Package 'HybridFS'

October 11, 2017

Title A Hybrid Filter-Wrapper Feature Selection Method

Version 0.1.2

Description A hybrid method of feature selection which combines both filter and wrapper methods. The first level involves feature reduction based on some of the important filter methods while the second level involves feature subset selection as in a wrapper method. Comparative analysis with the existing feature selection packages shows this package results in higher classification accuracy, reduced processing time and improved data handling capacity.

Depends R (>= 3.4.1)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports FSelector, caTools, woeBinning, ROCR, InformationValue

NeedsCompilation no

Author Yamini Pandari [aut, cre],
Prashanth Thangavel [aut],
Hemanth Senthamaraikannan [aut],
Sivaranjani Jagadeeswaran [aut],
Thirumaalavan Elumalai [aut]

Maintainer Yamini Pandari <yamini.pandari@latentview.com>

Repository CRAN

Date/Publication 2017-10-11 17:18:57 UTC

R topics documented:

FinalBinnedData *Intermediate Binned Dataset*

Description

Retrieves the transformed dataset returned from HybridFS function

Usage

```
FinalBinnedData(input.df, target.var.name)
```

Arguments

input.df	Input data frame that contains the target variable and predictor variables with no missing values. Predictors can be either categorical or continuous.
target.var.name	Name of binary target variable. Target variables should be numeric with only two distinct values (0, 1)

Value

TransformedData	A data frame that contains the transformed dataset used in the HybridFS function

Examples

```
BinnedData=FinalBinnedData(input.df=validation,target.var.name="Survived")
```

HybridFS *A Hybrid Feature Selection Function*

Description

HybridFS is a combination of filter and wrapper methods which uses a set of statistical tests for feature selection. Primary level feature reduction involves filtering based on statistical test such as Chi-Square test of Independence, Information value(IV) and Entropy-related methods. Features filtered at this level are further fed into a classification algorithm and final features of the optimal model is returned along with the feature importance.

Usage

```
HybridFS(input.df, target.var.name)
```

Arguments

`input.df`	Input data frame that contains the target variable and predictor variables with no missing values. Predictors can be either categorical or continuous.Unique identifier,if present should be named "ID".
`target.var.name`	Name of binary target variable. Target variables should be integer with only two distinct values (0, 1)

Details

Binning of Continuous Predictors
Supervised Binning of continuous predictors reduces computational time, improves model performance and predictive power. Binning is implemented based on similar weight of evidence (WOE) values and information value (IV). Transformed dataset with binned copy of continuous variables is then fed into the Hybrid filter-Wrapper algorithm. Continuous features selected are returned as binned variables (e.g. average_volume is returned as average_volume.binned). To retrieve the transformed dataset, use FinalBinnedData() function.

Level1 Feature Reduction - Filter Method
Chi-Square test of Independence, Information value(IV) and Entropy-related methods such as Information Gain, Gain Ratio and Symmetrical Uncertainty are used to generate variable importance scores. Top n features are dynamically selected and different subsets are formed based on relative ranking from each of the filter methods.

Level2 Feature Reduction - Wrapper Method
Different subsets of variables from the first level are trained using a classification algorithm. Optimum probability cut-off for the target class is determined by the K-S Statistic. Combination of Area Under the Curve(AUC) and F-score (F1 score) are used as the benchmark metrics to measure the model performance. Best set of features with variable importance and rank from the optimal model is returned. Out-of-Sample Validation results are also displayed to understand the stability of the optimal model selected.

Value

An object of class FS, which is a list with the following components:

`imp.features`	A data frame of the selected features from the optimal model retuned with the relative rank.Variable importance plot for top 10 variables selected is displayed.Continuous features selected are returned as binned variables (e.g. average_volume is returned as average_volume.binned)
`model.perf`	Performance metrics of the optimal model such as F1 Score, Accuracy, Precision and Recall are returned

Note

Requires latest version of Java(8 and above)

Examples

```
FS=HybridFS(input.df=validation,target.var.name="Survived")
```

`imp.features` *Displaying the Selected Features*

Description

Displays the selected set of features generated via the HybridFS function

Usage

```
imp.features(FS)
```

Arguments

`FS`	FS is the final list returned from HybridFS function.It contains a data frame of the selected features with the relative rank and the optimal model performance.

Details

Displays the final selected variables with relative rank and an additional plot of Top 10 significant variables.Continuous features selected are returned as binned variables (e.g. average_volume is returned as average_volume.binned).To retrieve the transformed dataset,use FinalBinnedData function

Examples

```
FS=HybridFS(input.df=validation,target.var.name="Survived")
imp.features(FS)
```

`model.perf` *Displaying the Optimal Model Performance*

Description

Displays the performance metrics of the optimal model generated via the HybridFS function

Usage

```
model.perf(FS)
```

Arguments

`FS`	FS is the final list returned from HybridFS function.It contains a data frame of the selected features with the relative rank and the optimal model performance.

Details

Displays the performance metrics of the optimal model returned from the HybridFS function such as F1 Score, Accuracy, Precision and Recall.Performance metrics of the Validation dataset is also displayed to understand model stability

Examples

```
FS=HybridFS(input.df=validation,target.var.name="Survived")
model.perf(FS)
```

`validation` *Survival of passengers on the Titanic*

Description

Survival of passengers on the Titanic

Format

A data frame with columns:

ID Unique Passenger ID

Pclass 1st,2nd,3rd,Crew

Sex Male,Female

Name Passenger Name

SibSp # of siblings / spouses aboard the Titanic

Parch # of parents / children aboard the Titanic

Ticket Ticket number

Fare Passenger fare

Cabin Cabin number

Embarked Port of Embarkation

Survived No,Yes

Source

https://www.amstat.org/publications/jse/v3n3/datasets.dawson.html

Index