Package ‘InformationValue’

October 12, 2022

Type Package

Title Performance Analysis and Companion Functions for Binary Classification Models

Version 1.2.3

Date 2016-10-29

Author Selva Prabhakaran

Maintainer Selva Prabhakaran <selva86@gmail.com>

URL http://r-statistics.co/Information-Value-With-R.html

Description Provides companion function for analysing the performance of classification models. Also, provides function to optimise probability cutoff score based on used specified objectives, Plot 'ROC' Curve in 'ggplot2', 'AUROC', 'IV', 'WOE' Calculation, 'KS Statistic' etc to aid accuracy improvement in binary classification models.

License GPL (>= 2)

BugReports https://github.com/selva86/InformationValue/issues License: GPL (>= 2)

LazyData TRUE

LazyLoad yes

Depends R (>= 3.0.0)

Imports ggplot2, data.table

Suggests knitr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-30 09:00:24

R topics documented:

`ActualsAndScores` *ActualsAndScores*

Description

A dataset containing the actuals for a simulated binary response variable as a numeric and the prediction probablity scores for a classification model like logistic regression.

Usage

```
data(ActualsAndScores)
```

Format

A data frame with 170 rows and 2 variables

Details

- Actuals. A simulated variable meant to serve as the actual binary response variable. The good/events are marked as 1 while the bads/non-events are marked 0.
- PredictedScores. The prediction probability scores based on a classification model.

AUROC *AUROC*

Description

Calculate the area uder ROC curve statistic for a given logit model.

Usage

```
AUROC(actuals, predictedScores)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

Details

For a given actuals and predicted probability scores, the area under the ROC curve shows how well the model performs at capturing the false events and false non-events. An best case model will have an area of 1. However that would be unrealistic, so the closer the aROC to 1, the better is the model.

Value

The area under the ROC curve for a given logit model.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
AUROC(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

`Concordance` *Concordance*

Description

Calculate concordance and discordance percentages for a logit model

Usage

```
Concordance(actuals, predictedScores)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

Details

Calculate the percentage of concordant and discordant pairs for a given logit model.

Value

a list containing percentage of concordant pairs, percentage discordant pairs, percentage ties and No. of pairs.

- Concordance The total proportion of pairs in concordance. A pair is said to be concordant when the predicted score of 'Good' (Event) is greater than that of the 'Bad'(Non-event)
- Discordance The total proportion of pairs that are discordant.
- Tied The proportion of pairs for which scores are tied.
- Pairs The total possible combinations of 'Good-Bad' pairs based on actual response (1/0) labels.

Author(s)

Selva Prabhakaran <`selva86@gmail.com`>

Examples

```
data('ActualsAndScores')
Concordance(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

`confusionMatrix` *confusionMatrix*

Description

Calculate the confusion matrix for the fitted values for a logistic regression model.

Usage

```
confusionMatrix(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals` The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.

`predictedScores` The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

`threshold` If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given actuals and predicted probability scores, the confusion matrix showing the count of predicted events and non-events against actual events and non events.

Value

For a given actuals and predicted probability scores, returns the confusion matrix showing the count of predicted events and non-events against actual events and non events.

Author(s)

Selva Prabhakaran <`selva86@gmail.com`>

Examples

```
data('ActualsAndScores')
confusionMatrix(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

IV *IV*

Description

Compute the IV for each group of a given categorical X and binary response Y. The resulting WOE can be used in place of the categorical X so as to be used as a continuous variable.

Usage

```
IV(X, Y, valueOfGood = 1)
```

Arguments

X	The categorical variable stored as factor for which Information Value (IV) is to be computed.
Y	The actual 1/0 flags for the binary response variable. It can take values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
valueOfGood	The value in Y that is used to represent 'Good' or the occurence of the event of interest. Defaults to 1.

Details

For a given actual for a Binary Y variable and a categorical X variable stored as factor, the information values are computed.

Value

The Information Value (IV) for each group in categorical X variable.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('SimData')
IV(X=SimData$X.Cat, Y=SimData$Y.Binary)
```

kappaCohen *kappaCohen*

Description

Calculate the Cohen's kappa statistic for a given logit model.

Usage

```
kappaCohen(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
`threshold`	If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given actuals and predicted probability scores, Cohen's kappa is calculated. Cohen's kappa is calculated as (probabiliity of agreement - probability of expected) / (1-(probability of expected)))

Value

The Cohen's kappa of the given actuals and predicted probability scores

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
kappaCohen(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

ks_plot *ks_plot*

Description

Plot the cumulative percentage of responders (ones) captured by the model

Usage

```
ks_plot(actuals, predictedScores)
```

Arguments

actuals	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
predictedScores	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

Details

Plot the cumulative percentage of responders (ones) captured by the model against the expected cumulative percentage of responders at random (i.e. had there been no model). The greater the distance between the random and model cumulatives, the better is the predictive ability of the model to effectively capture the responders (ones).

Value

The KS plot

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
ks_plot(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

`ks_stat` *ks_stat*

Description

Compute the Kolmogorov-Smirnov statistic

Usage

```
ks_stat(actuals, predictedScores, returnKSTable = FALSE)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
`returnKSTable`	If set to TRUE, returns the KS table used to calculate the KS statistic instead. Defaults to FALSE.

Details

Compute the KS statistic for a given actuals and predicted scores for a binary response variable. KS statistic is calculated as the maximum difference between the cumulative true positive and cumulative false positive rate. Set returnKSTable to TRUE to see the calculations from ks_table.

Value

The KS statistic for a given actual values of a binary response variable and the respective prediction probability scores.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
ks_stat(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

misClassError *misClassError*

Description

Calculate the percentage misclassification error for the given actuals and probaility scores.

Usage

```
misClassError(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
`threshold`	If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given binary response actuals and predicted probability scores, misclassfication error is the number of mismatches between the predicted and actuals direction of the binary y variable.

Value

The misclassification error, which tells what proportion of predicted direction did not match with the actuals.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
misClassError(actuals=ActualsAndScores$Actuals,
  predictedScores=ActualsAndScores$PredictedScores, threshold=0.5)
```

npv *npv*

Description

Calculate the negative predictive value for a given set of actuals and predicted probability scores.

Usage

```
npv(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
`threshold`	If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given given binary response actuals and predicted probability scores, negative predictive value is defined as the proportion of observations without the event out of the total negative predictions.

Value

The negative predictive value for a given set of actuals and probability scores, with the specified cutoff threshold.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
npv(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

optimalCutoff *optimalCutoff*

Description

Compute the optimal probability cutoff score, based on a user defined objective.

Usage

```
optimalCutoff(actuals, predictedScores, optimiseFor = "misclasserror",
  returnDiagnostics = FALSE)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
`optimiseFor`	The maximization criterion for which probability cutoff score needs to be optimised. Can take either of following values: "Ones" or "Zeros" or "Both" or "misclasserror"(default). If "Ones" is used, 'optimalCutoff' will be chosen to maximise detection of "One's". If 'Both' is specified, the probability cut-off that gives maximum Youden's Index is chosen. If 'misclasserror' is specified, the probability cut-off that gives minimum mis-clasification error is chosen.
`returnDiagnostics`	If TRUE, would return additional diagnostics such as 'sensitivityTable', 'misclassificationError', 'TPR', 'FPR' and 'specificity' for the chosen cut-off.

Details

Compute the optimal probability cutoff score for a given set of actuals and predicted probability scores, based on a user defined objective, which is specified by optimiseFor = "Ones" or "Zeros" or "Both" (default).

Value

The optimal probability score cutoff that maximises a given criterion. If 'returnDiagnostics' is TRUE, then the following items are returned in a list:

- optimalCutoff The optimal probability score cutoff that maximises a given criterion.
- sensitivityTable The dataframe that shows the TPR, FPR, Youden's Index and Specificity for variaous values of purbability cut-off scores.
- misclassificationError The percentage misclassification error for the given actuals and probaility scores.

- TPR The ’True Positive Rate’ (a.k.a ’sensitivity’)for the chosen probability cut-off score.
- FPR The ’False Positive Rate’ (a.k.a ’sensitivity’)for the chosen probability cut-off score.
- Specificity The specificity of the given actuals and probability scores, i.e. the ratio of number of observations without the event AND predicted to not have the event divided by the number of observations without the event.

Author(s)

Selva Prabhakaran `<selva86@gmail.com>`

Examples

```
data('ActualsAndScores')
optimalCutoff(actuals=ActualsAndScores$Actuals,
predictedScores=ActualsAndScores$PredictedScores, optimiseFor="Both", returnDiagnostics=TRUE)
```

`plotROC` *plotROC*

Description

Plot the Receiver Operating Characteristics(ROC) Curve based on ggplot2

Usage

```
plotROC(actuals, predictedScores, Show.labels = F, returnSensitivityMat = F)
```

Arguments

`actuals` The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the ’Good’ or ’Events’ while 0 represents ’Bad’ or ’Non-Events’.

`predictedScores` The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1’s and 0’s.

`Show.labels` Whether the probability scores should be printed at change points?. Defaults to False.

`returnSensitivityMat` Whether the sensitivity matrix (a dataframe) should be returned. Defaults to FALSE.

Details

For a given actuals and predicted probability scores, A ROC curve is plotted using the ggplot2 framework along the the area under the curve.

Value

Plots the ROC curve

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
plotROC(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

---

precision *precision*

Description

Calculate the precision or positive predictive value for a given set of actuals and predicted probability scores.

Usage

```
precision(actuals, predictedScores, threshold = 0.5)
```

Arguments

actuals	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
predictedScores	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
threshold	If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given given binary response actuals and predicted probability scores, precision is defined as the proportion of observations with the event out of the total positive predictions.

Value

The precision or the positive predictive value.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
precision(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

sensitivity *sensitivity*

Description

Calculate the sensitivity for a given logit model.

Usage

```
sensitivity(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.
`threshold`	If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given binary response actuals and predicted probability scores, sensitivity is defined as number of observations with the event AND predicted to have the event divided by the number of observations with the event. It can be used as an indicator to gauge how sensitive is your model in detecting the occurence of events, especially when you are not so concerned about predicting the non-events as true.

Value

The sensitivity of the given binary response actuals and predicted probability scores, which is, the number of observations with the event AND predicted to have the event divided by the nummber of observations with the event.

Author(s)

Selva Prabhakaran `<selva86@gmail.com>`

Examples

```
data('ActualsAndScores')
sensitivity(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

SimData *SimData*

Description

A dataset containing the actuals for a simulated binary response variable (Y) as a numeric and a categorical X variable with 9 groups, for which WOE calculation is performed.

Usage

```
data(SimData)
```

Format

A data frame with 30000 rows and 2 variables

Details

- Y.Binary. A simulated variable meant to serve as the actual binary response variable. The good/events are marked as 1 while the bads/non-events are marked 0.
- X.Cat. A categorical variable (factor) with 9 groups.

somersD *somersD*

Description

Calculate the Somers D statistic for a given logit model

Usage

```
somersD(actuals, predictedScores)
```

Arguments

`actuals`	The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
`predictedScores`	The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

Details

For a given binary response actuals and predicted probability scores, Somer's D is calculated as the number of concordant pairs less number of discordant pairs divided by total number of pairs.

Value

The Somers D statistic, which tells how many more concordant than discordant pairs exist divided by total number of pairs.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
somersD(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

---

specificity *specificity*

Description

Calculate the specificity for a given logit model.

Usage

```
specificity(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals` The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.

`predictedScores` The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

`threshold` If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given given binary response actuals and predicted probability scores, specificity is defined as number of observations without the event AND predicted to not have the event divided by the number of observations without the event. Specificity is particularly useful when you are extra careful not to predict a non event as an event, like in spam detection where you dont want to classify a genuine mail as spam(event) where it may be somewhat ok to occasionally classify a spam as a genuine mail(a non-event).

Value

The specificity of the given binary response actuals and predicted probability scores, which is, the number of observations without the event AND predicted to not have the event divided by the nummber of observations without the event.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
specificity(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

WOE *WOE*

Description

Compute the Weights Of Evidence (WOE) for each group of a given categorical X and binary response Y. The resulting WOE can be usued in place of the categorical X so as to be used as a continuous variable.

Usage

```
WOE(X, Y, valueOfGood = 1)
```

Arguments

X	The categorical variable stored as factor for which Weights of Evidence(WOE) is to be computed.
Y	The actual 1/0 flags for the binary response variable. It can take values of either 1 or 0, where 1 represents the ’Good’ or ’Events’ while 0 represents ’Bad’ or ’Non-Events’.
valueOfGood	The value in Y that is used to represent ’Good’ or the occurence of the event of interest. Defaults to 1.

Details

For a given actual for a Binary Y variable and a categorical X variable stored as factor, the WOE’s are computed.

Value

The Weights Of Evidence (WOE) for each group in categorical X variable.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('SimData')
WOE(X=SimData$X.Cat, Y=SimData$Y.Binary)
```

WOETable *WOETable*

Description

Compute the WOETable that shows the Weights Of Evidence (WOE) for each group and respeective Information Values (IVs).

Usage

```
WOETable(X, Y, valueOfGood = 1)
```

Arguments

X	The categorical variable stored as factor for which WOE Table is to be computed.
Y	The actual 1/0 flags for the binary response variable. It can take values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.
valueOfGood	The value in Y that is used to represent 'Good' or the occurence of the event of interest. Defaults to 1.

Details

For a given actual for a Binary Y variable and a categorical X variable stored as factor, the WOE table is generated with calculated WOE's and IV's

Value

The WOE table with the respective weights of evidence for each group and the IV's.

- CAT. The groups (levels) of the categorical X variable for which WOE is to be calculated.
- GOODS. The total number of "Goods" or "Events" in respective group.
- BADS. The total number of "Bads" or "Non-Events" in respective group.
- TOTAL. The total number of observations in respective group.
- PCT_G. The Percentage of 'Goods' or 'Events' accounted for by respective group.
- PCT_B. The Percentage of 'Bads' or 'Non-Events' accounted for by respective group.

- WOE. The computed weights of evidence(WOE) for respective group. The WOE values can be used in place of the actual group itself, thereby producing a 'continuous' alternative.
- IV. The information value contributed by each group in the X. The sum of IVs is the total information value of the categorical X variable.

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('SimData')
WOETable(X=SimData$X.Cat, Y=SimData$Y.Binary)
```

`youdensIndex` *youdensIndex*

Description

Calculate the specificity for a given logit model.

Usage

```
youdensIndex(actuals, predictedScores, threshold = 0.5)
```

Arguments

`actuals` The actual binary flags for the response variable. It can take a numeric vector containing values of either 1 or 0, where 1 represents the 'Good' or 'Events' while 0 represents 'Bad' or 'Non-Events'.

`predictedScores` The prediction probability scores for each observation. If your classification model gives the 1/0 predcitions, convert it to a numeric vector of 1's and 0's.

`threshold` If predicted value is above the threshold, it will be considered as an event (1), else it will be a non-event (0). Defaults to 0.5.

Details

For a given binary response actuals and predicted probability scores, Youden's index is calculated as sensitivity + specificity - 1

Value

The youdensIndex of the given binary response actuals and predicted probability scores, which is calculated as Sensitivity + Specificity - 1

Author(s)

Selva Prabhakaran <selva86@gmail.com>

Examples

```
data('ActualsAndScores')
youdensIndex(actuals=ActualsAndScores$Actuals, predictedScores=ActualsAndScores$PredictedScores)
```

Index