Package ‘pomodoro’

March 26, 2022

Type Package
Title Predictive Power of Linear and Tree Modeling
Version 3.8.0
Author Seyma Kalay <seymakalay@hotmail.com>
Maintainer Seyma Kalay <seymakalay@hotmail.com>
Description Runs generalized and multinominal logistic (GLM and MLM) models, as well as random forest (RF), Bagging (BAG), and Boosting (BOOST). This package prints out to predictive outcomes easy for the selected data and data splits.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.2
URL https://github.com/seymakalay/pomodoro,
       https://seymakalay.github.io/pomodoro/
BugReports https://github.com/seymakalay/pomodoro/issues
Suggests knitr, rmarkdown
VignetteBuilder knitr
Imports tibble, caret, gbm, stats, randomForest, pROC, ipred
Depends R (>= 2.10)
NeedsCompilation no
Repository CRAN
Date/Publication 2022-03-26 12:10:02 UTC

R topics documented:

   BAG_Model .................................................. 2
   Combined_Performance ..................................... 3
   Estimate_Models .......................................... 3
   GBM_Model ................................................. 4
**BAG_Model**

---

**Description**

Bagging Model

**Usage**

BAG_Model(Data, xvar, yvar)

**Arguments**

- **Data**
  - The name of the Dataset.
- **xvar**
  - X variables.
- **yvar**
  - Y variable.

**Details**

Decision trees suffer from high variance (If we split the training data-set randomly into two parts and set a decision tree to both parts, the results might be quite different). Bagging is an ensemble procedure which reduces the variance and increases the prediction accuracy of a statistical learning method by considering many training sets ($\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_B(x)$) from the population. Since we can not have multiple training-sets, from a single training data-set, we can generate $B$ different bootstrapped training data-sets ($f^*_1(x), f^*_2(x), \ldots, f^*_B(x)$) by each $B$ trees and take a majority vote. Therefore, bagging for classification problem defined as

$$\hat{f}(x) = \arg \max_k f^*_b(x)$$

**Value**

The output from BAG_Model.

**Examples**

```r
yvar <- c("Loan.Type")
sample_data <- sample_data[c(1:750),]
xvar <- c("sex", "married", "age", "havejob", "educ", "political.afl", "rural", "region", "fin.intermediaries", "fin.knowledge", "income")
BchMk.BAG <- BAG_Model(sample_data, c(xvar, "networth"), yvar )
BchMk.BAG$Roc$auc
```
**Combined_Performance**  

**Combined Performance of the Data Splits**

**Description**

Combined Performance of the Data Splits

**Usage**

Combined_Performance(Sub.Est.Mdls)

**Arguments**

Sub.Est.Mdls is the total performance of exog.

**Value**

The output from `Combined_Performance`.

**Examples**

```r
sample_data <- sample_data[c(1:750),]  
yvar <- c("Loan.Type")  
xvar <- c("sex", "married", "age", "havejob", "educ", "political.afl",  
"rural", "region", "fin.intermediaries", "fin.knowledge", "income")  
CCP.RF <- Estimate_Models(sample_data, yvar, xvec = xvar, exog = "political.afl",  
xadd = c("networth", "networth_homequity", "liquid.assets"),  
type = "RF", dnames = c("0","1"))  
Sub.CCP.RF <- list (Mdl.1 = CCP.RF$EstMdl$"D.1+networth",  
Mdl.0 = CCP.RF$EstMdl$"D.0+networth")  
CCP.NoCCP.RF <- Combined_Performance (Sub.CCP.RF)
```

---

**Estimate_Models**  

**Results of the Each Data and Data Splits**

**Description**

Results of the Each Data and Data Splits

**Usage**

`Estimate_Models(DataSet, yvar, exog = NULL, xvec, xadd, type, dnames)`

---

**Combined_Performance**  

**Combined Performance of the Data Splits**

**Description**

Combined Performance of the Data Splits

**Usage**

Combined_Performance(Sub.Est.Mdls)

**Arguments**

Sub.Est.Mdls is the total performance of exog.

**Value**

The output from `Combined_Performance`.

**Examples**

```r
sample_data <- sample_data[c(1:750),]  
yvar <- c("Loan.Type")  
xvar <- c("sex", "married", "age", "havejob", "educ", "political.afl",  
"rural", "region", "fin.intermediaries", "fin.knowledge", "income")  
CCP.RF <- Estimate_Models(sample_data, yvar, xvec = xvar, exog = "political.afl",  
xadd = c("networth", "networth_homequity", "liquid.assets"),  
type = "RF", dnames = c("0","1"))  
Sub.CCP.RF <- list (Mdl.1 = CCP.RF$EstMdl$"D.1+networth",  
Mdl.0 = CCP.RF$EstMdl$"D.0+networth")  
CCP.NoCCP.RF <- Combined_Performance (Sub.CCP.RF)
```
**Arguments**

- **DataSet** The name of the Dataset.
- **yvar** Y variable.
- **exog** is a vector to be subtract from the calculation.
- **xvec** is a vector of the variables to be used.
- **xadd** is an additional vector to be used.
- **type** can be RF, GLM, MLM, BAG, and GBM.
- **dnames** is the unique values of exog.

**Value**

The output from `Estimate_Models`.

**Examples**

```r
sample_data <- sample_data[c(1:750),]
m2.xvar0 <- c("sex","married","age","havejob","educ","rural","region","income")
CCP.RF <- Estimate_Models(sample_data, yvar = c("Loan.Type"),
exog = "political.afl", xvec = m2.xvar0,
xadd = "networth", type = "RF", dnames = c("0","1"))
```

---

**GBM_Model**  
*Gradient Boosting Model*

**Description**

Gradient Boosting Model

**Usage**

`GBM_Model(Data, xvar, yvar)`

**Arguments**

- **Data** The name of the Dataset.
- **xvar** X variables.
- **yvar** Y variable.
GLM_Model

Details

Unlike bagging trees, boosting does not use bootstrap sampling, rather each tree is fit using information from previous trees. An event probability of stochastic gradient boosting model is given by

\[ \hat{\pi}_i = \frac{1}{1 + \exp(-f(x))} \]

where \( f(x) \) is in the range of \([-\infty, \infty] \) and its initial estimate of the model is

\[ f_i^{(0)} = \log\left( \frac{\hat{\pi}_i}{1 - \hat{\pi}_i} \right) \]

where \( \hat{\pi} \) is the estimated sample proportion of a single class from the training set.

Value

The output from GBM_Model.

Examples

```r
yvar <- c("Loan.Type")
sample_data <- sample_data[c(1:120),]
xvar <- c("sex", "married", "age", "havejob", "educ", "political.afl", "rural", "region", "fin.intermediaries", "fin.knowledge", "income")
BchMk.GBM <- GBM_Model(sample_data, c(xvar, "networth"), yvar )
BchMk.GBM$finalModel
BchMk.GBM$Roc$auc
```

---

GLM_Model

Generalized Linear Model

Description

Generalized Linear Model

Usage

```
GLM_Model(Data, xvar, yvar)
```

Arguments

- **Data** The name of the Dataset.
- **xvar** X variables.
- **yvar** Y variable.
Details

Let y be a vector of response variable of accessing credit for each applicant n, such that \( y_i = 1 \) if the applicant-i has access to credit, and zero otherwise. Furthermore, let let \( x = x_{ij} \), where \( i = 1, \ldots, n \) and \( j = 1, \ldots, p \) characteristics of the applicants. The log-odds can be define as:

\[
\log\left( \frac{\pi_i}{1 - \pi_i} \right) = \beta_0 + \sum_{i=1}^{p} \beta_i x_{ij}
\]

\( \beta_0 \) is the intercept, \( \beta = (\beta_1, \ldots, \beta_p) \) is a \( p \times 1 \) vector of coefficients and \( x_{ij} \) is the \( i_{th} \) row of \( x \).

Value

The output from \texttt{GLM.Model}.

Examples

\[
yvar <- \text{c("multi.level")}
\text{sample.data} <- \text{sample.data[c(1:750),]}
xvar <- \text{c("sex", "married", "age", "havejob", "educ", "political.afl", "rural", "region", "fin.intermediaries", "fin.knowledge", "income")}
\text{BchMk.GLM} <- \text{GLM.Model(sample.data, c(xvar, "networth"), yvar )}
\text{BchMk.GLM$finalModel}
\text{BchMk.GLM$Roc$auc}
\]

---

### MLM.Model

**Multinominal Logistic Model**

**Description**

Multinominal Logistic Model

**Usage**

\[ \text{MLM.Model(Data, xvar, yvar)} \]

**Arguments**

- **Data** The name of the Dataset.
- **xvar** X variables.
- **yvar** Y variable.
Details

Multi-nominal model is the generalized form of generalized logistic model and can be defined as

\[ \pi_i^h = P(y_i^h = 1|x_i^h) \]

where \( h \) presents the class labels ("1-of-\( h \") on the basis of an input vector \( x_j \), in our case \( x_j \) is loan types ("Formal Loan", "Informal Loan", "Both Loan", and "No Loan"). Furthermore, \( y_i^h = 1 \) if the weight \( w \) of \( x_j \) corresponds to belong a class and \( y_i^h = 0 \) otherwise. For \( i \in 1, \ldots, h \) and the weight vectors \( w^i \) corresponds to class \( i \).

We set \( w^h = 0 \) and the parameters to be learned are the weight vectors \( w^i \) for \( i \in 1, \ldots, h - 1 \). And the class probabilities must satisfy

\[ \sum_{i=1}^{h} P(y_i^h = 1|x_i^h, w) = 1. \]

Value

The output from MLM_Model.

Examples

```r
yvar <- c("Loan.Type")
sample_data <- sample_data[1:750,]
xvar <- c("sex", "married", "age", "havejob", "educ", "political.afl", "rural", "region", "fin.intermediaries", "fin.knowledge", "income")
BchMk.MLM <- MLM_Model(sample_data, c(xvar, "networth"), yvar )
BchMk.MLM$finalModel
BchMk.MLM$Roc$auc
```

---

**RF_Model**

**Random Forest**

Description

Random Forest

Usage

`RF_Model(Data, xvar, yvar)`

Arguments

- **Data**: The name of the Dataset.
- **xvar**: X variables.
- **yvar**: Y variable.
Details

Rather than considering the random sample of \( m \) predictors from the total of \( p \) predictors in each split, random forest does not consider a majority of the \( p \) predictors, and considers in each split a fresh sample of \( m_{\text{try}} \) which we usually set to \( m_{\text{try}} \approx \sqrt{p} \). Random forests which de-correlate the trees by considering \( m_{\text{try}} \approx \sqrt{p} \) show an improvement over bagged trees \( m = p \).

Value

The output from \texttt{RF_Model}.

Examples

```r
sample_data <- sample_data[c(1:750),]
yvar <- c("Loan.Type")
xvar <- c("sex", "married", "age", "havejob", "educ", "political.afl", "rural", "region", "fin.intermediaries", "fin.knowledge", "income")
BchMk.RF <- RF_Model(sample_data, c(xvar, "networth"), yvar )
BchMk.RF
```

---

**sample_data**

Sample data for analysis. A dataset containing information of access to credit.

Description

Sample data for analysis.
A dataset containing information of access to credit.

Usage

```r
sample_data
```

Format

A \texttt{data.frame} with 53940 rows and 10 variables:

- **x1** hhid, household id number
- **x2** swgt, survey weight
- **x3** region, 3 factor level, west, east, and center
- **x4** No.Loan, if the household has no loan
- **x5** Formal, if the household has formal loan
- **x6** Both, if the household has both loan
- **x7** Informal, if the household has informal loan
- **x8** sex, if the household has male
- **y1** Loan.Type, 4 factor level type of the loan
- **y2** multi.level, 2 factor level if the household has access to loan or not ...
Index

* datasets
  sample_data, 8

BAG_Model, 2, 2

Combined_Performance, 3, 3

Estimate_Models, 3, 4

GBM_Model, 4, 5
GLM_Model, 5, 6

MLM_Model, 6, 7
RF_Model, 7, 8

sample_data, 8