Package ‘Rprofet’

April 1, 2020

Type Package

Title WOE Transformation and Scorecard Builder

Version 2.2.1

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 7.1.0

NeedsCompilation no

Author Thomas Brandenburger [cre],
Eric Stratman [aut],
Riaz Khan [aut],
Allison Lempola [aut]

Repository CRAN

Date/Publication 2020-04-01 15:50:02 UTC

R topics documented:

  BinProfet ................................................................. 2
  ScorecardProfet ......................................................... 3
  WOEClust_hclust ....................................................... 4
  WOEClust_kmeans ...................................................... 5
BinProfet

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

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

### Examples

```r
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 overriden
```

### ScorecardProfet

#### Description

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

#### Usage

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

#### Arguments

- **object**: A WOEProfet 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

```r
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 <- WOEProfet(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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A WOEProfet object containing dataframes with binned and WOE values.</td>
</tr>
<tr>
<td>id</td>
<td>ID variable.</td>
</tr>
<tr>
<td>target</td>
<td>A binary target variable.</td>
</tr>
<tr>
<td>num_clusts</td>
<td>Number of desired clusters.</td>
</tr>
<tr>
<td>method</td>
<td>Clustering method to be used. This should be one of &quot;ward.D&quot;, &quot;ward.D2&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;single&quot;, &quot;average&quot;, &quot;mcquitty&quot;, &quot;median&quot;, or &quot;centroid&quot;.</td>
</tr>
</tbody>
</table>

 Value

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

 Examples

```r
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 <- WOEProfet(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

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

- object: A WOEProfet 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

```r
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 <- WOEProfet(binned, "ID", "default")

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

Description

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

Usage

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

**Description**

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

**Usage**

```r
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.

---

### 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](http://www.sthda.com/sthda/RDoc/images/hextable.gif)

**Examples**

```r
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")
```
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

```r
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 <- WOEProfet(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

```r
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.
WOE_customFactor

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))
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