Package 'citrus'

November 17, 2021

Type Package

Title Customer Intelligence Tool for Rapid Understandable Segmentation

Version 1.0.1

Maintainer Dom Clarke <dom.clarke@peak.ai>

Copyright See the file COPYRIGHTS

Description A tool to easily run and visualise supervised and unsupervised state of the art customer segmentation.
It is built like a pipeline covering the 3 main steps in a segmentation project: pre-processing, modelling, and plotting.
Users can either run the pipeline as a whole, or choose to run any one of the three individual steps.
It is equipped with a supervised option (tree optimisation) and an unsupervised option (k-clustering) as default models.

License MIT + file LICENSE

Suggests testthat

Depends R (>= 3.5.0)

Imports ggplot2, GGally, clustMixType (>= 0.1-16), treeClust, rpart, tibble, rpart.plot, rpart.utils, stringr, dplyr, RColorBrewer, rlang

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

NeedsCompilation no

Author Dom Clarke [aut, cre], Cinzia Braglia [aut], Oskar Nummedal [aut], Leo McCarthy [aut], Rebekah Yates [aut], Joash Alonso [aut], PEAK AI LIMITED [cph]

Repository CRAN

Date/Publication 2021-11-17 21:10:08 UTC

R topics documented:

`citrus_pair_plot` *Creates pair plot from data table*

Description

Creates pair plot from data table

Usage

```
citrus_pair_plot(model, vars = NULL)
```

Arguments

`model`	list, a citrus segmentation model
`vars`	data.frame, the data to segment

Value

GGally object displaying the segment feature pair plots.

k_clusters *k-clusters model*

Description

k-clusters method for segmentation. It can handle segmentation for both numerical data types only, by using k-means algorithm, and mixed data types (numerical and categorical) by using k-prototypes algorithm

Usage

```
k_clusters(data, hyperparameters, verbose = TRUE)
```

Arguments

data	data.frame, the data to segment
hyperparameters	list of hyperparameters to pass. They include centers: number of clusters or a set of initial (distinct) cluster centers, or 'auto'. When 'auto' is chosen, the number of clusters is optimised; iter_max: the maximum number of iterations allowed; n_start: how many random sets of cluster centers should be tried; max_centers: maximum number of clusters when 'auto' option is selected for the centers; segmentation_variables: the columns to use to segment on. standardize: whether to standardize numeric columns.
verbose	logical whether information about the clustering procedure should be given.

Value

A class called "k-clusters" containing a list of the model definition, the hyper-parameters, a table of outliers, the elbow plot (ggplot object) used to determine the optimal no. of clusters, and a lookup table containing segment predictions for customers.

model_management *Model management function*

Description

Saves the model and its settings so that it can be recreated

Usage

```
model_management(model, hyperparameters)
```

Arguments

`model`	data.frame, the model to save
`hyperparameters`	list, list of hyperparameters of the model

Value

No return value. Called to save model and settings locally.

`output_table` *Output Table*

Description

Generates the output table for model and data

Usage

```
output_table(data, model)
```

Arguments

`data`	A dataframe generated from the pre-processing step
`model`	A model object used to classify ids with, generated from the model selection layer

Value

A tibble providing high-level segment attributes such as mean and max (numeric) or mode (categorical) for the segmentation features used.

`preprocess` *Preprocess Function*

Description

Transforms a transactional table into an id aggregated table with custom options for aggregation methods for numeric and categorical columns.

Usage

```
preprocess(
  df,
  samplesize = NA,
  numeric_operation_list = c("mean"),
  categories = NULL,
  target = NA,
  target_agg = "mean",
  verbose = TRUE
)
```

Arguments

df	data.frame, the data to preprocess
samplesize	numeric, the fraction of ids used to create a sub-sample of the input df
numeric_operation_list	list, a list of the aggregation functions to apply to numeric columns
categories	list, a list of the categorical columns to aggregate
target	character, the column to use as a response variable for supervised learning
target_agg	character, the aggregation function to use to aggregate the target column
verbose	logical whether information about the preprocessing should be given

Value

An id attributes data frame, e.g. customer attributes if the id represents customer IDs. A single row per unique id.

---

preprocessed_data *Segmentation preprocessed data*

Description

A sample customer dataset for the purpose of demonstrating the segmentation algorithm.

Usage

```
data(preprocessed_data)
```

Format

Data frame on a customer level. Contains 402 rows and 8 columns.

Examples

```
data(preprocessed_data)
```

`rpart.plot_pretty` *Plot a prettified rpart model*

Description

Plot an rpart model and prettifies it. Wrap around the rpart.plot::prp function

Usage

```
rpart.plot_pretty(
  model,
  main = "",
  sub,
  caption,
  palettes,
  type = 2,
  fontfamily = "sans",
  ...
)
```

Arguments

`model`	an rpart model object
`main`	main title
`sub`	fixing captions in line
`caption`	character, caption to use in the plot
`palettes`	list, list of colours to use in the plot
`type`	type of plot. Default is 2. Possible values are: 0 Default. Draw a split label at each split and a node label at each leaf. 1 Label all nodes, not just leaves. 2 Like 1 but draw the split labels below the node labels. 3 Draw separate split labels for the left and right directions. 4 Like 3 but label all nodes, not just leaves. 5 Show the split variable name in the interior nodes.
`fontfamily`	Names of the font family to use for the text in the plots.
`...`	Additional arguments.

Value

An rpart.plot object. This plot object can be plotted using the rpart::prp function.

segment *Segment Function*

Description

Segments the data by running all steps in the segmentation pipeline, including output table

Usage

```
segment(
  data,
  modeltype = c("tree", "k-clusters"),
  FUN = NULL,
  FUN_preprocess = NULL,
  steps = c("preprocess", "model"),
  prettify = FALSE,
  print_plot = FALSE,
  hyperparameters = NULL,
  force = FALSE,
  verbose = FALSE
)
```

Arguments

data	data.frame, the data to segment
modeltype	character, the type of model to use to segment choices are: 'tree', 'k-clusters'
FUN	function, A user specified function to segment, if the standard methods are not wanting to be used
FUN_preprocess	function, A user specified function to preprocess, if the standard methods are not wanting to be used
steps	list, names of the steps the user want to run the data on. Options are 'preprocess' and 'model'
prettify	logical, TRUE if want cleaned up outputs, FALSE for raw output
print_plot	logical, TRUE if want to print the plot
hyperparameters	list of hyperparameters to use in the model.
force	logical, TRUE to ignore errors in validation step and force model execution.
verbose	logical whether information about the segmentation pipeline should be given.

Value

A list of three objects. A tibble providing high-level segment attributes, a lookup table (data frame) with the id and predicted segment number, and an rpart object defining the model.

transactional_data *Segmentation transactional data*

Description

A sample customer dataset for the purpose of demonstrating the segmentation algorithm.

Usage

```
data(transactional_data)
```

Format

Data frame on a transactional level. Contains 10,000 rows and 6 columns.

Examples

```
data(transactional_data)
```

tree_abstract *Abstraction layer function*

Description

Organises the model outputs, predictions and settings in a general structure

Usage

```
tree_abstract(model, inputdata)
```

Arguments

model	The model to organise
inputdata	The data used to train the model

Value

A structure with the class name "tree_model" which contains a list of all the relevant model data, including the rpart model object, hyper-parameters, segment table, labelled customer lookup table, and the input data used to train the model.

tree_segment *Tree Segment Function*

Description

Runs decision tree optimisation on the data to segment ids.

Usage

```
tree_segment(data, hyperparameters, verbose = TRUE)
```

Arguments

data	data.frame, the data to segment
hyperparameters	list, list of hyperparameters to pass. They include segmentation_variables: a vector or list with variable names that will be used as segmentation variables; dependent_variable: a string with the name of the dependent variable that is used in the clustering; min_segmentation_fraction: integer, the minimum segment size as a proportion of the total data set; number_of_segments: integer, number of leaves you want the decision tree to have.
verbose	logical whether information about the segmentation procedure should be given.

Value

List of 4 objects. The rpart object defining the model, a data frame providing high-level segment attributes, a lookup table (data frame) with the id and predicted segment number, and a list of the hyperparameters used.

tree_segment_prettify *Tree Segment Prettify Function*

Description

Returns a prettier version of the decision tree.

Usage

```
tree_segment_prettify(tree, char_length = 20, print_plot = FALSE)
```

Arguments

tree	The decision tree model to prettify
char_length	integer, the character limit before truncating categories and putting them into an "other" group
print_plot	logical, indicates whether to print the generated plot or not

Value

A formatted and "prettified" rpart.plot object. This plot object can be plotted using the rpart::prp function.

`validate` *Validation function*

Description

Validates that the input data adheres to the expected format for modelling.

Usage

```
validate(df, supervised = TRUE, force, hyperparameters)
```

Arguments

`df`	data.frame, the data to validate
`supervised`	logical, TRUE for supervised learning, FALSE for k-clusters
`force`	logical, TRUE to ignore error on categorical columns
`hyperparameters`	list of hyperparameters used in the model

Value

'TRUE' if all checks are passed. Otherwise an error is raised.

Index