Package: rtemis.ml (via r-universe)

November 12, 2026

Title Performance Metrics for Classification

Version 0.1.0

Date 2026-11-10

Description Computes the area under the receiver operating characteristic curve (AUC) and the average precision of binary classifiers, and their one-vs-rest values for each class of a multiclass outcome. The AUC is the Mann-Whitney U statistic divided by the number of positive-negative pairs (Hanley and McNeil (1982) <doi:10.1148/radiology.143.1.7063747>), and the average precision is the area under the stepwise precision-recall curve. The computations are implemented in the 'rtemis' 'Rust' crate, which sorts the scores once per problem and reads the inputs without copying them.

URL https://www.rtemis.org, https://github.com/rtemis-org/rtemis.ml

BugReports https://github.com/rtemis-org/rtemis.ml/issues

License BSD_3_clause + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.2)

Imports utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/testthat/parallel true

Roxygen list(markdown = TRUE)

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.71.0, xz

Config/roxygen2/version 8.1.1

Config/pak/sysreqs xz-utils libclang-dev

Repository https://rtemis-org.r-universe.dev

Date/Publication 2026-11-12 01:06:27 UTC

RemoteUrl https://github.com/rtemis-org/rtemis.ml

RemoteRef HEAD

RemoteSha 22cb9d5db4a5b65767038ab99d0a85efaa04870a

Contents

average_precision 2
average_precision_ovr 3
roc_auc 4
roc_auc_ovr 5

Index **6**

`average_precision` *Average precision*

Description

Computes the average precision (AP) of a binary classifier: the area under the stepwise precision-recall curve. Each distinct score is one threshold, and the cases tied at it enter together. Thresholds are numbered from the highest score down; with recall R_k and precision P_k at threshold k, $AP = \sum_k (R_k - R_{k-1}) P_k$, with $R_0 = 0$.

Usage

```
average_precision(reference, scores)
```

Arguments

`reference`	Integer {0, 1}: Reference label of each case, `1L` for the positive class and `0L` for the negative class.
`scores`	Numeric: Score of each case, where a higher value indicates the positive class, such as the predicted probability of the positive class. A double vector the same length as `reference`.

Details

The AP depends on the scores only through their order. The value of a classifier that scores independently of the reference labels is expected to be near the prevalence of the positive class. The inputs are read in place, without copying. Extra memory is 8 bytes per case, and time is O(n log n).

Value

Double: the AP, between 0 and 1. `NaN` when `reference` contains no positive case; 1 when it contains no negative case. Errors as `roc_auc()`.

See Also

average_precision_ovr() for multiclass outcomes, roc_auc() for the area under the ROC curve.

Examples

```
average_precision(c(0L, 0L, 1L, 1L), c(0.1, 0.4, 0.35, 0.8))
```

---

average_precision_ovr *One-vs-rest average precision*

Description

Computes, for each class of a multiclass outcome, the average precision of that class against all other classes, as average_precision() does for a binary outcome. The macro-averaged AP is the mean of the result.

Usage

```
average_precision_ovr(reference, scores)
```

Arguments

reference	Factor: Reference label of each case.
scores	Numeric matrix: Score of each case for each class, where a higher value indicates that class, such as the predicted class probabilities. A double matrix with one row per case and one column per level of reference, in level order.

Details

The inputs are read in place, without copying. Extra memory is 8 bytes per case, reused across classes, and time is O(k n log n) for k classes.

Value

Named double vector: the AP of each class, named by the levels of reference. NaN for a class that is absent. Errors as roc_auc_ovr().

See Also

roc_auc_ovr()

Examples

```
reference <- factor(c("a", "b", "c", "a"))
scores <- matrix(
  c(0.8, 0.1, 0.1, 0.6, 0.1, 0.7, 0.2, 0.2, 0.1, 0.2, 0.7, 0.2),
  ncol = 3L
)
average_precision_ovr(reference, scores)
```

roc_auc *Area under the ROC curve*

Description

Computes the area under the receiver operating characteristic (ROC) curve of a binary classifier: the probability that a randomly chosen positive case receives a higher score than a randomly chosen negative case, with ties counted as one half.

Usage

```
roc_auc(reference, scores)
```

Arguments

`reference`	Integer {0, 1}: Reference label of each case, `1L` for the positive class and `0L` for the negative class.
`scores`	Numeric: Score of each case, where a higher value indicates the positive class, such as the predicted probability of the positive class. A double vector the same length as `reference`.

Details

The AUC depends on the scores only through their order, so any score where a higher value indicates the positive class gives the same result as its predicted probability. The inputs are read in place, without copying. Extra memory is 8 bytes per case, and time is O(n log n).

Value

Double: the AUC, between 0 and 1. `NaN` when `reference` contains only one class or is empty. An error is raised when the lengths differ, when `reference` contains a value other than 0 or 1, including `NA`, or when `scores` contains `NA` or `NaN`.

See Also

`roc_auc_ovr()` for multiclass outcomes, `average_precision()` for the area under the precision-recall curve.

Examples

```
roc_auc(c(0L, 0L, 1L, 1L), c(0.1, 0.4, 0.35, 0.8))
```

roc_auc_ovr *One-vs-rest area under the ROC curve*

Description

Computes, for each class of a multiclass outcome, the AUC of that class against all other classes, as `roc_auc()` does for a binary outcome. The macro-averaged AUC is the mean of the result.

Usage

```
roc_auc_ovr(reference, scores)
```

Arguments

`reference`	Factor: Reference label of each case.
`scores`	Numeric matrix: Score of each case for each class, where a higher value indicates that class, such as the predicted class probabilities. A double matrix with one row per case and one column per level of `reference`, in level order.

Details

The inputs are read in place, without copying. Extra memory is 8 bytes per case, reused across classes, and time is O(k n log n) for k classes.

Value

Named double vector: the AUC of each class, named by the levels of `reference`. `NaN` for a class that is absent or is the only class present. An error is raised when `reference` is not a factor or contains `NA`, when the dimensions of `scores` do not match, or when `scores` contains `NA` or `NaN`.

See Also

`average_precision_ovr()`

Examples

```
reference <- factor(c("a", "b", "c", "a"))
scores <- matrix(
  c(0.8, 0.1, 0.1, 0.6, 0.1, 0.7, 0.2, 0.2, 0.1, 0.2, 0.7, 0.2),
  ncol = 3L
)
roc_auc_ovr(reference, scores)
```

Index

average_precision, 2
average_precision(), *3, 4*
average_precision_ovr, 3
average_precision_ovr(), *3*, *5*

roc_auc, 4
roc_auc(), *2, 3*, *5*
roc_auc_ovr, 5
roc_auc_ovr(), *3, 4*