Package 'EFS'

July 24, 2017

Title Tool for Ensemble Feature Selection

Description Provides a function to check the
importance of a feature based on a dependent classification
variable. An ensemble of feature selection methods
is used to determine the normalized importance value of
all features. Combining these methods in one function
(building the cumulative importance values) provides a
stable feature selection tool. This selection
can also be viewed in a barplot using the barplot_fs() function
and proved using the evaluation function efs_eval().

Type Package

Version 1.0.3

Date 2016-11-18

License GPL (>= 2)

Encoding UTF-8

Author Nikita Genze, Ursula Neumann

Maintainer Ursula Neumann <u.neumann@wz-straubing.de>

LazyLoad yes

Imports party, pROC, randomForest, ROCR, grDevices, graphics, stats,
utils

Repository CRAN

RoxygenNote 6.0.1

NeedsCompilation no

Date/Publication 2017-07-24 15:45:14 UTC

R topics documented:

barplot_fs *Visualization of* ensemble_fs *in barplot*

Description

Generates a barplot from the output of ensemble_fs and produces a pdf-file. This file will be located in the working directory. A barplot will only be provided, when the number of features does not exceed 100.
x-axis: sum of all normed importance values of each feature ranging from 0 to 1
y-axis: names of features
If the number of features is greater or equal to 100, a barplot of the summed up importance over all FS method is created.
x-axis: features; y-axis: importance values
If order = TRUE the bars will be ordered in an increasing order bottom up (i.e., the most important parameter are on top).

Usage

```
barplot_fs(name, efs_table, order = TRUE)
```

Arguments

name	a character string giving the name of the file. If it is NULL, then no external file is created (effectively, no drawing occurs), but the device may still be queried.
efs_table	a table object of class matrix (retrieved from ensemble_fs)
order	a logical value indicating whether the bars should be sorted in descending order or not

Author(s)

Ursula Neumann

See Also

barplot, pdf

Examples

```
## Loading dataset in environment
data(efsdata)
## Generate a ranking based on inportance (with default
## NA_threshold = 0.7,cor_threshold = 0.2)
efs <- ensemble_fs(efsdata ,5 ,runs=2)
## Create a cumulative barplot based on the output from efs
barplot_fs("test", efs, order = TRUE)
```

efsdata *Meteorological data for feature selection analysis*

Description

A dataset with meteorological data from a weather station in Frankfurt (Oder), Germany from february 2016

Usage

```
data(efsdata)
```

Format

a data frame with 29 entries and following 7 variables

`date` index variable from 1 to 29

`Tmin` temperature minimum of the day

`Tmax` temperature maximum of the day

`SunAvg` sunshine duration of the day

`RainBool` classification variable: if it has not rained: 0, if it has rained: 1

`RelHumAvg` average relative humidity of the day

`WindForceAvg` average wind force of the day

References

modified data from http://wetterstationen.meteomedia.de/

efs_eval *Evaluation of Ensemble Features Selection*

Description

Provides several evaluation tests of the ouput of `ensemble_fs`. There are performance test, namely the logreg test and permutation test as well as tests of stability via the variance of feature importances and the Jaccard-index (see Details).

Usage

```
efs_eval(data, efs_table, file_name, classnumber, NA_threshold, logreg = TRUE,
  rf = TRUE, permutation = TRUE, p_num = 100, variances = TRUE,
  jaccard = TRUE, bs_num = 100, bs_percentage = 0.9)
```

Arguments

data	an object of class data.frame
efs_table	a table object of class matrix (retrieved from ensemble_fs)
file_name	a character string, name which is used for the two possible PDF files.
classnumber	a number indicating the index of variable for binary classification
NA_threshold	a number in range of [0,1]. Threshold for deletion of features with a greater proportion of NAs than NA_threshold.
logreg	a logical value indicating whether to conduct an evaluation via logistic regression or not
rf	a logical value indicating whether to conduct an evaluation via random forest or not
permutation	a logical value indicating whether to conduct a permutation of the class variable or not
p_num	number of permutations
variances	a logical value indicating whether to calculate the variances of importances retrieved from bootrapping or not
jaccard	a logical value indicating whether to calculate the jaccard-index or not
bs_num	a number of boostrap permutations of the importances
bs_percentage	a number in range of [0,1]. Proportion of randomly selected samples for boostraping

Details

A logistic regression model with leave-one-out cross-validation (LOOCV) of the selected features and of all feature is conducted by `logreg = TRUE`. Both AUC-values of the ROC curves are compared with `roc.test`. The ROC curves are illustrated on the PDF file "file_name" + "LG-ROC.pdf". By `rf = TRUE`, random forst model will be constructed and evaluated. Parallel to Logreg, the AUC-values of the two ROC curves of all features and a subset of the best ranked feautres are compared with `roc.test`. The ROC curves are illustrated on the PDF file "file_name" + "RF-ROC.pdf".

The permutation test (`permutation = TRUE`) compares the AUC outcome of an logistic regression with `p_num` AUCs from random permutations of the class variable by a `t.test`.

Variances of the importances after a bootstrapping analysis are calculated by `variances = TRUE`. Thereby the number and proportion of the bootstrapping can be set by `bs_num` and `bs_percentage`. The function also provides a PDF file "file_name" +"_Variances.pdf". Additionally, the Jaccard-index of this bootstrapped importances can be calculated by setting `jaccard=TRUE`.

Value

An object of class list, with the following components:
"AUC of LR with all parameters",
"AUC of LR with EFS parameter"
"P-value of LR-ROC test", #'
"AUC of RF with all parameters",

"AUC of RF with EFS parameter"
"P-value of RF-ROC test",
"P-value of permutation",
"Variances of feature importances",
"Jaccard-index".

Author(s)

Ursula Neumann

See Also

glm, roc,prediction, boxplot, tail, t.test

Examples

```
## Loading dataset in environment
data(efsdata)
## Generate a ranking based on importance (with default
## NA_threshold = 0.7,cor_threshold = 0.2)
efs<-ensemble_fs(efsdata,5,runs=2)
## Conduct AUC test and permutation test
eval_example <- efs_eval(data = efsdata, efs_table = efs, file_name = 'eval_test',
                     classnumber = 5, NA_threshold = 0.2,
                     logreg = TRUE,
                     rf = FALSE,
                     permutation = TRUE, p_num = 2,
                     variances = FALSE, jaccard = FALSE)
## Calculating variances and the Jaccard-index can take several minutes computation time
```

---

ensemble_fs *Ensemble Feature Selection*

Description

Uses an ensemble of feature selection methods to create a normalized quantitative score of all relevant features. Irrelevant features (e.g. features with too many missing values or variance = 1) will be deleted. See Details for a list of tests used in this function.

Usage

```
ensemble_fs(data, classnumber, NA_threshold = 0.2, cor_threshold = 0.7,
  runs = 100, selection = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, FALSE,
  FALSE))
```

Arguments

`data`	an object of class data.frame
`classnumber`	a number indicating the index of variable for binary classification
`NA_threshold`	a number in range of [0,1]. Threshold for deletion of features with a greater proportion of NAs than `NA_threshold`.
`cor_threshold`	a number used only for Spearman and Pearson correlation. Correlation threshold within features. If the correlation of 2 features is greater than `cor_threshold` the dependent feature is deleted.
`runs`	a number used only for randomForest and cforest. Amount of runs to gain higher robustness.
`selection`	a vector of length eight with TRUE or FALSE values. Selection of feature selection methods to be conducted.

Details

Following methods are provided in the `ensemble_fs`:

- Median: p-values from Wilcoxon signed-rank test (wilcox.test)
- Spearman: Spearman's rank correlation test arccording to Yu et al. (2004) (cor)
- Pearson: Pearson's product moment correlation test arccording to Yu et al. (2004) (cor)
- LogReg: beta-Values of logistic regression (glm)
- Accuracy//Error-rate randomForest: Error-rate-based variable importance measure embedded in randomForest according to Breiman (2001) (randomForest)
- Gini randomForest: Gini-index-based variable importance measure embedded in randomForest according to Breiman (2001) (randomForest)
- Error-rate cforest: Error-rate-based variable importance measure embedded in cforest according Strobl et al. (2009) (cforest)
- AUC cforest: AUC-based variable importance measure embedded in cforest according to Janitza et al. (2013) (cforest)

By the argument `selection` the user decides which feature selection methods are used in `ensemble_fs`. Default value is `selection = c(TRUE, TRUE, TRUE,TRUE, TRUE, TRUE, FALSE, FALSE)`, i.e., the function does not use either of the cforest variable importance measures. The maximum score for features depends on the input of `selection`. The scores are always divided through the amount of selected feature selection, respectively the amount of TRUEs.

Value

table of normalized importance values of class matrix (used methods as rows and features of the imported file as columns).

Author(s)

Ursula Neumann

References

- Yu, L. and Liu H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. 2004, 5:1205-1224.

- Breiman, L.: Random Forests, Machine Learning. 2001, 45(1): 5-32.

- Strobl, C., Malley, J. anpercentaged Tutz, G.: An Introduction to Recursive Partitioning: Rationale, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random forests. Psychological Methods. 2009, 14(4), 323–348.

- Janitza, S., Strobl, C. and Boulesteix AL.: An AUC-based Permutation Variable Importance Measure for Random Forests. BMC Bioinformatics.2013, 14, 119.

See Also

wilcox.test, randomForest, cforest, cor, glm

Examples

```
## Loading dataset in environment
data(efsdata)
## Generate a ranking based on importance (with default NA_threshold = 0.2,
## cor_threshold = 0.7, selection = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, FALSE))
efs <- ensemble_fs(efsdata, 5, runs=2)
```

Index