Accessors\_forest

**Accessor method for forest class**

**Description**
Methods to access public slots in the forest class.

**Usage**
```r
## S4 method for signature 'forest'
x[i]
```

**Arguments**
- `x` Object of class inheriting from "forest-class".
- `i` String value that is either "n", "p", "min\_node\_obs", "max\_depth", "numsamps", "numvars", "numboots", "numnodes", "model", "x", "y", or "fmla"

---

easy\_2var\_data

**Easy to fit dataset for decision trees**

**Description**
Fake data that should be easy for a decision tree to fit

**Usage**
easy\_2var\_data

**Format**
Data frame containing 3 variables and 100 observations.

**Source**
Author's creation.
Description

A forest of decision tree classifiers to be used for ensemble prediction.

Objects from the Class

Objects can be created by calls of the form new("forest", ...).

Slots

n: Number of observations in dataset used to fit this forest.
p: Number of independent variables in dataset used to fit this forest.
min_node_obs: Leaf of any tree in this forest will not be split unless it has more observations than this value.
max_depth: Maximum depth of any tree in this forest
numsamps: Number of observations randomly drawn with replacement used to fit a tree in this forest.
numvars: Number of independent variables randomly drawn without replacement used to fit a tree in this forest.
umboots: Number of trees in this forest.
numnodes: Vector with the number of nodes that each tree has in this forest.
flattened.nodes: Data frame containing information on the nodes of the trees in this forest.
model: Model frame used to fit this forest.
x: Design (independent variables) matrix used to fit this forest.
y: Dependent variable vector used to fit this forest.
fmla: Formula used to construct the model frame from the data.
depvar.restore.info: This is a slot that the package needs internally.

grow.forest Growing random decision forest classifier

Description

Grow random decision forest classifier
Usage

```r
grow.forest(formula, data, subset, na.action, impurity.function = "gini", model = FALSE, x = FALSE, y = FALSE, min_node_obs, max_depth, numsamps, numvars, numboots)
```

Arguments

- **formula**: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
- **data**: an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `grow.forest` is called.
- **subset**: an optional vector specifying a subset of observations to be used in the fitting process.
- **na.action**: a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The ‘factory-fresh’ default is `na.omit`. Another possible value is `null`, no action.
- **impurity.function**: the impurity function to be used to fit decision trees, currently only "gini" is supported.
- **model, x, y**: logistics. If `TRUE` the corresponding components of the fit (the model frame, the model matrix, the response) are returned.
- **min_node_obs**: the minimum number of observations required for a node to be split. If not provided as input, the package will attempt to choose a reasonable value.
- **max_depth**: the deepest that a tree should be fit (root node is at depth 0). If not provided as input, the package will attempt to choose a reasonable value.
- **numsamps**: number of samples to draw with replacement for each tree in the forest (bootstrapped sample). If not provided as input, the package will attempt to choose a reasonable value.
- **numvars**: number of variables to be randomly selected without replacement for each tree in the forest. If not provided as input, the package will attempt to choose a reasonable value.
- **numboots**: number of trees in the forest. If not provided as input, the package will attempt to choose a reasonable value.

Details

Bootstrapped samples will be automatically balanced between dependent variable classes. The number of sampled observations per tree will be increased as necessary to achieve a number that can divide the number of dependent variable classes so that bootstrapped samples will be balanced. The number of distinct values that the dependent variable has must be exactly two. Predictor variables should only be continuous, ordinal, or categorical with only two categories (do not include nominal variables or categorical variables with three or more categories).
low_high_earners

Examples

data(easy_2var_data)

fforest = grow.forest(Y~X1+X2, data=easy_2var_data,
  min_node_obs=5, max_depth=10,
  numsamps=90, numvars=1, numboots=5)

---

low_high_earners  Low earners and high earners, training dataset

Description

Dataset of low earners and high earners for classification. low_high_earners is the training dataset and low_high_earners_test is the testing dataset.

Usage

low_high_earners

Format

Data frame containing 8 variables and 199,522 observations.

Source

Prepared by keeping only the dependent variable, continuous variables, ordinal variables, and binary categorical variables from http://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29

---

low_high_earners_test  Low earners and high earners, testing dataset

Description

Dataset of low earners and high earners for classification. low_high_earners is the training dataset and low_high_earners_test is the testing dataset.

Usage

low_high_earners_test

Format

Data frame containing 8 variables and 99,761 observations.

Source

Prepared by keeping only the dependent variable, continuous variables, ordinal variables, and binary categorical variables from http://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29
predict.forest

Predict method for random decision forest classifier fits

Description

Predict method for random decision forest classifier fits

Usage

## S4 method for signature 'forest'
predict(object, newdata, ...)

Arguments

- **object**: Object of class inheriting from "forest-class".
- **newdata**: A data frame in which to look for variables with which to predict.
- **...**: further arguments passed to or from other methods.

Examples

data(easy_2var_data)

fforest = grow.forest(Y~X1+X2, data=easy_2var_data,
    min_node_obs=5, max_depth=10,
    numsamps=90, numvars=1, numboots=5)

xnew = data.frame(
    X1 = c(0.06, 0.05, 0.05, 0.01, 0.09, 0.05, 0.05,-1000, 1000),
    X2 = c(0.03, 0.02, 0.05, 0.03, 0.04,-1000, 1000, 0.04, 0.03)
)

fforest_ynewwhat = predict(fforest, xnew)
Index

* Topic **classes**
  forest-class, 3
* Topic **datasets**
  easy_2var_data, 2
  low_high_earners, 5
  low_high_earners_test, 5
  [, forest-method (accessors.forest), 2

  accessors.forest, 2
  as.data.frame, 4

  easy_2var_data, 2

  forest-class, 3
  formula, 4

  grow.forest, 3

  low_high_earners, 5
  low_high_earners_test, 5

  na.fail, 4
  na.omit, 4

  options, 4

  predict, forest-method (predict.forest), 6
  predict, tree-method (predict.forest), 6
  predict.forest, 6
  predict.tree (predict.forest), 6