Package 'mlr3measures'

October 13, 2022

Title Performance Measures for 'mlr3'

Version 0.5.0

Description Implements multiple performance measures for supervised learning. Includes over 40 measures for regression and classification. Additionally, meta information about the performance measures can be queried, e.g. what the best and worst possible performances scores are.

License LGPL-3

URL https:///mlr3measures.mlr-org.com, https://github.com/mlr-org/mlr3measures

BugReports https://github.com/mlr-org/mlr3measures/issues

Depends R (>= 3.1.0)

Imports checkmate, PRROC

Suggests testthat (>= 3.0.0)

Encoding UTF-8

Config/testthat/edition 3

RoxygenNote 7.2.1

Collate 'assertions.R' 'bibentries.R' 'measures.R' 'binary_auc.R' 'binary_bbrier.R' 'binary_dor.R' 'binary_fbeta.R' 'binary_fdr.R' 'binary_fn.R' 'binary_fnr.R' 'binary_fomr.R' 'binary_fp.R' 'binary_fpr.R' 'binary_mcc.R' 'binary_npv.R' 'binary_ppv.R' 'binary_prauc.R' 'binary_tn.R' 'binary_tnr.R' 'binary_tp.R' 'binary_tpr.R' 'classif_acc.R' 'classif_auc.R' 'classif_bacc.R' 'classif_ce.R' 'classif_logloss.R' 'classif_mbrier.R' 'classif_zero_one.R' 'confusion_matrix.R' 'helper.R' 'regr_ae.R' 'regr_ape.R' 'regr_bias.R' 'regr_ktau.R' 'regr_mae.R' 'regr_mape.R' 'regr_maxae.R' 'regr_maxse.R' 'regr_medae.R' 'regr_medse.R' 'regr_mse.R' 'regr_msle.R' 'regr_pbias.R' 'regr_rae.R' 'regr_rmse.R' 'regr_rmsle.R' 'regr_rrse.R' 'regr_rse.R' 'regr_rsq.R' 'regr_sae.R' 'regr_se.R' 'regr_sle.R' 'regr_smape.R' 'regr_srho.R' 'regr_sse.R' 'roxygen.R' 'similarity_jaccard.R' 'similarity_phi.R' 'zzz.R'

NeedsCompilation no

Author Michel Lang [cre, aut] (<https://orcid.org/0000-0001-9754-0393>),
Martin Binder [ctb]

Maintainer Michel Lang <michellang@gmail.com>

Repository CRAN

Date/Publication 2022-08-05 12:50:02 UTC

R topics documented:

`mlr3measures-package` *mlr3measures: Performance Measures for 'mlr3'*

Description

Implements multiple performance measures for supervised learning. Includes over 40 measures for regression and classification. Additionally, meta information about the performance measures can be queried, e.g. what the best and worst possible performances scores are.

Author(s)

Maintainer: Michel Lang <michellang@gmail.com> (ORCID)

Other contributors:

- Martin Binder <mlr.developer@mb706.com> [contributor]

See Also

Useful links:

- https:///mlr3measures.mlr-org.com
- https://github.com/mlr-org/mlr3measures
- Report bugs at https://github.com/mlr-org/mlr3measures/issues

acc *Classification Accuracy*

Description

Measure to compare true observed labels with predicted labels in multiclass classification tasks.

Usage

```
acc(truth, response, sample_weights = NULL, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the same levels and length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the same levels and length as `truth`.
`sample_weights`	(`numeric()`) Vector of non-negative and finite sample weights. Must have the same length as `truth`. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
`...`	(any) Additional arguments. Currently ignored.

Details

The Classification Accuracy is defined as

$$\frac{1}{n} \sum_{i=1}^{n} w_i \left(t_i = r_i\right).$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"classif"`
- Range: $[0, 1]$
- Minimize: `FALSE`
- Required prediction: `response`

See Also

Other Classification Measures: `bacc()`, `ce()`, `logloss()`, `mauc_aunu()`, `mbrier()`, `zero_one()`

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
acc(truth, response)
```

---

ae *Absolute Error (per observation)*

Description

Calculates the per-observation absolute error as

$$|t_i - r_i| \,.$$

Measure to compare true observed response with predicted response in regression tasks.

Note that this is an unaggregated measure, returning the losses per observation.

Usage

```
ae(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(any) Additional arguments. Currently ignored.

Value

Performance value as `numeric(length(truth))`.

Meta Information

- Type: `"regr"`
- Range (per observation): $[0, \infty)$
- Minimize (per observation): `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ape()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

ape *Absolute Percentage Error (per observation)*

Description

Calculates the per-observation absolute percentage error as

$$\left|\frac{t_i - r_i}{t_i}\right|.$$

Measure to compare true observed response with predicted response in regression tasks.

Note that this is an unaggregated measure, returning the losses per observation.

Usage

```
ape(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Value

Performance value as `numeric(length(truth))`.

Meta Information

- Type: `"regr"`
- Range (per observation): $[0, \infty)$
- Minimize (per observation): `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ae()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

auc *Area Under the ROC Curve*

Description

Measure to compare true observed labels with predicted probabilities in binary classification tasks.

Usage

```
auc(truth, prob, positive, na_value = NaN, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the exactly same two levels and the same length as `response`.
`prob`	(`numeric()`) Predicted probability for positive class. Must have exactly same length as `truth`.
`positive`	(`character(1)`) Name of the positive class.
`na_value`	(`numeric(1)`) Value that should be returned if the measure is not defined for the input (as described in the note). Default is `NaN`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

Computes the area under the Receiver Operator Characteristic (ROC) curve. The AUC can be interpreted as the probability that a randomly chosen positive observation has a higher predicted probability than a randomly chosen negative observation.

This measure is undefined if the true values are either all positive or all negative.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, 1]$
- Minimize: `FALSE`
- Required prediction: `prob`

References

Youden WJ (1950). "Index for rating diagnostic tests." *Cancer*, **3**(1), 32–35. doi:10.1002/1097-0142(1950)3:1<32::aidcncr2820030106>3.0.co;23.

See Also

Other Binary Classification Measures: `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
truth = factor(c("a", "a", "a", "b"))
prob = c(.6, .7, .1, .4)
auc(truth, prob, "a")
```

---

bacc *Balanced Accuracy*

Description

Measure to compare true observed labels with predicted labels in multiclass classification tasks.

Usage

```
bacc(truth, response, sample_weights = NULL, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the same levels and length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the same levels and length as `truth`.
`sample_weights`	(`numeric()`) Vector of non-negative and finite sample weights. Must have the same length as `truth`. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Balanced Accuracy computes the weighted balanced accuracy, suitable for imbalanced data sets. It is defined analogously to the definition in sklearn.

First, the sample weights w are normalized per class:

$$\hat{w_i} = \frac{w_i}{\sum_j 1(y_j = y_i) w_i}.$$

The balanced accuracy is calculated as

$$\frac{1}{\sum_i \hat{w}_i} \sum_i 1(r_i = t_i)\hat{w}_i.$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"classif"`
- Range: $[0, 1]$
- Minimize: `FALSE`
- Required prediction: `response`

References

Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010). "The Balanced Accuracy and Its Posterior Distribution." In *2010 20th International Conference on Pattern Recognition*. doi:10.1109/icpr.2010.764.

Guyon I, Bennett K, Cawley G, Escalante HJ, Escalera S, Ho TK, Macia N, Ray B, Saeed M, Statnikov A, Viegas E (2015). "Design of the 2015 ChaLearn AutoML challenge." In *2015 International Joint Conference on Neural Networks (IJCNN)*. doi:10.1109/ijcnn.2015.7280767.

See Also

Other Classification Measures: `acc()`, `ce()`, `logloss()`, `mauc_aunu()`, `mbrier()`, `zero_one()`

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
bacc(truth, response)
```

---

bbrier *Binary Brier Score*

Description

Measure to compare true observed labels with predicted probabilities in binary classification tasks.

Usage

```
bbrier(truth, prob, positive, sample_weights = NULL, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
prob	(numeric()) Predicted probability for positive class. Must have exactly same length as truth.
positive	(character(1)) Name of the positive class.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
...	(any) Additional arguments. Currently ignored.

Details

The Binary Brier Score is defined as

$$\frac{1}{n}\sum_{i=1}^{n} w_i(I_i - p_i)^2.$$

w_i are the sample weights, I_i is 1 if observation i belongs to the positive class, and 0 otherwise.

Note that this (more common) definition of the Brier score is equivalent to the original definition of the multi-class Brier score (see mbrier()) divided by 2.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: TRUE
- Required prediction: prob

References

https://en.wikipedia.org/wiki/Brier_score

Brier GW (1950). "Verification of forecasts expressed in terms of probability." *Monthly Weather Review*, **78**(1), 1–3. doi:10.1175/15200493(1950)078<0001:vofeit>2.0.co;2.

See Also

Other Binary Classification Measures: auc(), dor(), fbeta(), fdr(), fnr(), fn(), fomr(), fpr(), fp(), mcc(), npv(), ppv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = runif(10)
bbrier(truth, prob, positive = "a")
```

`bias` *Bias*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
bias(truth, response, sample_weights = NULL, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`sample_weights`	(`numeric()`) Vector of non-negative and finite sample weights. Must have the same length as `truth`. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Bias is defined as

$$\frac{1}{n} \sum_{i=1}^{n} w_i \left(t_i - r_i \right).$$

Good predictions score close to 0.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $(-\infty, \infty)$
- Minimize: `NA`
- Required prediction: `response`

See Also

Other Regression Measures: ae(), ape(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
bias(truth, response)
```

---

ce *Classification Error*

Description

Measure to compare true observed labels with predicted labels in multiclass classification tasks.

Usage

```
ce(truth, response, sample_weights = NULL, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the same levels and length as response.
response	(factor()) Predicted response labels. Must have the same levels and length as truth.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
...	(any) Additional arguments. Currently ignored.

Details

The Classification Error is defined as

$$\frac{1}{n} \sum_{i=1}^{n} w_i \left(t_i \neq r_i\right).$$

Value

Performance value as numeric(1).

Meta Information

- Type: "classif"
- Range: $[0, 1]$
- Minimize: TRUE
- Required prediction: response

See Also

Other Classification Measures: acc(), bacc(), logloss(), mauc_aunu(), mbrier(), zero_one()

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
ce(truth, response)
```

---

confusion_matrix *Calculate Binary Confusion Matrix*

Description

Calculates the confusion matrix for a binary classification problem once and then calculates all confusion measures of this package.

Usage

```
confusion_matrix(truth, response, positive, na_value = NaN, relative = FALSE)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
relative	(logical(1)) If TRUE, the returned confusion matrix contains relative frequencies instead of absolute frequencies.

Value

List with two elements:

- `matrix` stores the calculated confusion matrix.
- `measures` stores the metrics as named numeric vector.

Examples

```
set.seed(123)
lvls = c("a", "b")
truth = factor(sample(lvls, 20, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 20, replace = TRUE), levels = lvls)

confusion_matrix(truth, response, positive = "a")
confusion_matrix(truth, response, positive = "a", relative = TRUE)
confusion_matrix(truth, response, positive = "b")
```

---

dor *Diagnostic Odds Ratio*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
dor(truth, response, positive, na_value = NaN, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the exactly same two levels and the same length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the exactly same two levels and the same length as `truth`.
`positive`	(`character(1)`) Name of the positive class.
`na_value`	(`numeric(1)`) Value that should be returned if the measure is not defined for the input (as described in the note). Default is `NaN`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Diagnostic Odds Ratio is defined as

$$\frac{\mathrm{TP}/\mathrm{FP}}{\mathrm{FN}/\mathrm{TN}}.$$

This measure is undefined if FP = 0 or FN = 0.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, \infty)$
- Minimize: `FALSE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
dor(truth, response, positive = "a")
```

---

fbeta *F-beta Score*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fbeta(truth, response, positive, beta = 1, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
beta	(numeric(1)) Parameter to give either precision or recall more weight. Default is 1, resulting in balanced weights.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

With P as precision() and R as recall(), the F-beta Score is defined as

$$(1 + \beta^2) \frac{P \cdot R}{(\beta^2 P) + R}.$$

It measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision. For $\beta = 1$, this measure is called "F1" score.

This measure is undefined if precision or recall is undefined, i.e. TP + FP = 0 or TP + FN = 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: FALSE
- Required prediction: response

References

Rijsbergen, Van CJ (1979). *Information Retrieval*, 2nd edition. Butterworth-Heinemann, Newton, MA, USA. ISBN 408709294.

Goutte C, Gaussier E (2005). "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation." In *Lecture Notes in Computer Science*, 345–359. doi:10.1007/9783-540318651_25.

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fdr(), fnr(), fn(), fomr(), fpr(), fp(), mcc(), npv(), ppv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fbeta(truth, response, positive = "a")
```

---

fdr *False Discovery Rate*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fdr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The False Discovery Rate is defined as

$$\frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}}.$$

This measure is undefined if TP + FP = 0.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, 1]$
- Minimize: `TRUE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fdr(truth, response, positive = "a")
```

---

fn *False Negatives*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fn(truth, response, positive, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the exactly same two levels and the same length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the exactly same two levels and the same length as `truth`.

positive	(character(1)) Name of the positive class.
...	(any) Additional arguments. Currently ignored.

Details

This measure counts the false negatives (type 2 error), i.e. the number of predictions indicating a negative class label while in fact it is positive. This is sometimes also called a "false alarm".

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fomr(), fpr(), fp(), mcc(), npv(), ppv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fn(truth, response, positive = "a")
```

---

fnr *False Negative Rate*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fnr(truth, response, positive, na_value = NaN, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the exactly same two levels and the same length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the exactly same two levels and the same length as `truth`.
`positive`	(`character(1)`) Name of the positive class.
`na_value`	(`numeric(1)`) Value that should be returned if the measure is not defined for the input (as described in the note). Default is `NaN`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The False Negative Rate is defined as

$$\frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}.$$

Also know as "miss rate".

This measure is undefined if TP + FN = 0.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, 1]$
- Minimize: `TRUE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fnr(truth, response, positive = "a")
```

---

fomr *False Omission Rate*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fomr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The False Omission Rate is defined as

$$\frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}}.$$

This measure is undefined if FN + TN = 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: TRUE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fn(), fpr(), fp(), mcc(), npv(), ppv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fomr(truth, response, positive = "a")
```

---

fp *False Positives*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fp(truth, response, positive, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
...	(any) Additional arguments. Currently ignored.

Details

This measure counts the false positives (type 1 error), i.e. the number of predictions indicating a positive class label while in fact it is negative.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fp(truth, response, positive = "a")
```

---

fpr *False Positive Rate*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
fpr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The False Positive Rate is defined as

$$\frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

Also know as fall out or probability of false alarm.

This measure is undefined if FP + TN = 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: TRUE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fn(), fomr(), fp(), mcc(), npv(), ppv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fpr(truth, response, positive = "a")
```

---

jaccard *Jaccard Similarity Index*

Description

Measure to compare two or more sets w.r.t. their similarity.

Usage

```
jaccard(sets, na_value = NaN, ...)
```

Arguments

sets	(list()) List of character or integer vectors. sets must have at least 2 elements.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

For two sets A and B, the Jaccard Index is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

If more than two sets are provided, the mean of all pairwise scores is calculated.

This measure is undefined if two or more sets are empty.

Value

Performance value as numeric(1).

Meta Information

- Type: "similarity"
- Range: $[0, 1]$
- Minimize: FALSE

References

Jaccard, Paul (1901). "Étude comparative de la distribution florale dans une portion des Alpes et du Jura." *Bulletin de la Société Vaudoise des Sciences Naturelles*, **37**, 547-579. doi:10.5169/SEALS-266450.

Bommert A, Rahnenführer J, Lang M (2017). "A Multicriteria Approach to Find Predictive and Sparse Models with Stable Feature Selection for High-Dimensional Data." *Computational and Mathematical Methods in Medicine*, **2017**, 1–18. doi:10.1155/2017/7907163.

Bommert A, Lang M (2021). "stabm: Stability Measures for Feature Selection." *Journal of Open Source Software*, **6**(59), 3010. doi:10.21105/joss.03010.

See Also

Package **stabm** which implements many more stability measures with included correction for chance.

Other Similarity Measures: `phi()`

Examples

```
set.seed(1)
sets = list(
  sample(letters[1:3], 1),
  sample(letters[1:3], 2)
)
jaccard(sets)
```

`ktau` *Kendall's tau*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
ktau(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

Kendall's tau is defined as Kendall's rank correlation coefficient between truth and response. Calls `stats::cor()` with `method` set to `"kendall"`.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[-1, 1]$
- Minimize: `FALSE`
- Required prediction: `response`

References

Rosset S, Perlich C, Zadrozny B (2006). "Ranking-based evaluation of regression models." *Knowledge and Information Systems*, **12**(3), 331–353. doi:10.1007/s1011500600373.

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
ktau(truth, response)
```

---

logloss *Log Loss*

Description

Measure to compare true observed labels with predicted probabilities in multiclass classification tasks.

Usage

```
logloss(truth, prob, sample_weights = NULL, eps = 1e-15, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the same levels and length as response.
prob	(matrix()) Matrix of predicted probabilities, each column is a vector of probabilities for a specific class label. Columns must be named with levels of truth.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
eps	(numeric(1)) Probabilities are clipped to max(eps, min(1 - eps, p)). Otherwise the measure would be undefined for probabilities p = 0 and p = 1.
...	(any) Additional arguments. Currently ignored.

Details

The Log Loss is defined as

$$-\frac{1}{n}\sum_{i=1}^{n} w_i \log\left(p_i\right)$$

where p_i is the probability for the true class of observation i.

Value

Performance value as numeric(1).

Meta Information

- Type: "classif"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: prob

See Also

Other Classification Measures: acc(), bacc(), ce(), mauc_aunu(), mbrier(), zero_one()

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = matrix(runif(3 * 10), ncol = 3, dimnames = list(NULL, lvls))
prob = t(apply(prob, 1, function(x) x / sum(x)))
logloss(truth, prob)
```

mae *Mean Absolute Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
mae(truth, response, sample_weights = NULL, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`sample_weights`	(`numeric()`) Vector of non-negative and finite sample weights. Must have the same length as `truth`. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Mean Absolute Error is defined as

$$\frac{1}{n} \sum_{i=1}^{n} w_i \left| t_i - r_i \right| .$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
mae(truth, response)
```

---

mape *Mean Absolute Percent Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
mape(truth, response, sample_weights = NULL, na_value = NaN, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Mean Absolute Percent Error is defined as

$$\frac{1}{n} \sum_{i=1}^{n} w_i \left| \frac{t_i - r_i}{t_i} \right|.$$

This measure is undefined if any element of t is 0.

Value

Performance value as numeric(1).

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

References

de Myttenaere, Arnaud, Golden, Boris, Le Grand, Bénédicte, Rossi, Fabrice (2016). "Mean Absolute Percentage Error for regression models." *Neurocomputing*, **192**, 38-48. ISSN 0925-2312, doi:10.1016/j.neucom.2015.12.114.

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mae()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
mape(truth, response)
```

---

mauc_aunu *Multiclass AUC Scores*

Description

Measure to compare true observed labels with predicted probabilities in multiclass classification tasks.

Usage

```
mauc_aunu(truth, prob, na_value = NaN, ...)

mauc_aunp(truth, prob, na_value = NaN, ...)

mauc_au1u(truth, prob, na_value = NaN, ...)

mauc_au1p(truth, prob, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the same levels and length as response.
prob	(matrix()) Matrix of predicted probabilities, each column is a vector of probabilities for a specific class label. Columns must be named with levels of truth.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

Multiclass AUC measures.

- *AUNU*: AUC of each class against the rest, using the uniform class distribution. Computes the AUC treating a c-dimensional classifier as c two-dimensional 1-vs-rest classifiers, where classes are assumed to have uniform distribution, in order to have a measure which is independent of class distribution change (Fawcett 2001).
- *AUNP*: AUC of each class against the rest, using the a-priori class distribution. Computes the AUC treating a c-dimensional classifier as c two-dimensional 1-vs-rest classifiers, taking into account the prior probability of each class (Fawcett 2001).
- *AU1U*: AUC of each class against each other, using the uniform class distribution. Computes something like the AUC of c(c - 1) binary classifiers (all possible pairwise combinations). See Hand (2001) for details.
- *AU1P*: AUC of each class against each other, using the a-priori class distribution. Computes something like AUC of c(c - 1) binary classifiers while considering the a-priori distribution of the classes as suggested in Ferri (2009). Note we deviate from the definition in Ferri (2009) by a factor of c. The person implementing this function and writing this very documentation right now cautions against using this measure because it is an imperfect generalization of AU1U.

Value

Performance value as numeric(1).

Meta Information

- Type: "classif"
- Range: $[0, 1]$
- Minimize: FALSE
- Required prediction: prob

References

Fawcett, Tom (2001). "Using rule sets to maximize ROC performance." In *Proceedings 2001 IEEE international conference on data mining*, 131–138. IEEE.

Ferri, César, Hernández-Orallo, José, Modroiu, R (2009). "An experimental comparison of performance measures for classification." *Pattern Recognition Letters*, **30**(1), 27–38. doi:10.1016/j.patrec.2008.08.010.

Hand, J D, Till, J R (2001). "A simple generalisation of the area under the ROC curve for multiple class classification problems." *Machine learning*, **45**(2), 171–186.

See Also

Other Classification Measures: `acc()`, `bacc()`, `ce()`, `logloss()`, `mbrier()`, `zero_one()`

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = matrix(runif(3 * 10), ncol = 3)
colnames(prob) = levels(truth)
mauc_aunu(truth, prob)
```

---

maxae *Max Absolute Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
maxae(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(any) Additional arguments. Currently ignored.

Details

The Max Absolute Error is defined as

$$\max\left(|t_i - r_i|\right).$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
maxae(truth, response)
```

---

maxse *Max Squared Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
maxse(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Max Squared Error is defined as

$$\max \left(t_i - r_i\right)^2 .$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
maxse(truth, response)
```

---

mbrier *Multiclass Brier Score*

Description

Measure to compare true observed labels with predicted probabilities in multiclass classification tasks.

Usage

```
mbrier(truth, prob, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the same levels and length as `response`.
`prob`	(`matrix()`) Matrix of predicted probabilities, each column is a vector of probabilities for a specific class label. Columns must be named with levels of `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

Brier score for multi-class classification problems with r labels defined as

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{r} (I_{ij} - p_{ij})^2.$$

I_{ij} is 1 if observation i has true label j, and 0 otherwise.

Note that there also is the more common definition of the Brier score for binary classification problems in `bbrier()`.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"classif"`
- Range: $[0, 2]$
- Minimize: `TRUE`
- Required prediction: `prob`

References

Brier GW (1950). "Verification of forecasts expressed in terms of probability." *Monthly Weather Review*, **78**(1), 1–3. doi:10.1175/15200493(1950)078<0001:vofeit>2.0.co;2.

See Also

Other Classification Measures: `acc()`, `bacc()`, `ce()`, `logloss()`, `mauc_aunu()`, `zero_one()`

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = matrix(runif(3 * 10), ncol = 3)
colnames(prob) = levels(truth)
mbrier(truth, prob)
```

mcc *Matthews Correlation Coefficient*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
mcc(truth, response, positive, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
...	(any) Additional arguments. Currently ignored.

Details

The Matthews Correlation Coefficient is defined as

$$\frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}.$$

This above formula is undefined if any of the four sums in the denominator is 0. The denominator is then set to 1.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[-1, 1]$
- Minimize: FALSE
- Required prediction: response

References

Matthews BW (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." *Biochimica et Biophysica Acta (BBA) - Protein Structure*, **405**(2), 442–451. doi:10.1016/00052795(75)901099.

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
mcc(truth, response, positive = "a")
```

`measures` *Measure Registry*

Description

The `environment()` `measures` keeps track of all measures in this package. It stores meta information such as minimum, maximum or if the measure must be minimized or maximized. The following information is available for each measure:

- `id`: Name of the measure.
- `title`: Short descriptive title.
- `type`: `"binary"` for binary classification, `"classif"` for binary or multi-class classification, `"regr"` for regression and `"similarity"` for similarity measures.
- `lower`: lower bound.
- `upper`: upper bound.
- `predict_type`: prediction type the measure operates on. `"response"` corresponds to class labels for classification and the numeric response for regression. `"prob"` corresponds to class probabilities, provided as a matrix with class labels as column names. `"se"` corresponds to to the vector of predicted standard errors for regression.
- `minimize`: If `TRUE` or `FALSE`, the objective is to minimize or maximize the measure, respectively. Can also be `NA`.
- `obs_loss`: Name of the function which is called to calculate the (unaggregated) loss per observation.
- `aggregated`: If `TRUE`, this function aggregates the losses to a single numeric value. Otherwise, a vector of losses is returned.
- `sample_weights`: If `TRUE`, it is possible calculate a weighted measure.

Usage

```
measures
```

Format

An object of class `environment` of length 59.

Examples

```
names(measures)
measures$tpr
```

`medae` *Median Absolute Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
medae(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Median Absolute Error is defined as

$$\underset{i}{\text{median}} \left| t_i - r_i \right| .$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
medae(truth, response)
```

---

medse *Median Squared Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
medse(truth, response, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
...	(any) Additional arguments. Currently ignored.

Details

The Median Squared Error is defined as

$$\underset{i}{\text{median}} \left[(t_i - r_i)^2 \right].$$

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
medse(truth, response)
```

---

mse *Mean Squared Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
mse(truth, response, sample_weights = NULL, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
...	(any) Additional arguments. Currently ignored.

Details

The Mean Squared Error is defined as

$$\frac{1}{n} w_i \sum_{i=1}^{n} (t_i - r_i)^2 .$$

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
mse(truth, response)
```

---

msle *Mean Squared Log Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
msle(truth, response, sample_weights = NULL, na_value = NaN, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Mean Squared Log Error is defined as

$$\frac{1}{n} \sum_{i=1}^{n} w_i \left(\ln(1 + t_i) - \ln(1 + r_i) \right)^2 .$$

This measure is undefined if any element of t or r is less than or equal to -1.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
msle(truth, response)
```

---

npv *Negative Predictive Value*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
npv(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Negative Predictive Value is defined as

$$\frac{\mathrm{TN}}{\mathrm{FN} + \mathrm{TN}}.$$

This measure is undefined if FN + TN = 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: FALSE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fn(), fomr(), fpr(), fp(), mcc(), ppv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
npv(truth, response, positive = "a")
```

---

pbias *Percent Bias*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
pbias(truth, response, sample_weights = NULL, na_value = NaN, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`sample_weights`	(`numeric()`) Vector of non-negative and finite sample weights. Must have the same length as `truth`. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
`na_value`	(`numeric(1)`) Value that should be returned if the measure is not defined for the input (as described in the note). Default is `NaN`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Percent Bias is defined as

$$\frac{1}{n}\sum_{i=1}^{n} w_i \frac{(t_i - r_i)}{|t_i|}.$$

Good predictions score close to 0.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $(-\infty, \infty)$
- Minimize: NA
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
pbias(truth, response)
```

---

phi *Phi Coefficient Similarity*

Description

Measure to compare two or more sets w.r.t. their similarity.

Usage

```
phi(sets, p, na_value = NaN, ...)
```

Arguments

sets	(list()) List of character or integer vectors. sets must have at least 2 elements.
p	(integer(1)) Total number of possible elements.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Phi Coefficient is defined as the Pearson correlation between the binary representation of two sets A and B. The binary representation for A is a logical vector of length p with the i-th element being 1 if the corresponding element is in A, and 0 otherwise.

If more than two sets are provided, the mean of all pairwise scores is calculated.

This measure is undefined if one set contains none or all possible elements.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"similarity"`
- Range: $[-1, 1]$
- Minimize: `FALSE`

References

Nogueira S, Brown G (2016). "Measuring the Stability of Feature Selection." In *Machine Learning and Knowledge Discovery in Databases*, 442–457. Springer International Publishing. doi:10.1007/9783319462271_28.

Bommert A, Rahnenführer J, Lang M (2017). "A Multicriteria Approach to Find Predictive and Sparse Models with Stable Feature Selection for High-Dimensional Data." *Computational and Mathematical Methods in Medicine*, **2017**, 1–18. doi:10.1155/2017/7907163.

Bommert A, Lang M (2021). "stabm: Stability Measures for Feature Selection." *Journal of Open Source Software*, **6**(59), 3010. doi:10.21105/joss.03010.

See Also

Package **stabm** which implements many more stability measures with included correction for chance.

Other Similarity Measures: `jaccard()`

Examples

```
set.seed(1)
sets = list(
  sample(letters[1:3], 1),
  sample(letters[1:3], 2)
)
phi(sets, p = 3)
```

ppv *Positive Predictive Value*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
ppv(truth, response, positive, na_value = NaN, ...)

precision(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Positive Predictive Value is defined as

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$

Also know as "precision".

This measure is undefined if TP + FP = 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: FALSE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

Goutte C, Gaussier E (2005). "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation." In *Lecture Notes in Computer Science*, 345–359. doi:10.1007/9783-540318651_25.

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fn(), fomr(), fpr(), fp(), mcc(), npv(), prauc(), tnr(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
ppv(truth, response, positive = "a")
```

---

prauc *Area Under the Precision-Recall Curve*

Description

Measure to compare true observed labels with predicted probabilities in binary classification tasks.

Usage

```
prauc(truth, prob, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
prob	(numeric()) Predicted probability for positive class. Must have exactly same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

Computes the area under the Precision-Recall curve (PRC). The PRC can be interpreted as the relationship between precision and recall (sensitivity), and is considered to be a more appropriate measure for unbalanced datasets than the ROC curve. The PRC is computed by integration of the piecewise function.

This measure is undefined if the true values are either all positive or all negative.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, 1]$
- Minimize: `FALSE`
- Required prediction: `prob`

References

Davis J, Goadrich M (2006). "The relationship between precision-recall and ROC curves." In *Proceedings of the 23rd International Conference on Machine Learning*. ISBN 9781595933836.

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
truth = factor(c("a", "a", "a", "b"))
prob = c(.6, .7, .1, .4)
prauc(truth, prob, "a")
```

---

rae *Relative Absolute Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
rae(truth, response, na_value = NaN, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Relative Absolute Error is defined as

$$\frac{\sum_{i=1}^{n} |t_i - r_i|}{\sum_{i=1}^{n} |t_i - \bar{t}|}.$$

Can be interpreted as absolute error of the predictions relative to a naive model predicting the mean.

This measure is undefined for constant t.

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rae(truth, response)
```

rmse *Root Mean Squared Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
rmse(truth, response, sample_weights = NULL, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
sample_weights	(numeric()) Vector of non-negative and finite sample weights. Must have the same length as truth. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
...	(any) Additional arguments. Currently ignored.

Details

The Root Mean Squared Error is defined as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} w_i \left(t_i - r_i\right)^2}.$$

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rmse(truth, response)
```

`rmsle` *Root Mean Squared Log Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
rmsle(truth, response, sample_weights = NULL, na_value = NaN, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`sample_weights`	(`numeric()`) Vector of non-negative and finite sample weights. Must have the same length as `truth`. The vector gets automatically normalized to sum to one. Defaults to equal sample weights.
`na_value`	(`numeric(1)`) Value that should be returned if the measure is not defined for the input (as described in the note). Default is `NaN`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Root Mean Squared Log Error is defined as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} w_i \left(\ln(1+t_i) - \ln(1+r_i)\right)^2}.$$

This measure is undefined if any element of t or r is less than or equal to -1.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rmsle(truth, response)
```

---

rrse *Root Relative Squared Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
rrse(truth, response, na_value = NaN, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Root Relative Squared Error is defined as

$$\sqrt{\frac{\sum_{i=1}^{n} (t_i - r_i)^2}{\sum_{i=1}^{n} (t_i - \bar{t})^2}}.$$

Can be interpreted as root of the squared error of the predictions relative to a naive model predicting the mean.

This measure is undefined for constant t.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $[0, \infty)$
- Minimize: `TRUE`
- Required prediction: `response`

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `srho()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rrse(truth, response)
```

---

rse *Relative Squared Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
rse(truth, response, na_value = NaN, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Relative Squared Error is defined as

$$\frac{\sum_{i=1}^{n} (t_i - r_i)^2}{\sum_{i=1}^{n} (t_i - \bar{t})^2}.$$

Can be interpreted as squared error of the predictions relative to a naive model predicting the mean.

This measure is undefined for constant t.

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rsq(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rse(truth, response)
```

`rsq` *R Squared*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
rsq(truth, response, na_value = NaN, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`na_value`	(`numeric(1)`) Value that should be returned if the measure is not defined for the input (as described in the note). Default is `NaN`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

R Squared is defined as

$$1 - \frac{\sum_{i=1}^{n} (t_i - r_i)^2}{\sum_{i=1}^{n} (t_i - \bar{t})^2}.$$

Also known as coefficient of determination or explained variation. Subtracts the `rse()` from 1, hence it compares the squared error of the predictions relative to a naive model predicting the mean.

This measure is undefined for constant t.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"regr"`
- Range: $(-\infty, 1]$
- Minimize: `FALSE`
- Required prediction: `response`

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), sae(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rsq(truth, response)
```

---

sae *Sum of Absolute Errors*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
sae(truth, response, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
...	(any) Additional arguments. Currently ignored.

Details

The Sum of Absolute Errors is defined as

$$\sum_{i=1}^{n} |t_i - r_i| \,.$$

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), se(), sle(), smape(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
sae(truth, response)
```

---

se *Squared Error (per observation)*

Description

Calculates the per-observation squared error as

$$(t_i - r_i)^2 .$$

Measure to compare true observed response with predicted response in regression tasks.

Note that this is an unaggregated measure, returning the losses per observation.

Usage

```
se(truth, response, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
...	(any) Additional arguments. Currently ignored.

Value

Performance value as numeric(length(truth)).

Meta Information

- Type: "regr"
- Range (per observation): $[0, \infty)$
- Minimize (per observation): TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), sle(), smape(), srho(), sse()

---

sle *Squared Log Error (per observation)*

Description

Calculates the per-observation squared error as

$$(\ln(1 + t_i) - \ln(1 + r_i))^2 .$$

Measure to compare true observed response with predicted response in regression tasks.

Note that this is an unaggregated measure, returning the losses per observation.

Usage

```
sle(truth, response, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
...	(any) Additional arguments. Currently ignored.

Value

Performance value as numeric(length(truth)).

Meta Information

- Type: "regr"
- Range (per observation): $[0, \infty)$
- Minimize (per observation): TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), smape(), srho(), sse()

---

smape *Symmetric Mean Absolute Percent Error*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
smape(truth, response, na_value = NaN, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The Symmetric Mean Absolute Percent Error is defined as

$$\frac{2}{n} \sum_{i=1}^{n} \frac{|t_i - r_i|}{|t_i| + |r_i|}.$$

This measure is undefined if if any $|t| + |r|$ is 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, 2]$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), srho(), sse()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
smape(truth, response)
```

---

srho *Spearman's rho*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
srho(truth, response, ...)
```

Arguments

truth	(numeric()) True (observed) values. Must have the same length as response.
response	(numeric()) Predicted response values. Must have the same length as truth.
...	(any) Additional arguments. Currently ignored.

Details

Spearman's rho is defined as Spearman's rank correlation coefficient between truth and response. Calls stats::cor() with method set to "spearman".

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[-1, 1]$
- Minimize: FALSE
- Required prediction: response

References

Rosset S, Perlich C, Zadrozny B (2006). "Ranking-based evaluation of regression models." *Knowledge and Information Systems*, **12**(3), 331–353. doi:10.1007/s1011500600373.

See Also

Other Regression Measures: `ae()`, `ape()`, `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `se()`, `sle()`, `smape()`, `sse()`

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
srho(truth, response)
```

---

sse *Sum of Squared Errors*

Description

Measure to compare true observed response with predicted response in regression tasks.

Usage

```
sse(truth, response, ...)
```

Arguments

`truth`	(`numeric()`) True (observed) values. Must have the same length as `response`.
`response`	(`numeric()`) Predicted response values. Must have the same length as `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Details

The Sum of Squared Errors is defined as

$$\sum_{i=1}^{n} (t_i - r_i)^2 .$$

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: ae(), ape(), bias(), ktau(), mae(), mape(), maxae(), maxse(), medae(), medse(), mse(), msle(), pbias(), rae(), rmse(), rmsle(), rrse(), rse(), rsq(), sae(), se(), sle(), smape(), srho()

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
sse(truth, response)
```

---

tn *True Negatives*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
tn(truth, response, positive, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
...	(any) Additional arguments. Currently ignored.

Details

This measure counts the true negatives, i.e. the number of predictions correctly indicating a negative class label.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, \infty)$
- Minimize: `FALSE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tn(truth, response, positive = "a")
```

---

tnr *True Negative Rate*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
tnr(truth, response, positive, na_value = NaN, ...)

specificity(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The True Negative Rate is defined as

$$\frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}.$$

Also know as "specificity".

This measure is undefined if FP + TN = 0.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: $[0, 1]$
- Minimize: FALSE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fn(), fomr(), fpr(), fp(), mcc(), npv(), ppv(), prauc(), tn(), tpr(), tp()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tnr(truth, response, positive = "a")
```

---

tp *True Positives*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
tp(truth, response, positive, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the exactly same two levels and the same length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the exactly same two levels and the same length as `truth`.
`positive`	(`character(1)`) Name of the positive class.
`...`	(`any`) Additional arguments. Currently ignored.

Details

This measure counts the true positives, i.e. the number of predictions correctly indicating a positive class label.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, \infty)$
- Minimize: `FALSE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: auc(), bbrier(), dor(), fbeta(), fdr(), fnr(), fn(), fomr(), fpr(), fp(), mcc(), npv(), ppv(), prauc(), tnr(), tn(), tpr()

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tp(truth, response, positive = "a")
```

---

tpr *True Positive Rate*

Description

Measure to compare true observed labels with predicted labels in binary classification tasks.

Usage

```
tpr(truth, response, positive, na_value = NaN, ...)

recall(truth, response, positive, na_value = NaN, ...)

sensitivity(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	(factor()) True (observed) labels. Must have the exactly same two levels and the same length as response.
response	(factor()) Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	(character(1)) Name of the positive class.
na_value	(numeric(1)) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	(any) Additional arguments. Currently ignored.

Details

The True Positive Rate is defined as

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

Also know as "recall" or "sensitivity".

This measure is undefined if TP + FN = 0.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: `"binary"`
- Range: $[0, 1]$
- Minimize: `FALSE`
- Required prediction: `response`

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

Goutte C, Gaussier E (2005). "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation." In *Lecture Notes in Computer Science*, 345–359. doi:10.1007/9783-540318651_25.

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `tnr()`, `tn()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tpr(truth, response, positive = "a")
```

`zero_one` *Zero-One Classification Loss (per observation)*

Description

Calculates the per-observation 0/1 loss as

$$t_i \neq r_1.$$

Measure to compare true observed labels with predicted labels in multiclass classification tasks.

Note that this is an unaggregated measure, returning the losses per observation.

Usage

```
zero_one(truth, response, ...)
```

Arguments

`truth`	(`factor()`) True (observed) labels. Must have the same levels and length as `response`.
`response`	(`factor()`) Predicted response labels. Must have the same levels and length as `truth`.
`...`	(`any`) Additional arguments. Currently ignored.

Value

Performance value as `numeric(length(truth))`.

Meta Information

- Type: `"classif"`
- Range (per observation): $[0, 1]$
- Minimize (per observation): `TRUE`
- Required prediction: `response`

See Also

Other Classification Measures: `acc()`, `bacc()`, `ce()`, `logloss()`, `mauc_aunu()`, `mbrier()`

Index