Package 'ai'

October 2, 2024

Title Build, Predict and Analyse Artificial Intelligence Models

Version 1.0.4.44

Description An interface for data processing, building models, predicting values and analysing outcomes. Fitting Linear Models, Robust Fitting of Linear Models, k-Nearest Neighbor Classification, 1-Nearest Neighbor Classification, and Conditional Inference Trees are available.

Depends R (>= 4.4.0)

License GPL-3

Encoding UTF-8

URL https://github.com/urniaz/ai

BugReports https://github.com/urniaz/ai/issues

biocViews Software

Imports base, class, stats, caTools, MASS, party, Metrics

Suggests testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Rafal Urniaz [aut, cre] (<https://orcid.org/0000-0003-0192-2165>)

Maintainer Rafal Urniaz <rafal.urniaz@cantab.net>

Repository CRAN

Date/Publication 2024-10-02 13:30:06 UTC

Contents

config *Models parameters*

Description

The config function sets additional models parameters

Usage

```
config(formula = NULL, k = NULL)
```

Arguments

formula	formula parameter for eg. linear models including lm, rlm, read more: lm
k	number of neighbors considered from knn models, read more: knn

Value

configuration list contains models parameters different than defaults

Examples

```
config(formula = "Status ~ Value")
```

model *AI/ML models*

Description

The model function generates AI/ML models

Usage

```
model(data, type = "lm", config = NULL, verbose = FALSE)
```

Arguments

data	data object with data to be modeled, read more prodata
type	model type, lm (Fitting Linear Models) by default; available are lm, rlm, ctree, knn, knn1
config	additional parameters for model, read more config
verbose	if true the messages are displayed in console, false by default

Value

model list contains model, predicted, and expected values for all generated models

Examples

```
model_data <- data.frame(a = c(1,2,3,4,5,6),
                         b = c(1,2,3,4,5,6),
                         s = c(1,2,3,4,5,6))

config <- config(formula = "a ~ b + s")

model_data <- prodata(model_data, status_colname = "s")

model(model_data, config)
```

`prodata` *Data processing*

Description

The `prodata` function generates an data list for models. It additionally splits data for training and testing set by split ratio.

Usage

```
prodata(data, status_colname, SplitRatio = 0.75)
```

Arguments

`data`	data.frame with data to be modeled
`status_colname`	name of the column in data where the true results (true positive, expected) values are listed
`SplitRatio`	Splitting ratio; 0.75 means 75% data for training and 25% for testing, more: sample.split

Value

data list

Examples

```
model_data <- data.frame(a = c(1,2,3,4,5,6),
                         b = c(1,2,3,4,5,6),
                         s = c(1,2,3,4,5,6))

prodata(data = model_data, status_colname = "s")
```

`stats` *Models statistics*

Description

The `stats` function calculates models statistics. Read more auc

Usage

```
stats(modelA, modelB = NULL)
```

Arguments

`modelA`	Model generated by model function
`modelB`	Model generated by model function

Value

modified model list contains additional statistics

Examples

```
model_data <- data.frame(a = c(1,2,3,4,5,6),
                         b = c(1,2,3,4,5,6),
                         s = c(1,2,3,4,5,6))

model_data <- prodata(model_data, status_colname = "s")

config <- config(formula = "a ~ b + s")

model <- model(model_data, config)

stats(model)
```

stats_compare_models *stats_compare_models()*

Description

stats_compare_models()

Usage

```
stats_compare_models(modelA, modelB)
```

Arguments

modelA	modelA
modelB	modelB

Value

data.frame contains comparison of both models statistics

stats_model *stats_model()*

Description

stats_model()

Usage

```
stats_model(model)
```

Arguments

model	model

Value

list contains model statistics

Index