

Package ‘pmml’

November 4, 2009

Type Package

Title Generate PMML for various models

Version 1.2.20

Date 2009-11-04

Author Graham Williams, Michael Hahsler (arules), Zementis Inc, Hemant Ishwaran, Udaya B. Kogalur, Rajarshi Guha

Maintainer Graham Williams <Graham.Williams@togaware.com>

Depends XML

Suggests arules, nnet, rpart, randomSurvivalForest, randomForest, kernlab

License GPL (>= 2)

Description The Predictive Modelling Markup Language (PMML) is a language for representing models, in an application independent way. Such models can then be loaded into other applications supporting PMML, including ADAPA from Zementis, Teradata Warehouse Miner and IBM’s DB2. The package provides a generic pmml function to generate pmml for an object. Using a S3 generic function the appropriate method for the class of the supplied object is dispatched. The package currently supports the export of PMML for linear regression, SVMs, rpart classification trees, randomSurvivalForest forest models, and kmeans clusters. This package is part of the Rattle toolkit.

URL <http://rattle.togaware.com/>

Repository CRAN

Date/Publication 2009-11-04 07:53:29

R topics documented:

pmml	2
pmml.coxph	3
pmml.hclust	4

pmml.kmeans	5
pmml.ksvm	6
pmml.lm	7
pmml.nnet	8
pmml.rpart	10
pmml.rsf	12
pmml.rules	14
pmmlCanExport	15
pmmltoc	16

Index	17
--------------	-----------

pmml	<i>Generate PMML for an R object</i>
------	--------------------------------------

Description

'pmml' is a generic function implementing S3 methods used to produce the PMML (Predictive Model Markup Language) representation of an R model. The PMML can then be imported into other systems that accept PMML.

Usage

```
pmml(model, model.name="Rattle_Model", app.name="Rattle/PMML",
      description=NULL, copyright=NULL, transforms=NULL, dataset=NULL, ...)
```

Arguments

model	an object to be converted to PMML.
model.name	a name to give to the model in the PMML.
app.name	the name of the application that generated the PMML.
description	a descriptive text for the header of the PMML.
copyright	the copyright notice for the model.
transforms	a coded list of transforms performed.
dataset	the original training dataset, if available.
...	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application, such as Teradata Warehouse Miner and DB2. Generally, these applications convert the PMML into SQL for execution across a database.

Author(s)

⟨Graham.Williams@togaware.com⟩

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

See Also

`pmml.rules`, `pmml.hclust`, `pmml.kmeans`, `pmml.ksvm`, `pmml.lm`, `pmml.nnet`, `pmml.rpart`,
`pmml.rsrf`, `pmmltoc`.

Examples

```
# Build a simple lm model
(iris.lm <- lm(Sepal.Length ~ ., data=iris))

# Convert to pmml
pmml(iris.lm)
```

`pmml.coxph`

Generate PMML for a coxph object

Description

Generate the PMML (Predictive Model Markup Language) representation of a **coxph** object. The PMML can then be imported into other systems that accept PMML.

Usage

```
## S3 method for class 'coxph':
pmml(model, model.name="CoxPH_Survival_Regression_Model", app.name="Rattle/PMML",
      description="CoxPH Survival Regression Model", copyright=NULL, transforms=NULL)
```

Arguments

<code>model</code>	a <code>coxph</code> object.
<code>model.name</code>	a name to give to the model in the PMML.
<code>app.name</code>	the name of the application that generated the PMML.
<code>description</code>	a descriptive text for the header of the PMML.
<code>copyright</code>	the copyright notice for the model.
<code>transforms</code>	a coded list of transforms performed.
<code>...</code>	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application, such as Zementis' ADAPA.

Only numeric risk regression is supported currently.

Author(s)

⟨Graham.Williams@togaware.com⟩

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

Zementis' useful PMML convert: http://www.zementis.com/pmml_converters.htm

See Also

[pmml.](#)

pmml.hclust

Generate PMML for a hclust object

Description

Generate a PMML representation for a hierarchical cluster object. The hclust object will be approximated by k centroids and is converted into a PMML representation for kmeans clusters. The PMML can then be imported into other systems that accept PMML.

Usage

```
## S3 method for class 'hclust':
pmml(model, model.name="HClust_Model", app.name="Rattle/PMML",
      description="Hierarchical cluster model", copyright=NULL,
      transforms=NULL, dataset=NULL, centers, ...)
```

Arguments

model	a hclust object.
model.name	a name to give to the model in the PMML.
app.name	the name of the application that generated the PMML.
description	a descriptive text for the header of the PMML.
copyright	the copyright notice for the model.
transforms	a coded list of transforms performed.
dataset	the original training dataset, if available.
centers	a list of means to represent the clusters.
...	further arguments passed to or from other methods.

Author(s)

⟨Graham.Williams@togaware.com⟩

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

See Also

[pmml](#), [hclust](#).

pmml.kmeans

Generate PMML for a kmeans object

Description

Generate the PMML representation for a KMeans object (cluster). The kmeans object (a cluster described by k centroids) is converted into a PMML representation. The PMML can then be imported into other systems that accept PMML.

Usage

```
## S3 method for class 'kmeans':
pmml(model, model.name="KMeans_Model", app.name="Rattle/PMML",
      description="KMeans cluster model", copyright=NULL, transforms=NULL, ...)
```

Arguments

model	a kmeans object.
model.name	a name to give to the model in the PMML.
app.name	the name of the application that generated the PMML.
description	a descriptive text for the header of the PMML.
copyright	the copyright notice for the model.
transforms	a coded list of transforms performed.
...	further arguments passed to or from other methods.

Author(s)

⟨Graham.Williams@togaware.com⟩

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

See Also

[pmml](#), [kmeans](#).

Examples

```
ds <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
            matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
colnames(ds) <- c("Dimension1", "Dimension2")
cl <- kmeans(ds, 2)
pmml(cl)
```

pmml.ksvm

Generate PMML for a ksvm object

Description

Generate the PMML representation for a ksvm object (SVM). The PMML can then be imported into other systems that accept PMML. With this code, a PMML representation can be obtained for SVMs implementing classification (multi-class and binary) as well as regression.

Usage

```
## S3 method for class 'ksvm':
pmml(model, model.name="SVM_model", app.name="Rattle/PMML",
      description="Support Vector Machine PMML Model", copyright=NULL,
      transforms=NULL, dataset=NULL, ...)
```

Arguments

model	a ksvm object.
dataset	the original dataset used to train the SVM model in ksvm - required since the ksvm object does not appear to record information about the used categorical variables.
model.name	a name to give to the model in the PMML.
app.name	the name of the application that generated the PMML.
description	a descriptive text for the header of the PMML.
copyright	the copyright notice for the model.
transforms	a coded list of transforms performed.
...	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application that recognizes PMML 3.2. An example is ADAPA (Adaptive Decision and Predictive Analytics), a lightweight decision engine with batch and real-time scoring of predictive models (also supporting neural networks and linear and logistic regression).

Author(s)

<info@zementis.com>

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

Zementis' useful PMML convert: http://www.zementis.com/pmml_converters.htm

ADAPA home page: <http://www.zementis.com/adapa.htm>

See Also

[pmml.ksvm](#).

Examples

```
# Train a support vector machine to perform binary classification.
require(kernlab)
data(spam)
index <- sample(1:dim(spam)[1])
ds <- spam[index[1:300],] # For illustration only use a small dataset.
fit <- ksvm(type~., data=ds, kenrel="rbfdot")

# Genetate the PMML.
pmml(fit, dataset=ds)
```

pmml.lm

Generate PMML for an lm object

Description

Generate the PMML (Predictive Model Markup Language) representation of an **lm** object. The PMML can then be imported into other systems that accept PMML.

Usage

```
## S3 method for class 'lm':
pmml(model, model.name="Linear_Regression_Model", app.name="Rattle/PMML",
      description="Linear Regression Model", copyright=NULL, transforms=NULL, ...)
```

Arguments

<code>model</code>	an lm object.
<code>model.name</code>	a name to give to the model in the PMML.
<code>app.name</code>	the name of the application that generated the PMML.
<code>description</code>	a descriptive text for the header of the PMML.
<code>copyright</code>	the copyright notice for the model.
<code>transforms</code>	a coded list of transforms performed.
<code>...</code>	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application, such as Zementis' ADAPA.

Currently, the resultant PMML document will not encode interaction terms.

Only numeric regression is supported currently. Generalised linear models (logistic regression) are not yet supported.

Author(s)

<rguha@indiana.edu>

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

Zementis' useful PMML convert: http://www.zementis.com/pmml_converters.htm

See Also

[pmml.](#)

Examples

```
# Build a simple lm model
(iris.lm <- lm(Sepal.Length ~ ., data=iris))

# Convert to pmml
pmml(iris.lm)
```

pmml.nnet

Generate PMML for a nnet object

Description

Generate the PMML representation for a nnet object (Neural Network). The nnet object is converted into a PMML representation. The PMML can then be imported into other systems that accept PMML. With this code, a PMML representation can be obtained for Neural Networks implementing classification (multi-class and binary) as well as regression.

Usage

```
## S3 method for class 'nnet':
pmml(model, model.name="NeuralNet_model", app.name="Rattle/PMML",
      description="Neural Network PMML Model", copyright=NULL,
      transforms=NULL, ...)
```

Arguments

<code>model</code>	a nnet object.
<code>model.name</code>	a name to give to the model in the PMML.
<code>app.name</code>	the name of the application that generated the PMML.
<code>description</code>	a descriptive text for the header of the PMML.
<code>copyright</code>	the copyright notice for the model.
<code>transforms</code>	a coded list of transforms performed.
<code>...</code>	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application that recognizes PMML 3.2. An example is ADAPA. ADAPA (Adaptive Decision and Predictive Analytics) is a lightweight decision engine that offers at its core batch and real-time scoring of predictive models as well as fast execution of business rules. ADAPA supports an extensive collection of PMML elements, including the following predictive techniques: 1) Neural Networks (Backpropagation and Neural Gas); 2) Support Vector Machines; 3) Linear and Logistic Regression as well as all general regression PMML models: a) Multinomial Logistic; b) General Linear; 3) Ordinal Multinomial; 4) Simple Regression; and 5) Generalized Linear Model. ADAPA provides a reliable and fast way to manage, deploy, and execute a multitude of models and decision strategies.

Author(s)

`<info@zementis.com>`

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

Zementis' useful PMML convert: http://www.zementis.com/pmml_converters.htm

ADAPA home page: <http://www.zementis.com/adapa.htm>

See Also

[pmml.](#)

pmml.rpart

Generate PMML for an rpart object

Description

Generate the PMML (Predictive Model Markup Language) representation of an **rpart** object (classification tree). The rpart object (currently expected to be a classification tree) is converted into a PMML representation. The PMML can then be imported into other systems that accept PMML.

Usage

```
## S3 method for class 'rpart':
pmml(model, model.name="RPart_Model", app.name="Rattle/PMML",
      description="RPart Decision Tree Model", copyright=NULL,
      transforms=NULL, dataset=NULL, ...)
```

Arguments

model	an rpart object.
model.name	a name to give to the model in the PMML.
app.name	the name of the application that generated the PMML.
description	a descriptive text for the header of the PMML.
copyright	the copyright notice for the model.
transforms	a coded list of transforms performed.
dataset	the original training dataset, if available.
...	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application, such as Teradata Warehouse Miner and DB2. Generally, these applications convert the PMML into SQL for execution across a database.

Teradata, for example, generates a single SELECT statement to implement a decision tree. In the Examples section below, we use the rpart example to build a model stored in the variable fit. A segment of the PMML for this model is:

```
<Node score="absent" recordCount="81">
  <True/>
  <Node score="absent" recordCount="62">
    <SimplePredicate field="Start" operator="greaterOrEqual"
      value="8.5"/>
    <Node score="absent" recordCount="29">
      <SimplePredicate field="Start" operator="greaterOrEqual"
        value="14.5"/>
```

```

</Node>
<Node score="absent" recordCount="33">
  <SimplePredicate field="Start" operator="lessThan"
    value="14.5"/>
  <Node score="absent" recordCount="12">
    <SimplePredicate field="Age" operator="lessThan"
      value="55"/>
  </Node>
  <Node score="absent" recordCount="21">
    <SimplePredicate field="Age" operator="greaterOrEqual"
      value="55"/>
    <Node score="absent" recordCount="14">
      <SimplePredicate field="Age" operator="greaterOrEqual"
        value="111"/>
    </Node>
    <Node score="present" recordCount="7">
      <SimplePredicate field="Age" operator="lessThan"
        value="111"/>
    </Node>
  </Node>
</Node>
</Node>
</Node>
</Node>
<Node score="present" recordCount="19">
  <SimplePredicate field="Start" operator="lessThan"
    value="8.5"/>
</Node>
</Node>

```

The resulting SQL from Teradata includes:

```

CREATE TABLE "MyScores" AS (
  SELECT "UserID",
    (CASE WHEN _node = 0 THEN 'absent'
      WHEN _node = 1 THEN 'absent'
      WHEN _node = 2 THEN 'absent'
      WHEN _node = 3 THEN 'present'
      WHEN _node = 4 THEN 'present'
      ELSE NULL END)
    (VARCHAR(8)) AS "Kyphosis"
  FROM
    (SELECT "UserID",
      (CASE WHEN ("Start" >= 8.5) AND ("Start" >= 14.5)
        THEN 0
        WHEN ("Start" >= 8.5) AND ("Start" < 14.5)
          AND ("Age" < 55)
        THEN 1
        WHEN ("Start" >= 8.5) AND ("Start" < 14.5)
          AND ("Age" >= 55) AND ("Age" >= 111)
        THEN 2

```

```

        WHEN ("Start" >= 8.5) AND ("Start" < 14.5)
        AND ("Age" >= 55) AND ("Age" < 111)
        THEN 3
        WHEN ("Start" < 8.5)
        THEN 4
        ELSE -1 END) AS _node
    FROM "MyData" WHERE _node IS NOT NULL) A
    WHERE "Kyphosis" IS NOT NULL)
WITH DATA UNIQUE PRIMARY INDEX ("UserID");

```

Author(s)

⟨Graham.Williams@togaware.com⟩

References

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

Zementis' useful PMML convert: http://www.zementis.com/pmml_converters.htm

See Also

[pmml](#).

Examples

```

library(rpart)
(iris.rpart <- rpart(Species ~ ., data=iris))
pmml(iris.rpart)

```

pmml.rsf

Generate PMML for a Random Survival Forest (rsf) object

Description

Generate the Predictive Model Markup Language (PMML) representation of a **randomSurvival-Forest** forest object. In particular, this function gives the user the ability to save the geometry of a forest as a PMML XML document.

Usage

```

## S3 method for class 'rsf':
pmml(model, model.name="rsfForest_Model", app.name="Rattle/PMML",
      description="Random Survival Forest Tree Model", copyright, ...)

```

Arguments

<code>model</code>	the <code>forest</code> object contained in an object of class randomSurvivalForest , as that contained in the object returned by the function <code>rsf</code> with the parameter “forest=TRUE”.
<code>model.name</code>	a name to give to the model in the PMML.
<code>app.name</code>	the name of the application that generated the PMML.
<code>description</code>	a descriptive text for the header of the PMML.
<code>copyright</code>	the copyright notice for the model.
<code>...</code>	further arguments passed to or from other methods.

Details

The Predictive Model Markup Language is an XML based language which provides a way for applications to define statistical and data mining models and to share models between PMML compliant applications. More information about PMML and the Data Mining Group can be found at <http://www.dmg.org>.

Use of PMML and `pmml.rsfl` requires the **XML** package. Be aware that XML is a very verbose data format. Reasonably sized trees and data sets can lead to extremely large text files. XML, while achieving interoperability, is not an efficient data storage mechanism in this case.

This function is used to export the geometry of the forest to other PMML compliant applications, including graphics packages that are capable of printing binary trees. In addition, the user may wish to save the geometry of the forest for later retrieval and prediction on new data sets using `pmml.rsfl` together with `pmml_to_rsfl`.

Value

An object of class `XMLNode` as that defined by the **XML** package. This represents the top level, or root node, of the XML document and is of type PMML. It can be written to file with `saveXML`.

Note

One cautionary note is in order. The PMML representation of the **randomSurvivalForest** forest object is incomplete, in that the object needs to be massaged in order for prediction to be possible. This will be clear in the examples. This deficiency will be addressed in future releases of this package. However, it was felt that the current functionality was important enough and mature enough to warrant release in this version of the product.

Author(s)

Hemant Ishwaran (hemant.ishwaran@gmail.com) and Udaya B. Kogalur (ubk2101@columbia.edu) with incorporation into the pmml package by (Graham.Williams@togaware.com)

References

H. Ishwaran and Udaya B. Kogalur (2006). Random Survival Forests. *Cleveland Clinic Technical Report*.

PMML home page: <http://www.dmg.org>

See Also

[pmml.](#)

Examples

```
library(randomSurvivalForest)
data(veteran, package = "randomSurvivalForest")
veteran.out <- rsf(Survrsf(time, status)~.,
  data = veteran,
  ntree = 5,
  forest = TRUE)
veteran.forest <- veteran.out$forest
pmml(veteran.forest)
```

pmml.rules

Generate PMML for arules objects

Description

Generate the PMML (Predictive Model Markup Language) representation of a rules or an itemset object from package **arules**. The PMML can then be imported into other systems that accept PMML.

Usage

```
## S3 method for class 'rules':
pmml(model, model.name="arules_Model",
  app.name="Rattle/PMML",
  description="arules association rules model", copyright=NULL, ...)
## S3 method for class 'itemsets':
pmml(model, model.name="arules_Model",
  app.name="Rattle/PMML",
  description="arules frequent itemsets model", copyright=NULL, ...)
```

Arguments

model	an rules or itemsets object.
model.name	a name to give to the model in the PMML.
app.name	the name of the application that generated the PMML.
description	a descriptive text for the header of the PMML.
copyright	the copyright notice for the model.
...	further arguments passed to or from other methods.

Details

The generated PMML can be imported into any PMML consuming application.

Author(s)

Michael Hahsler (michael@hahsler.net)

References

Package **arules** home page: <http://r-forge.r-project.org/projects/arules>

Package home page: <http://rattle.togaware.com>

PMML home page: <http://www.dmg.org>

See Also

[pmml.](#)

pmmlCanExport	<i>Can this installation export PMML variables (particularly transforms).</i>
---------------	---

Description

This function is designed to be overridden by other packages that implement pmml export, particularly of transforms.

Usage

```
pmmlCanExport(vname)
```

Arguments

vname a variable name to check whether it is exportable.

Author(s)

Graham.Williams@togaware.com

See Also

[pmml.](#)

`pmmltoc`*Generate C code from a PMML object - dummy function*

Description

This is a dummy function that does nothing. Plugins for Rattle are starting to appear which implement this for specific environments. This is experimental.

Usage

```
pmmltoc(p, name=NULL, includePMML=TRUE, includeMetaData=TRUE, exportClass=TRUE)
```

Arguments

<code>p</code>	<code>pmml</code> .
<code>name</code>	a name to give to the model in the C code.
<code>includePMML</code>	include the actual PMML as comments.
<code>includeMetaData</code>	include model information as comments.
<code>exportClass</code>	whether to export class or probability.

Author(s)

`<Graham.Williams@togaware.com>`

See Also

[pmml](#).

Index

*Topic **interface**

- pmml, 2
- pmml.coxph, 3
- pmml.hclust, 4
- pmml.kmeans, 5
- pmml.ksvm, 6
- pmml.lm, 7
- pmml.nnet, 8
- pmml.rpart, 9
- pmml.rsf, 12
- pmml.rules, 14
- pmmlCanExport, 15
- pmmltoc, 15

*Topic **survival**

- pmml.rsf, 12

*Topic **tree**

- pmml.rpart, 9
- pmml.rsf, 12

hclust, 4

kmeans, 5

ksvm, 7

pmml, 2, 4, 5, 7–9, 12–16

pmml.coxph, 3

pmml.hclust, 2, 4

pmml.itemsets (*pmml.rules*), 14

pmml.kmeans, 2, 5

pmml.ksvm, 2, 6

pmml.lm, 2, 7

pmml.nnet, 2, 8

pmml.rpart, 2, 9

pmml.rsf, 2, 12

pmml.rules, 2, 14

pmmlCanExport, 15

pmmltoc, 2, 15