Package: mltest (via r-universe)

September 17, 2024

Title Classification Evaluation Metrics

Version 1.0.1

Description A fast, robust and easy-to-use calculation of multi-class classification evaluation metrics based on confusion matrix.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation no

Author G. Dudnik [aut, cre, cph]

Maintainer G. Dudnik <gl.dudnik@gmail.com>

Repository CRAN

Date/Publication 2018-11-16 17:00:03 UTC

Contents

`ml_test` *multi-class classifier evaluation metrics based on a confusion matrix (contingency table)*

Description

Calculates multi-class classification evaluation metrics: **balanced.accuracy**, balanced accuracy (**balanced.accuracy**), diagnostic odds ratio (**DOR**), error rate (**error.rate**), F.beta (**F0.5**, **F1** (F-measure, F-score), **F2** with where beta is 0.5, 1 and 2 respectively), false positive rate (**FPR**), false negative rate (**FNR**), false omission rate ((**FOR**)), false discovery rate (**FDR**), geometric mean (**geometric.mean**), **Jaccard**, positive likelihood ratio (p+, LR(+) or simply **L**), negative likelihood ratio

(p-, LR(-) or simply **lambda**), Matthews corellation coefficient (**MCC**), markedness (**MK**), negative predictive value (**NPV**), optimization precision **OP**, **precision**, **recall** (sensitivity), **specificity** and finally **Youden**'s index. The function calculates the aforementioned metrics from a confusion matrix (contingency matrix) where *TP*, *TN*, *FP FN* are abbreviations for *true positives*, *true negatives*, *false positives* and *false negatives* respectively.

Usage

```
ml_test(predicted, true, output.as.table = FALSE)
```

Arguments

`predicted`	class labels predicted by the classifier model (a set of classes convertible into type factor with levels representing labels)
`true`	true class labels (a set of classes convertible into type factor of the same length and with the same levels as predicted)
`output.as.table`	the function returns all metrics except for **accuracy** and **error.rate** in a tabular format if this argument is set to *TRUE*

Value

the function returns a list of following metrics:

`\strong{accuracy}` *calculated as:* (TP+TN) / (TP+FP+TN+FN) *(doesn't show up when output.as.table = TRUE)*

`\strong{balanced.accuracy}` *calculated as:* (TP / (TP+FN)+TN / (TN+FP)) / 2 = (recall+specificity) / 2

`\strong{DOR}` *calculated as:* TP*TN / (FP*FN) = L / lambda

`\strong{error.rate}` *calculated as:* (FP+FN) / (TP+TN+FP+FN) = 1-accuracy *(doesn't show up when output.as.table = TRUE)*

`\strong{F0.5}` *calculated as:* 1.25*(recall*precision/(0.25*precision+recall))

`\strong{F1}` *calculated as:* 2*(precision*recall / (precision+recall))

`\strong{F2}` *calculated as:* 5*(precision*recall / (4*precision+recall))

`\strong{FDR}` *calculated as:* 1-precision

`\strong{FNR}` *calculated as:* 1-recall

`\strong{FOR}` *calculated as:* 1-NPV

`\strong{FPR}` *calculated as:* 1-specificity

`\strong{geometric.mean}` *calculated as:* (recall*specificity)^0.5

`\strong{Jaccard}` *calculated as:* TP / (TP+FP+FN)

`\strong{L}` *calculated as:* recall / (1-specificity)

\strong{lambda}
: *calculated as:* (1-recall) / (specificity)

\strong{MCC}
: *calculated as:* (TP*TN-FP*FN) / (((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))^0.5)

\strong{MK}
: *calculated as:* precision + NPV - 1

\strong{NPV}
: *calculated as:* TN / (TN+FN)

\strong{OP}
: *calculated as:* accuracy - |recall-specificity| / (recall+specificity)

\strong{precision}
: *calculated as:* TP / (TP+FP)

\strong{recall}
: *calculated as:* TP / (TP+FN)

\strong{specificity}
: *calculated as:* TN / (TN+FP)

\strong{Youden}
: *calculated as:* recall+specificity-1

Author(s)

G. Dudnik

References

1. Sasaki Y. (2007). The truth of the F-measure.:1–5. https://www.researchgate.net/publication/268185911_The_truth_of_the_F-measure.
2. Powers DMW. (2011). Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness & Correlation. Arch Geschwulstforsch. 2(1):37–63. https://www.researchgate.net/publication/313610493_Evaluation_From_precision_recall_and_fmeasure_to_roc_informedness_markedness_and_correlation.
3. Bekkar M, Djemaa HK, Alitouche TA. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. J Inf Eng Appl. 3(10):27–38. https://www.iiste.org/Journals/index.php/JIEA/article/view/7633.
4. Jeni LA, Cohn JF, De La Torre F. (2013). Facing Imbalanced Data Recommendations for the Use of Performance Metrics. Conference on Affective Computing and Intelligent Interaction. IEEE. p. 245–51. http://ieeexplore.ieee.org/document/6681438/.
5. López V, Fernández A, García S, Palade V, Herrera F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf Sci. 250:113–41. http://dx.doi.org/10.1016/j.ins.2013.07.007.
6. Tharwat A. (2018). Classification assessment methods. Appl Comput Informatics . https://linkinghub.elsevier.com/retrieve/pii/S2210832718301546.

Examples

```
library(mltest)

# class labels ("cat, "dog" and "rat") predicted by the classifier model
predicted_labels <- as.factor(c("dog", "cat", "dog", "rat", "rat"))
```

```
# true labels (test set)
true_labels <- as.factor(c("dog", "cat", "dog", "rat", "dog"))

classifier_metrics <- ml_test(predicted_labels, true_labels, output.as.table = FALSE)

# overall classification accuracy
accuracy <- classifier_metrics$accuracy

# F1-measures for classes "cat", "dog" and "rat"
F1 <- classifier_metrics$F1

# tabular view of the metrics (except for 'accuracy' and 'error.rate')
classifier_metrics <- ml_test(predicted_labels, true_labels, output.as.table = TRUE)
```

Index