

# Package ‘FFTrees’

May 4, 2017

**Type** Package

**Title** Generate, Visualise, and Compare Fast and Frugal Decision Trees

**Version** 1.2.3

**Date** 2017-5-4

**Maintainer** Nathaniel Phillips <Nathaniel.D.Phillips.is@gmail.com>

**Description** Create, visualise, and test fast and frugal decision trees (FFTrees). FFTrees are very simple decision trees for classifying cases (i.e.; breast cancer patients) into one of two classes (e.g.; no cancer vs. true cancer) based on a small number of cues (e.g.; test results). FFTrees can be preferable to more complex algorithms because they are easy to communicate, require very little information, and are robust against overfitting.

**LazyData** TRUE

**Depends** R(>= 2.10)

**Imports** rpart, yarr, MASS, circlize, mlbench, graphics, randomForest, igraph, snow, snowfall, e1071

**License** CC0

**BugReports** <https://github.com/ndphillips/FFTrees/issues>

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Nathaniel Phillips [aut, cre]

**Repository** CRAN

**Date/Publication** 2017-05-04 16:23:34 UTC

## R topics documented:

apply.tree . . . . .	2
auc . . . . .	3

bank . . . . .	4
blood . . . . .	5
breastcancer . . . . .	5
car . . . . .	6
classtable . . . . .	7
comp.pred . . . . .	7
contraceptive . . . . .	8
creditapproval . . . . .	9
cuerank . . . . .	10
factclean . . . . .	11
fertility . . . . .	11
FFForest . . . . .	12
FFTrees . . . . .	13
FFTrees.guide . . . . .	15
forestfires . . . . .	15
grow.FFTrees . . . . .	16
heartdisease . . . . .	18
income . . . . .	19
iris.v . . . . .	20
mushrooms . . . . .	20
plot.FFForest . . . . .	21
plot.FFTrees . . . . .	22
predict.FFForest . . . . .	23
predict.FFTrees . . . . .	24
print.FFForest . . . . .	25
print.FFTrees . . . . .	25
showcues . . . . .	26
sonar . . . . .	26
summary.FFTrees . . . . .	27
titanic . . . . .	27
voting . . . . .	28
wine . . . . .	28
<b>Index</b>	<b>29</b>

---

apply.tree

*Applies a fast and frugal tree to a dataset.*

---

### Description

Applies a fast and frugal tree to a dataset.

### Usage

```
apply.tree(data, formula, tree.definitions, sens.w = 0.5)
```

**Arguments**

data	dataframe. A model training dataset. An m x n dataframe containing n cue values for each of the m exemplars.
formula	A formula
tree.definitions	dataframe. Definitions of one or more trees. The dataframe must contain the columns: cues, classes, thresholds, directions, exits.
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. Only used for calculating wacc values.

**Value**

A list of length 4 containing

**Examples**

```
tree.definitions <- data.frame("tree" = 1,
                              "cues" = "sex;age",
                              "thresholds" = "male;adult",
                              "directions" = ">=;="
                              )
```

---

auc	<i>Calculates AUC (Area under the Curve) using trapezoidal approximation</i>
-----	--

---

**Description**

Calculates AUC (Area under the Curve) using trapezoidal approximation

**Usage**

```
auc(sens.v, spec.v)
```

**Arguments**

sens.v	a vector of sensitivities
spec.v	A vector of specificities

**Examples**

```
# Calculate the AUC for a vector of hit rates and false alarm rates
auc(sens.v = c(.1, .3, .5, .7), spec.v = c(.05, .1, .15, .3))
```

---

bank	<i>A bank marketing dataset</i>
------	---------------------------------

---

**Description**

A bank marketing dataset

**Usage**

bank

**Format**

A data frame containing 45,211 rows and 17 columns

**age** numeric

**job** character

**marital** character - marital status

**education** character - education level

**default** character - has credit in default

**balance** numeric - balance of the loan

**housing** character - housing status

**loan** character - has a personal loan

**contact** character - contact communication type

**day** numeric - day in month last contacted

**month** character - last contact month

**duration** numeric - last contact duration in seconds

**campaign** numeric - number of contacts performed during this campaign for this client

**pdays** numeric - number of days that passed after the client was last contacted from a previous campaign (-1 means not previously contacted)

**previous** numeric - number of previous contacts

**poutcome** character - outcome of previous contact

**y** character - did the client subscribe to a term deposit? ...

**Source**

<http://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

---

blood	<i>Blood donation dataset</i>
-------	-------------------------------

---

**Description**

Blood donation dataset

**Usage**

blood

**Format**

A data frame containing 748 rows and 5 columns

**recency** Months since last donation

**frequency** Total number of donations

**total** Total blood donated in c.c.

**time** Months since first donation

**donation.crit** Did he/she donated blood in March 2007? ...

**Source**

<https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center>

---

breastcancer	<i>Dataset: Physiological dataset for 699 patients tested for breast cancer.</i>
--------------	--

---

**Description**

Dataset: Physiological dataset for 699 patients tested for breast cancer.

**Usage**

breastcancer

**Format**

A data frame containing 699 rows and 9 columns

**thickness** Clump Thickness  
**cellsize.unif** Uniformity of Cell Size  
**cellshape.unif** Uniformity of Cell Shape  
**adhesion** Marginal Adhesion  
**epithelial** Single Epithelial Cell Size  
**nuclei.bare** Bare Nuclei  
**chromatin** Bland Chromatin  
**nucleoli** Normal Nucleoli  
**mitoses** Mitoses  
**diagnosis** Is cancer present? T or F  
 ...

**Source**

[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))

---

car	<i>Car acceptability data</i>
-----	-------------------------------

---

**Description**

Car acceptability data

**Usage**

car

**Format**

A data frame containing 1728 rows and 7 columns

**buying.price** Numeric  
**maint.price** Factor  
**doors** Factor  
**persons** Numeric  
**luggage** Numeric  
**safety** Factor  
**acceptability** Factor  
 ...

**Source**

<http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

---

classtable	<i>Calculates several classification statistics from binary prediction and criterion (e.g.; truth) vectors</i>
------------	--

---

**Description**

Calculates several classification statistics from binary prediction and criterion (e.g.; truth) vectors

**Usage**

```
classtable(prediction.v, criterion.v, sens.w = 0.5)
```

**Arguments**

prediction.v	A binary vector of predictions
criterion.v	A binary vector of criterion (true) values
sens.w	numeric. Weight given to sensitivity, must range from 0 to 1.

**Examples**

```
# classification statistics for 5 cases
classtable(prediction.v = c(0, 0, 0, 1, 1),
           criterion.v = c(0, 0, 1, 0, 1))
```

---

comp.pred	<i>Wrapper for classification algorithms</i>
-----------	--

---

**Description**

This function is a wrapper for many classification algorithms such as CART (rpart::rpart), logistic regression (glm), support vector machines (svm::svm) and random forests (randomForest::randomForest)

**Usage**

```
comp.pred(formula, data.train, data.test = NULL, algorithm = NULL,
          new.factors = "exclude")
```

**Arguments**

formula	a formula
data.train	dataframe. A training dataset
data.test	dataframe. A testing dataset
algorithm	string. An algorithm in the set "lr" – logistic regression, "cart" – decision trees, "rlr" – regularised logistic regression, "svm" – support vector machines, "rf" – random forests
new.factors	string. What should be done if new factor values are discovered in the test set? "exclude" = exclude (i.e.; remove these cases), "base" = predict the base rate of the criterion.

**Examples**

```
# Fit many alternative algorithms to the mushrooms dataset

mushrooms.cart.pred <- comp.pred(formula = poisonous ~.,
                                data.train = mushrooms[1:100,],
                                data.test = mushrooms[101:nrow(mushrooms),],
                                algorithm = "cart")

mushrooms.rf.pred <- comp.pred(formula = poisonous ~.,
                               data.train = mushrooms[1:100,],
                               data.test = mushrooms[101:nrow(mushrooms),],
                               algorithm = "rf")

mushrooms.svm.pred <- comp.pred(formula = poisonous ~.,
                                data.train = mushrooms[1:100,],
                                data.test = mushrooms[101:nrow(mushrooms),],
                                algorithm = "svm")
```

---

 contraceptive

*Contraceptive use data*


---

**Description**

Contraceptive use data

**Usage**

contraceptive



**Format**

A data frame containing 1473 rows and 10 columns

**wife.age** Numeric

**wife.edu** Factor

**hus.ed** Factor

**children** Numeric

**wife.rel** Numeric

**wife.work** Factor

**hus.occ** Factor

**sol** Factor

**media** Numeric

**cont.crit** numeric

...

**Source**

<https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice>

---

creditapproval

*Credit approval data*

---

**Description**

Credit approval data

**Usage**

creditapproval

**Format**

A data frame containing 690 rows and 15 columns

**Source**

<https://archive.ics.uci.edu/ml/datasets/Credit+Approval>

---

cuerank	<i>Calculate the marginal accuracy of all cues in a dataframe. For each cue, the threshold that maximizes the criterion is selected.</i>
---------	--

---

### Description

Calculate the marginal accuracy of all cues in a dataframe. For each cue, the threshold that maximizes the criterion is selected.

### Usage

```
cuerank(formula = NULL, data = NULL, goal = "bacc", sens.w = 0.5,
        numthresh.method = "o", rounding = NULL, verbose = FALSE,
        cue.rules = NULL)
```

### Arguments

formula	formula. A formula specifying a binary criterion as a function of multiple variables
data	dataframe. A dataframe containing variables in formula
goal	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "bacc" = balanced accuracy, "wacc" = weighted accuracy, "d" = dprime
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity.
numthresh.method	character. A string indicating how to calculate cue splitting thresholds. "m" = median split, "o" = split that maximizes the goal,
rounding	integer. An integer indicating digit rounding for non-integer numeric cue thresholds. The default is NULL which means no rounding. A value of 0 rounds all possible thresholds to the nearest integer, 1 rounds to the nearest .1 (etc.).
verbose	logical. A logical value indicating whether or not to print ongoing diagnostics
cue.rules	dataframe. An optional df specifying how to make decisions for each cue. Must contain columns "cue", "class", "threshold" and "direction"

### Value

A dataframe containing best thresholds and marginal classification statistics for each cue

### Examples

```
## Not run:
# What are the best thresholds for each cue in the mushrooms dataset?
mushrooms.cues <- cuerank(formula = poisonous ~.,
                          data = mushrooms)
```

```
## End(Not run)
```

---

factclean	<i>Does miscellaneous cleaning of prediction datasets</i>
-----------	---

---

**Description**

Does miscellaneous cleaning of prediction datasets

**Usage**

```
factclean(data.train, data.test, show.warning = T)
```

**Arguments**

data.train	A training dataset
data.test	A testing dataset
show.warning	...

---

fertility	<i>Fertility data set</i>
-----------	---------------------------

---

**Description**

Fertility data set

**Usage**

```
fertility
```

**Format**

A data frame containing 100 rows and 10 columns

**Source**

<https://archive.ics.uci.edu/ml/datasets/Fertility>

---

 FFForest

*Creates a forest of fast and frugal decision trees*


---

### Description

This function is currently in development. The idea is to generate a random forest of fast and frugal trees from many splits of the training dataset.

### Usage

```
FFForest(formula = NULL, data = NULL, data.test = NULL, max.levels = 5,
         ntree = 10, train.p = 0.5, algorithm = "m", goal = "wacc",
         sens.w = 0.5, verbose = TRUE, cpus = 1, do.lr = TRUE,
         do.cart = TRUE, do.rf = TRUE, do.svm = TRUE, rank.method = NULL,
         hr.weight = NULL)
```

### Arguments

formula	formula. A formula specifying a binary criterion as a function of multiple variables
data	dataframe. A dataframe containing variables in formula
data.test	dataframe. An optional dataframe of test data
max.levels	integer. Maximum number of levels considered for the trees.
ntree	integer. Number of trees to create.
train.p	numeric. What percentage of the data should be used to fit each tree? Smaller values will result in more diverse trees.
algorithm	string. How to rank cues during tree construction. "m" (for marginal) means that cues will only be ranked once with the entire training dataset. "c" (conditional) means that cues will be ranked after each level in the tree with the remaining unclassified training exemplars. This also means that the same cue can be used multiple times in the trees. Note that the "c" method will take (much) longer and may be prone to overfitting.
goal	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "bacc" = balanced accuracy, "d" = d-prime
sens.w	numeric. How much weight to give to maximizing hits versus minimizing false alarms (between 0 and 1)
verbose	logical. Should progress reports be printed?
cpus	integer. Number of cpus to use. Any value larger than 1 will initiate parallel calculations in snowfall.
do.lr, do.cart, do.rf, do.svm	logical. Should logistic regression, cart, regularized logistic regression, random forests and/or support vector machines be calculated for comparison?
rank.method, hr.weight	deprecated arguments

**Value**

An object of class FFForest with the following elements...

**Examples**

```
## Not run:
cancer.fff <- FFForest(formula = diagnosis ~.,
                      data = breastcancer,
                      ntree = 10,
                      cpus = 1)

## End(Not run)
```

---

FFTrees

*Creates a Fast and Frugal Trees (FFTrees) object.*


---

**Description**

This is the workhorse function for the FFTrees package. It creates a set of fast and frugal decision trees trained on a training dataset and tested on an optional test dataset.

**Usage**

```
FFTrees(formula = NULL, data = NULL, data.test = NULL, train.p = 1,
        algorithm = "m", goal = "wacc", sens.w = 0.5, max.levels = 4,
        tree.definitions = NULL, verbose = FALSE, do.cart = TRUE,
        do.lr = TRUE, do.rf = TRUE, do.svm = TRUE, store.data = FALSE,
        object = NULL, rank.method = NULL, force = FALSE)
```

**Arguments**

formula	formula. A formula specifying a logical criterion as a function of 1 or more predictors.
data	dataframe. A training dataset.
data.test	dataframe. An optional testing dataset with the same structure as data.
train.p	numeric. What percentage of the data to use for training when data.test is not specified? For example, train.p = .5 will randomly split data into a 50% training set and a 50% test set. train.p = 1, the default, uses all data for training.
algorithm	character. How should cues be ranked during tree construction. "m" (for marginal) means that cues will only be ranked once with the entire training dataset. "c" (conditional) means that cues will be re-ranked after each level in the tree with the remaining unclassified training exemplars. This also means that the same cue can be used multiple times in the trees. Note that the "c" method can take (much) longer and may be prone to overfitting.

<code>goal</code>	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "wacc" = weighted accuracy
<code>sens.w</code>	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. Only relevant when <code>goal = 'wacc'</code>
<code>max.levels</code>	integer. The maximum number of levels considered for the trees. Because all permutations of exit structures are considered, the larger <code>max.levels</code> is, the more trees will be created.
<code>tree.definitions</code>	dataframe. An optional hard-coded definition of trees (see details below). If specified, no new trees are created.
<code>verbose</code>	logical. Should progress reports be printed? Can be helpful for diagnosis when the function is running slowly.
<code>do.cart, do.lr, do.rf, do.svm</code>	logical. Should alternative algorithms be created for comparison? <code>cart</code> = regression trees, <code>lr</code> = logistic regression, <code>rf</code> = random forests, <code>svm</code> = support vector machines.
<code>store.data</code>	logical. Should training / test data be stored in the object? Default is FALSE.
<code>object</code>	FFTrees. An optional existing FFTrees object. When specified, no new trees are fitted and the existing trees are applied to <code>data</code> and <code>data.test</code> .
<code>rank.method</code>	deprecated arguments.
<code>force</code>	logical. If TRUE, forces some parameters (like <code>goal</code> ) to be as specified by the user even when the algorithm thinks those specifications don't make sense.

## Details

`tree.definitions` should be a dataframe defining trees with each row. At least 4 columns should be present: `cues`, the names of the cues, `thresholds`, thresholds determining cue splits, `directions`, directions pointing towards positive classifications, `classes`, classes of the cues, and `exits`, the exit directions where 0 means a negative exit, 1 means a positive exit, and .5 means a bi-directional exit. Different levels within a tree should be separated by semicolons.

## Value

An FFTrees object with the following elements

**data, data.test** The original training and test data.

**cue.accuracy** Marginal accuracies of each cue given a threshold that maximizes balanced accuracy for the training data. These are calculated using the `cuerank()` function.

**tree.definitions** Definitions of each tree created by FFTrees. Each row corresponds to one tree. Different levels within a tree are separated by semi-colons. See above for more details.

**tree.stats** Tree definitions and classification statistics. Training and test data are stored separately

**level.stats** Cumulative classification statistics at each tree level. Training and test data are stored separately

**auc** Area under the curve statistics

**decision** Final classification decisions. Each row is a case and each column is a tree. For example, row 1 in column 2 is the classification decision of tree number 2 for the first case. Training and test data are stored separately.

**levelout** The level at which each case is classified in each tree. Rows correspond to cases and columns correspond to trees. Training and test data are stored separately.

**params** A list of control parameters (e.g.; `algorithm`, `goal`)

**comp** Models and classification statistics for competitive classification algorithms: Regularized logistic regression, CART, and random forest.

## Examples

```
# Create ffts for heart disease
heart.fft <- FFTrees(formula = diagnosis ~.,
                    data = heartdisease)

# Print the result for summary info
heart.fft

# Plot the best tree
plot(heart.fft)
```

---

FFTrees.guide

*Opens the FFTrees package guide*

---

## Description

Opens the FFTrees package guide

## Usage

```
FFTrees.guide()
```

---

forestfires

*forestfires*

---

## Description

A dataset of forest fire statistics.

## Usage

```
forestfires
```

**Format**

A data frame containing 517 rows and 13 columns

**X** Integer -x-axis spatial coordinate within the Montesinho park map: 1 to 9

**Y** Integer - y-axis spatial coordinate within the Montesinho park map: 2 to 9

**month** Factor - month of the year: "jan" to "dec"

**day** Factor -day of the week: "mon" to "sun"

**FFMC** Numeric -FFMC index from the FWI system: 18.7 to 96.20

**DMC** Numeric - DMC index from the FWI system: 1.1 to 291.3

**DC** Numeric - DC index from the FWI system: 7.9 to 860.6

**ISI** Numeric - ISI index from the FWI system: 0.0 to 56.10

**temp** Numeric - temperature in Celsius degrees: 2.2 to 33.30

**RH** Numeric - relative humidity in percent: 15.0 to 100

**wind** Numeric - wind speed in km/h: 0.40 to 9.40

**rain** Numeric - outside rain in mm/m2 : 0.0 to 6.4

**area** Numeric - the burned area of the forest (in ha): 0.00 to 1090.84

...

**Source**

<http://archive.ics.uci.edu/ml/datasets/Forest+Fires>

---

grow.FFTrees

*Grows fast and frugal trees using an algorithm specified by algorithm.*

---

**Description**

Grows fast and frugal trees using an algorithm specified by `algorithm`.

**Usage**

```
grow.FFTrees(formula, data, max.levels = 4, algorithm = "m",
  goal = "bacc", sens.w = 0.5, stopping.rule = "exemplars",
  stopping.par = 0.1, verbose = FALSE, rank.method = NULL, ...)
```



**Arguments**

formula	formula. A formula
data	dataframe. A dataset
max.levels	integer. The maximum number of levels in the tree(s)
algorithm	character. A string indicating how to rank cues during tree construction. "m" (for marginal) means that cues will only be ranked once with the entire training dataset. "c" (conditional) means that cues will be ranked after each level in the tree with the remaining unclassified training exemplars.
goal	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "bacc" = balanced accuracy, "wacc" = weighted accuracy
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity.
stopping.rule	character. A string indicating the method to stop growing trees. "levels" means the tree grows until a certain level. "exemplars" means the tree grows until a certain number of unclassified exemplars remain. "statdelta" means the tree grows until the change in the criterion statistic is less than a specified level.
stopping.par	numeric. A number indicating the parameter for the stopping rule. For stopping.rule == "levels", this is the number of levels. For stopping rule == "exemplars", this is the smallest percentage of exemplars allowed in the last level.
verbose	logical. Should tree growing progress be displayed?
rank.method	deprecated arguments
...	Currently ignored

**Value**

A list of length 4. `tree.definitions` contains definitions of the tree(s). `tree.stats` contains classification statistics for the tree(s). `levelout` shows which level in the tree(s) each exemplar is classified. Finally, `decision` shows the classification decision for each tree for each exemplar

**Examples**

```
titanic.trees <- grow.FFTrees(formula = survived ~.,
                             data = titanic)

# Tree definitions are stored in tree.definitions

titanic.trees$tree.definitions

# Tree classification statistics are in tree.stats

titanic.trees$tree.stats

# The level at which each exemplar is classified for each tree is in levelout

titanic.trees$levelout
```

```
# The decision for each exemplar for each tree is in decision
titanic.trees$decision
```

---

heartdisease	<i>Heart disease dataset</i>
--------------	------------------------------

---

### Description

Data for 303 patients tested for heart disease

### Usage

```
heartdisease
```

### Format

A data frame containing 303 rows and 14 columns

**age** Age

**sex** Sex, 1 = male, 0 = female

**cp** Chest pain type: ta = typical angina, aa = atypical angina, np = non-anginal pain, a = asymptomatic

**trestbps** Resting blood pressure (in mm Hg on admission to the hospital)

**chol** Serum cholesterol in mg/dl

**fbs** Fasting blood sugar > 120 mg/dl: 1 = true, 0 = false

**restecg** Resting electrocardiographic results. "normal" = normal, "abnormal" = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), "hypertrophy" = showing probable or definite left ventricular hypertrophy by Estes' criteria.

**thalach** Maximum heart rate achieved

**exang** Exercise induced angina: 1 = yes, 0 = no

**oldpeak** ST depression induced by exercise relative to rest

**slope** The slope of the peak exercise ST segment.

**ca** Number of major vessels (0-3) colored by fluoroscopy

**thal** "normal" = normal, "fd" = fixed defect, "rd" = reversible defect

**diagnosis** 1 = Heart disease, 0 = No Heart disease

...

### Source

<https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

---

income	<i>Income dataset</i>
--------	-----------------------

---

**Description**

Census Data used to predict who has an income greater than 50K

**Usage**

income

**Format**

A data frame containing 32561 rows and 15 columns

**age** Numeric

**workclass** Factor

**fnlwgt** Numeric

**education** Factor

**education.num** Numeric

**marital.status** Factor

**occupation** Factor

**relationship** Factor

**race** Factor

**sex** Factor

**capital.gain** Numeric

**capital.loss** Numeric

**hours.per.week** Numeric

**native.country** Factor

**income.gt50** Factor - 1 = >= 50K, 0 = < 50K

...

**Source**

<http://archive.ics.uci.edu/ml/datasets/Census+Income>

---

iris.v	<i>Iris data set</i>
--------	----------------------

---

**Description**

Iris data set

**Usage**

iris.v

**Format**

A data frame containing 150 rows and 4 columns

**Source**

<https://archive.ics.uci.edu/ml/datasets/Iris>

---

mushrooms	<i>Mushrooms dataset</i>
-----------	--------------------------

---

**Description**

Mushrooms dataset

**Usage**

mushrooms

**Format**

A data frame containing 8,124 rows and 23 columns (see <http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.names> for column descriptions)

**poisonous** numeric

**cshape** character

**csurface** character

**ccolor** character

**bruises** character

**odor** numeric

**gattach** character

**gspace** character

**gsize** character

**gcolor** character  
**sshape** character  
**sroot** character  
**ssaring** character  
**ssbring** character  
**scaring** character  
**scbring** character  
**vtype** character  
**vcolor** character  
**ringnum** character  
**ringtype** character  
**sporepc** character  
**population** character  
**habitat** character ...

### Source

<http://archive.ics.uci.edu/ml/datasets/Mushroom>

---

plot.FFForest	<i>Creates a network plot.</i>	<i>Code taken from Dirk Wulff</i> <i>(www.dirkwulff.org)</i>
---------------	--------------------------------	---

---

### Description

Creates a network plot. Code taken from Dirk Wulff ([www.dirkwulff.org](http://www.dirkwulff.org))

### Usage

```
## S3 method for class 'FFForest'
plot(x, node.cex.lim = c(1, 10), line.cex.lim = c(0.3,
  5), mincon = 0, lo = "kk", palette = NULL, ...)
```

### Arguments

x	FFForest. An FFForest object created from FFForest()
node.cex.lim	numeric. Nodesize adjustment
line.cex.lim	numeric. Edgesize adjustment
mincon	integer. Minimum connection cutoff
lo	string. The layout of the network plot. Either 'kk' (Kamada-Kawai, the default), 'dh' (Davidson-Harel) or 'fr' (Fruchterman-Reingold)
palette	string. A string vector of colors
...	currently ignored

---

plot.FFTrees                      *Draws a FFTrees object.*

---

### Description

The primary purpose of this function is to visualize a Fast and Frugal Tree (FFT) for data that has already been classified using the FFTrees() function. However, if the data have not yet been classified, the function can also implement a tree specified by the user. Inputs with the (M) header are mandatory. If the tree has already been implemented, then only inputs with the (A) header should be entered. If the tree has not been implemented, then only inputs with the (B) header should be entered.

### Usage

```
## S3 method for class 'FFTrees'
plot(x = NULL, data = "train", what = "tree",
     tree = "best.train", main = "Data", decision.names = c("Noise",
     "Signal"), cue.cex = NULL, threshold.cex = NULL, comp = TRUE,
     stats = TRUE, n.per.icon = NULL, which.tree = NULL, ...)
```

### Arguments

x	A FFTrees object created from "FFTrees()"
data	Either a dataframe of new data, or one of two strings 'train' or 'test'. In this case, the corresponding dataset in the x object will be used.
what	string. What should be plotted? 'tree' (the default) shows one tree (specified by 'tree'). 'cues' shows the marginal accuracy of cues in an ROC space.
tree	integer. An integer indicating which tree to plot (only valid when the tree argument is non-empty). To plot the best training (or test) tree with respect to v (sens - spec), use "best.train" or "best.test"
main	character. The main plot label.
decision.names	character. A string vector of length 2 indicating the content-specific name for noise and signal cases.
cue.cex	numeric. A numeric vector specifying the size of the cue labels.
threshold.cex	numeric. A numeric vector specifying the size of the decision thresholds.
comp	logical. Should the performance of competitive algorithms (e.g.; logistic regression, random forests etc.) be shown in the ROC plot (if available?)
stats	logical. Should statistical information be plotted? If FALSE, then only the tree (without any reference to statistics) will be plotted.
n.per.icon	Number of exemplars per icon
which.tree	deprecated argument, only for backwards compatibility, use "tree" instead.
...	Currently ignored.

## Examples

```
# Create FFTrees of the heart disease data
heart.fft <- FFTrees(formula = diagnosis ~.,
  data = heartdisease)

# Visualise the tree
plot(heart.fft,
  main = "Heart Disease Diagnosis",
  decision.names = c("Absent", "Present"))

# See the vignette for more details
vignette("FFTrees_plot", package = "FFTrees")
```

---

predict.FFForest	<i>Predict outcoms from a test dataset using an FFForest object</i>
------------------	---

---

## Description

Predict outcoms from a test dataset using an FFForest object

## Usage

```
## S3 method for class 'FFForest'
predict(object = NULL, data = NULL, threshold = 0.5,
  ...)
```

## Arguments

object	FFForest. An FFForest object created from the FFForest() function.
data	dataframe. A dataframe of test data
threshold	numeric. A threshold value
...	Additional arguments passed on to predict()

## Value

A logical vector of predictions

---

predict.FFTrees      *Predict new data from an FFTrees x*

---

### Description

Predict new data from an FFTrees x

### Usage

```
## S3 method for class 'FFTrees'
predict(object = NULL, data = NULL, tree = "best.train",
        sens.w = NULL, ...)
```

### Arguments

object	An FFTrees object created from the FFTrees() function.
data	dataframe. A dataframe of test data
tree	Which tree in the FFTrees x should be used? Can be an integer or "best.train" (the default) to use the tree with the best training statistics (according the goal specified in tree construction).
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. If specified, the tree with the highest weighted accuracy (wacc) given the specified value of sens.w will be selected
...	Additional arguments passed on to predict()

### Value

A logical vector of predictions

### Examples

```
# Create training and test data

set.seed(100)
breastcancer <- breastcancer[sample(nrow(breastcancer)),]
breast.train <- breastcancer[1:150,]
breast.test <- breastcancer[151:303,]

# Create an FFTrees x from the training data

breast.fft <- FFTrees(formula = diagnosis ~.,
                     data = breast.train)

# Predict results for test data
breast.fft.pred <- predict(breast.fft,
                          data = breast.test)
```



---

print.FFForest	<i>Prints summary information from an FFForest x</i>
----------------	--

---

**Description**

Printing function for an FFTrees x

**Usage**

```
## S3 method for class 'FFForest'  
print(x = NULL, ...)
```

**Arguments**

x	FFForest. An FFForest x created from FFForest()
...	additional arguments passed to print.

---

print.FFTrees	<i>Prints summary information from an FFTrees x</i>
---------------	---

---

**Description**

Printing function for an FFTrees x

**Usage**

```
## S3 method for class 'FFTrees'  
print(x = NULL, ...)
```

**Arguments**

x	FFTrees. A FFTrees x created from FFTrees()
...	additional arguments passed to print.

---

showcues	<i>Visualizes cue accuracies from an FFTrees object in a ROC space</i>
----------	--

---

**Description**

Visualizes cue accuracies from an FFTrees object in a ROC space

**Usage**

```
showcues(x = NULL, data = "train", cue.accuracy = NULL, main = NULL,
         top = 5)
```

**Arguments**

x	An FFTrees object
data	A string indicating whether or not to show training ("train") or testing ("test") cue accuracies
cue.accuracy	dataframe. An optional dataframe specifying cue accuracies directly (without specifying an FFTrees object x)
main	Main plot description
top	An integer indicating how many of the top cues to highlight

---

sonar	<i>Sonar data set</i>
-------	-----------------------

---

**Description**

Sonar data set

**Usage**

```
sonar
```

**Format**

A data frame containing 208 rows and 60 columns

**Source**

[https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+\(Sonar,+Mines+vs.+Rocks\)](https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks))

---

summary.FFTrees	<i>Returns a summary of an fft object</i>
-----------------	---

---

**Description**

Returns a summary of an fft object

**Usage**

```
## S3 method for class 'FFTrees'
summary(object, ...)
```

**Arguments**

object	An FFTrees object
...	Additional arguments (currently ignored)

---

titanic	<i>Titanic dataset</i>
---------	------------------------

---

**Description**

A dataset indicating who survived on the Titanic

**Usage**

```
titanic
```

**Format**

A data frame containing 2,201 rows and 4 columns

**class** Factor - Class (first, second, third, or crew)

**age** Factor - Age group (child or adult)

**sex** Factor - Sex (male or female)

**survived** Factor - Whether the passenger survived (1) or not (0)

...

**Source**

<https://www.amstat.org/publications/jse/datasets/titanic.dat.txt>

---

voting	<i>Voting data set</i>
--------	------------------------

---

**Description**

Voting data set

**Usage**

voting

**Format**

A data frame containing 435 rows and 16 columns

**Source**

<https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records>

---

wine	<i>Wine tasting dataset</i>
------	-----------------------------

---

**Description**

Chemical and tasting data from wines in North Portugal

**Usage**

wine

**Format**

A data frame containing 6497 rows and 13 columns

**Source**

<http://archive.ics.uci.edu/ml/datasets/Wine+Quality>

# Index

## \*Topic **datasets**

- bank, 4
- blood, 5
- breastcancer, 5
- car, 6
- contraceptive, 8
- creditapproval, 9
- fertility, 11
- forestfires, 15
- heartdisease, 18
- income, 19
- iris.v, 20
- mushrooms, 20
- sonar, 26
- titanic, 27
- voting, 28
- wine, 28

apply.tree, 2

auc, 3

bank, 4

blood, 5

breastcancer, 5

car, 6

classtable, 7

comp.pred, 7

contraceptive, 8

creditapproval, 9

cuerank, 10

factclean, 11

fertility, 11

FFForest, 12

FFTrees, 13

FFTrees.guide, 15

forestfires, 15

grow.FFTrees, 16

heartdisease, 18

income, 19

iris.v, 20

mushrooms, 20

plot.FFForest, 21

plot.FFTrees, 22

predict.FFForest, 23

predict.FFTrees, 24

print.FFForest, 25

print.FFTrees, 25

showcues, 26

sonar, 26

summary.FFTrees, 27

titanic, 27

voting, 28

wine, 28