

Package ‘ModelMetrics’

January 13, 2020

Title Rapid Calculation of Model Metrics

Version 1.2.2.1

Date 2018-11-03

Description Collection of metrics for evaluating models written in C++ using 'Rcpp'. Popular metrics include area under the curve, log loss, root mean square error, etc.

Depends R (>= 3.2.2)

License GPL (>= 2)

Encoding UTF-8

LazyData true

LinkingTo Rcpp

Imports Rcpp, data.table

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation yes

Author Tyler Hunt [aut, cre]

Maintainer Tyler Hunt <thunt@snapfinance.com>

Repository CRAN

Date/Publication 2020-01-13 17:36:03 UTC

R topics documented:

auc	2
brier	3
ce	4
confusionMatrix	5
f1Score	5
fScore	6
gini	6
kappa	7
logLoss	8

mae	9
mauc	9
mcc	10
mlogLoss	11
mse	11
msle	12
npv	13
ppv	13
recall	14
rmse	15
rmsle	15
testDF	16
tnr	17
Index	18

auc	<i>Area Under the Curve</i>
-----	-----------------------------

Description

Calculates the area under the curve for a binary classification model

Usage

```

auc(...)

## Default S3 method:
auc(actual, predicted, ...)

## S3 method for class 'glm'
auc(modelObject, ...)

## S3 method for class 'randomForest'
auc(modelObject, ...)

## S3 method for class 'glmerMod'
auc(modelObject, ...)

## S3 method for class 'gbm'
auc(modelObject, ...)

## S3 method for class 'rpart'
auc(modelObject, ...)

```

Arguments

... additional parameters to be passed the the s3 methods
 actual A vector of the labels. Can be numeric, character, or factor
 predicted A vector of predicted values
 modelObject the model object. Currently supported glm, randomForest, glmerMod, gbm

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')

auc(testDF$y, Preds)
# using s3 method for glm
auc(glmModel)
```

 brier

Brier Score

Description

Calculates the Brier score

Usage

```
brier(...)

## Default S3 method:
brier(actual, predicted, ...)

## S3 method for class 'glm'
brier(modelObject, ...)

## S3 method for class 'randomForest'
brier(modelObject, ...)

## S3 method for class 'glmerMod'
brier(modelObject, ...)

## S3 method for class 'gbm'
brier(modelObject, ...)

## S3 method for class 'rpart'
brier(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported glm, randomForest, glmerMod, gbm

ce	<i>Classification error</i>
----	-----------------------------

Description

Calculates the classification error

Usage

```
ce(...)
```

```
## Default S3 method:
ce(actual, predicted, ...)
```

```
## S3 method for class 'lm'
ce(modelObject, ...)
```

```
## S3 method for class 'glm'
ce(modelObject, ...)
```

```
## S3 method for class 'randomForest'
ce(modelObject, ...)
```

```
## S3 method for class 'glmerMod'
ce(modelObject, ...)
```

```
## S3 method for class 'gbm'
ce(modelObject, ...)
```

```
## S3 method for class 'rpart'
ce(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported lm, glm, randomForest, glmerMod, gbm, rpart

confusionMatrix	<i>Confusion Matrix</i>
-----------------	-------------------------

Description

Create a confusion matrix given a specific cutoff.

Usage

```
confusionMatrix(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

f1Score	<i>F1 Score</i>
---------	-----------------

Description

Calculates the f1 score

Usage

```
f1Score(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

fScore	<i>F Score</i>
--------	----------------

Description

Calculates the F score and allows different specifications of the beta value (F0.5)

Usage

```
fScore(actual, predicted, cutoff = 0.5, beta = 1)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values
beta	the desired beta value (lower increases weight of precision over recall). Defaults to 1

gini	<i>GINI Coefficient</i>
------	-------------------------

Description

Calculates the GINI coefficient for a binary classification model

Usage

```
gini(...)

## Default S3 method:
gini(actual, predicted, ...)

## S3 method for class 'glm'
gini(modelObject, ...)

## S3 method for class 'randomForest'
gini(modelObject, ...)

## S3 method for class 'glmerMod'
gini(modelObject, ...)

## S3 method for class 'gbm'
gini(modelObject, ...)

## S3 method for class 'rpart'
gini(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels. Can be numeric, character, or factor
predicted	A vector of predicted values
modelObject	the model object. Currently supported glm, randomForest, glmerMod, gbm

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')
```

```
gini(testDF$y, Preds)
# using s3 method for glm
gini(glmModel)
```

kappa	<i>kappa statistic</i>
-------	------------------------

Description

Calculates kappa statistic. Currently build to handle binary values in actual vector.

Usage

```
kappa(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

logLoss	<i>Log Loss</i>
---------	-----------------

Description

Calculates the log loss or entropy loss for a binary outcome

Usage

```
logLoss(...)  
  
## Default S3 method:  
logLoss(actual, predicted, distribution = "binomial", ...)  
  
## S3 method for class 'glm'  
logLoss(modelObject, ...)  
  
## S3 method for class 'randomForest'  
logLoss(modelObject, ...)  
  
## S3 method for class 'glmerMod'  
logLoss(modelObject, ...)  
  
## S3 method for class 'gbm'  
logLoss(modelObject, ...)  
  
## S3 method for class 'rpart'  
logLoss(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	a binary vector of the labels
predicted	a vector of predicted values
distribution	the distribution of the loss function needed binomial, poisson
modelObject	the model object. Currently supported glm, randomForest, glmerMod, gbm

Examples

```
data(testDF)  
glmModel <- glm(y ~ ., data = testDF, family="binomial")  
Preds <- predict(glmModel, type = 'response')  
  
logLoss(testDF$y, Preds)  
# using s3 method for glm  
logLoss(glmModel)
```

mae	<i>Mean absolute error</i>
-----	----------------------------

Description

Calculates the mean absolute error

Usage

```
mae(...)  
  
## Default S3 method:  
mae(actual, predicted, ...)  
  
## S3 method for class 'glm'  
mae(modelObject, ...)  
  
## S3 method for class 'randomForest'  
mae(modelObject, ...)  
  
## S3 method for class 'glmerMod'  
mae(modelObject, ...)  
  
## S3 method for class 'gbm'  
mae(modelObject, ...)  
  
## S3 method for class 'rpart'  
mae(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported glm, randomForest, glmerMod, gbm

mauc	<i>Multiclass Area Under the Curve</i>
------	--

Description

Calculates the area under the curve for a binary classification model

Usage

```
mauc(actual, predicted)
```

Arguments

actual A vector of the labels. Can be numeric, character, or factor
predicted A data.frame of predicted values. Can be matrix, data.frame

Examples

```
setosa <- glm(I(Species == 'setosa') ~ Sepal.Length, data = iris, family = 'binomial')
versicolor <- glm(I(Species == 'versicolor') ~ Sepal.Length, data = iris, family = 'binomial')
virginica <- glm(I(Species == 'virginica') ~ Sepal.Length, data = iris, family = 'binomial')

Pred <-
  data.frame(
    setosa = predict(setosa, type = 'response')
    ,versicolor = predict(versicolor, type = 'response')
    ,virginica = predict(virginica, type = 'response')
  )

Predicted = Pred/rowSums(Pred)
Actual = iris$Species

macc(Actual, Predicted)
```

mcc

Matthews Correlation Coefficient

Description

Calculates the Matthews Correlation Coefficient

Usage

```
mcc(actual, predicted, cutoff)
```

Arguments

actual A vector of the labels
predicted A vector of predicted values
cutoff A cutoff for the predicted values

mlogLoss	<i>Multiclass Log Loss</i>
----------	----------------------------

Description

Calculated the multi-class log loss

Usage

```
mlogLoss(actual, predicted)
```

Arguments

actual	A vector of the labels. Can be numeric, character, or factor
predicted	matrix of predicted values. Can be matrix, data.frame

mse	<i>Mean Square Error</i>
-----	--------------------------

Description

Calculates the mean square error

Usage

```
mse(...)  
  
## Default S3 method:  
mse(actual, predicted, ...)  
  
## S3 method for class 'lm'  
mse(modelObject, ...)  
  
## S3 method for class 'glm'  
mse(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported lm

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')

mse(testDF$y, Preds)
```

msle

Mean Squared Log Error

Description

Calculates the mean square log error

Usage

```
msle(...)

## Default S3 method:
msle(actual, predicted, ...)

## S3 method for class 'lm'
msle(modelObject, ...)

## S3 method for class 'glm'
msle(modelObject, ...)

## S3 method for class 'randomForest'
msle(modelObject, ...)

## S3 method for class 'glmerMod'
msle(modelObject, ...)

## S3 method for class 'gbm'
msle(modelObject, ...)

## S3 method for class 'rpart'
msle(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported glm, randomForest, glmerMod, gbm

npv *Negative Predictive Value*

Description

True Negatives / (True Negatives + False Negatives)

Usage

```
npv(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')

npv(testDF$y, Preds, cutoff = 0)
```

ppv *Positive Predictive Value*

Description

True Positives / (True Positives + False Positives)

Usage

```
ppv(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')

ppv(testDF$y, Preds, cutoff = 0)
precision(testDF$y, Preds, cutoff = 0)
```

recall	<i>Recall, Sensitivity, tpr</i>
--------	---------------------------------

Description

True Positives / (True Positives + False Negatives)

Usage

```
recall(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')

recall(testDF$y, Preds, cutoff = 0)
sensitivity(testDF$y, Preds, cutoff = 0)
tpr(testDF$y, Preds, cutoff = 0)
```

rmse	<i>Root-Mean Square Error</i>
------	-------------------------------

Description

Calculates the root mean square error

Usage

```
rmse(...)  
  
## Default S3 method:  
rmse(actual, predicted, ...)  
  
## S3 method for class 'lm'  
rmse(modelObject, ...)  
  
## S3 method for class 'glm'  
rmse(modelObject, ...)
```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported lm

Examples

```
data(testDF)  
glmModel <- glm(y ~ ., data = testDF, family="binomial")  
Preds <- predict(glmModel, type = 'response')  
  
rmse(testDF$y, Preds)
```

rmsle	<i>Root Mean Squared Log Error</i>
-------	------------------------------------

Description

Calculates the mean square log error

Usage

```

rmsle(...)

## Default S3 method:
rmsle(actual, predicted, ...)

## S3 method for class 'lm'
rmsle(modelObject, ...)

## S3 method for class 'glm'
rmsle(modelObject, ...)

## S3 method for class 'randomForest'
rmsle(modelObject, ...)

## S3 method for class 'glmerMod'
rmsle(modelObject, ...)

## S3 method for class 'gbm'
rmsle(modelObject, ...)

## S3 method for class 'rpart'
rmsle(modelObject, ...)

```

Arguments

...	additional parameters to be passed the the s3 methods
actual	A vector of the labels
predicted	A vector of predicted values
modelObject	the model object. Currently supported glm, randomForest, glmerMod, gbm

testDF	<i>Test data</i>
--------	------------------

Description

Test data

tnr	<i>Specificity, True negative rate</i>
-----	--

Description

True Negatives / (True Negatives + False Positives)

Usage

```
tnr(actual, predicted, cutoff = 0.5)
```

Arguments

actual	A vector of the labels
predicted	A vector of predicted values
cutoff	A cutoff for the predicted values

Examples

```
data(testDF)
glmModel <- glm(y ~ ., data = testDF, family="binomial")
Preds <- predict(glmModel, type = 'response')

tnr(testDF$y, Preds, cutoff = 0)
specificity(testDF$y, Preds, cutoff = 0)
```

Index

auc, [2](#)

brier, [3](#)

ce, [4](#)

confusionMatrix, [5](#)

f1Score, [5](#)

fScore, [6](#)

gini, [6](#)

kappa, [7](#)

logLoss, [8](#)

mae, [9](#)

mauc, [9](#)

mcc, [10](#)

mlogLoss, [11](#)

mse, [11](#)

msle, [12](#)

npv, [13](#)

ppv, [13](#)

precision (ppv), [13](#)

recall, [14](#)

rmse, [15](#)

rmsle, [15](#)

sensitivity (recall), [14](#)

specificity (tnr), [17](#)

testDF, [16](#)

tnr, [17](#)

tpr (recall), [14](#)