

Package: conformalpvalue (via r-universe)

October 9, 2024

Title Computes Conformal p-Values

Version 0.1.0

Description Computes marginal conformal p-values using conformal prediction in binary classification tasks. Conformal prediction is a framework that augments machine learning algorithms with a measure of uncertainty, in the form of prediction regions that attain a user-specified level of confidence. This package specifically focuses on providing conformal p-values that can be used to assess the confidence of the classification predictions. For more details, see Tyagi and Guo (2023) <<https://proceedings.mlr.press/v204/tyagi23a.html>>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports e1071, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Chhavi Tyagi [aut, cre]

Maintainer Chhavi Tyagi <tyagi.chhavi2222@gmail.com>

Date/Publication 2023-10-05 07:10:02 UTC

Repository <https://chhavi-tyagi.r-universe.dev>

RemoteUrl <https://github.com/cran/conformalpvalue>

RemoteRef HEAD

RemoteSha 393908433cc3abd40e014700d0daa9efaf9d1e8b

Contents

conformal_pvalues	2
Index	3

conformal_pvalues *Conformal P-values Calculation*

Description

This function calculates conformal p-values based of binary class labels for test data.

Usage

```
conformal_pvalues(train_data, calib_data, test_data, target_col, method)
```

Arguments

train_data	A data frame containing the training data with the target variable.
calib_data	A data frame containing the calibration data with the target variable.
test_data	A data frame containing the test data.
target_col	The name of the target variable column.
method	A character string specifying the classification method to use. Options are 'naiveBayes', 'svm', and 'glm'. This function trains a Naive Bayes classifier, computes non-conformity scores on the calibration data and test data, and calculates conformal p-values of both classes "0" and "1" using the conformal prediction for a binary classification problem.

Value

A matrix containing p-values for each test case and class.

Examples

```
# Create dummy train_data, calib_data, and test_data
train_data <- data.frame(
  x1 = as.numeric(rnorm(50, 1, 2)),
  x2 = as.numeric(rnorm(50, 2.5, 3)),
  target = as.factor(rbinom(50, 1, 0.5))
)
calib_data <- data.frame(
  x1 = as.numeric(rnorm(50, 1, 2)),
  x2 = as.numeric(rnorm(50, 2.5, 3)),
  target = as.factor(rbinom(50, 1, 0.5))
)
test_data <- data.frame(
  x1 = as.numeric(rnorm(50, 1, 2)),
  x2 = as.numeric(rnorm(50, 2.5, 3))
)
p_values <- conformal_pvalues(train_data, calib_data, test_data, target="target", method="svm")
```

Index

`conformal_pvalues`, [2](#)