- [superclass \(PMML44Super.NormDiscrete attribute\), 172](#)
- [superclass \(PMML44Super.Numerator attribute\), 174](#)
- [superclass \(PMML44Super.NumericInfo attribute\), 175](#)
- [superclass \(PMML44Super.NumericPredictor attribute\), 176](#)
- [superclass \(PMML44Super.Optimizers attribute\), 178](#)
- [superclass \(PMML44Super.OptimumLiftGraph attribute\), 178](#)
- [superclass \(PMML44Super.OutlierEffect attribute\), 179](#)
- [superclass \(PMML44Super.Output attribute\), 180](#)
- [superclass \(PMML44Super.OutputField attribute\), 182](#)
- [superclass \(PMML44Super.PairCounts attribute\), 191](#)
- [superclass \(PMML44Super.Parameter attribute\), 193](#)
- [superclass \(PMML44Super.ParameterField attribute\), 193](#)
- [superclass \(PMML44Super.ParameterList attribute\), 194](#)
- [superclass \(PMML44Super.ParamMatrix attribute\), 192](#)
- [superclass \(PMML44Super.ParentValue attribute\), 195](#)
- [superclass \(PMML44Super.Partition attribute\), 196](#)
- [superclass \(PMML44Super.PartitionFieldStats attribute\), 197](#)
- [superclass \(PMML44Super.PastVariances attribute\), 197](#)
- [superclass \(PMML44Super.PCell attribute\), 183](#)
- [superclass \(PMML44Super.PCovCell attribute\), 184](#)
- [superclass \(PMML44Super.PCovMatrix attribute\), 185](#)
- [superclass \(PMML44Super.PMML attribute\), 189](#)
- [superclass \(PMML44Super.PoissonDistribution attribute\), 198](#)
- [superclass \(PMML44Super.PolynomialKernelType attribute\), 199](#)
- [superclass \(PMML44Super.PPCell attribute\), 190](#)
- [superclass \(PMML44Super.PPMatrix attribute\), 190](#)
- [superclass \(PMML44Super.PredictiveModelQuality attribute\), 201](#)
- [superclass \(PMML44Super.Predictor attribute\), 202](#)
- [superclass \(PMML44Super.PredictorTerm attribute\), 203](#)
- [superclass \(PMML44Super.PsiVector attribute\), 204](#)
- [superclass \(PMML44Super.Quantile attribute\), 204](#)
- [superclass \(PMML44Super.RadialBasisKernel attribute\), 209](#)
- [superclass \(PMML44Super.RadialBasisKernelType attribute\), 210](#)
- [superclass \(PMML44Super.RandomLiftGraph attribute\), 211](#)
- [superclass \(PMML44Super.REAL_Entries attribute\), 205](#)
- [superclass \(PMML44Super.REAL_SparseArray attribute\), 206](#)
- [superclass \(PMML44Super.Regression attribute\), 212](#)
- [superclass \(PMML44Super.RegressionModel attribute\), 214](#)
- [superclass \(PMML44Super.RegressionTable attribute\), 215](#)
- [superclass \(PMML44Super.RegressorValues attribute\), 216](#)
- [superclass \(PMML44Super.Residuals attribute\), 217](#)
- [superclass \(PMML44Super.ResidualSquareCoefficients attribute\), 216](#)
- [superclass \(PMML44Super.ResultField attribute\), 218](#)
- [superclass \(PMML44Super.RMSprop attribute\), 207](#)
- [superclass \(PMML44Super.ROC attribute\), 207](#)
- [superclass \(PMML44Super.ROCGraph attribute\), 208](#)
- [superclass \(PMML44Super.row attribute\), 303](#)
- [superclass \(PMML44Super.RuleSelectionMethod attribute\), 219](#)
- [superclass \(PMML44Super.RuleSet attribute\), 220](#)
- [superclass \(PMML44Super.RuleSetModel attribute\), 221](#)
- [superclass \(PMML44Super.Scorecard attribute\), 225](#)
- [superclass \(PMML44Super.ScoreDistribution attribute\), 223](#)
- [superclass \(PMML44Super.SeasonalComponent attribute\), 226](#)
- [superclass \(PMML44Super.SeasonalFactor attribute\), 227](#)
- [superclass \(PMML44Super.Seasonality_ExpoSmooth attribute\), 228](#)
- [superclass \(PMML44Super.SeasonalTrendDecomposition attribute\), 227](#)
- [superclass \(PMML44Super.Segment attribute\), 230](#)
- [superclass \(PMML44Super.Segmentation attribute\), 231](#)
- [superclass \(PMML44Super.Sequence attribute\), 232](#)
- [superclass \(PMML44Super.SequenceModel attribute\), 235](#)

- superclass (*PMML44Super.SequenceReference attribute*), 236
- superclass (*PMML44Super.SequenceRule attribute*), 237
- superclass (*PMML44Super.SetPredicate attribute*), 238
- superclass (*PMML44Super.SetReference attribute*), 238
- superclass (*PMML44Super.SGD attribute*), 222
- superclass (*PMML44Super.SigmoidKernelType attribute*), 239
- superclass (*PMML44Super.simpleMatching attribute*), 303
- superclass (*PMML44Super.SimplePredicate attribute*), 240
- superclass (*PMML44Super.SimpleRule attribute*), 241
- superclass (*PMML44Super.SimpleSetPredicate attribute*), 242
- superclass (*PMML44Super.SpectralAnalysis attribute*), 243
- superclass (*PMML44Super.squaredEuclidean attribute*), 304
- superclass (*PMML44Super.StateSpaceModel attribute*), 244
- superclass (*PMML44Super.StateVector attribute*), 245
- superclass (*PMML44Super.SupportVector attribute*), 245
- superclass (*PMML44Super.SupportVectorMachine attribute*), 246
- superclass (*PMML44Super.SupportVectorMachineModel attribute*), 249
- superclass (*PMML44Super.SupportVectors attribute*), 250
- superclass (*PMML44Super.TableLocator attribute*), 250
- superclass (*PMML44Super.tanimoto attribute*), 304
- superclass (*PMML44Super.Target attribute*), 251
- superclass (*PMML44Super.Targets attribute*), 256
- superclass (*PMML44Super.TargetValue attribute*), 252
- superclass (*PMML44Super.TargetValueCount attribute*), 253
- superclass (*PMML44Super.TargetValueCounts attribute*), 254
- superclass (*PMML44Super.TargetValueStat attribute*), 255
- superclass (*PMML44Super.TargetValueStats attribute*), 255
- superclass (*PMML44Super.Taxonomy attribute*), 257
- superclass (*PMML44Super.TestDistributions attribute*), 258
- superclass (*PMML44Super.TextCorpus attribute*), 259
- superclass (*PMML44Super.TextDictionary attribute*), 260
- superclass (*PMML44Super.TextDocument attribute*), 260
- superclass (*PMML44Super.TextIndex attribute*), 262
- superclass (*PMML44Super.TextIndexNormalization attribute*), 264
- superclass (*PMML44Super.TextModel attribute*), 265
- superclass (*PMML44Super.TextModelNormalization attribute*), 266
- superclass (*PMML44Super.TextModelSimiliarity attribute*), 267
- superclass (*PMML44Super.Theta attribute*), 267
- superclass (*PMML44Super.ThetaRecursionState attribute*), 268
- superclass (*PMML44Super.Time attribute*), 269
- superclass (*PMML44Super.TimeAnchor attribute*), 270
- superclass (*PMML44Super.TimeCycle attribute*), 271
- superclass (*PMML44Super.TimeException attribute*), 271
- superclass (*PMML44Super.TimeSeries attribute*), 272
- superclass (*PMML44Super.TimeSeriesModel attribute*), 274
- superclass (*PMML44Super.Timestamp attribute*), 276
- superclass (*PMML44Super.TimeValue attribute*), 275
- superclass (*PMML44Super.TrainingInstances attribute*), 277
- superclass (*PMML44Super.TrainingParameters attribute*), 278
- superclass (*PMML44Super.TransferFunctionValues attribute*), 278
- superclass (*PMML44Super.TransformationDictionary attribute*), 279
- superclass (*PMML44Super.TransitionMatrix attribute*), 280
- superclass (*PMML44Super.TreeModel attribute*), 281
- superclass (*PMML44Super.Trend_ExpoSmooth attribute*), 283
- superclass (*PMML44Super.TrendCoefficients attribute*), 282
- superclass (*PMML44Super.TriangularDistributionForBN attribute*), 284
- superclass (*PMML44Super.True_ attribute*), 284
- superclass (*PMML44Super.UniformDistribution attribute*), 285
- superclass (*PMML44Super.UniformDistributionForBN attribute*), 286
- superclass (*PMML44Super.UnivariateStats attribute*), 287

- superclass (*PMML44Super.Upper* attribute), 288
 - superclass (*PMML44Super.Value* attribute), 289
 - superclass (*PMML44Super.ValueProbability* attribute), 290
 - superclass (*PMML44Super.VariableWeight* attribute), 290
 - superclass (*PMML44Super.Variance* attribute), 291
 - superclass (*PMML44Super.VarianceCoefficients* attribute), 292
 - superclass (*PMML44Super.VectorDictionary* attribute), 293
 - superclass (*PMML44Super.VectorFields* attribute), 294
 - superclass (*PMML44Super.VectorInstance* attribute), 295
 - superclass (*PMML44Super.VerificationField* attribute), 296
 - superclass (*PMML44Super.VerificationFields* attribute), 297
 - superclass (*PMML44Super.XCoordinates* attribute), 297
 - superclass (*PMML44Super.YCoordinates* attribute), 298
 - SupportVector (class in *PMML44Super*), 245
 - SupportVectorMachine (class in *PMML44Super*), 245
 - SupportVectorMachineModel (class in *PMML44Super*), 246
 - SupportVectors (class in *PMML44Super*), 249
- ## T
- TableLocator (class in *PMML44Super*), 250
 - tanimoto (class in *PMML44Super*), 304
 - Target (class in *PMML44Super*), 250
 - Targets (class in *PMML44Super*), 255
 - TargetValue (class in *PMML44Super*), 251
 - TargetValueCount (class in *PMML44Super*), 252
 - TargetValueCounts (class in *PMML44Super*), 253
 - TargetValueStat (class in *PMML44Super*), 254
 - TargetValueStats (class in *PMML44Super*), 255
 - Taxonomy (class in *PMML44Super*), 256
 - TestDistributions (class in *PMML44Super*), 257
 - TextCorpus (class in *PMML44Super*), 258
 - TextDictionary (class in *PMML44Super*), 259
 - TextDocument (class in *PMML44Super*), 260
 - TextIndex (class in *PMML44Super*), 260
 - TextIndexNormalization (class in *PMML44Super*), 262
 - TextModel (class in *PMML44Super*), 264
 - TextModelNormalization (class in *PMML44Super*), 265
 - TextModelSimilarity (class in *PMML44Super*), 266
 - Theta (class in *PMML44Super*), 267
 - ThetaRecursionState (class in *PMML44Super*), 267
 - Time (class in *PMML44Super*), 268
 - TimeAnchor (class in *PMML44Super*), 269
 - TimeCycle (class in *PMML44Super*), 270
 - TimeException (class in *PMML44Super*), 271
 - TimeSeries (class in *PMML44Super*), 271
 - TimeSeriesModel (class in *PMML44Super*), 273
 - Timestamp (class in *PMML44Super*), 275
 - TimeValue (class in *PMML44Super*), 274
 - to_etree () (*PMML44Super.AbsoluteExponentialKernel* method), 8
 - to_etree () (*PMML44Super.Adadelata* method), 9
 - to_etree () (*PMML44Super.Adagrad* method), 9
 - to_etree () (*PMML44Super.Adam* method), 10
 - to_etree () (*PMML44Super.Adamax* method), 11
 - to_etree () (*PMML44Super.Aggregate* method), 12
 - to_etree () (*PMML44Super.Alternate* method), 13
 - to_etree () (*PMML44Super.Annotation* method), 13
 - to_etree () (*PMML44Super.AnomalyDetectionModel* method), 16
 - to_etree () (*PMML44Super.Anova* method), 17
 - to_etree () (*PMML44Super.AnovaRow* method), 18
 - to_etree () (*PMML44Super.AntecedentSequence* method), 19
 - to_etree () (*PMML44Super.AnyDistribution* method), 19
 - to_etree () (*PMML44Super.Application* method), 20
 - to_etree () (*PMML44Super.Apply* method), 22
 - to_etree () (*PMML44Super.AR* method), 4
 - to_etree () (*PMML44Super.ARDSquaredExponentialKernel* method), 4
 - to_etree () (*PMML44Super.ARIMA* method), 6
 - to_etree () (*PMML44Super.ARMAPart* method), 7
 - to_etree () (*PMML44Super.ArrayType* method), 23
 - to_etree () (*PMML44Super.AssociationModel* method), 25
 - to_etree () (*PMML44Super.AssociationRule* method), 26
 - to_etree () (*PMML44Super.Attribute* method), 27
 - to_etree () (*PMML44Super.BaseCumHazardTables* method), 28
 - to_etree () (*PMML44Super.Baseline* method), 29
 - to_etree () (*PMML44Super.BaselineCell* method), 30
 - to_etree () (*PMML44Super.BaselineModel* method), 31
 - to_etree () (*PMML44Super.BaselineStratum* method), 32
 - to_etree () (*PMML44Super.BayesianNetworkModel* method), 36
 - to_etree () (*PMML44Super.BayesianNetworkNodes* method), 37
 - to_etree () (*PMML44Super.BayesInput* method), 33
 - to_etree () (*PMML44Super.BayesInputs* method), 34

`to_etree()` (PMML44Super.BayesOutput method), 35
`to_etree()` (PMML44Super.binarySimilarity method), 299
`to_etree()` (PMML44Super.BlockIndicator method), 38
`to_etree()` (PMML44Super.BoundaryValueMeans method), 39
`to_etree()` (PMML44Super.BoundaryValues method), 39
`to_etree()` (PMML44Super.CategoricalPredictor method), 41
`to_etree()` (PMML44Super.Categories method), 42
`to_etree()` (PMML44Super.Category method), 43
`to_etree()` (PMML44Super.Characteristic method), 44
`to_etree()` (PMML44Super.Characteristics method), 44
`to_etree()` (PMML44Super.chebychev method), 300
`to_etree()` (PMML44Super.ChildParent method), 45
`to_etree()` (PMML44Super.cityBlock method), 300
`to_etree()` (PMML44Super.ClassLabels method), 46
`to_etree()` (PMML44Super.Cluster method), 47
`to_etree()` (PMML44Super.ClusteringField method), 48
`to_etree()` (PMML44Super.ClusteringModel method), 50
`to_etree()` (PMML44Super.ClusteringModelQuality method), 51
`to_etree()` (PMML44Super.Coefficient method), 51
`to_etree()` (PMML44Super.Coefficients method), 52
`to_etree()` (PMML44Super.ComparisonMeasure method), 54
`to_etree()` (PMML44Super.Comparisons method), 54
`to_etree()` (PMML44Super.ComplexPartialScore method), 56
`to_etree()` (PMML44Super.CompoundPredicate method), 57
`to_etree()` (PMML44Super.CompoundRule method), 58
`to_etree()` (PMML44Super.Con method), 59
`to_etree()` (PMML44Super.ConfusionMatrix method), 60
`to_etree()` (PMML44Super.ConsequentSequence method), 61
`to_etree()` (PMML44Super.Constant method), 61
`to_etree()` (PMML44Super.Constraints method), 63
`to_etree()` (PMML44Super.ContinuousConditionalProbability method), 65
`to_etree()` (PMML44Super.ContinuousDistribution method), 66
`to_etree()` (PMML44Super.ContinuousNode method), 67
`to_etree()` (PMML44Super.ContStats method), 64
`to_etree()` (PMML44Super.CorrelationFields method), 68
`to_etree()` (PMML44Super.CorrelationMethods method), 68
`to_etree()` (PMML44Super.Correlations method), 70
`to_etree()` (PMML44Super.CorrelationValues method), 69
`to_etree()` (PMML44Super.COUNT_TABLE_TYPE method), 40
`to_etree()` (PMML44Super.Counts method), 70
`to_etree()` (PMML44Super.Covariances method), 71
`to_etree()` (PMML44Super.CovariateList method), 72
`to_etree()` (PMML44Super.DataDictionary method), 73
`to_etree()` (PMML44Super.DataField method), 74
`to_etree()` (PMML44Super.Decision method), 75
`to_etree()` (PMML44Super.Decisions method), 77
`to_etree()` (PMML44Super.DecisionTree method), 76
`to_etree()` (PMML44Super.DeepNetwork method), 79
`to_etree()` (PMML44Super.DefineFunction method), 81
`to_etree()` (PMML44Super.Delimiter method), 81
`to_etree()` (PMML44Super.Denominator method), 82
`to_etree()` (PMML44Super.DerivedField method), 84
`to_etree()` (PMML44Super.DiscreteConditionalProbability method), 86
`to_etree()` (PMML44Super.DiscreteNode method), 87
`to_etree()` (PMML44Super.Discretize method), 88
`to_etree()` (PMML44Super.DiscretizeBin method), 89
`to_etree()` (PMML44Super.DiscrStats method), 85
`to_etree()` (PMML44Super.DocumentTermMatrix method), 89
`to_etree()` (PMML44Super.DynamicRegressor method), 91
`to_etree()` (PMML44Super.euclidean method), 301
`to_etree()` (PMML44Super.EventValues method), 92
`to_etree()` (PMML44Super.ExponentialSmoothing method), 92
`to_etree()` (PMML44Super.Extension method), 93
`to_etree()` (PMML44Super.FactorList method), 94
`to_etree()` (PMML44Super.False_ method), 94
`to_etree()` (PMML44Super.FieldColumnPair method), 95
`to_etree()` (PMML44Super.FieldRef method), 96
`to_etree()` (PMML44Super.FieldValue method), 97
`to_etree()` (PMML44Super.FieldValueCount method), 98

`to_etree()` (*PMML44Super.FinalNoise method*), 98
`to_etree()` (*PMML44Super.FinalNu method*), 99
`to_etree()` (*PMML44Super.FinalOmega method*), 99
`to_etree()` (*PMML44Super.FinalPredictedNoise method*), 100
`to_etree()` (*PMML44Super.FinalStateVector method*), 100
`to_etree()` (*PMML44Super.FinalTheta method*), 101
`to_etree()` (*PMML44Super.GARCH method*), 102
`to_etree()` (*PMML44Super.GARCHPart method*), 103
`to_etree()` (*PMML44Super.GaussianDistribution method*), 103
`to_etree()` (*PMML44Super.GaussianProcessModel method*), 105
`to_etree()` (*PMML44Super.GeneralizedExponentialKernel method*), 110
`to_etree()` (*PMML44Super.GeneralRegressionModel method*), 108
`to_etree()` (*PMML44Super.Header method*), 111
`to_etree()` (*PMML44Super.HVector method*), 110
`to_etree()` (*PMML44Super.Indices method*), 113
`to_etree()` (*PMML44Super.InlineTable method*), 114
`to_etree()` (*PMML44Super.InstanceField method*), 114
`to_etree()` (*PMML44Super.InstanceFields method*), 115
`to_etree()` (*PMML44Super.INT_Entries method*), 112
`to_etree()` (*PMML44Super.INT_SparseArray method*), 112
`to_etree()` (*PMML44Super.Interval method*), 116
`to_etree()` (*PMML44Super.Item method*), 117
`to_etree()` (*PMML44Super.ItemRef method*), 118
`to_etree()` (*PMML44Super.Itemset method*), 118
`to_etree()` (*PMML44Super.jaccard method*), 301
`to_etree()` (*PMML44Super.KalmanState method*), 121
`to_etree()` (*PMML44Super.KNNInput method*), 119
`to_etree()` (*PMML44Super.KNNInputs method*), 120
`to_etree()` (*PMML44Super.KohonenMap method*), 121
`to_etree()` (*PMML44Super.Lag method*), 122
`to_etree()` (*PMML44Super.Lambda method*), 123
`to_etree()` (*PMML44Super.LayerBias method*), 124
`to_etree()` (*PMML44Super.LayerParameters method*), 128
`to_etree()` (*PMML44Super.LayerWeights method*), 129
`to_etree()` (*PMML44Super.Level method*), 129
`to_etree()` (*PMML44Super.LiftData method*), 130
`to_etree()` (*PMML44Super.LiftGraph method*), 131
`to_etree()` (*PMML44Super.LinearKernelType method*), 132
`to_etree()` (*PMML44Super.LinearNorm method*), 132
`to_etree()` (*PMML44Super.LocalTransformations method*), 133
`to_etree()` (*PMML44Super.LognormalDistributionForBN method*), 134
`to_etree()` (*PMML44Super.Losses method*), 135
`to_etree()` (*PMML44Super.Lower method*), 136
`to_etree()` (*PMML44Super.MA method*), 137
`to_etree()` (*PMML44Super.MACoefficients method*), 137
`to_etree()` (*PMML44Super.MapValues method*), 138
`to_etree()` (*PMML44Super.MatCell method*), 139
`to_etree()` (*PMML44Super.Matrix method*), 140
`to_etree()` (*PMML44Super.MaximumLikelihoodStat method*), 141
`to_etree()` (*PMML44Super.Mean method*), 142
`to_etree()` (*PMML44Super.MeasurementMatrix method*), 143
`to_etree()` (*PMML44Super.Metrics method*), 143
`to_etree()` (*PMML44Super.MiningBuildTask method*), 144
`to_etree()` (*PMML44Super.MiningField method*), 145
`to_etree()` (*PMML44Super.MiningModel method*), 147
`to_etree()` (*PMML44Super.MiningSchema method*), 148
`to_etree()` (*PMML44Super.minkowski method*), 302
`to_etree()` (*PMML44Super.MissingValueWeights method*), 149
`to_etree()` (*PMML44Super.ModelExplanation method*), 149
`to_etree()` (*PMML44Super.ModelLiftGraph method*), 150
`to_etree()` (*PMML44Super.ModelStats method*), 151
`to_etree()` (*PMML44Super.ModelVerification method*), 152
`to_etree()` (*PMML44Super.MultivariateStat method*), 153
`to_etree()` (*PMML44Super.MultivariateStats method*), 154
`to_etree()` (*PMML44Super.Nadam method*), 155
`to_etree()` (*PMML44Super.NaiveBayesModel method*), 157
`to_etree()` (*PMML44Super.NearestNeighborModel method*), 159
`to_etree()` (*PMML44Super.NetworkLayer method*), 160
`to_etree()` (*PMML44Super.NeuralInput method*), 161
`to_etree()` (*PMML44Super.NeuralInputs method*), 161
`to_etree()` (*PMML44Super.NeuralLayer method*),

[163](#)
[to_etree\(\)](#) (*PMML44Super.NeuralNetwork method*), [165](#)
[to_etree\(\)](#) (*PMML44Super.NeuralOutput method*), [165](#)
[to_etree\(\)](#) (*PMML44Super.NeuralOutputs method*), [166](#)
[to_etree\(\)](#) (*PMML44Super.Neuron method*), [167](#)
[to_etree\(\)](#) (*PMML44Super.Node method*), [169](#)
[to_etree\(\)](#) (*PMML44Super.NonseasonalComponent method*), [170](#)
[to_etree\(\)](#) (*PMML44Super.NonseasonalFactor method*), [171](#)
[to_etree\(\)](#) (*PMML44Super.NormalDistributionForBN method*), [173](#)
[to_etree\(\)](#) (*PMML44Super.NormContinuous method*), [172](#)
[to_etree\(\)](#) (*PMML44Super.NormDiscrete method*), [172](#)
[to_etree\(\)](#) (*PMML44Super.Numerator method*), [174](#)
[to_etree\(\)](#) (*PMML44Super.NumericInfo method*), [175](#)
[to_etree\(\)](#) (*PMML44Super.NumericPredictor method*), [176](#)
[to_etree\(\)](#) (*PMML44Super.Optimizers method*), [178](#)
[to_etree\(\)](#) (*PMML44Super.OptimumLiftGraph method*), [178](#)
[to_etree\(\)](#) (*PMML44Super.OutlierEffect method*), [179](#)
[to_etree\(\)](#) (*PMML44Super.Output method*), [180](#)
[to_etree\(\)](#) (*PMML44Super.OutputField method*), [182](#)
[to_etree\(\)](#) (*PMML44Super.PairCounts method*), [191](#)
[to_etree\(\)](#) (*PMML44Super.Parameter method*), [193](#)
[to_etree\(\)](#) (*PMML44Super.ParameterField method*), [193](#)
[to_etree\(\)](#) (*PMML44Super.ParameterList method*), [194](#)
[to_etree\(\)](#) (*PMML44Super.ParamMatrix method*), [192](#)
[to_etree\(\)](#) (*PMML44Super.ParentValue method*), [195](#)
[to_etree\(\)](#) (*PMML44Super.Partition method*), [196](#)
[to_etree\(\)](#) (*PMML44Super.PartitionFieldStats method*), [197](#)
[to_etree\(\)](#) (*PMML44Super.PastVariances method*), [197](#)
[to_etree\(\)](#) (*PMML44Super.PCell method*), [183](#)
[to_etree\(\)](#) (*PMML44Super.PCovCell method*), [184](#)
[to_etree\(\)](#) (*PMML44Super.PCovMatrix method*), [185](#)
[to_etree\(\)](#) (*PMML44Super.PMML method*), [189](#)
[to_etree\(\)](#) (*PMML44Super.PoissonDistribution method*), [198](#)
[to_etree\(\)](#) (*PMML44Super.PolynomialKernelType method*), [199](#)
[to_etree\(\)](#) (*PMML44Super.PPCell method*), [190](#)
[to_etree\(\)](#) (*PMML44Super.PPMatrix method*), [190](#)
[to_etree\(\)](#) (*PMML44Super.PredictiveModelQuality method*), [201](#)
[to_etree\(\)](#) (*PMML44Super.Predictor method*), [202](#)
[to_etree\(\)](#) (*PMML44Super.PredictorTerm method*), [203](#)
[to_etree\(\)](#) (*PMML44Super.PsiVector method*), [204](#)
[to_etree\(\)](#) (*PMML44Super.Quantile method*), [204](#)
[to_etree\(\)](#) (*PMML44Super.RadialBasisKernel method*), [209](#)
[to_etree\(\)](#) (*PMML44Super.RadialBasisKernelType method*), [210](#)
[to_etree\(\)](#) (*PMML44Super.RandomLiftGraph method*), [211](#)
[to_etree\(\)](#) (*PMML44Super.REAL_Entries method*), [205](#)
[to_etree\(\)](#) (*PMML44Super.REAL_SparseArray method*), [206](#)
[to_etree\(\)](#) (*PMML44Super.Regression method*), [212](#)
[to_etree\(\)](#) (*PMML44Super.RegressionModel method*), [214](#)
[to_etree\(\)](#) (*PMML44Super.RegressionTable method*), [215](#)
[to_etree\(\)](#) (*PMML44Super.RegressorValues method*), [216](#)
[to_etree\(\)](#) (*PMML44Super.Residuals method*), [217](#)
[to_etree\(\)](#) (*PMML44Super.ResidualSquareCoefficients method*), [216](#)
[to_etree\(\)](#) (*PMML44Super.ResultField method*), [218](#)
[to_etree\(\)](#) (*PMML44Super.RMSprop method*), [207](#)
[to_etree\(\)](#) (*PMML44Super.ROC method*), [208](#)
[to_etree\(\)](#) (*PMML44Super.ROCGraph method*), [208](#)
[to_etree\(\)](#) (*PMML44Super.row method*), [303](#)
[to_etree\(\)](#) (*PMML44Super.RuleSelectionMethod method*), [219](#)
[to_etree\(\)](#) (*PMML44Super.RuleSet method*), [220](#)
[to_etree\(\)](#) (*PMML44Super.RuleSetModel method*), [222](#)
[to_etree\(\)](#) (*PMML44Super.Scorecard method*), [225](#)
[to_etree\(\)](#) (*PMML44Super.ScoreDistribution method*), [223](#)
[to_etree\(\)](#) (*PMML44Super.SeasonalComponent method*), [226](#)
[to_etree\(\)](#) (*PMML44Super.SeasonalFactor method*), [227](#)
[to_etree\(\)](#) (*PMML44Super.Seasonality_ExpoSmooth method*), [228](#)
[to_etree\(\)](#) (*PMML44Super.SeasonalTrendDecomposition method*), [227](#)
[to_etree\(\)](#) (*PMML44Super.Segment method*), [230](#)
[to_etree\(\)](#) (*PMML44Super.Segmentation method*),

- 231
- `to_etree()` (*PMML44Super.Sequence* method), 233
- `to_etree()` (*PMML44Super.SequenceModel* method), 235
- `to_etree()` (*PMML44Super.SequenceReference* method), 236
- `to_etree()` (*PMML44Super.SequenceRule* method), 237
- `to_etree()` (*PMML44Super.SetPredicate* method), 238
- `to_etree()` (*PMML44Super.SetReference* method), 238
- `to_etree()` (*PMML44Super.SGD* method), 222
- `to_etree()` (*PMML44Super.SigmoidKernelType* method), 239
- `to_etree()` (*PMML44Super.simpleMatching* method), 303
- `to_etree()` (*PMML44Super.SimplePredicate* method), 240
- `to_etree()` (*PMML44Super.SimpleRule* method), 241
- `to_etree()` (*PMML44Super.SimpleSetPredicate* method), 242
- `to_etree()` (*PMML44Super.SpectralAnalysis* method), 243
- `to_etree()` (*PMML44Super.squaredEuclidean* method), 304
- `to_etree()` (*PMML44Super.StateSpaceModel* method), 244
- `to_etree()` (*PMML44Super.StateVector* method), 245
- `to_etree()` (*PMML44Super.SupportVector* method), 245
- `to_etree()` (*PMML44Super.SupportVectorMachine* method), 246
- `to_etree()` (*PMML44Super.SupportVectorMachineModel* method), 249
- `to_etree()` (*PMML44Super.SupportVectors* method), 250
- `to_etree()` (*PMML44Super.TableLocator* method), 250
- `to_etree()` (*PMML44Super.tanimoto* method), 304
- `to_etree()` (*PMML44Super.Target* method), 251
- `to_etree()` (*PMML44Super.Targets* method), 256
- `to_etree()` (*PMML44Super.TargetValue* method), 252
- `to_etree()` (*PMML44Super.TargetValueCount* method), 253
- `to_etree()` (*PMML44Super.TargetValueCounts* method), 254
- `to_etree()` (*PMML44Super.TargetValueStat* method), 255
- `to_etree()` (*PMML44Super.TargetValueStats* method), 255
- `to_etree()` (*PMML44Super.Taxonomy* method), 257
- `to_etree()` (*PMML44Super.TestDistributions* method), 258
- `to_etree()` (*PMML44Super.TextCorpus* method), 259
- `to_etree()` (*PMML44Super.TextDictionary* method), 260
- `to_etree()` (*PMML44Super.TextDocument* method), 260
- `to_etree()` (*PMML44Super.TextIndex* method), 262
- `to_etree()` (*PMML44Super.TextIndexNormalization* method), 264
- `to_etree()` (*PMML44Super.TextModel* method), 265
- `to_etree()` (*PMML44Super.TextModelNormalization* method), 266
- `to_etree()` (*PMML44Super.TextModelSimiliarity* method), 267
- `to_etree()` (*PMML44Super.Theta* method), 267
- `to_etree()` (*PMML44Super.ThetaRecursionState* method), 268
- `to_etree()` (*PMML44Super.Time* method), 269
- `to_etree()` (*PMML44Super.TimeAnchor* method), 270
- `to_etree()` (*PMML44Super.TimeCycle* method), 271
- `to_etree()` (*PMML44Super.TimeException* method), 271
- `to_etree()` (*PMML44Super.TimeSeries* method), 272
- `to_etree()` (*PMML44Super.TimeSeriesModel* method), 274
- `to_etree()` (*PMML44Super.Timestamp* method), 276
- `to_etree()` (*PMML44Super.TimeValue* method), 275
- `to_etree()` (*PMML44Super.TrainingInstances* method), 277
- `to_etree()` (*PMML44Super.TrainingParameters* method), 278
- `to_etree()` (*PMML44Super.TransferFunctionValues* method), 278
- `to_etree()` (*PMML44Super.TransformationDictionary* method), 279
- `to_etree()` (*PMML44Super.TransitionMatrix* method), 280
- `to_etree()` (*PMML44Super.TreeModel* method), 282
- `to_etree()` (*PMML44Super.Trend_ExpoSmooth* method), 283
- `to_etree()` (*PMML44Super.TrendCoefficients* method), 282
- `to_etree()` (*PMML44Super.TriangularDistributionForBN* method), 284
- `to_etree()` (*PMML44Super.True_* method), 284
- `to_etree()` (*PMML44Super.UniformDistribution* method), 285
- `to_etree()` (*PMML44Super.UniformDistributionForBN* method), 286
- `to_etree()` (*PMML44Super.UnivariateStats* method), 287
- `to_etree()` (*PMML44Super.Upper* method), 288
- `to_etree()` (*PMML44Super.Value* method), 289

[to_etree\(\)](#) (*PMML44Super.ValueProbability method*), 290
[to_etree\(\)](#) (*PMML44Super.VariableWeight method*), 290
[to_etree\(\)](#) (*PMML44Super.Variance method*), 292
[to_etree\(\)](#) (*PMML44Super.VarianceCoefficients method*), 292
[to_etree\(\)](#) (*PMML44Super.VectorDictionary method*), 293
[to_etree\(\)](#) (*PMML44Super.VectorFields method*), 294
[to_etree\(\)](#) (*PMML44Super.VectorInstance method*), 295
[to_etree\(\)](#) (*PMML44Super.VerificationField method*), 296
[to_etree\(\)](#) (*PMML44Super.VerificationFields method*), 297
[to_etree\(\)](#) (*PMML44Super.XCoordinates method*), 297
[to_etree\(\)](#) (*PMML44Super.YCoordinates method*), 298
[TrainingInstances](#) (*class in PMML44Super*), 276
[TrainingParameters](#) (*class in PMML44Super*), 277
[TransferFunctionValues](#) (*class in PMML44Super*), 278
[TransformationDictionary](#) (*class in PMML44Super*), 278
[TransitionMatrix](#) (*class in PMML44Super*), 279
[TreeModel](#) (*class in PMML44Super*), 280
[Trend_ExpoSmooth](#) (*class in PMML44Super*), 282
[TrendCoefficients](#) (*class in PMML44Super*), 282
[TriangularDistributionForBN](#) (*class in PMML44Super*), 283
[True_](#) (*class in PMML44Super*), 284

U

[UniformDistribution](#) (*class in PMML44Super*), 284
[UniformDistributionForBN](#) (*class in PMML44Super*), 285
[UnivariateStats](#) (*class in PMML44Super*), 286
[Upper](#) (*class in PMML44Super*), 287

V

[validate_ACTIVATION_FUNCTION\(\)](#) (*PMML44Super.NeuralLayer method*), 163
[validate_ACTIVATION_FUNCTION\(\)](#) (*PMML44Super.NeuralNetwork method*), 165
[validate_ALGORITHM_TYPE\(\)](#) (*PMML44Super.AnomalyDetectionModel method*), 16
[validate_ARCHITECTURENAME\(\)](#) (*PMML44Super.DerivedField method*), 84
[validate_BACKEND\(\)](#) (*PMML44Super.DerivedField method*), 84
[validate_BASELINE_TEST_STATISTIC\(\)](#) (*PMML44Super.TestDistributions method*), 258
[validate_BN_TYPE\(\)](#) (*PMML44Super.BayesianNetworkModel method*), 36
[validate_CAT_SCORING_METHOD\(\)](#) (*PMML44Super.NearestNeighborModel method*), 159
[validate_COMPARE_FUNCTION\(\)](#) (*PMML44Super.ClusteringField method*), 48
[validate_COMPARE_FUNCTION\(\)](#) (*PMML44Super.ComparisonMeasure method*), 54
[validate_COMPARE_FUNCTION\(\)](#) (*PMML44Super.KNNInput method*), 119
[validate_CONT_SCORING_METHOD\(\)](#) (*PMML44Super.NearestNeighborModel method*), 159
[validate_CUMULATIVE_LINK_FUNCTION\(\)](#) (*PMML44Super.GeneralRegressionModel method*), 109
[validate_DATATYPE\(\)](#) (*PMML44Super.Constant method*), 61
[validate_DATATYPE\(\)](#) (*PMML44Super.DataField method*), 74
[validate_DATATYPE\(\)](#) (*PMML44Super.DefineFunction method*), 81
[validate_DATATYPE\(\)](#) (*PMML44Super.DerivedField method*), 84
[validate_DATATYPE\(\)](#) (*PMML44Super.Discretize method*), 88
[validate_DATATYPE\(\)](#) (*PMML44Super.MapValues method*), 138
[validate_DATATYPE\(\)](#) (*PMML44Super.OutputField method*), 182
[validate_DATATYPE\(\)](#) (*PMML44Super.ParameterField method*), 193
[validate_DATATYPE\(\)](#) (*PMML44Super.ResultField method*), 218
[validate_DELIMITER\(\)](#) (*PMML44Super.Delimiter method*), 82
[validate_DNN_ACTIVATION_FUNCTION\(\)](#) (*PMML44Super.LayerParameters method*), 128
[validate_ELEMENT_ID\(\)](#) (*PMML44Super.Sequence method*), 233
[validate_ELEMENT_ID\(\)](#) (*PMML44Super.SequenceReference method*), 236

[validate_ELEMENT_ID\(\)](#)
 (PMML44Super.SequenceRule method), [237](#)
[validate_ELEMENT_ID\(\)](#)
 (PMML44Super.SetPredicate method), [238](#)
[validate_ELEMENT_ID\(\)](#)
 (PMML44Super.SetReference method), [238](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.Aggregate method), [12](#)
[validate_FIELD_NAME\(\)](#) (PMML44Super.Anova method), [17](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.BayesInput method), [33](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.BayesOutput method), [35](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.BlockIndicator method), [38](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.CategoricalPredictor method), [41](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.ClusteringField method), [48](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.ContinuousNode method), [67](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.DataField method), [74](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.DerivedField method), [84](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.DiscreteNode method), [87](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.Discretize method), [88](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.DynamicRegressor method), [91](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.FieldColumnPair method), [95](#)
[validate_FIELD_NAME\(\)](#) (PMML44Super.FieldRef method), [96](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.FieldValue method), [97](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.FieldValueCount method), [98](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.GeneralRegressionModel method), [109](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.InstanceField method), [114](#)
[validate_FIELD_NAME\(\)](#) (PMML44Super.Item method), [117](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.KNNInput method), [119](#)
[validate_FIELD_NAME\(\)](#) (PMML44Super.Lag method), [122](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.MiningField method), [145](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.NearestNeighborModel method), [159](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.NetworkLayer method), [160](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.NormContinuous method), [172](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.NormDiscrete method), [172](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.NumericPredictor method), [176](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.OutputField method), [182](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.ParameterField method), [193](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.ParentValue method), [195](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.PartitionFieldStats method), [197](#)
[validate_FIELD_NAME\(\)](#) (PMML44Super.PPCell method), [190](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.PredictiveModelQuality method), [201](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.Predictor method), [202](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.PredictorTerm method), [203](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.RegressionModel method), [214](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.ResultField method), [218](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.SetPredicate method), [238](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.SimplePredicate method), [240](#)
[validate_FIELD_NAME\(\)](#)
 (PMML44Super.SimpleSetPredicate method),

[242](#)
 validate_FIELD_NAME() (PMML44Super.Target method), [251](#)
 validate_FIELD_NAME() (PMML44Super.TestDistributions method), [258](#)
 validate_FIELD_NAME() (PMML44Super.TextIndex method), [262](#)
 validate_FIELD_NAME() (PMML44Super.TimeSeries method), [272](#)
 validate_FIELD_NAME() (PMML44Super.UnivariateStats method), [287](#)
 validate_FIELD_NAME() (PMML44Super.VariableWeight method), [290](#)
 validate_FIELD_NAME() (PMML44Super.VerificationField method), [296](#)
 validate_FIELD_USAGE_TYPE() (PMML44Super.MiningField method), [145](#)
 validate_GAP() (PMML44Super.Delimiter method), [82](#)
 validate_Indices() (PMML44Super.INT_SparseArray method), [112](#)
 validate_Indices() (PMML44Super.REAL_SparseArray method), [206](#)
 validate_INFERENCE_TYPE() (PMML44Super.BayesianNetworkModel method), [36](#)
 validate_INT_Entries() (PMML44Super.INT_SparseArray method), [112](#)
 validate_INT_NUMBER() (PMML44Super.ARMAPart method), [7](#)
 validate_INT_NUMBER() (PMML44Super.ArrayType method), [23](#)
 validate_INT_NUMBER() (PMML44Super.AssociationModel method), [25](#)
 validate_INT_NUMBER() (PMML44Super.ClusteringModel method), [50](#)
 validate_INT_NUMBER() (PMML44Super.Coefficients method), [52](#)
 validate_INT_NUMBER() (PMML44Super.Constraints method), [63](#)
 validate_INT_NUMBER() (PMML44Super.DynamicRegressor method), [91](#)
 validate_INT_NUMBER() (PMML44Super.GARCHPart method), [103](#)
 validate_INT_NUMBER() (PMML44Super.GeneralRegressionModel method), [109](#)
 validate_INT_NUMBER() (PMML44Super.INT_SparseArray method), [112](#)
 validate_INT_NUMBER() (PMML44Super.LayerParameters method), [128](#)
 validate_INT_NUMBER() (PMML44Super.MatCell method), [139](#)
 validate_INT_NUMBER() (PMML44Super.Matrix method), [140](#)
 validate_INT_NUMBER() (PMML44Super.MaximumLikelihoodStat method), [141](#)
 validate_INT_NUMBER() (PMML44Super.Metrics method), [143](#)
 validate_INT_NUMBER() (PMML44Super.ModelVerification method), [152](#)
 validate_INT_NUMBER() (PMML44Super.MultivariateStat method), [153](#)
 validate_INT_NUMBER() (PMML44Super.NearestNeighborModel method), [159](#)
 validate_INT_NUMBER() (PMML44Super.NonseasonalComponent method), [170](#)
 validate_INT_NUMBER() (PMML44Super.NonseasonalFactor method), [171](#)
 validate_INT_NUMBER() (PMML44Super.NumericPredictor method), [176](#)
 validate_INT_NUMBER() (PMML44Super.OutputField method), [182](#)
 validate_INT_NUMBER() (PMML44Super.PCell method), [183](#)
 validate_INT_NUMBER() (PMML44Super.REAL_SparseArray method), [206](#)
 validate_INT_NUMBER() (PMML44Super.SeasonalComponent method), [226](#)
 validate_INT_NUMBER() (PMML44Super.SeasonalFactor method), [227](#)
 validate_INT_NUMBER() (PMML44Super.Seasonality_ExpoSmooth method), [228](#)
 validate_INT_NUMBER() (PMML44Super.Sequence method), [233](#)
 validate_INT_NUMBER()

(PMML44Super.SequenceModel method), 235
 validate_INT_NUMBER() (PMML44Super.SequenceRule method), 237
 validate_INT_NUMBER() (PMML44Super.SupportVectors method), 250
 validate_INT_NUMBER() (PMML44Super.TestDistributions method), 258
 validate_INT_NUMBER() (PMML44Super.TextDocument method), 260
 validate_INT_NUMBER() (PMML44Super.Theta method), 267
 validate_INT_NUMBER() (PMML44Super.TimeAnchor method), 270
 validate_INT_NUMBER() (PMML44Super.TimeCycle method), 271
 validate_INT_NUMBER() (PMML44Super.TimeException method), 271
 validate_INT_NUMBER() (PMML44Super.TimeValue method), 275
 validate_INT_NUMBER() (PMML44Super.TrainingInstances method), 277
 validate_INT_NUMBER() (PMML44Super.VectorDictionary method), 293
 validate_INT_NUMBER() (PMML44Super.VectorFields method), 294
 validate_INTERPOLATION_METHOD() (PMML44Super.TimeSeries method), 272
 validate_INVALID_VALUE_TREATMENT_METHOD() (PMML44Super.Apply method), 22
 validate_INVALID_VALUE_TREATMENT_METHOD() (PMML44Super.MiningField method), 145
 validate_LAYER_TYPE() (PMML44Super.NetworkLayer method), 160
 validate_LAYEROP_TYPE() (PMML44Super.LayerParameters method), 128
 validate_LINK_FUNCTION() (PMML44Super.GeneralRegressionModel method), 109
 validate_LOSS_TYPE() (PMML44Super.Losses method), 135
 validate_METRIC_TYPE() (PMML44Super.Metrics method), 143
 validate_MINING_FUNCTION() (PMML44Super.AnomalyDetectionModel method), 16
 validate_MINING_FUNCTION() (PMML44Super.AssociationModel method), 25
 validate_MINING_FUNCTION() (PMML44Super.BaselineModel method), 31
 validate_MINING_FUNCTION() (PMML44Super.BayesianNetworkModel method), 36
 validate_MINING_FUNCTION() (PMML44Super.ClusteringModel method), 50
 validate_MINING_FUNCTION() (PMML44Super.DecisionTree method), 76
 validate_MINING_FUNCTION() (PMML44Super.DeepNetwork method), 79
 validate_MINING_FUNCTION() (PMML44Super.GaussianProcessModel method), 105
 validate_MINING_FUNCTION() (PMML44Super.GeneralRegressionModel method), 109
 validate_MINING_FUNCTION() (PMML44Super.MiningModel method), 147
 validate_MINING_FUNCTION() (PMML44Super.NaiveBayesModel method), 157
 validate_MINING_FUNCTION() (PMML44Super.NearestNeighborModel method), 159
 validate_MINING_FUNCTION() (PMML44Super.NeuralNetwork method), 165
 validate_MINING_FUNCTION() (PMML44Super.Regression method), 212
 validate_MINING_FUNCTION() (PMML44Super.RegressionModel method), 214
 validate_MINING_FUNCTION() (PMML44Super.RuleSetModel method), 222
 validate_MINING_FUNCTION() (PMML44Super.Scorecard method), 225
 validate_MINING_FUNCTION() (PMML44Super.SequenceModel method), 235
 validate_MINING_FUNCTION() (PMML44Super.SupportVectorMachineModel method), 249
 validate_MINING_FUNCTION() (PMML44Super.TextModel method), 265
 validate_MINING_FUNCTION() (PMML44Super.TimeSeriesModel method),

274
 validate_MINING_FUNCTION() (PMML44Super.TreeModel method), 282
 validate_MISSING_VALUE_STRATEGY() (PMML44Super.DecisionTree method), 76
 validate_MISSING_VALUE_STRATEGY() (PMML44Super.TreeModel method), 282
 validate_MISSING_VALUE_TREATMENT_METHOD() (PMML44Super.MiningField method), 145
 validate_MULTIPLE_MODEL_METHOD() (PMML44Super.Segmentation method), 231
 validate_NN_NEURON_ID() (PMML44Super.NeuralInput method), 161
 validate_NN_NEURON_ID() (PMML44Super.Neuron method), 167
 validate_NN_NEURON_IDREF() (PMML44Super.Con method), 59
 validate_NN_NEURON_IDREF() (PMML44Super.NeuralOutput method), 165
 validate_NN_NORMALIZATION_METHOD() (PMML44Super.DeepNetwork method), 79
 validate_NN_NORMALIZATION_METHOD() (PMML44Super.NetworkLayer method), 160
 validate_NN_NORMALIZATION_METHOD() (PMML44Super.NeuralLayer method), 163
 validate_NN_NORMALIZATION_METHOD() (PMML44Super.NeuralNetwork method), 165
 validate_NO_TRUE_CHILD_STRATEGY() (PMML44Super.DecisionTree method), 76
 validate_NO_TRUE_CHILD_STRATEGY() (PMML44Super.TreeModel method), 282
 validate_NUMBER() (PMML44Super.AnovaRow method), 18
 validate_NUMBER() (PMML44Super.Attribute method), 27
 validate_NUMBER() (PMML44Super.binarySimilarity method), 299
 validate_NUMBER() (PMML44Super.Characteristic method), 44
 validate_NUMBER() (PMML44Super.ClusteringModelQuality method), 51
 validate_NUMBER() (PMML44Super.ComparisonMeasure method), 54
 validate_NUMBER() (PMML44Super.ContStats method), 64
 validate_NUMBER() (PMML44Super.COUNT_TABLE_TYPE method), 40
 validate_NUMBER() (PMML44Super.Counts method), 70
 validate_NUMBER() (PMML44Super.FieldValueCount method), 98
 validate_NUMBER() (PMML44Super.Interval method), 116
 validate_NUMBER() (PMML44Super.LiftData method), 130
 validate_NUMBER() (PMML44Super.LinearNorm method), 133
 validate_NUMBER() (PMML44Super.MiningField method), 145
 validate_NUMBER() (PMML44Super.minkowski method), 302
 validate_NUMBER() (PMML44Super.MultivariateStat method), 154
 validate_NUMBER() (PMML44Super.Node method), 169
 validate_NUMBER() (PMML44Super.NormContinuous method), 172
 validate_NUMBER() (PMML44Super.NormDiscrete method), 172
 validate_NUMBER() (PMML44Super.NumericInfo method), 175
 validate_NUMBER() (PMML44Super.Partition method), 196
 validate_NUMBER() (PMML44Super.PredictiveModelQuality method), 201
 validate_NUMBER() (PMML44Super.Quantile method), 204
 validate_NUMBER() (PMML44Super.RuleSet method), 220
 validate_NUMBER() (PMML44Super.Scorecard method), 225
 validate_NUMBER() (PMML44Super.ScoreDistribution method), 223
 validate_NUMBER() (PMML44Super.Segment method), 230
 validate_NUMBER() (PMML44Super.SimpleRule method), 242
 validate_NUMBER() (PMML44Super.TargetValue method), 252
 validate_NUMBER() (PMML44Super.Time method), 269
 validate_NUMBER() (PMML44Super.TimeValue method), 275
 validate_OPTYPE() (PMML44Super.DataField method), 74
 validate_OPTYPE() (PMML44Super.DefineFunction method), 81

`validate_OPTYPE()` (*PMML44Super.DerivedField method*), 84
`validate_OPTYPE()` (*PMML44Super.MiningField method*), 145
`validate_OPTYPE()` (*PMML44Super.OutputField method*), 182
`validate_OPTYPE()` (*PMML44Super.ParameterField method*), 193
`validate_OPTYPE()` (*PMML44Super.ResultField method*), 218
`validate_OPTYPE()` (*PMML44Super.Target method*), 251
`validate_OUTLIER_TREATMENT_METHOD()` (*PMML44Super.MiningField method*), 145
`validate_OUTLIER_TREATMENT_METHOD()` (*PMML44Super.NormContinuous method*), 172
`validate_PERCENTAGE_NUMBER()` (*PMML44Super.Quantile method*), 204
`validate_PROB_NUMBER()` (*PMML44Super.AnovaRow method*), 18
`validate_PROB_NUMBER()` (*PMML44Super.AssociationModel method*), 25
`validate_PROB_NUMBER()` (*PMML44Super.AssociationRule method*), 26
`validate_PROB_NUMBER()` (*PMML44Super.DecisionTree method*), 77
`validate_PROB_NUMBER()` (*PMML44Super.Itemset method*), 118
`validate_PROB_NUMBER()` (*PMML44Super.MiningField method*), 146
`validate_PROB_NUMBER()` (*PMML44Super.MultivariateStat method*), 154
`validate_PROB_NUMBER()` (*PMML44Super.ScoreDistribution method*), 223
`validate_PROB_NUMBER()` (*PMML44Super.Segmentation method*), 231
`validate_PROB_NUMBER()` (*PMML44Super.TargetValue method*), 252
`validate_PROB_NUMBER()` (*PMML44Super.TreeModel method*), 282
`validate_PROB_NUMBER()` (*PMML44Super.ValueProbability method*), 290
`validate_REAL_Entries()` (*PMML44Super.REAL_SparseArray method*), 206
`validate_REAL_NUMBER()` (*PMML44Super.AbsoluteExponentialKernel method*), 8
`validate_REAL_NUMBER()` (*PMML44Super.Adadelta method*), 9
`validate_REAL_NUMBER()` (*PMML44Super.Adagrad method*), 9
`validate_REAL_NUMBER()` (*PMML44Super.Adam method*), 10
`validate_REAL_NUMBER()` (*PMML44Super.Adamax method*), 11
`validate_REAL_NUMBER()` (*PMML44Super.AnyDistribution method*), 19
`validate_REAL_NUMBER()` (*PMML44Super.ARDSquaredExponentialKernel method*), 4
`validate_REAL_NUMBER()` (*PMML44Super.ARIMA method*), 6
`validate_REAL_NUMBER()` (*PMML44Super.ARMAPart method*), 7
`validate_REAL_NUMBER()` (*PMML44Super.AssociationModel method*), 25
`validate_REAL_NUMBER()` (*PMML44Super.BaseCumHazardTables method*), 28
`validate_REAL_NUMBER()` (*PMML44Super.BaselineCell method*), 30
`validate_REAL_NUMBER()` (*PMML44Super.BaselineStratum method*), 32
`validate_REAL_NUMBER()` (*PMML44Super.CategoricalPredictor method*), 41
`validate_REAL_NUMBER()` (*PMML44Super.ClusteringField method*), 48
`validate_REAL_NUMBER()` (*PMML44Super.Coefficient method*), 51
`validate_REAL_NUMBER()` (*PMML44Super.Coefficients method*), 52
`validate_REAL_NUMBER()` (*PMML44Super.Con method*), 59
`validate_REAL_NUMBER()` (*PMML44Super.Constraints method*), 63
`validate_REAL_NUMBER()` (*PMML44Super.ContinuousConditionalProbability method*), 65
`validate_REAL_NUMBER()` (*PMML44Super.ContinuousNode method*), 67
`validate_REAL_NUMBER()` (*PMML44Super.DiscreteConditionalProbability method*), 86
`validate_REAL_NUMBER()` (*PMML44Super.DiscreteNode method*), 87
`validate_REAL_NUMBER()`

(PMML44Super.ExponentialSmoothing method), 92
 validate_REAL_NUMBER() *(PMML44Super.GARCHPart method)*, 103
 validate_REAL_NUMBER() *(PMML44Super.GaussianDistribution method)*, 103
 validate_REAL_NUMBER() *(PMML44Super.GeneralizedExponentialKernel method)*, 110
 validate_REAL_NUMBER() *(PMML44Super.GeneralRegressionModel method)*, 109
 validate_REAL_NUMBER() *(PMML44Super.Item method)*, 117
 validate_REAL_NUMBER() *(PMML44Super.KNNInput method)*, 119
 validate_REAL_NUMBER() *(PMML44Super.LayerParameters method)*, 128
 validate_REAL_NUMBER() *(PMML44Super.Level method)*, 129
 validate_REAL_NUMBER() *(PMML44Super.Matrix method)*, 140
 validate_REAL_NUMBER() *(PMML44Super.Nadam method)*, 155
 validate_REAL_NUMBER() *(PMML44Super.NaiveBayesModel method)*, 157
 validate_REAL_NUMBER() *(PMML44Super.NearestNeighborModel method)*, 159
 validate_REAL_NUMBER() *(PMML44Super.NeuralLayer method)*, 163
 validate_REAL_NUMBER() *(PMML44Super.NeuralNetwork method)*, 165
 validate_REAL_NUMBER() *(PMML44Super.Neuron method)*, 167
 validate_REAL_NUMBER() *(PMML44Super.NumericPredictor method)*, 176
 validate_REAL_NUMBER() *(PMML44Super.Optimizers method)*, 178
 validate_REAL_NUMBER() *(PMML44Super.OutlierEffect method)*, 179
 validate_REAL_NUMBER() *(PMML44Super.OutputField method)*, 182
 validate_REAL_NUMBER() *(PMML44Super.Parameter method)*, 193
 validate_REAL_NUMBER() *(PMML44Super.PCell method)*, 183
 validate_REAL_NUMBER() *(PMML44Super.PCovCell method)*, 184
 validate_REAL_NUMBER() *(PMML44Super.PoissonDistribution method)*, 198
 validate_REAL_NUMBER() *(PMML44Super.PolynomialKernelType method)*, 199
 validate_REAL_NUMBER() *(PMML44Super.PredictorTerm method)*, 203
 validate_REAL_NUMBER() *(PMML44Super.RadialBasisKernel method)*, 209
 validate_REAL_NUMBER() *(PMML44Super.RadialBasisKernelType method)*, 210
 validate_REAL_NUMBER() *(PMML44Super.REAL_SparseArray method)*, 206
 validate_REAL_NUMBER() *(PMML44Super.RegressionTable method)*, 215
 validate_REAL_NUMBER() *(PMML44Super.RMSprop method)*, 207
 validate_REAL_NUMBER() *(PMML44Super.Seasonality_ExpoSmooth method)*, 228
 validate_REAL_NUMBER() *(PMML44Super.Sequence method)*, 233
 validate_REAL_NUMBER() *(PMML44Super.SequenceModel method)*, 235
 validate_REAL_NUMBER() *(PMML44Super.SequenceRule method)*, 237
 validate_REAL_NUMBER() *(PMML44Super.SGD method)*, 222
 validate_REAL_NUMBER() *(PMML44Super.SigmoidKernelType method)*, 239
 validate_REAL_NUMBER() *(PMML44Super.StateSpaceModel method)*, 244
 validate_REAL_NUMBER() *(PMML44Super.SupportVectorMachine method)*, 246
 validate_REAL_NUMBER() *(PMML44Super.SupportVectorMachineModel method)*, 249
 validate_REAL_NUMBER() *(PMML44Super.TargetValueCount method)*, 253
 validate_REAL_NUMBER() *(PMML44Super.TestDistributions method)*, 258

`validate_REAL_NUMBER()`
(*PMML44Super.TimeSeries* method), 272

`validate_REAL_NUMBER()`
(*PMML44Super.TimeValue* method), 275

`validate_REAL_NUMBER()`
(*PMML44Super.Trend_ExpoSmooth* method), 283

`validate_REAL_NUMBER()`
(*PMML44Super.UniformDistribution* method), 285

`validate_REGRESSIONNORMALIZATIONMETHOD()`
(*PMML44Super.Regression* method), 212

`validate_REGRESSIONNORMALIZATIONMETHOD()`
(*PMML44Super.RegressionModel* method), 214

`validate_RESULT_FEATURE()`
(*PMML44Super.OutputField* method), 182

`validate_RESULT_FEATURE()`
(*PMML44Super.ResultField* method), 218

`validate_RULE_FEATURE()`
(*PMML44Super.OutputField* method), 182

`validate_SVM_CLASSIFICATION_METHOD()`
(*PMML44Super.SupportVectorMachineModel* method), 249

`validate_SVM_REPRESENTATION()`
(*PMML44Super.SupportVectorMachineModel* method), 249

`validate_TIME_ANCHOR()`
(*PMML44Super.TimeAnchor* method), 270

`validate_TIME_EXCEPTION_TYPE()`
(*PMML44Super.TimeException* method), 271

`validate_TIMESERIES_ALGORITHM()`
(*PMML44Super.TimeSeriesModel* method), 274

`validate_TIMESERIES_USAGE()`
(*PMML44Super.TimeSeries* method), 273

`validate_VALID_TIME_SPEC()`
(*PMML44Super.TimeCycle* method), 271

`validate_VECTOR_ID()`
(*PMML44Super.SupportVector* method), 245

`validate_VECTOR_ID()`
(*PMML44Super.VectorInstance* method), 295

Value (class in *PMML44Super*), 288

ValueProbability (class in *PMML44Super*), 289

VariableWeight (class in *PMML44Super*), 290

Variance (class in *PMML44Super*), 290

VarianceCoefficients (class in *PMML44Super*), 292

VectorDictionary (class in *PMML44Super*), 292

VectorFields (class in *PMML44Super*), 293

VectorInstance (class in *PMML44Super*), 294

VerificationField (class in *PMML44Super*), 295

VerificationFields (class in *PMML44Super*), 296

X

XCoordinates (class in *PMML44Super*), 297

Y

YCoordinates (class in *PMML44Super*), 297