Package ‘GMDH2’

April 28, 2021

Type Package

Title Binary Classification via GMDH-Type Neural Network Algorithms

Version 1.6

Date 2021-04-28

Depends R (>= 3.2.2)

Imports e1071, utils, graphics, stats, randomForest, glmnet, nnet, MASS, plotly, magrittr, xtable

Suggests mlbench

Author Osman Dag [aut, cre], Erdem Karabulut [aut], Reha Alpar [aut], Merve Kasikci [ctb]

Maintainer Osman Dag <`osman.dag@outlook.com`>

URL `http://www.softmed.hacettepe.edu.tr/GMDH2`

Description Performs binary classification via Group Method of Data Handling (GMDH) - type neural network algorithms. There exist two main algorithms available in GMDH() and dceGMDH() functions. GMDH() performs classification via GMDH algorithm for a binary response and returns important variables. dceGMDH() performs classification via diverse classifiers ensemble based on GMDH (dce-GMDH) algorithm. Also, the package produces a well-formatted table of descriptives for a binary response. Moreover, it produces confusion matrix, its related statistics and scatter plot (2D and 3D) with classification labels of binary classes to assess the prediction performance. All 'GMDH2' functions are designed for a binary response (Dag et al., 2019, <https://download.atlantis-press.com/article/125911202.pdf>).

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-28 04:20:02 UTC

R topics documented:

`GMDH2-package` *Binary Classification via GMDH-Type Neural Network Algorithms*

Description

Performs binary classification via Group Method of Data Handling (GMDH) - type neural network algorithms. There exist two main algorithms available in GMDH() and dceGMDH() functions. GMDH() performs classification via GMDH algorithm for a binary response and returns important variables. dceGMDH() performs classification via diverse classifiers ensemble based on GMDH (dce-GMDH) algorithm. Also, the package produces a well-formatted table of descriptives for a binary response. Moreover, it produces confusion matrix, its related statistics and scatter plot (2D and 3D) with classification labels of binary classes to assess the prediction performance. All 'GMDH2' functions are designed for a binary response (Dag et al., 2019, <https://download.atlantis-press.com/article/125911202.pdf>).

Details

Package:	GMDH
Type:	Package
License:	GPL (>=2)

`confMat` *Confusion Matrix*

Description

`confMat` constructs a 2×2 confusion matrix and returns some statistics related to confusion matrix.

Usage

```
confMat(data, ...)

## Default S3 method:
```

```
confMat(data, reference, positive = NULL, verbose = TRUE, ...)

## S3 method for class 'table'
confMat(data, positive = NULL, verbose = TRUE, ...)
```

Arguments

`data`	a factor of predicted classes (for the default method) or an object of class `table`.
`...`	option to be passed to `table`. Note: do not include `reference` here.
`reference`	a factor of classes to be used as the true results.
`positive`	an optional character string for the factor level that corresponds to a "positive" result.
`verbose`	a logical for printing output to R console.

Details

The `confMat` function requires that the factors have exactly the same levels. The function constructs 2×2 confusion matrix and calculates accuracy, no information rate (NIR), unweighted Kappa statistic, Matthews correlation coefficient, sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), prevalence, balanced accuracy, youden index, detection rate, detection prevalence, precision, recall and F1 measure.

Suppose a 2×2 table with notation

	Reference	
Predicted	Event	No Event
Event	TP	FP
No Event	FN	TN

TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives and TN is the number of true negatives.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$NIR = max(Prevalence, 1 - Prevalence)$$

$$Kappa = \frac{Accuracy - \frac{(TP+FP)(TP+FN)+(FN+TN)(FP+TN)}{(TP+FP+FN+TN)^2}}{1 - \frac{(TP+FP)(TP+FN)+(FN+TN)(FP+TN)}{(TP+FP+FN+TN)^2}}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (FN + TN) \times (TP + FN) \times (FP + TN)}}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$PPV = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$Prevalence = \frac{TP + FN}{TP + FP + FN + TN}$$

$$Balanced\ accuracy = \frac{Sensitivity + Specificity}{2}$$

$$Youden\ index = Sensitivity + Specificity - 1$$

$$Detection\ rate = \frac{TP}{TP + FP + FN + TN}$$

$$Detection\ prevalence = \frac{TP + FP}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Value

Returns a list containing following elements:

`table`	confusion matrix
`accuracy`	accuracy
`NIR`	no information rate
`kappa`	unweighted kappa
`MCC`	Matthews correlation coefficient
`sensitivity`	sensitivity
`specificity`	specificity
`PPV`	positive predictive value
`NPV`	negative predictive value
`prevalence`	prevalence
`baccuracy`	balanced accuracy
`youden`	youden index
`detectRate`	detection rate
`detectPrev`	detection prevalence
`precision`	precision
`recall`	recall
`F1`	F1 measure
`all`	returns a matrix containing all statistics

Note

If the factors, reference and data, have the same levels, but in the incorrect order, the `confMat` will reorder data with the order of reference.

Author(s)

Osman Dag

See Also

`confusionMatrix`

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]
```

```
x.test <- x[test.indices,]
y.test <- y[test.indices]


set.seed(seed)
# to construct model via dce-GMDH algorithm
model <- dceGMDH(x.train, y.train, x.valid, y.valid)

# to obtain predicted classes for test set
y.test_pred <- predict(model, x.test, type = "class")

# to obtain confusion matrix and some statistics for test set
confMat(y.test_pred, y.test, positive = "malignant")

# to obtain statistics from table
result <- table(y.test_pred, y.test)
confMat(result, positive = "malignant")
```

`cplot2d` *2D Scatter Plot with Classification Labels*

Description

`cplot2d` produces two dimensional scatter plot with classification labels of binary classes.

Usage

```
cplot2d(x1, x2, ypred, yobs, colors = c("red", "blue"), symbols = c("circle", "o"),
  size = 10, xlab = NULL, ylab = NULL, title = NULL)
```

Arguments

`x1`	the horizontal coordinate of points in the plot.
`x2`	the vertical coordinate of points in the plot.
`ypred`	a factor of the predicted binary response variable.
`yobs`	a factor of the observed binary response variable.
`colors`	a vector of two colors specifying the levels of the observed binary response variable.
`symbols`	a vector of two symbols specifying the levels of the predicted binary response variable.
`size`	the size of symbols.
`xlab`	a label for the x axis, defaults to a description of x1.
`ylab`	a label for the y axis, defaults to a description of x2.
`title`	a main title of the plot.

Details

Symbols indicate the observed classes of binary response. Colors show TRUE or FALSE classification of the observations.

Value

An object with class "plotly" and "htmlwidget".

Author(s)

Osman Dag

See Also

`plot_ly`

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]
```

```
x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]

set.seed(seed)
# to construct model via dce-GMDH algorithm
model <- dceGMDH(x.train, y.train, x.valid, y.valid)

# to obtain predicted classes for test set
y.test_pred <- predict(model, x.test, type = "class")

# to obtain confusion matrix and some statistics for test set
confMat(y.test_pred, y.test, positive = "malignant")

# to produce 2D scatter plot with classification labels for test set
cplot2d(x.test[,1], x.test[,6], y.test_pred, y.test, symbols = c("x", "o"))
cplot2d(x.test[,1], x.test[,6], y.test_pred, y.test, colors = c("red", "black"))
```

`cplot3d` *3D Scatter Plot with Classification Labels*

Description

`cplot3d` produces three dimensional scatter plot with classification labels of binary classes.

Usage

```
cplot3d(x1, x2, x3, ypred, yobs, colors = c("red", "blue"),
  symbols = c("circle", "o"), size = 10, xlab = NULL, ylab = NULL, zlab = NULL,
  title = NULL)
```

Arguments

`x1`	the x coordinate of points in the plot.
`x2`	the y coordinate of points in the plot.
`x3`	the z coordinate of points in the plot.
`ypred`	a factor of the predicted binary response variable.
`yobs`	a factor of the observed binary response variable.
`colors`	a vector of two colors specifying the levels of the observed binary response variable.
`symbols`	a vector of two symbols specifying the levels of the predicted binary response variable.
`size`	the size of symbols.

`xlab`	a label for the x axis, defaults to a description of x1.
`ylab`	a label for the y axis, defaults to a description of x2.
`zlab`	a label for the z axis, defaults to a description of x3.
`title`	a main title of the plot.

Details

Symbols indicate the observed classes of binary response. Colors show TRUE or FALSE classification of the observations.

Value

An object with class "plotly" and "htmlwidget".

Author(s)

Osman Dag

See Also

`plot_ly`

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)
```

```
train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]


set.seed(seed)
# to construct model via dce-GMDH algorithm
model <- dceGMDH(x.train, y.train, x.valid, y.valid)

# to obtain predicted classes for test set
y.test_pred <- predict(model, x.test, type = "class")

# to obtain confusion matrix and some statistics for test set
confMat(y.test_pred, y.test, positive = "malignant")

# to produce 3D scatter plot with classification labels for test set
cplot3d(x.test[,1], x.test[,6], x.test[,3], y.test_pred, y.test,
colors = c("red", "black"))
```

`dceGMDH` *Diverse Classifiers Ensemble Based on GMDH-Type Neural Network Algorithm for Binary Classification*

Description

`dceGMDH` makes a binary classification via diverse classifiers ensemble Based on GMDH-Type Neural Network (dce-GMDH) Algorithm.

Usage

```
dceGMDH(x.train, y.train, x.valid, y.valid, alpha = 0.6, maxlayers = 10,
  maxneurons = 15, exCriterion = "MSE", verbose = TRUE, svm_options,
  randomForest_options, naiveBayes_options, cv.glmnet_options, nnet_options, ...)
```

Arguments

`x.train`	a n1xp matrix to be included in model construction, n1 is the number of observations and p is the number of variables.

`y.train`	a factor of binary response variable to be included in model construction.
`x.valid`	a n2xp matrix to be used for neuron selection, n2 is the number of observations and p is the number of variables.
`y.valid`	a factor of binary response variable to be used for neuron selection.
`alpha`	the selection pressure in a layer. Defaults alpha = 0.6.
`maxlayers`	the number of maximum layers. Defaults maxlayers = 10.
`maxneurons`	the number of maximum neurons selected in each layer. Defaults maxneurons = 15.
`exCriterion`	a character string to select an external criteria. "MSE": Mean Square Error, "MAE": Mean Absolute Error. Default is set to "MSE".
`verbose`	a logical for printing summary output to R console.
`svm_options`	a list for options of `svm`.
`randomForest_options`	a list for options of `randomForest`.
`naiveBayes_options`	a list for options of `naiveBayes`.
`cv.glmnet_options`	a list for options of `cv.glmnet`.
`nnet_options`	a list for options of `nnet`.
`...`	not used currently.

Value

A list with class "dceGMDH" and "GMDHplot" containing the following components:

`architecture`	all objects stored in construction process of network
`nlayer`	the number of layers
`neurons`	the number of neurons in layers
`sneurons`	the number of selected neurons in layers
`structure`	the summary structure of the process
`levels`	the levels of binary response
`base_perf`	the performances of the classifiers on validation set at base training
`base_models`	the constructed base classifiers models
`classifiers`	the names of assembled classifiers
`plot_list`	the list of objects to be used in `plot.GMDHplot`

Author(s)

Osman Dag, Erdem Karabulut, Reha Alpar

References

Dag, O., Karabulut, E., Alpar, R. (2019). GMDH2: Binary Classification via GMDH-Type Neural Network Algorithms - R Package and Web-Based Tool. *International Journal of Computational Intelligence Systems*, **12:2**, 649-660.

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]

set.seed(seed)
# to construct model via dce-GMDH algorithm
model <- dceGMDH(x.train, y.train, x.valid, y.valid)

# to obtain predicted classes for test set
predict(model, x.test)
```

`GMDH` *Feature Selection and Classification via GMDH-Type Neural Network Algorithm for a Binary Response*

Description

`GMDH` makes feature selection and classification via GMDH-type neural network algorithm.

Usage

```
GMDH(x.train, y.train, x.valid, y.valid, alpha = 0.6, maxlayers = 10, maxneurons = 15,
  exCriterion = "MSE", verbose = TRUE, ...)
```

Arguments

`x.train`	a n1xp matrix to be included in model construction, n1 is the number of observations and p is the number of variables.
`y.train`	a factor of binary response variable to be included in model construction.
`x.valid`	a n2xp matrix to be used for neuron selection, n2 is the number of observations and p is the number of variables.
`y.valid`	a factor of binary response variable to be used for neuron selection.
`alpha`	the selection pressure in a layer. Defaults alpha = 0.6.
`maxlayers`	the number of maximum layers. Defaults maxlayers = 10.
`maxneurons`	the number of maximum neurons selected in each layer. Defaults maxneurons = 15.
`exCriterion`	a character string to select an external criteria. "MSE": Mean Square Error, "MAE": Mean Absolute Error. Default is set to "MSE".
`verbose`	a logical for printing summary output to R console.
`...`	not used currently.

Value

A list with class "GMDH" and "GMDHplot" containing the following components:

`architecture`	all objects stored in construction process of network
`nlayer`	the number of layers
`neurons`	the number of neurons in layers
`sneurons`	the number of selected neurons in layers
`structure`	the summary structure of the process
`levels`	the levels of binary response
`features`	the names of variables used in GMDH algorithm
`pfeatures`	the column number of variables used in GMDH algorithm
`nvar`	the number of variables in the data set
`plot_list`	the list of objects to be used in `plot.GMDHplot`

Author(s)

Osman Dag, Erdem Karabulut, Reha Alpar

References

Dag, O., Karabulut, E., Alpar, R. (2019). GMDH2: Binary Classification via GMDH-Type Neural Network Algorithms - R Package and Web-Based Tool. *International Journal of Computational Intelligence Systems*, **12:2**, 649-660.

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]
```

```
set.seed(seed)
# to construct model via GMDH algorithm
model <- GMDH(x.train, y.train, x.valid, y.valid)
predict(model, x.test)
```

`plot.GMDHplot` *Plotting Using GMDH and dce-GMDH Algorithms for Binary Classification*

Description

This function plots minimum specified external criterion across layers based upon a model trained by `GMDH` or `dceGMDH`. This is plotted for validation set.

Usage

```
## S3 method for class 'GMDHplot'
plot(x, ...)
```

Arguments

`x`	an object of class created by `GMDH` or `dceGMDH`.
`...`	currently not used.

Author(s)

Osman Dag

See Also

`GMDH`, `dceGMDH`

Examples

```

library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]
```

```
seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets
indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]

set.seed(seed)
# to construct model via GMDH algorithm
model <- GMDH(x.train, y.train, x.valid, y.valid)
plot(model)

# to construct model via dce-GMDH algorithm
model2 <- dceGMDH(x.train, y.train, x.valid, y.valid)
plot(model2)
```

`predict.dceGMDH` *Predicting Using dce-GMDH Algorithm for Binary Classification*

Description

This function predicts values based upon a model trained by `dceGMDH`.

Usage

```
## S3 method for class 'dceGMDH'
predict(object, x, type = "class", ...)
```

Arguments

`object`	an object of class `"dceGMDH"`, created by `dceGMDH`.
`x`	a matrix containing the new input data.
`type`	a character string to return predicted output. If type = "class", the function returns the predicted classes. If type = "probability", it returns the predicted probabilities. Default is set to "class".
`...`	currently not used.

Value

A vector of predicted values of corresponding classes depending on type specified.

Author(s)

Osman Dag, Erdem Karabulut, Reha Alpar

See Also

`dceGMDH`

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
```

```
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]

set.seed(seed)
# to construct model via dce-GMDH algorithm
model <- dceGMDH(x.train, y.train, x.valid, y.valid)

# to obtain predicted classes for test set
predict(model, x.test, type = "class")
# to obtain predicted probabilities for test set
predict(model, x.test, type = "probability")
```

`predict.GMDH` *Predicting Using GMDH Algorithm for Binary Classification*

Description

This function predicts values based upon a model trained by `GMDH`.

Usage

```
## S3 method for class 'GMDH'
predict(object, x, type = "class", ...)
```

Arguments

`object`	an object of class `"GMDH"`, created by `GMDH`.
`x`	a matrix containing the new input data.
`type`	a character string to return predicted output. If type = "class", the function returns the predicted classes. If type = "probability", it returns the predicted probabilities. Default is set to "class".
`...`	currently not used.

Value

A vector of predicted values of corresponding classes depending on type specified.

Author(s)

Osman Dag, Erdem Karabulut, Reha Alpar

See Also

`GMDH`

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)

data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data.matrix(data[,2:10])
y <- data[,11]

seed <- 12345
set.seed(seed)
nobs <- length(y)

# to split train, validation and test sets

indices <- sample(1:nobs)

ntrain <- round(nobs*0.6,0)
nvalid <- round(nobs*0.2,0)
ntest <- nobs-(ntrain+nvalid)

train.indices <- sort(indices[1:ntrain])
valid.indices <- sort(indices[(ntrain+1):(ntrain+nvalid)])
test.indices <- sort(indices[(ntrain+nvalid+1):nobs])


x.train <- x[train.indices,]
y.train <- y[train.indices]

x.valid <- x[valid.indices,]
y.valid <- y[valid.indices]

x.test <- x[test.indices,]
y.test <- y[test.indices]

set.seed(seed)
# to construct model via GMDH algorithm
```

```
model <- GMDH(x.train, y.train, x.valid, y.valid)

# to obtain predicted classes for test set
predict(model, x.test, type = "class")
# to obtain predicted probabilities for test set
predict(model, x.test, type = "probability")
```

`Table` *Table for Simple Descriptive Statistics*

Description

`Table` produces a table for simple descriptive statistics for a binary response.

Usage

```
Table(x, y, option = "min-max", percentages = "column", ndigits = c(2,1),
  output = "R")
```

Arguments

`x`	a data frame including all variables.
`y`	a factor of binary response variable.
`option`	an option to return "min-max" or "Q1-Q3". Default is set to "min-max".
`percentages`	a character string to select the desired percentages. To use column or row percentages, percentages should be set to "column" or "row", respectively. The percentages argument is set to "total" to obtain total percentages. Default is set to "column".
`ndigits`	a vector of two numbers. The first one is used to specify the number of digit for numeric/integer variables. The second one specifies the number of digits for percentages of factor/ordered variables. Default is set to ndigits = c(2,1).
`output`	a character string to specify the format of descriptive statistics. If output = "LaTeX", it returns the table as latex format. If output = "HTML", it produces the table as html format. If output = "R", it returns the table in R console.

Author(s)

Osman Dag

Examples

```
library(GMDH2)

library(mlbench)
data(BreastCancer)
```

```
data <- BreastCancer

# to obtain complete observations
completeObs <- complete.cases(data)
data <- data[completeObs,]

x <- data[,2:10]
y <- data[,11]

Table(x, y)
Table(x, y, output = "LaTeX")
```

Index