

# Package ‘jForest’

June 3, 2015

**Encoding** UTF-8

**Type** Package

**Title** This is a wrapper to the jForest java framework

**Version** 1.0

**Date** 2015-06-02

**Depends** rJava (>= 0.9-6)

**Author** Jérôme Paul

**Maintainer** Jérôme Paul <jerome.paul@uclouvain.be>

**Description** Provides R functions to build custom tree ensembles with jForest java framework

**License** GPL-3

## R topics documented:

averageSplitImportance . . . . .	1
bootstrapSampler . . . . .	2
CARTSplitter . . . . .	2
CARTwithPriorSplitter . . . . .	3
extraTreesSplitter . . . . .	3
formatData . . . . .	4
fullSetSampler . . . . .	4
importance . . . . .	5
jForest . . . . .	6
predict.jForest . . . . .	7
rndFromSeed . . . . .	7
subsetSampler . . . . .	8
<b>Index</b>	<b>9</b>

---

averageSplitImportance

*Creates an aggregator of feature importance measures*

---

## Description

Creates an aggregator of feature importance measures

**Usage**

```
averageSplitImportance(p)
```

**Arguments**

p                      the number of features in the dataset

**Value**

an object of class `importance.internal.InternalImportanceIF`

---

<code>bootstrapSampler</code>	<i>Creates an instance sampler that creates bootstrap samples of the data</i>
-------------------------------	---

---

**Description**

Creates an instance sampler that creates bootstrap samples of the data

**Usage**

```
bootstrapSampler(seed)
```

**Arguments**

seed                      an integer

**Value**

an object of class `sampler.RandomSampler` initialized with seed

---

<code>CARTSplitter</code>	<i>Creates a splitter that follows the CART method</i>
---------------------------	--

---

**Description**

Creates a splitter that follows the CART method

**Usage**

```
CARTSplitter()
```

**Value**

an object of class `splitting.CARTSplitter`

---

**CARTwithPriorSplitter** *Creates a splitter that follows the RRF method. The Gini drop of features which are not favoured is multiplied by coef.*

---

### Description

Creates a splitter that follows the RRF method. The Gini drop of features which are not favoured is multiplied by coef.

### Usage

```
CARTwithPriorSplitter(coef, favoured.features)
```

### Arguments

**coef** a number between 0 and 1.  
**favoured.features** a logical vector which indicates for each feature if it is favoured or not. The features must be in the same order than in the x data.frame given to jForest.

### Value

an object of class `splitting.PriorKnowledgeCARTSplitter`

### References

Deng, Houtao, and George Runger. Feature selection via regularized trees. Neural Networks (IJCNN), The 2012 International Joint Conference on. IEEE, 2012.

---

**extraTreesSplitter** *Creates a splitter that follows the Extra-trees method i.e. choose one random split per feature and keep the best*

---

### Description

Creates a splitter that follows the Extra-trees method i.e. choose one random split per feature and keep the best

### Usage

```
extraTreesSplitter()
```

### Value

an object of class `splitting.ExtraTreesSplitter`

### References

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3-42.

---

formatData	<i>Transforms a dataframe with continuous and categorical attributes to a matrix. The levels of a categorical variable are mapped to numeric values</i>
------------	---

---

**Description**

Transforms a dataframe with continuous and categorical attributes to a matrix. The levels of a categorical variable are mapped to numeric values

**Usage**

```
formatData(d)
```

**Arguments**

d                      A data.frame

**Value**

a matrix of numerics

---

fullSetSampler	<i>Creates an instance sampler that always returns the full set of data</i>
----------------	---

---

**Description**

Creates an instance sampler that always returns the full set of data

**Usage**

```
fullSetSampler()
```

**Value**

an object of class sampler.RandomSampler

---

`importance`*Computes the importances of the variables in a jForest model*

---

## Description

Computes the importances of the variables in a jForest model

## Usage

```
importance(model, type = "internal", pval = TRUE)
```

## Arguments

<code>model</code>	a jForest model
<code>type</code>	takes values in <code>c("internal", "Ja", "Jp", "Jchisq", "Jks", "Jks.bcr")</code> . "internal" computes the importance specified by <code>inImportance</code> in <code>jForest</code> function. "Ja" computes Breiman's mean decrease in accuracy feature importance. "Jp" computes the mean difference in class prediction. "Jchisq" computes the significance of the features through a chi-squared test on the class vote distributions. "Jks" computes the significance of the features through a Kolmogorov-Smirnov test on the accuracies. "Jks.bcr" computes the significance of the features through a Kolmogorov-Smirnov test on the BCRs.
<code>pval</code>	a boolean indicating if "Jchisq", "Jks" or "Jks.bcr" computes p-values or the value of the chi-squared statistic.

## Value

a vector containing the importance of each variables.

## References

Jerome Paul, Pierre Dupont, Inferring statistically significant features from random forests, *Neurocomputing*, Volume 150, Part B, 20 February 2015, Pages 471-480, ISSN 0925-2312, <http://dx.doi.org/10.1016/j.neucom.2014.07.067>.

## Examples

```
m = jForest(iris[,1:4], iris$Species, ntree=1000, seed=42)
predict(m, iris[,1:4])
importance(m)
importance(m, "Ja")
importance(m, "Jchisq")
importance(m, "Jks")
```

jForest

*jForest' R interface***Description**

Wrapper to jForest java framework

Builds a jForest classification model. The default parameters corresponds to Breiman's Random Forest.

**Usage**

```
jForest(x, y, ntree = 500, mtry = sqrt(ncol(x)), seed = NULL,
        instanceSampler = bootstrapSampler(seed),
        featureSampler = subsetSampler(if (is.null(seed)) seed else seed + 1, mtry),
        splitCriterion = CARTSplitter(),
        inImportance = averageSplitImportance(ncol(x)), maxDepth = -1L)
```

**Arguments**

x	a n*p data.frame containing n samples in p dimensions
y	a vector of factors containing the n class labels
ntree	the number of trees to be grown
mtry	the number of candidate variables to be sampled in each node
seed	an integer to initialize the randomization
instanceSampler	an object of class sampler.RandomSampler
featureSampler	an object of class sampler.SubsetSampler
splitCriterion	an object of class splitting.CARTSplitter
inImportance	an object of class importance.internal.InternalImportanceIF
maxDepth	an integer defining the maximal depth of the trees. If it is set to a negative value, trees are fully grown.

**Value**

a jForest predictive model

**Author(s)**

Jérôme Paul, ICTEAM/Université catholique de Louvain

**Examples**

```
m = jForest(iris[,1:4], iris$Species, ntree=100, seed=42)
```

---

predict.jForest	<i>Classifies new data samples</i>
-----------------	------------------------------------

---

**Description**

Classifies new data samples

**Usage**

```
## S3 method for class 'jForest'  
predict(object, newdata, ...)
```

**Arguments**

object	a jForest object
newdata	a data.frame of new data
...	ignored

**Value**

the class labels corresponding to the samples in newdata

**Examples**

```
m = jForest(iris[,1:4],iris$Species,ntree=100,seed=42)  
predict(m,iris[,1:4])
```

---

rndFromSeed	<i>Creates a random number generator for Java</i>
-------------	---

---

**Description**

Creates a random number generator for Java

**Usage**

```
rndFromSeed(seed)
```

**Arguments**

seed	an integer or NULL
------	--------------------

**Value**

an object of class java.util.Random initialized with seed

---

subsetSampler	<i>Creates an instance sampler that creates sub-samples of the data</i>
---------------	---

---

**Description**

Creates an instance sampler that creates sub-samples of the data

**Usage**

```
subsetSampler(seed, mtry)
```

**Arguments**

seed	an integer
mtry	the number of instances to be sampled

**Value**

an object of class `sampler.SubsetSampler` initialized with seed



# Index

averageSplitImportance, [1](#)  
bootstrapSampler, [2](#)  
CARTSplitter, [2](#)  
CARTwithPriorSplitter, [3](#)  
extraTreesSplitter, [3](#)  
formatData, [4](#)  
fullSetSampler, [4](#)  
importance, [5](#)  
jForest, [6](#)  
jForest-package (jForest), [6](#)  
predict.jForest, [7](#)  
rndFromSeed, [7](#)  
subsetSampler, [8](#)