

Package ‘jenga’

July 22, 2025

Type Package

Title Fast Extrapolation of Time Features using K-Nearest Neighbors

Version 1.3.0

Author Giancarlo Vercellino

Maintainer Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

Description Fast extrapolation of univariate and multivariate time features using K-Nearest Neighbors. The compact set of hyper-parameters is tuned via grid or random search.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 4.1)

Imports purrr (>= 0.3.4), abind (>= 1.4-5), ggplot2 (>= 3.3.5), readr (>= 2.1.2), lubridate (>= 1.4.0), narray (>= 0.4.1.1), imputeTS (>= 3.2), scales (>= 1.1.1), tictoc (>= 1.0.1), modeest (>= 2.4.0), moments (>= 0.14), philentropy (>= 0.5.0), greybox (>= 1.0.1), Rfast (>= 2.0.6), dplyr (>= 1.0.7), fastDummies (>= 1.6.3), fANCOVA (>= 0.6-1), entropy (>= 1.3.1)

URL https://rpubs.com/giancarlo_vercellino/jenga

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-18 08:10:02 UTC

Contents

covid_in_europe	2
jenga	2
Index	5

covid_in_europe	<i>covid_in_europe data set</i>
-----------------	---------------------------------

Description

A data frame with with daily and cumulative cases of Covid infections and deaths in Europe since March 2021.

Usage

```
covid_in_europe
```

Format

A data frame with 5 columns and 163 rows.

Source

www.ecdc.europa.eu

jenga	<i>jenga: automatic projections of time features using KNN</i>
-------	--

Description

Automatic projections of time features using KNN

Usage

```
jenga(  
  df,  
  seq_len = NULL,  
  smoother = FALSE,  
  k = NULL,  
  method = NULL,  
  kernel = NULL,  
  ci = 0.8,  
  n_windows = 10,  
  mode = NULL,  
  n_sample = 30,  
  search = "random",  
  dates = NULL,  
  error_scale = "naive",  
  error_benchmark = "naive",  
  seed = 42  
)
```

Arguments

<code>df</code>	A data frame with time features on columns (numerical or categorical features, but not both).
<code>seq_len</code>	Positive integer. Time-step number of the projected sequence
<code>smoother</code>	Logical. Perform optimal smoothing using standard loess (only for numerical features). Default: FALSE
<code>k</code>	Positive integer. Number of neighbors to consider when applying kernel average. Min number is 3. Default: NULL (automatic selection).
<code>method</code>	Positive integer. Distance method for calculating neighbors. Possible options are: "euclidean", "manhattan", "minkowski". Default: NULL (automatic selection).
<code>kernel</code>	String. Distribution used to calculate kernel densities. Possible options are: "norm", "cauchy", "unif", "t". Default: NULL (automatic selection).
<code>ci</code>	Confidence interval. Default: 0.8
<code>n_windows</code>	Positive integer. Number of validation tests to measure/sample error. Default: 10.
<code>mode</code>	String. Sequencing method: deterministic ("segmented"), or non-deterministic ("sampled"). Default: NULL (automatic selection).
<code>n_sample</code>	Positive integer. Number of samples for grid or random search. Default: 30.
<code>search</code>	String. Two option available: "grid", "random". Default: "random".
<code>dates</code>	Date. Vector with dates for time features.
<code>error_scale</code>	String. Scale for the scaled error metrics. Two options: "naive" (average of naive one-step absolute error for the historical series) or "deviation" (standard error of the historical series). Default: "naive".
<code>error_benchmark</code>	String. Benchmark for the relative error metrics. Two options: "naive" (sequential extension of last value) or "average" (mean value of true sequence). Default: "naive".
<code>seed</code>	Positive integer. Random seed. Default: 42.

Value

This function returns a list including:

- `exploration`: list of all models, complete with predictions, test metrics, prediction stats and plot
- `history`: a table with the sampled models, hyper-parameters, validation errors
- `best_model`: results for the best model, including:
 - `predictions`: min, max, q25, q50, q75, quantiles at selected ci, and different statics for numerical and categorical variables
 - `testing_errors`: training and testing errors for one-step and sequence for each ts feature (different measures for numerical and categorical variables)
- `time_log`

Author(s)

Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

See Also

Useful links:

- https://rpubs.com/giancarlo_vercellino/jenga

Examples

```
jenga(covid_in_europe[, c(2, 3)], n_sample = 1)
jenga(covid_in_europe[, c(4, 5)], n_sample = 1)
```

Index

* **datasets**

covid_in_europe, [2](#)

covid_in_europe, [2](#)

jenga, [2](#)

jenga-package (jenga), [2](#)