Package ‘PredictTestbench’

December 20, 2016

Type Package
Title Test Bench for Comparison of Data Prediction Models
Version 1.1.3
Author Neeraj Bokde
Maintainer Neeraj Bokde <neerajdhanraj@gmail.com>
Description Provides a Testbench for comparison of prediction models. This package is inspired from ‘imputeTestbench’ package <https://cran.r-project.org/package=imputeTestbench>. It compares prediction models with reference to RMSE, MAE or MAPE parameters. It allows to add new proposed methods to test bench and to compare with other methods. The function ‘prediction_append()’ allows to add multiple numbers of methods to the existing methods available in test bench. One/two step ahead prediction is also possible in the testbench.
License GPL

BugReports https://github.com/neerajdhanraj/PredictTestbench/issues
URL http://www.neerajbokde.com/cran/predicttestbench
Imports ggplot2, reshape2, PSF, forecast, methods, stats, imputeTestbench
LazyData TRUE
RoxygenNote 5.0.1
Suggests knitr, rmarkdown
NeedsCompilation no
Repository CRAN
Date/Publication 2016-12-20 13:02:09

R topics documented:

plot_predictions .................................................. 2
prediction_append ................................................. 2
prediction_errors .................................................. 3
prediction_remove ............................................... 4
step_ahead_forecast ............................................ 4
plot_predictions  

**Function to plot the Error Comparison**

**Description**

Function to plot the Error Comparison

**Usage**

```r
dataIn
```

**Arguments**

- `dataIn` as input data in list format (returned by function `prediction_errors()`)

**Value**

It returns the Error comparison for different methods

**Examples**

```r
# aa <- prediction_errors()
# bb <- plot_predictions(aa)
# bb
```

prediction_append  

**To attach and compare new method to existing comparison study done with function 'impute_errors()'**

**Description**

To attach and compare new method to existing comparison study done with function 'impute_errors()'  

**Usage**

```r
dataIn
```

**Arguments**

- `existing_method` as Error observations for different methods
- `dataIn` as imput time series for testing
- `nextVal` as an integer to decide number of values to predict
- `errorParameter` as type of error calculation (RMSE, MAE or MAPE)
- `MethodPath` as location of function for proposed imputation method
- `MethodName` as name for function for proposed imputation method
**prediction_errors**

**Value**

Returns error comparison for imputation methods

**Examples**

# Kindly, refer "Vignette" document

---

**Description**

Function working as testbench for comparison of Prediction algorithms

**Usage**

prediction_errors(dataIn, nextVal, errorParameter, MethodPath, MethodName)

**Arguments**

- **dataIn**: as input time series for testing
- **nextVal**: as an integer to decide number of values to predict
- **errorParameter**: as type of error calculation (RMSE, MAE or MAPE)
- **MethodPath**: as location of function for the proposed imputation method
- **MethodName**: as name for function for the proposed imputation method

**Value**

Returns error comparison for imputation methods

**Examples**

```r
# aa <- prediction_errors(nextVal = 10)
# aa
```
**prediction_remove**

*Removes unwanted method from already existing methods in Test bench*

**Description**

Removes unwanted method from already existing methods in Test bench

**Usage**

```python
prediction_remove(existing_method, index_number)
```

**Arguments**

- `existing_method` as Error observations for different methods
- `index_number` as index number of unwanted method in study

**Value**

It removes unwanted method from test bench and returns other method errors

---

**step_ahead_forecast**

*Function to calculate the step ahead forecasting for a proposed Prediction method*

**Description**

Function to calculate the step ahead forecasting for a proposed Prediction method

**Usage**

```python
step_ahead_forecast(dataIn, trainedData, MethodPath, errorParameter, stepSize)
```

**Arguments**

- `dataIn` as input time series for testing
- `trainedData` as partition point of input data ‘dataIn’
- `MethodPath` as location of function for the proposed imputation method
- `errorParameter` as type of error calculation (RMSE, MAE or MAPE)
- `stepSize` as interval at which step by step prediction will be done (Possible values are 1 & 2)

**Value**

Returns the plot for one/two step ahead prediction along with error values decided by ‘errorParameter’
Index

plot_predictions, 2
prediction_append, 2
prediction_errors, 3
prediction_remove, 4

step_ahead_forecast, 4