Package

May 28, 2019

Type Package

Title Automated Machine Learning System in R

Description Provides web UI to train regression and classification models.

Version 0.0.17

Date 2019-04-21

License GPL-3

Maintainer Thiloshon Nagarajah <thiloshon@gmail.com>

Imports rmarkdown, knitr, shiny, shinydashboard, shinyjs, DT, data.table, mlr, graphics, methods, stats, utils

Depends R (>= 2.10)

RoxygenNote 6.1.1

LazyData true

R topics documented:

category_known *find if data is known category*

Description

find if data is known category

Usage

```
category_known(type)
```

category_pred *find if data is categorical.*

Description

find if data is categorical.

Usage

```
category_pred(type)
```

factorPre *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
factorPre(data, perform = T)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

getProperties *Get properties to filter algorithms.*

Description

Get properties to filter algorithms.

Usage

```
getProperties(data, type, target)
```

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

identifierPre *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
identifierPre(data, perform = T)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

imputePre *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
imputePre(data, perform = T)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

label_data *find if data is labelled*

Description

find if data is labelled

Usage

```
label_data(type)
```

MLPlan-class *MLPlan r6 class.*

Description

MLPlan r6 class.

Value

MLPlan object with default settings

normalizePre *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
normalizePre(data, perform = T)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

outlierPre *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
outlierPre(data, perform = T)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

PipeLine-class *PipeLine r6 class.*

Description

PipeLine r6 class.

Value

PipeLine object with default settings

recommend_preprocessing *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
recommend_preprocessing(data, algorithm, breed)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

run_mlr *Run Shiny app.*

Description

Run Shiny app.

Usage

```
run_mlr()
```

Value

Web App

skewPre *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
skewPre(data, perform = T)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

split_data *Create the package default Questionnaire.*

Description

Create the package default Questionnaire.

Usage

```
split_data(data)
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

suggest_learner *Suggest algorithms based on hybrid approach.*

Description

Suggest algorithms based on hybrid approach.

Usage

```
suggest_learner(dataset, type = "regression", predictor)
```

Value

expected performance score for the dataset and algorithms

Examples

```
val <- suggest_learner(data, "classsification", "Species")
```

suggest_learner_manual *Suggest algorithms based on Decision tree method.*

Description

Suggest algorithms based on Decision tree method.

Usage

```
suggest_learner_manual(dataset, type = "regression", target)
```

Value

expected performance score for the dataset and algorithm

Examples

```
val <- suggest_learner_manual(data, "classsification", "Species")
```

suggest_learner_meta *Suggest algorithms based on meta learninng technique.*

Description

Suggest algorithms based on meta learninng technique.

Usage

```
suggest_learner_meta(dataset, type = "classification", predictor,
  algorithms)
```

Value

expected performance score for the dataset and algorithm

Examples

```
val <- suggest_learner_meta(data, "classsification", "Species", "nnet")
```

Index