Package ‘FFTrees’

June 8, 2020

**Type** Package

**Title** Generate, Visualise, and Evaluate Fast-and-Frugal Decision Trees

**Version** 1.5.5

**Date** 2020-06-07

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

**Description** Create, visualize, and test fast-and-frugal decision trees (FFTs). FFTs are very simple decision trees for binary classification problems. FFTs 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(>= 3.5.0)

**Imports** rpart, circlize, parallel, graphics, randomForest, igraph, e1071, crayon, scales, tibble, stringr, progress, pROC, caret, dplyr, testthat, magrittr, tidyselect

**License** CC0

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

**RoxygenNote** 7.1.0

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Nathaniel Phillips [aut, cre], Hansjoerg Neth [aut], Jan Woike [aut], Wolfgang Gaismaer [aut]

**Repository** CRAN

**Date/Publication** 2020-06-08 15:00:24 UTC
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add_Stats</td>
</tr>
<tr>
<td>blood</td>
</tr>
<tr>
<td>breastcancer</td>
</tr>
<tr>
<td>car</td>
</tr>
<tr>
<td>classtable</td>
</tr>
<tr>
<td>comp.pred</td>
</tr>
<tr>
<td>contraceptive</td>
</tr>
<tr>
<td>creditapproval</td>
</tr>
<tr>
<td>factclean</td>
</tr>
<tr>
<td>fertility</td>
</tr>
<tr>
<td>FFTrees</td>
</tr>
<tr>
<td>FFTrees.guide</td>
</tr>
<tr>
<td>fftrees_apply</td>
</tr>
<tr>
<td>fftrees_create</td>
</tr>
<tr>
<td>fftrees_cuerank</td>
</tr>
<tr>
<td>fftrees_define</td>
</tr>
<tr>
<td>fftrees_ffttowords</td>
</tr>
<tr>
<td>fftrees_fitcomp</td>
</tr>
<tr>
<td>fftrees_grow_fan</td>
</tr>
<tr>
<td>fftrees_ranktrees</td>
</tr>
<tr>
<td>fftrees_threshold_factor_grid</td>
</tr>
<tr>
<td>fftrees_threshold_numeric_grid</td>
</tr>
<tr>
<td>fftrees_wordstofftrees</td>
</tr>
<tr>
<td>forestfires</td>
</tr>
<tr>
<td>heart.cost</td>
</tr>
<tr>
<td>heart.test</td>
</tr>
<tr>
<td>heart.train</td>
</tr>
<tr>
<td>heartdisease</td>
</tr>
<tr>
<td>inwords</td>
</tr>
<tr>
<td>iris.v</td>
</tr>
<tr>
<td>mushrooms</td>
</tr>
<tr>
<td>plot.FFTrees</td>
</tr>
<tr>
<td>predict.FFTrees</td>
</tr>
<tr>
<td>print.FFTrees</td>
</tr>
<tr>
<td>showcues</td>
</tr>
<tr>
<td>sonar</td>
</tr>
<tr>
<td>summary.FFTrees</td>
</tr>
<tr>
<td>titanic</td>
</tr>
<tr>
<td>voting</td>
</tr>
<tr>
<td>wine</td>
</tr>
</tbody>
</table>
Add_Stats

Adddecisionstatistics toadataframe containing hr, cr, mi and fa

Description

Addssionstatistics toadataframe containing hr, cr, mi and fa

Usage

Add_Stats(
   data,
   sens.w = 0.5,
   cost.each = NULL,
   cost.outcomes = list(hi = 0, fa = 1, mi = 1, cr = 0)
)

Arguments

data dataframe. With named (integer) columns hi, cr, mi, fa
sens.w numeric. Sensitivity weight
cost.each numeric. An optional fixed cost added to all outputs (e.g., the cost of the cue)
cost.outcomes list. A list of length 4 with names 'hi', 'fa', 'mi', and 'cr' specifying the costs of a hit, false alarm, miss, and correct rejection respectively. E.g.; cost.outcomes = listc("hi" = 0,"fa" = 10,"mi" = 20,"cr" = 0) means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.

blood

Blood donation dataset

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
totaltotal blood donated in c.c.
time Months since first donation
donation.crit Did he/she donated blood in March 2007? ...
breastcancer

Source

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

breastcancer

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

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

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


classtable

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

Description

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

Usage

classtable(
  prediction_v = NULL,
  criterion_v,
  sens.w = 0.5,
  cost.v = NULL,
  correction = 0.25,
  cost.outcomes = list(hi = 0, fa = 1, mi = 1, cr = 0),
  na_prediction_action = "ignore"
)
Arguments

- **prediction_v**: logical. A logical vector of predictions
- **criterion_v**: logical. A logical vector of criterion (true) values
- **sens.w**: numeric. Weight given to sensitivity, must range from 0 to 1.
- **cost.v**: list. An optional list of additional costs to be added to each case.
- **correction**: numeric. Correction added to all counts for calculating dprime
- **cost.outcomes**: list. A list of length 4 with names 'hi', 'fa', 'mi', and 'cr' specifying the costs of a hit, false alarm, miss, and correct rejection respectively. E.g.; `cost.outcomes = listc("hi" = 0,"fa" = 10,"mi" = 20,"cr" = 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
- **na_prediction_action**: not sure.

---

`comp.pred`  
*Wrapper for classification algorithms*

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

```r
comp.pred(
  formula,
  data.train,
  data.test = NULL,
  algorithm = NULL,
  model = 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
- **model**: model. An optional existing model applied to test data
- **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.
### Contraceptive

**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

### Credit approval

**Description**
Credit approval data

**Usage**
creditapproval

**Source**
https://archive.ics.uci.edu/ml/datasets/creditapproval
**Format**
A data frame containing 690 rows and 15 columns

**Source**
https://archive.ics.uci.edu/ml/datasets/Credit+Approval

---

**factclean**
*Does miscellaneous cleaning of prediction datasets*

**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**
*Fertility data set*

**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
FFTrees

Creates a fast-and-frugal trees (FFTrees) object.

Description

This is the workhorse function for the FFFtrees package. It creates (one or more) fast-and-frugal decision trees trained on a training dataset and tested on an optional test dataset.

Create fast and frugal trees

Usage

FFTrees(
  formula = NULL,
  data = NULL,
  data.test = NULL,
  algorithm = "ifan",
  max.levels = NULL,
  sens.w = 0.5,
  cost.outcomes = NULL,
  cost.cues = NULL,
  stopping.rule = "exemplars",
  stopping.par = 0.1,
  goal = NULL,
  goal.chase = NULL,
  goal.threshold = "bacc",
  numthresh.method = "o",
  numthresh.n = 10,
  decision.labels = c("False", "True"),
  main = NULL,
  train.p = 1,
  rounding = NULL,
  repeat.cues = TRUE,
  my.tree = NULL,
  tree.definitions = NULL,
  do.comp = TRUE,
  do.cart = TRUE,
  do.lr = TRUE,
  do.rf = TRUE,
  do.svm = TRUE,
  store.data = FALSE,
  object = NULL,
  rank.method = NULL,
  force = FALSE,
  verbose = NULL,
  comp = NULL,
  quiet = 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.
- **algorithm**: character. The algorithm to create FFTs. Can be 'ifan', 'dfan', 'max', or 'zigzag'.
- **max.levels**: integer. The maximum number of levels considered for the trees. Because all permutations of exit structures are considered, the larger max.levels is, the more trees will be created.
- **sens.w**: numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. Only relevant when goal = 'wacc'.
- **cost.outcomes**: list. A list of length 4 with names 'hi', 'fa', 'mi', and 'cr' specifying the costs of a hit, false alarm, miss, and correct rejection respectively. E.g.; cost.outcomes = listc("hi" = 0,"fa" = 10,"mi" = 20,"cr" = 0) means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
- **cost.cues**: list. A list containing containing costs for each cue. Each element should have a name corresponding to a column in data, and each entry should be a single (positive) number. Cues not present in cost.cues are assume to have 0 cost.
- **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.
- **goal**: character. A string indicating the statistic to maximize when selecting final trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy
- **goal.chase**: character. A string indicating the statistic to maximize when constructing trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy, "cost" = cost.
- **goal.threshold**: character. A string indicating the statistic to maximize when calculating cue thresholds: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy
- **numthresh.method**: character. How should thresholds for numeric cues be determined? "o" will optimize thresholds, while "m" will always use the median.
- **numthresh.n**: integer. Number of numeric thresholds to try.
- **decision.labels**: string. A vector of strings of length 2 indicating labels for negative and positive cases. E.g.; decision.labels = c("Healthy","Diseased")
- **main**: string. An optional label for the dataset. Passed on to other functions like plot.FFTrees(), and print.FFTrees()
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.

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.).

repeat.cues logical. Can cues occur multiple times within a tree?

my.tree string. A string representing an FFT in words. For example, my.tree = “If age > 20, predict TRUE. If sex = {m}, predict FALSE. Otherwise, predict TRUE”

tree.definitions dataframe. An optional hard-coded definition of trees (see details below). If specified, no new trees are created.

do.comp, do.cart, do.lr, do.rf, do.svm logical. Should alternative algorithms be created for comparison? cart = regular (non-frugal) trees with rpart, lr = logistic regression with glm, rf = random forests with randomForest, svm = support vector machines with e1071. Setting comp = FALSE sets all these arguments to FALSE.

store.data logical. Should training / test data be stored in the object? Default is FALSE.

object FFTrees. An optional existing FFTrees object. When specified, no new trees are fitted and the existing trees are applied to data and data.test.

rank.method, verbose, comp deprecated arguments.

force logical. If TRUE, forces some parameters (like goal) to be as specified by the user even when the algorithm thinks those specifications don’t make sense.

quiet logical. Should progress reports be printed? Can be helpful for diagnosis when the function is running slowly.

Value

An FFTrees object with the following elements

formula The formula specified when creating the FFTs.
data.desc Descriptive statistics of the data
cue.accuracies Marginal accuracies of each cue given a decision threshold calculated with the specified algorithm
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
cost A list of cost information for each case in each tree.
level.stats Cumulative classification statistics at each tree level. Training and test data are stored separately
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.

tree.max The index of the 'final' tree specified by the algorithm. For algorithms that only return a single tree, this value is always 1.

inwords A verbal definition of tree.max.

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

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

data The original training and test data (only included when store.data = TRUE)

Examples

# Create fast-and-frugal trees for heart disease
heart.fft <- FFTrees(formula = diagnosis ~ .,
data = heart.train,
data.test = heart.test,
main = "Heart Disease",
decision.labels = c("Healthy", "Diseased"))

# Print the result for summary info
heart.fft

# Plot the tree applied to training data
plot(heart.fft, stats = FALSE)
plot(heart.fft)
plot(heart.fft, data = "test")  # Now for testing data
plot(heart.fft, data = "test", tree = 2)  # Look at tree number 2

### Predict classes and probabilities for new data
predict(heart.fft, newdata = heartdisease)
predict(heart.fft, newdata = heartdisease, type = "prob")

### Create your own custom tree with my.tree

custom.fft <- FFTrees(formula = diagnosis ~ .,
data = heartdisease,
my.tree = 'If chol > 300, predict True.
If sex = {m}, predict False,
If age > 70, predict True, otherwise predict False'
)

# Plot the custom tree (it's pretty terrible)
plot(custom.fft)
**FFTrees.guide**

Opens the FFTrees package guide

**Description**

Opens the FFTrees package guide

**Usage**

`FFTrees.guide()`

---

**fftrees_apply**

Applies a fast-and-frugal tree to a dataset and generates several accuracy statistics

**Description**

Applies a fast-and-frugal tree to a dataset and generates several accuracy statistics

**Usage**

`fftrees_apply(x, mydata = NULL, newdata = NULL, allNA.pred = FALSE)`

**Arguments**

- `x` : FFTrees.
- `mydata` : dataframe.
- `newdata` : dataframe.
- `allNA.pred` : logical. What should be predicted if all cue values in tree are NA? Default is FALSE

**Value**

A list of length 4 containing
fftrees_create

Create an FFTrees object

Description

Create an FFTrees object

Usage

```r
fftrees_create(
  data = NULL,
  formula = NULL,
  algorithm = NULL,
  goal = NULL,
  goal.chase = NULL,
  goal.threshold = NULL,
  numthresh.method = NULL,
  numthresh.n = NULL,
  sens.w = NULL,
  max.levels = NULL,
  cost.outcomes = NULL,
  cost.cues = NULL,
  stopping.rule = NULL,
  stopping.par = NULL,
  decision.labels = NULL,
  main = NULL,
  my.tree = NULL,
  data.test = NULL,
  repeat.cues = NULL,
  quiet = NULL,
  do.lr = TRUE,
  do.svm = TRUE,
  do.cart = TRUE,
  do.rf = TRUE,
  do.comp = TRUE
)
```

Arguments

- `data`: dataframe. Training data
- `formula`: formula. A formula
- `algorithm`: string.
- `goal`: string.
- `goal.chase`: string.
- `goal.threshold`: string.
fftrees_cuerank

numthresh.method
numthresh.n
sens.w
max.levels
cost.outcomes
cost.cues
stopping.rule
stopping.par
decision.labels
main
my.tree
data.test
repeat.cues
quiet
do.lr
do.svm
do.cart
do.rf
do.comp

Value
An FFTrees object.

fftrees_cuerank Calculates thresholds that maximize a statistic (goal) for cues.

Description
Calculates thresholds that maximize a statistic (goal) for cues.

Usage
fftrees_cuerank(x = NULL, newdata = NULL, data = "train", rounding = NULL)

Arguments
x FFTrees.
newdata dataframe.
data dataframe.
rounding integer.
fftrees_ffttowords

Value

A dataframe containing thresholds and marginal classification statistics for each cue

fftrees_define

Create definitions of FFTrees

Description

Create definitions of FFTrees

Usage

fftrees_define(x, object = NULL)

Arguments

x

FFTrees.

object

FFTrees.

fftrees_ffttowords

Describes an FFT in words

Description

Describes an FFT in words

Usage

fftrees_ffttowords(x = NULL, digits = 2)

Arguments

x

FFTrees. An FFTrees object created with FFTrees()

digits

integer. How many digits to round numeric values

Value

A list of string vectors

Examples

heart.fft <- FFTrees(diagnosis ~.,
                      data = heartdisease,
                      decision.labels = c("Healthy", "Disease"))

inwords(heart.fft)
**fftrees_fitcomp**

Fit competitive algorithms

**Description**
Fit competitive algorithms

**Usage**
`fftrees_fitcomp(x)`

**Arguments**
- `x` FF Trees.

---

**fftrees_grow_fan**
Grows fast-and-frugal trees using the fan algorithm

**Description**
Grows fast-and-frugal trees using the fan algorithm

**Usage**
`fftrees_grow_fan(x, repeat.cues = TRUE)`

**Arguments**
- `x` FF Trees. An FFTrees object
- `repeat.cues` logical.

---

**fftrees_ranktrees**
Rank trees by goal

**Description**
Rank trees by goal

**Usage**
`fftrees_ranktrees(x, data = "train")`

**Arguments**
- `x` FF Trees.
- `data` character.
fftrees_threshold_factor_grid

Performs a grid search over factor and returns accuracy statistics for a given factor cue

Description

Performs a grid search over factor and returns accuracy statistics for a given factor cue

Usage

```r
fftrees_threshold_factor_grid(
  thresholds = NULL,
  cue_v = NULL,
  criterion_v = NULL,
  directions = "",
  sens.w = 0.5,
  cost.outcomes = list(hi = 0, fa = 1, mi = 1, cr = 0),
  cost.each = 0,
  goal.threshold = "bacc"
)
```

Arguments

- `thresholds` numeric. A vector of factor thresholds to consider
- `cue_v` numeric. Feature values
- `criterion_v` logical. Criterion values
- `directions` character. Character vector of threshold directions to consider.
- `sens.w` numeric.
- `cost.outcomes` list. A list of length 4 with names 'hi', 'fa', 'mi', and 'cr' specifying the costs of a hit, false alarm, miss, and correct rejection respectively. E.g.: `cost.outcomes = listc("hi" = 0,"fa" = 10,"mi" = 20,"cr" = 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
- `cost.each` numeric.
- `goal.threshold` character.
**fftrees_threshold_numeric_grid**

*Performs a grid search over thresholds and returns accuracy statistics for a given numeric cue*

**Description**

Performs a grid search over thresholds and returns accuracy statistics for a given numeric cue

**Usage**

```r
fftrees_threshold_numeric_grid(
  thresholds,
  cue_v,
  criterion_v,
  directions = c(">", "<="),
  sens.w = 0.5,
  cost.each = 0,
  cost.outcomes = list(hi = 0, fa = 1, mi = 1, cr = 0),
  goal.threshold = "bacc"
)
```

**Arguments**

- `thresholds` numeric. A vector of thresholds to consider
- `cue_v` numeric. Feature values
- `criterion_v` logical. Criterion values
- `directions` character. Possible directions to consider
- `sens.w` numeric. Cost to add to each value (e.g.; cost of the cue)
- `cost.each` numeric. Cost to add to each value (e.g.; cost of the cue)
- `cost.outcomes` list. A list of length 4 with names 'hi', 'fa', 'mi', and 'cr' specifying the costs of a hit, false alarm, miss, and correct rejection respectively. E.g.; `cost.outcomes = listc("hi" = 0,"fa" = 10,"mi" = 20,"cr" = 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
- `goal.threshold` character. A string indicating the statistic to maximize when calculating cue thresholds: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy
fftrees_wordstofftrees

*Converts text describing an FFT into an FFT definition.*

**Description**

Converts text describing an FFT into an FFT definition.

**Usage**

```r
fftrees_wordstofftrees(x, my.tree)
```

**Arguments**

- `x` FFTrees.
- `my.tree` string. A string defining an FFT

---

**forestfires**

**Description**

A dataset of forest fire statistics.

**Usage**

```r
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

...
heart.cost

Source

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

heart.cost  Cue costs for the heartdisease dataa

Description

Cue costs for the heartdisease dataa

Usage

heart.cost

Format

A data frame containing 153 rows and 14 columns

cue  The name of the cue

cost  The cost of the cue

...  

Source


heart.test  Heartdisease testing dataset

Description

Testing data for a heart disease dataset. These data are used to test the prediction performance of a model trained on the heart.train data. The dataset heartdisease contains both datasets.

Usage

heart.test
Format

A data frame containing 153 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 flourosopy
- **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

heart.train  

*Heartdisease training dataset.*

Description

Training data for a heart disease dataset used to train a model. The corresponding dataset for model testing is heart.test. The dataset heartdisease contains both datasets.

Usage

heart.train
heartdisease

Format

A data frame containing 150 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 cholestoral 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 flourosopy
thal  "normal" = normal, "fd" = fixed defect, "rd" = reversable defect
diagnosis  1 = Heart disease, 0 = No Heart disease

Source

https://archive.ics.uci.edu/ml/datasets/Heart+Disease
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 flourosopy
thal  "normal" = normal, "fd" = fixed defect, "rd" = reversable defect
diagnosis 1 = Heart disease, 0 = No Heart disease
...

Source

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

inwords(x, tree = 1)

Display a verbal description of a tree in an FFTrees object

Description

Display a verbal description of a tree in an FFTrees object

Usage

inwords(x, tree = 1)

Arguments

x  FFTrees.
tree  numeric. The tree to display
**Iris data set**

**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 dataset**

**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
plot.FFTrees

Plots an FFTrees object.

Source

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

Description

Plots an FFTrees object created by the FFTrees() function.

Usage

## S3 method for class 'FFTrees'
plot(
  x = NULL,
  data = "train",
  what = "tree",
  tree = 1,
  main = NULL,
  hlines = TRUE,
  cue.labels = NULL,
  decision.labels = NULL,
  cue.cex = NULL,
  threshold.cex = NULL,
  decision.cex = 1,
  comp = TRUE,
  stats = TRUE,
)
Arguments

। **x**  A FFTrees object created from "FFTrees()"

## data
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, "roc" shows an roc curve of the tree(s)

## 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 the goal specified during FFT construction, use "best.train" or "best.test"

## main
character. The main plot label.

## hlines
logical. Should horizontal panel separation lines be shown?

cue.labels
character. An optional string of labels for the cues / nodes.

decision.labels
character. A string vector of length 2 indicating the content-specific name for noise and signal cases.

cue.cex
numeric. The size of the cue labels.

## threshold.cex
numeric. The size of the threshold labels.

## decision.cex
numeric. The size of the decision labels.

## 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.

## show.header, show.tree, show.confusion, show.levels, show.roc, show.icons, show.iconguide
logical. Logical arguments indicating which specific elements of the plot to show.

## label.tree, label.performance
string. Optional arguments to define lables for the tree and performance section(s).
predict.FFTrees

Predict classifications from newdata using an FFTrees object

Description

Predict classifications from newdata using an FFTrees object

Usage

## S3 method for class 'FFTrees'
predict(
  object = NULL,
  newdata = NULL,
  tree = 1,
  type = "class",
  sens.w = NULL,
  method = "laplace",
  data = NULL,
  ...
)

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.labels = c("Absent", "Present"))

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

- **object**: An FFTrees object created from the FFTrees() function.
- **newdata**: dataframe. A dataframe of test data
- **tree**: integer. Which tree in the object should be used? By default, tree = 1 is used
- **type**: string. What should be predicted? Can be "class", which returns a vector of class predictions, "prob" which returns a matrix of class probabilities, or "both" which returns a matrix with both class and probability predictions.
- **sens.w, data**: deprecated
- **method**: string. Method of calculating class probabilities. Either 'laplace', which applies the Laplace correction, or 'raw' which applies no correction.
- **...**: Additional arguments passed on to predict()

Value

Either a logical vector of predictions, or a matrix of class probabilities.

Examples

```r
# 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 classes of test data
breast.fft.pred <- predict(breast.fft,
                           newdata = breast.test)

# Predict class probabilities
breast.fft.pred <- predict(breast.fft,
                           newdata = breast.test,
                           type = "prob")
```
print.FFTrees

*Prints summary information from an FFTrees object*

**Description**

Printing function for an FFTrees object

**Usage**

```r
## S3 method for class 'FFTrees'
print(x = NULL, tree = 1, ...)
```

**Arguments**

- `x`: FFTrees. A FFTrees x created from FFTrees()
- `tree`: integer. The tree to explore.
- `...`: additional arguments passed to print.

---

showcues

*Visualizes cue accuracies from an FFTrees object in a ROC space*

**Description**

Visualizes cue accuracies from an FFTrees object in a ROC space

**Usage**

```r
showcues(x = NULL, data = "train", cue.accuracies = 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`: 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

---

**sonar**  
*Sonar data set*

**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)

---

**summary.FFTrees**  
*Returns summary information about an FFTrees x*

**Description**

Returns summary information about an FFTrees x

**Usage**

```r
## S3 method for class 'FFTrees'
summary(object, tree = 1, 
```

**Arguments**

- `object`: FFTrees.
- `tree`: integer. The tree to summarise
- `...`: additional arguments (currently ignored)
titanic  

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  

Description
Voting data set

Usage
voting

Format
A data frame containing 435 rows and 16 columns

Source
wine

---

text

**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
  blood, 3
  breastcancer, 4
  car, 5
  contraceptive, 7
  creditapproval, 7
  fertility, 8
  forestfires, 20
  heart.cost, 21
  heart.test, 21
  heart.train, 22
  heartdisease, 23
  iris.v, 25
  mushrooms, 25
  sonar, 31
  titanic, 32
  voting, 32
  wine, 33

Add_Stats, 3

blood, 3
breastcancer, 4

car, 5
classtable, 5
comp.pred, 6
contraceptive, 7
creditapproval, 7

factclean, 8
fertility, 8
FFTrees, 9
FFTrees.guide, 13
fftrees_apply, 13
fftrees_create, 14
fftrees_cuerank, 15
fftrees_define, 16
fftrees_ffttowords, 16
fftrees_fitcomp, 17
fftrees_grow_fan, 17
fftrees_ranktrees, 17
fftrees_threshold_factor_grid, 18
fftrees_threshold_numeric_grid, 19
fftrees_wordstofftrees, 20
forestfires, 20

heart.cost, 21
heart.test, 21
heart.train, 22
heartdisease, 23

inwords, 24
iris.v, 25

mushrooms, 25

plot.FFTrees, 26
predict.FFTrees, 28
print.FFTrees, 30

showcues, 30
sonar, 31
summary.FFTrees, 31

titanic, 32

voting, 32

wine, 33

34