

Package ‘Rprofet’

April 14, 2018

Type Package

Title WOE Transformation and Scorecard Builder

Version 2.2.0

Maintainer Thomas Brandenburger <thomas.brandenburger@sdstate.edu>

Description

Performs all steps in the credit scoring process. This package allows the user to follow all the necessary steps for building an effective scorecard. It provides the user functions for coarse binning of variables, Weights of Evidence (WOE) transformation, variable clustering, custom binning, visualization, and scaling of logistic regression coefficients. The results will generate a scorecard that can be used as an effective credit scoring tool to evaluate risk. For complete details on the credit scoring process, see Siddiqi (2005, ISBN:047175451X).

License GPL-3

Imports binr, ClustOfVar, ggplot2, gridExtra, plyr, sqldf, stringr

Suggests ISLR, MASS

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Thomas Brandenburger [cre],

Eric Stratman [aut],

Riaz Khan [aut],

Allison Lempola [aut]

Repository CRAN

Date/Publication 2018-04-14 14:53:09 UTC

R topics documented:

BinProfet	2
ScorecardProfet	3
WOEClust_hclust	4
WOEClust_kmeans	5

WOEplotter	6
WOEProfet	7
WOE_custom	8
WOE_customFactor	9

Index	11
--------------	-----------

BinProfet	<i>Binning Variable(s)</i>
-----------	----------------------------

Description

Function that bins selected variable(s) and returns a dataframe with binned values. Uses greedy binning algorithm to perform coarse binning of selected variable(s).

Usage

```
BinProfet(dat, id, target, varcol, minimum.cat = 4, num.bins = 10,
  min.pts.bin = 25, bracket = "left", special.values = NULL)
```

Arguments

<code>dat</code>	Dataframe of that contains ID, binary target and variables to be binned.
<code>id</code>	ID variable. See 'Details'.
<code>target</code>	The binary target/response variable for WOE. See 'Details'.
<code>varcol</code>	Vector of variables to be binned.
<code>minimum.cat</code>	Minimum number of bins.
<code>num.bins</code>	Target number of bins. Overridden by the number of levels if <code>varcol</code> is factor.
<code>min.pts.bin</code>	Minimum number of observations in a bin.
<code>bracket</code>	Indicating if the intervals should be closed on the right or left. Options include left and right.
<code>special.values</code>	A vector of values that should have their own bin. See 'Details'.

Details

The `id` and the `target` variables must be provided. Works for numeric, factor and binary target. To build a scorecard, a binary target is required.

Actual number of bins exceeds `num.bins` if `special.values` specified.

Value

A dataframe containing the ID, target, and binned variable(s) with corresponding binned vlues.

Examples

```

mydata <- ISLR::Default
head(mydata)

mydata$ID <- seq(1:nrow(mydata)) ## make an ID variable
mydata$default <- ifelse(mydata$default=="Yes", 1, 0) ## target coded with 1, 0

## bin balance and income
binned1 <- BinProfet(mydata, id="ID", target="default",
                    varcol = c("balance", "income"), num.bins = 5)
head(binned1)

## bin categorical variable-----
binned2 <- BinProfet(mydata, id="ID", target="default",
                    varcol = "student", num.bins = 5)
head(binned2)
summary(binned2$student_Bins) ## num.bins overridden

```

ScorecardProfet

Scorecard Builder

Description

Function that fits a logistic regression models and scores points for each bin and calculates observations' total score.

Usage

```
ScorecardProfet(object, target, id, varcol, PDO = 100, BaseOdds = 10,
               BasePts = 1000, reverse = FALSE)
```

Arguments

object	A WOEPProfet object containing dataframes with binned and WOE values.
target	A binary target variable.
id	ID variable.
varcol	Vector of WOE variables to be used in the logistic regression model.
PDO	Points to Double Odds.
BaseOdds	Base Odds.
BasePts	Base Points.
reverse	Logical. If true, higher points corresponds to a lower probability of being target.

Value

A list with the following components.

Scorecard	The actual scorecard model. Table with the attribute bins and their corresponding WOE values and the points assigned to each bin.
Results	Dataframe with the bin, WOE value, and points assigned to each attribute and the total score for each observation.
GLMSummary	The summary of the logistic regression model fitted to build the scorecard.

Examples

```
mydata <- ISLR::Default

mydata$ID = seq(1:nrow(mydata)) ## make the ID variable
mydata$default<-ifelse(mydata$default=="Yes",1,0) ## Creating numeric binary target variable

binned <- BinProfet(mydata, id= "ID", target= "default", num.bins = 5) ## Binning variables

WOE_dat <- WOEPProfet(binned, "ID","default", 3:5) ## WOE transformation of bins

Score_dat <- ScorecardProfet(WOE_dat, target="default",
                             id= "ID", PDO = 50, BaseOdds = 10, BasePts = 1000, reverse = TRUE)

Score_dat$GLMSummary
head(Score_dat$Scorecard) ## Less points means more likely to default
```

WOEClust_hclust *Variable Clustering*

Description

Function that implements hierarchical clustering on the variables to be used as a form of variable selection.

Usage

```
WOEClust_hclust(object, id, target, num_clusts, method = "ward.D")
```

Arguments

object	A WOEPProfet object containing dataframes with binned and WOE values.
id	ID variable.
target	A binary target variable.
num_clusts	Number of desired clusters.
method	Clustering method to be used. This should be one of "ward.D", "ward.D2", "single", "average", "mcquitty", "median", or "centroid".

Value

A dataframe with the name of all the variables to be clustered, the corresponding cluster and the information value for each variable.

Examples

```
mydata <- ISLR::Default
mydata$ID = seq(1:nrow(mydata)) ## make the ID variable
mydata$default<-ifelse(mydata$default=="Yes",1,0) ## Creating numeric binary target variable

## create two new variables from bivariate normal
sigma <- matrix(c(45000,-3000,-3000, 55000), nrow = 2)
set.seed(10)
newvars <- MASS::mvrnorm(nrow(mydata),
                        mu=c(1000,200), Sigma=sigma)

mydata$newvar1 <- newvars[,1]
mydata$newvar2 <- newvars[,2]

binned <- BinProfet(mydata, id= "ID", target= "default", num.bins = 5) ## Binning variables

WOE_dat <- WOEPProfet(binned, "ID","default")

## Cluster variables by WOEClust_hclust
clusters <- WOEClust_hclust(WOE_dat, id="ID", target="default", num_clusts=3)
clusters
```

WOEClust_kmeans

Kmeans Variable Clustering

Description

Function that implements kmeans variable clustering to be used as a form of variable selection.

Usage

```
WOEClust_kmeans(object, id, target, num_clusts)
```

Arguments

object	A WOEPProfet object containing dataframes with binned and WOE values.
id	ID variable.
target	A binary target variable.
num_clusts	Number of desired clusters.

Value

A dataframe with the name of all the variables to be clustered, the corresponding cluster and the information value for each variable.

Examples

```
mydata <- ISLR::Default
mydata$ID = seq(1:nrow(mydata)) ## make the ID variable
mydata$default<-ifelse(mydata$default=="Yes",1,0) ## Creating numeric binary target variable

## create two new variables from bivariate normal
sigma <- matrix(c(45000,-3000,-3000, 55000), nrow = 2)
set.seed(10)
newvars <- MASS::mvrnorm(nrow(mydata),
                          mu=c(1000,200), Sigma=sigma)

mydata$newvar1 <- newvars[,1]
mydata$newvar2 <- newvars[,2]

binned <- BinProfet(mydata, id= "ID", target= "default", num.bins = 5) ## Binning variables

WOE_dat <- WOEPofet(binned, "ID","default")

## Cluster variables by WOEClust_kmeans
clusters <- WOEClust_kmeans(WOE_dat, id="ID", target="default", num_clusts=3)
clusters
```

WOEplotter

Visualizing WOE and Target Rates

Description

Function generating three plots: WOE value for each bin, target rate for each bin, and the frequency for each bin

Usage

```
WOEplotter(dataset, target, var, color = "#0066CC")
```

Arguments

dataset	Dataframe containing binned values and a binary target variable.
target	A numeric binary target variable.
var	The desired WOE binned attribute to visualize.
color	A hexadecimal value representing a specific color.

Details

A list of the hexadecimal colors can be found at this link <http://www.sthda.com/sthda/RDoc/images/hextable.gif>

Examples

```
mydata <- ISLR::Default

mydata$ID = seq(1:nrow(mydata)) ## make the ID variable
mydata$default<-ifelse(mydata$default=="Yes",1,0) ## Creating numeric binary target variable

binned <- BinProfet(mydata, id= "ID", target= "default", num.bins = 5) ## Binning variables

WOEplotter(binned, target= "default", var= "income_Bins")

##--Changing Colors-----
WOEplotter(binned, target= "default", var= "income_Bins", color = "#33FF33")
```

 WOEProfet

WOE Transformation

Description

Function that calculates the WOE for each bin and the information value for each variable.

Usage

```
WOEProfet(dat, id, target, varcol)
```

Arguments

dat	Dataframe of binned variables.
id	ID variable.
target	A binary target variable.
varcol	Vector of variables to have WOE transformation.

Details

The id and the target variables must be provided. The target variable must be a numeric binary variable.

Value

A list with the following components.

BinWOE	Dataframe with the binned variables and their WOE values.
WOE	Dataframe with the WOE values.
IV	Each attribute and their associated information values.
vars	A list containing the different WOE values for each attribute.

Examples

```

mydata <- ISLR::Default

mydata$ID = seq(1:nrow(mydata)) ## make the ID variable
mydata$default<-ifelse(mydata$default=="Yes",1,0) ## Creating numeric binary target variable

binned <- BinProfet(mydata, id= "ID", target= "default", num.bins = 5) ## Binning variables

WOE_dat <- WOEPrefet(binned, "ID","default", 3:5)

head(WOE_dat$BinWOE)
head(WOE_dat$WOE)
WOE_dat$IV
head(WOE_dat$vars$income)

```

WOE_custom

Custom Binning Numeric Variables

Description

Function that bins a numeric variable based on user inputted breaks, plots the information on the new bins, and returns a vector of the newly binned values

Usage

```

WOE_custom(dataset, var, target, breaks, right_bracket = F,
           color = "#0066CC")

```

Arguments

dataset	Dataframe containing the target variable and desired numeric variables to be binned.
var	A specific numeric attribute to be binned. Must be specified.
target	A binary target variable. Must be specified.
breaks	A vector of breakpoints for the desired bins. Must be specified.
right_bracket	Logical. Specifying whether the intervals are closed on the right or the left.
color	A hexadecimal value representing a specific color.

Value

A vector containing the newly binned values. Generates three barplots displaying the worth of evidence (WoE), target rate and frequency across all bins. The bars are arranged in an ascending order of WoE.

Examples

```

mydata <- ISLR::Default
mydata$default <- ifelse(mydata$default=="Yes", 1, 0) ## target coded with 1, 0
WC_1 <- WOE_custom(dataset=mydata, var="balance", target = "default",
                   breaks=seq(0,3000,1000))
levels(factor(WC_1))
WC_2 <- WOE_custom(dataset=mydata, var="income", target = "default",
                   breaks=seq(0,75000, 15000))
levels(factor(WC_2))

```

WOE_customFactor	<i>Custom Binning Factor Variables</i>
------------------	--

Description

Function that bins a factor variable based on user inputted factor levels, plots the information on the new bins, and returns a vector of the newly binned values

Usage

```
WOE_customFactor(dataset, var, target, new_levels, color = "#0066CC")
```

Arguments

dataset	Dataframe containing the target variable and desired factor variables to be binned.
var	A specific factor attribute to be binned.
target	A binary target variable. Must be specified.
new_levels	A vector the same length as the number of levels for the categorical variable containing the new factor levels. Must be specified.
color	A hexadecimal value representing a specific color.

Value

A vector containing the newly binned values. Generates three barplots displaying the worth of evidence (WoE), target rate and frequency across all bins. The bars are arranged in an ascending order of WoE.

Examples

```

mydata <- ISLR::Default
mydata$default <- ifelse(mydata$default=="Yes", 1, 0) ## target coded with 1, 0
## WOE_customFactor
custom1 <- WOE_customFactor(mydata, var="student", target="default",
                            new_levels=c("Student : No", "Student : Yes"))
levels(custom1)
## -----

```

```
mydata$balance_cat <- cut(mydata$balance, breaks = c(-1,400,800,1200,1600,2000,2400,2800),  
  labels = c("Very-Low","Low","Med-Low","Med","Med-High","High","Very-High"))  
custom2 <- WOE_customFactor(mydata, var="balance_cat", target="default",  
  new_levels=c(1,1,2,2,2,3,3))  
levels(custom2)
```

Index

BinProfet, [2](#)

ScorecardProfet, [3](#)

WOE_custom, [8](#)

WOE_customFactor, [9](#)

WOEClust_hclust, [4](#)

WOEClust_kmeans, [5](#)

WOEplotter, [6](#)

WOEProfet, [7](#)