Package: mob (via r-universe)

August 23, 2024

Title Monotonic Optimal Binning

Version 0.4.2

Description Generate the monotonic binning and perform the woe (weight of evidence) transformation for the logistic regression used in the consumer credit scorecard development. The woe transformation is a piecewise transformation that is linear to the log odds. For a numeric variable, all of its monotonic functional transformations will converge to the same woe transformation.

License GPL (>= 2)

URL `https://github.com/statcompute/mob`

Author WenSui Liu

Maintainer WenSui Liu <`liuwensui@gmail.com`>

Depends R (>= 3.3.3)

Imports stats, gbm, Rborist

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Date/Publication 2021-07-31 04:30:07 UTC

Repository https://statcompute.r-universe.dev

RemoteUrl https://github.com/cran/mob

RemoteRef HEAD

RemoteSha 1e5cb564adeeb6643fe23dd75057a3d551f45f40

Contents

`arb_bin` *Monotonic binning based on decision tree model*

Description

The function `arb_bin` implements the monotonic binning based on the decision tree.

Usage

```
arb_bin(x, y)
```

Arguments

`x`	A numeric vector
`y`	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
arb_bin(hmeq$DEROG, hmeq$BAD)
```

`bad_bin` *Monotonic binning by quantile with cases Y = 1*

Description

The function `bad_bin` implements the quantile-based monotonic binning by the iterative discretization based on cases with Y = 1.

Usage

```
bad_bin(x, y)
```

Arguments

`x`	A numeric vector
`y`	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
bad_bin(hmeq$DEROG, hmeq$BAD)
```

`batch_bin` *Apply monotonic binning to all vectors in dataframe*

Description

The function `batch_bin` applies multiple binning algorithms in batch to each vector in the dataframe.

Usage

```
batch_bin(y, xs, method = 1)
```

Arguments

`y`	A numeric vector with 0/1 binary values.
`xs`	A dataframe with numeric vectors to discretize.
`method`	A integer from 1 to 7 referring to implementations below: 1. Implementation of iso_bin() 2. Implementation of qtl_bin() 3. Implementation of bad_bin() 4. Implementation of rng_bin() 5. Implementation of gbm_bin() 6. Implementation of kmn_bin() 7. Implementation of arb_bin()

Value

A list of binning outcomes with 2 dataframes: bin_sum: A dataframe of binning summary. bin_out: A list of binning output from binning functions, e.g. qtl_bin().

Examples

```
data(hmeq)
batch_bin(hmeq$BAD, hmeq[, c('DEROG', 'DELINQ')])
```

---

batch_woe *Apply WoE transformations to vectors in dataframe*

Description

The function `batch_woe` applies WoE transformations to vectors in the dataframe.

Usage

```
batch_woe(xs, bin_out)
```

Arguments

`xs`	A dataframe with numeric vectors to discretize.
`bin_out`	A binning output from the function batch_bin().

Value

A dataframe with identical headers as the input xs. However, values of each variable have been transformed to WoE values.

Examples

```
data(hmeq)
bin_out <- batch_bin(hmeq$BAD, hmeq[, c('DEROG', 'DELINQ')])$bin_out
head(batch_woe(hmeq[, c('DEROG', 'DELINQ')], bin_out))
```

cal_woe — *Perform WoE transformation of a numeric variable*

Description

The function `cal_woe` applies the WoE transformation to a numeric vector based on the binning outcome from a binning function, e.g. qtl_bin() or iso_bin().

Usage

```
cal_woe(x, bin)
```

Arguments

`x`	A numeric vector that will be transformed to WoE values.
`bin`	A list with the binning outcome from the binning function, e.g. qtl_bin() or iso_bin()

Value

A numeric vector with WoE transformed values.

Examples

```
data(hmeq)
bin_out <- qtl_bin(hmeq$DEROG, hmeq$BAD)
cal_woe(hmeq$DEROG[1:10], bin_out)
```

gbm_bin — *Monotonic binning based on generalized boosted model*

Description

The function `gbm_bin` implements the monotonic binning based on the generalized boosted model (GBM).

Usage

```
gbm_bin(x, y)
```

Arguments

`x`	A numeric vector
`y`	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
gbm_bin(hmeq$DEROG, hmeq$BAD)
```

---

hmeq *Credit attributes of 5,960 home equity loans*

Description

A dataset containing characteristics and delinquency information for 5,960 home equity loans.

Usage

```
hmeq
```

Format

A data frame with 5960 rows and 13 variables:

BAD indicator of applicant defaulted on loan or seriously delinquent

LOAN Amount of the loan request, in dollar

MORTDUE Amount due on existing mortgage, in dollar

VALUE Value of current property, in dollar

REASON DebtCon = debt consolidation; HomeImp = home improvement

JOB Occupational categories

YOJ Years at present job

DEROG Number of major derogatory reports

DELINQ Number of delinquent credit lines

CLAGE Age of oldest credit line in months

NINQ Number of recent credit inquiries

CLNO Number of credit lines

DEBTINC Debt-to-income ratio

Source

http://www.creditriskanalytics.net/datasets-private2.html

`iso_bin` *Monotonic binning based on isotonic regression*

Description

The function `iso_bin` implements the monotonic binning based on the isotonic regression.

Usage

```
iso_bin(x, y)
```

Arguments

`x`	A numeric vector
`y`	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
iso_bin(hmeq$DEROG, hmeq$BAD)
```

`kmn_bin` *Monotonic binning based on k-means clustering*

Description

The function `kmn_bin` implements the monotonic binning based on the k-means clustering

Usage

```
kmn_bin(x, y)
```

Arguments

`x`	A numeric vector
`y`	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
kmn_bin(hmeq$DEROG, hmeq$BAD)
```

---

pool_bin *Monotonic binning for the pool data*

Description

The function `pool_bin` implements the monotonic binning for the pool data based on the generalized boosted model (GBM).

Usage

```
pool_bin(x, num, den, log = FALSE)
```

Arguments

`x`	A numeric vector
`num`	A numeric vector with integer values for numerators to calculate bad rates
`den`	A numeric vector with integer values for denominators to calculate bad rates
`log`	A logical constant either TRUE or FALSE. The default is FALSE

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
df <- rbind(Reduce(rbind,
                   lapply(split(hmeq, floor(hmeq$CLAGE)),
                          function(d) data.frame(AGE = unique(floor(d$CLAGE)),
                                                 NUM = sum(d$BAD),
                                                 DEN = nrow(d)))),
            data.frame(AGE = NA,
                       NUM = sum(hmeq[is.na(hmeq$CLAGE), ]$BAD),
                       DEN = nrow(hmeq[is.na(hmeq$CLAGE), ])))
pool_bin(df$AGE, df$NUM, df$DEN, log = TRUE)
```

qcut *Discretizing a numeric vector*

Description

The function qcut discretizes a numeric vector into N pieces based on quantiles.

Usage

```
qcut(x, n)
```

Arguments

x	A numeric vector.
n	An integer indicating the number of categories to discretize.

Value

A numeric vector to divide the vector x into n categories.

Examples

```
x <- 1:10
# [1]  1  2  3  4  5  6  7  8  9 10
v <- qcut(1:10, 4)
# [1] 3 5 8
findInterval(x, sort(c(v, -Inf, Inf)), left.open = TRUE)
# [1] 1 1 1 2 2 3 3 3 4 4
```

qtl_bin *Monotonic binning by quantile*

Description

The function qtl_bin implements the quantile-based monotonic binning by the iterative discretization

Usage

```
qtl_bin(x, y)
```

Arguments

x	A numeric vector
y	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
qtl_bin(hmeq$DEROG, hmeq$BAD)
```

`rng_bin` *Monotonic binning by quantile based on value range*

Description

The function `rng_bin` implements the quantile-based monotonic binning by the iterative discretization based on the equal-width range of values.

Usage

```
rng_bin(x, y)
```

Arguments

`x`	A numeric vector
`y`	A numeric vector with 0/1 binary values

Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

Examples

```
data(hmeq)
rng_bin(hmeq$DEROG, hmeq$BAD)
```

Index