Package ‘LLM’

May 8, 2020

Title Logit Leaf Model Classifier for Binary Classification
Version 1.1.0
Date 2020-05-05
Author Arno De Caigny [aut, cre],
Kristof Coussement [aut],
Koen W. De Bock [aut]
Maintainer Arno De Caigny <a.de-caigny@ieseg.fr>
Description Fits the Logit Leaf Model, makes predictions and visualizes the output. (De Caigny et al., (2018) <DOI:10.1016/j.ejor.2018.02.009>.
Depends R (>= 4.0.0)
License GPL (>= 3)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Suggests mlbench
Imports partykit, stats, stringr, RWeka, survey, reghelper, scales
NeedsCompilation no
Repository CRAN
Date/Publication 2020-05-08 06:30:03 UTC

R topics documented:

llm .......................................................... 2
llm.cv .................................................. 3
predict.llm ............................................. 4
table.cat.llm.html ................................. 6
table.llm.html ....................................... 7

Index 9
llm  

*Create Logit Leaf Model*

**Description**

This function creates the logit leaf model. It takes a dataframe with numeric values as input and a corresponding vector with dependent values. Decision tree parameters threshold for pruning and number of observations per leaf can be set.

**Usage**

```r
llm(X, Y, threshold_pruning = 0.25, nbr_obs_leaf = 100)
```

**Arguments**

- `X`  
  Dataframe containing numerical independent variables.
- `Y`  
  Numerical vector of dependent variable. Currently only binary classification is supported.
- `threshold_pruning`  
  Set confidence threshold for pruning. Default 0.25.
- `nbr_obs_leaf`  
  The minimum number of observations in a leaf node. Default 100.

**Value**

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

- **Segment Rules**  
  The decision rules that define segments. Use `table.llm.html` to visualize.
- **Coefficients**  
  The segment specific logistic regression coefficients. Use `table.llm.html` to visualize.
- **Full decision tree for segmentation**  
  The raw decision tree. Use `table.llm.html` to visualize.
- **Observations per segment**  
  The raw decision tree. Use `table.llm.html` to visualize.
- **Incidence of dependent per segment**  
  The raw decision tree. Use `table.llm.html` to visualize.

**Author(s)**

Arno De Caigny, <a.de-caigny@ieseg.fr>, Kristof Coussement, <k.coussement@ieseg.fr>
and Koen W. De Bock, <kdebock@audencia.com>

**References**

## Load PimaIndiansDiabetes dataset from mlbench package

```r
if (requireNamespace("mlbench", quietly = TRUE)) {
  library("mlbench")
}
data("PimaIndiansDiabetes")
```

## Split in training and test (2/3 - 1/3)
```r
idtrain <- c(sample(1:768,512))
PimaTrain <- PimaIndiansDiabetes[idtrain,]
PimaTest <- PimaIndiansDiabetes[-idtrain,]
```

## Create the LLM
```r
Pima.llm <- llm(X = PimaTrain[,-c(9)], Y = PimaTrain$diabetes,
                 threshold_pruning = 0.25, nbr_obs_leaf = 100)
```

---

**llm.cv**  
*Runs v-fold cross validation with LLM*

**Description**

In v-fold cross validation, the data are divided into v subsets of approximately equal size. Subsequently, one of the v data parts is excluded while the remainder of the data is used to create a logitleafmodel object. Predictions are generated for the excluded data part. The process is repeated v times.

**Usage**

```r
llm.cv(X, Y, cv, threshold_pruning = 0.25, nbr_obs_leaf = 100)
```

**Arguments**

- **X**  
  Dataframe containing numerical independent variables.

- **Y**  
  Numerical vector of dependent variable. Currently only binary classification is supported.

- **cv**  
  An integer specifying the number of folds in the cross-validation.

- **threshold_pruning**  
  Set confidence threshold for pruning. Default 0.25.

- **nbr_obs_leaf**  
  The minimum number of observations in a leaf node. Default 100.
Value

An object of class llm.cv, which is a list with the following components:

- `foldpred`: a data frame with, per fold, predicted class membership probabilities for the left-out observations.
- `pred`: a data frame with predicted class membership probabilities.
- `foldclass`: a data frame with, per fold, predicted classes for the left-out observations.
- `class`: a data frame with the predicted classes.
- `conf`: the confusion matrix which compares the real versus the predicted class memberships based on the class object.

Author(s)

Arno De Caigny, <a.de-caigny@ieseg.fr>, Kristof Coussement, <k.coussement@ieseg.fr> and Koen W. De Bock, <kdebock@audencia.com>

References


See Also

predict.llm, table.llm.html, llm

Examples

```r
## Load PimaIndiansDiabetes dataset from mlbench package
if (requireNamespace("mlbench", quietly = TRUE)) {
  library("mlbench")
}
data("PimaIndiansDiabetes")
## Create the LLM with 5-cv
Pima.llm <- llm.cv(X = PimaIndiansDiabetes[-c(9)], Y = PimaIndiansDiabetes$diabetes, cv=5, threshold_pruning = 0.25, nbr_obs_leaf = 100)
```

---

**Description**

This function creates a prediction for an object of class logitleafmodel. It assumes a dataframe with numeric values as input and an object of class logitleafmodel, which is the result of the `llm` function. Currently only binary classification is supported.
predict.llm

Usage

## S3 method for class 'llm'
predict(object, X, ...)

Arguments

object An object of class logitleafmodel, as that created by the function llm.
X Dataframe containing numerical independent variables.
... further arguments passed to or from other methods.

Value

Returns a dataframe containing a probability for every instance based on the LLM model. Optional rownumbers can be added.

Author(s)

Arno De Caigny, <a.de-caigny@ieseg.fr>, Kristof Coussement, <k.coussement@ieseg.fr> and Koen W. De Bock, <kdebock@audencia.com>

References


See Also

llm, table.llm.html, lllm.cv

Examples

## Load PimaIndiansDiabetes dataset from mlbench package
if (requireNamespace("mlbench", quietly = TRUE)) {
  library("mlbench")
}
data("PimaIndiansDiabetes")
## Split in training and test (2/3 - 1/3)
idtrain <- c(sample(1:768,512))
PimaTrain <- PimaIndiansDiabetes[idtrain,]
PimaTest <- PimaIndiansDiabetes[-idtrain,]
## Create the LLM
Pima.llm <- llm(X = PimaTrain[,-c(9)], Y = PimaTrain$diabetes,
  threshold_pruning = 0.25, nbr_obs_leaf = 100)
## Use the model on the test dataset to make a prediction
PimaPrediction <- predict.llm(object = Pima.llm, X = PimaTest[,-c(9)])
## Optionally add the dependent to calculate performance statistics such as AUC
# PimaPrediction <- cbind(PimaPrediction, "diabetes" = PimaTest[, "diabetes"])}
Create the HTML code for Logit Leaf Model visualization

Description

This function generates HTML code for a visualization of the logit leaf model based on the variable importance per variable category.

Usage

```r
table.cat.llm.html(
  object,
  category_var_df,
  headertext = "The Logit Leaf Model",
  footertext = "A table footer comment",
  roundingnumbers = 2,
  methodvarimp = "Coef"
)
```

Arguments

- `object`: An object of class logitleafmodel, as that created by the function llm.
- `category_var_df`: dataframe containing a column called "iv" with the independent variables and a column called "cat" with the variable category names that is associated with every iv
- `headertext`: Allows to provide the table with a header.
- `footertext`: Allows to provide the table with a custom footer.
- `roundingnumbers`: An integer stating the number of decimals in the visualization.
- `methodvarimp`: Allows to determine the method to calculate the variable importance. There are 4 options: 1/ Variable coefficient (method = 'Coef') 2/ Standardized beta ('Beta') 3/ Wald statistic ('Wald') 4/ Likelihood Rate Test ('LRT')

Value

Generates HTML code for a visualization.

Author(s)

Arno De Caigny, <a.de-caigny@ieseg.fr>, Kristof Coussement, <k.coussement@ieseg.fr> and Koen W. De Bock, <kdebock@audencia.com>

References

### See Also

`predict.llm, llm, llm.cv`

### Examples

```r
## Load PimaIndiansDiabetes dataset from mlbench package
if (requireNamespace("mlbench", quietly = TRUE)) {
  library("mlbench")
}
data("PimaIndiansDiabetes")
## Split in training and test (2/3 - 1/3)
idtrain <- c(sample(1:768,512))
PimaTrain <- PimaIndiansDiabetes[idtrain,]
Pimatest <- PimaIndiansDiabetes[-idtrain,]
## Create the LLM
Pima.llm <- llm(X = PimaTrain[,-c(9)],Y = PimaTrain$diabetes,
                threshold_pruning = 0.25,nbr_obs_leaf = 100)
## Define the variable categories (note: the categories are only created for demonstration)
var_cat_df <- as.data.frame(cbind(names(PimaTrain[,-c(9)]),
                              c("cat_a","cat_a","cat_a","cat_a","cat_b","cat_b","cat_b","cat_b"),
                              stringsAsFactors = FALSE)
                           names(var_cat_df) <- c("iv", "cat")
## Save the output of the model to a html file
Pima.Viz <- table.cat.llm.html(object = Pima.llm,
                                category_var_df= var_cat_df,
                                headertext = "This is an example of the LLM model",
                                footertext = "Enjoy the package!"
)
## Optionally write it to your working directory
# write(Pima.Viz, "Visualization_LLM_on_PimaIndiansDiabetes.html")
```

---

**Description**

This function generates HTML code for a visualization of the logit leaf model.

**Usage**

```r
table.llm.html(
  object,
  headertext = "The Logit Leaf Model",
  footertext = "A table footer comment",
  roundingnumbers = 2
)
```

**Arguments**

- `object`: An object of class logitleafmodel, as that created by the function `llm`.
- `headertext`: Allows to provide the table with a header.
footertext Allows to provide the table with a custom footer.
roundingnumbers An integer stating the number of decimals in the visualization.

Value
Generates HTML code for a visualization.

Author(s)
Arno De Caigny, <a.de-caigny@ieseg.fr>, Kristof Coussement, <k.coussement@ieseg.fr> and Koen W. De Bock, <kdebock@audencia.com>

References

See Also
predict.llm, llm, llm.cv

Examples
```r
## Load PimaIndiansDiabetes dataset from mlbench package
if (requireNamespace("mlbench", quietly = TRUE)) {
  library("mlbench")
}
data("PimaIndiansDiabetes")
## Split in training and test (2/3 - 1/3)
idtrain <- c(sample(1:768,512))
PimaTrain <- PimaIndiansDiabetes[idtrain,]
Pimatest <- PimaIndiansDiabetes[-idtrain,]
## Create the LLM
Pima.llm <- llm(X = PimaTrain[, -c(9)], Y = PimaTrain$diabetes,
  threshold_pruning = 0.25, nbr_obs_leaf = 100)
## Save the output of the model to a html file
Pima.Viz <- table.llm.html(object = Pima.llm, headertext = "This is an example of the LLM model",
  footertext = "Enjoy the package!")
## Optionally write it to your working directory
# write(Pima.Viz, "Visualization_LLM_on_PimaIndiansDiabetes.html")
```
Index

llm, 2, 4, 5, 7, 8
llm.cv, 3, 3, 5, 7, 8
predict.llm, 3, 4, 4, 7, 8

table.cat.llm.html, 6
table.llm.html, 2–5, 7