# Package 'tsfknn'

April 5, 2021

**Type** Package

**Title** Time Series Forecasting Using Nearest Neighbors

**Version** 0.5.0

**Description** Allows to forecast time series using nearest neighbors regression
Francisco Martinez, Maria P. Frias, Maria D. Perez-Godoy and Antonio J.
Rivera (2017) <doi:10.1007/s10462-017-9593-z>. When the forecasting horizon
is higher than 1, two multi-step ahead forecasting strategies can be used.
The model built is autoregressive, that is, it is only based on the
observations of the time series. The nearest neighbors used in a prediction
can be consulted and plotted.

**Maintainer** Francisco Martinez <fmartin@ujaen.es>

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Depends** R (>= 3.6.0)

**Suggests** knitr, rmarkdown, testthat (>= 2.0.1)

**Imports** ggplot2 (>= 3.1.1), graphics, Rcpp, stats, utils

**VignetteBuilder** knitr

**URL** https://github.com/franciscomartinezdelrio/tsfknn

**BugReports** https://github.com/franciscomartinezdelrio/tsfknn/issues

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Author** Francisco Martinez [aut, cre]

**Repository** CRAN

**Date/Publication** 2021-04-05 08:10:02 UTC

## R **topics documented:**

---

autoplot.knnForecast    *Create a ggplot object from a knnForecast object*

---

#### Description

It uses a knnForecast object to create a ggplot object that plots a time series and its forecast using KNN regression.

#### Usage

```
## S3 method for class 'knnForecast'
autoplot(forecast, highlight = "none", faceting = TRUE)
```

#### Arguments

| | |
|---|---|
| forecast | The knnForecast object. |
| highlight | A string value indicating what elements should be highlighted. Possible values are "none", "points" and "neighbors". |
| faceting | Logical. This applies only if the highlight parameter is set to "neighbors". It indicates whether the different nearest neighbors should be seen in different plots (TRUE) or in one plot. |

#### Value

The ggplot object representing a plotting with the forecast.

#### Examples

```
pred <- knn_forecasting(USAccDeaths, h = 12, lags = 1:12, k = 2)
library(ggplot2)
autoplot(pred)
autoplot(pred, highlight = "neighbors")
```

---

knn_examples                    *Examples of the model associated with a prediction*

---

### Description

It allows to see the examples of the model associated to a knnForecast object.

### Usage

```
knn_examples(forecast)
```

### Arguments

forecast        A knnForecast object.

### Value

A matrix including the features and targets of the examples associated with the model of a knnForecast object.

### Examples

```
pred <- knn_forecasting(ts(1:8), h = 1, lags = 1:2, k = 2)
knn_examples(pred)
```

---

knn_forecasting                 *Time series forecasting using KNN regression*

---

### Description

It applies KNN regression to forecast the future values of a time series. The lags used as autoregressive variables are set with the lags parameter. If the user does not set the number of nearest neighbors or the lags, these values are selected automatically.

### Usage

```
knn_forecasting(
  timeS,
  h,
  lags = NULL,
  k = c(3, 5, 7),
  msas = c("recursive", "MIMO"),
  cf = c("mean", "median", "weighted"),
  transform = c("additive", "multiplicative", "none")
)
```

## Arguments

| | |
|---|---|
| timeS | A numeric vector or time series of class ts. |
| h | A positive integer. Number of values to forecast. |
| lags | An integer vector in increasing order expressing the lags used as autoregressive variables. |
| k | A positive integer. The k parameter in KNN regression. A vector of k values can also be used. In that case, the forecast is the average of the forecasts produced by the different models with the different k parameters. |
| msas | A string indicating the Multiple-Step Ahead Strategy used when more than one value is predicted. It can be "recursive" or "MIMO" (the default). |
| cf | A string. It indicates the combination function used to aggregate the targets associated with the nearest neighbors. It can be "median", "weighted" or "mean" (the default). |
| transform | A character value indicating whether the training samples are transformed. If the time series has a trend it is recommended. By default is ″multiplicative″ (multiplicative transformation). It is also possible a multiplicative transformation or no transformation. |

## Value

An object of class ″knnForecast″. The function [summary](#) can be used to obtain or print a summary of the results.

An object of class ″knnForecast″ is a list containing at least the following components:

| | |
|---|---|
| call | the matched call. |
| msas | the Multi-Step Ahead Strategy. |
| prediction | a time series with the forecast. |
| model | an object of class ″knnModel″ with the KNN model |

## Examples

```
pred <- knn_forecasting(USAccDeaths, h = 12, lags = 1:12, k = 2)
pred$prediction # To see a time series with the forecasts
plot(pred) # To see a plot with the forecast
```

---

| nearest_neighbors | *Nearest neighbors associated with predictions* |
|---|---|

---

## Description

It allows to check the new instances and their nearest neighbors used in a prediction associated with a "knnForecast" object.

## Usage

```
nearest_neighbors(forecast)
```

## Arguments

| | |
|---|---|
| forecast | A knnForecast object. |

## Value

A list including the new instances used in KNN regression and their nearest neighbors.

## Examples

```
pred <- knn_forecasting(UKgas, h = 4, lags = 1:4, k = 2, msas = "MIMO")
nearest_neighbors(pred)
```

---

n_training_examples     *Number of training examples*

---

## Description

It computes the number of training examples that would have a KNN model with the specified parameters.

## Usage

```
n_training_examples(timeS, h, lags, msas = c("MIMO", "recursive"))
```

## Arguments

| | |
|---|---|
| timeS | A numeric vector or time series of class ts. |
| h | A positive integer. Number of values to forecast. |
| lags | An integer vector in increasing order expressing the lags used as autoregressive variables. |
| msas | A string indicating the Multiple-Step Ahead Strategy used when more than one value is predicted. It can be "recursive" or "MIMO" (the default). |

## Value

An integer.

## Examples

```
n_training_examples(ts(1:10), h = 2, lags = 1:3, msas = "MIMO")
n_training_examples(ts(1:10), h = 2, lags = 1:3, msas = "recursive")
```

---

plot.knnForecastRO          *Plot a prediction of a test set*

---

### Description

It uses a test set generated with the function [rolling_origin](rolling_origin) and plots its forecast.

### Usage

```
## S3 method for class 'knnForecastRO'
plot(x, h = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | the object obtained from a call to [rolling_origin](rolling_origin). |
| h | an integer. The forecasting horizon. If NULL, the maximum forecasting horizon of all the test sets is used. |
| ... | Other plotting parameters to affect the plot. |

---

predict.knnForecast          *Predict method for KNN models for time series forecasting.*

---

### Description

Predicted values based on a KNN model for time series forecasting.

### Usage

```
## S3 method for class 'knnForecast'
predict(object, h, ...)
```

### Arguments

| | |
|---|---|
| object | a knnForecast object obtained by a call to the [knn_forecasting](knn_forecasting) function. |
| h | an integer. The forecasting horizon. |
| ... | further arguments passed to or from other methods. |

### Details

If the models uses the MIMO strategy for multiple-step ahead prediction, the forecasting horizon is fixed to the model forecasting horizon.

### Value

a knnForecast object with the prediction and information about the KNN model, see the documentation of [knn_forecasting](knn_forecasting) for the structure of knnForecast objects.

## Examples

```
pred <- knn_forecasting(UKgas, h = 4, k = 1, msas = "recursive")
new_pred <- predict(pred, h = 6)
print(new_pred$prediction)
plot(new_pred) # To see a plot with the forecast
```

---

| rolling_origin | *Assessing forecasting accuracy with rolling origin* |
| --- | --- |

---

## Description

It uses the model and the time series associated with the knnForecast object to asses the forecasting accuracy of the model using the last h values of the time series to build test sets applying a rolling origin evaluation.

## Usage

```
rolling_origin(knnf, h = NULL, rolling = TRUE)
```

## Arguments

| | |
| --- | --- |
| knnf | A knnForecast object. |
| h | A positive integer. The forecast horizon. If NULL the prediction horizon of the knnForecast object is used. |
| rolling | A logical. If TRUE (the default), forecasting horizons from 1 to h are used. Otherwise, only horizon h is used. |

## Details

This function assesses the forecast accuracy of the model used by the knnForecast object. It uses h different test and training sets. The first test set consists of the last h values of the time series (the training set is formed by the previous values). The next test set consists of the last $h - 1$ values of the time series and so on (the last test set is formed by the last value of the time series).

## Value

A list containing at least the following fields:

| | |
| --- | --- |
| test_sets | a matrix containing the test sets used in the evaluation. Every row contains a different test set. |
| predictions | The predictions for the test sets. |
| errors | The errors for the test sets. |
| global_accu | Different measures of accuracy applied to all the errors. |
| h_accu | Different measures of accuracy applied to all the errors for every forecasting horizon. |

## Examples

```
pred <- knn_forecasting(UKgas, h = 4, lags = 1:4, k = 2)
ro <- rolling_origin(pred)
print(ro$global_accu)
```

---

tsfknn                          *tsfknn: A package for time series forecasting using KNN regression.*

---

## Description

The tsfknn package allows univariate time series forecasting using KNN regression.

## Functions

**knnForecasting** It is used to forecast a time series

**n_training_examples** To compute how many training examples would have a model

**nearest_neighbors** To see the nearest neighbors used to forecast a times series

**predict** To make new forecasts

**knn_examples** To see the examples used by the KNN model

**rolling_origin** To assess forecasting accuracy using rolling origin evaluation

**autoplot** To plot a prediction and the nearest neighbors used in the prediction

# Index