

Package ‘trendeval’

October 14, 2022

Title Evaluate Trending Models

Version 0.0.1

Description Provides a coherent interface for evaluating models fit with the trending package. This package is part of the RECON (<<https://www.repidemicsconsortium.org/>>) toolkit for outbreak analysis.

URL <https://github.com/reconhub/trendeval>

BugReports <https://github.com/reconhub/trendeval/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports trending, yardstick, rsample, ellipsis, stats, tidyr, tibble

Suggests testthat, dplyr, outbreaks

NeedsCompilation no

Author Dirk Schumacher [aut],
Thibaut Jombart [aut],
Tim Taylor [ctb, cre] (<<https://orcid.org/0000-0002-8587-7113>>)

Maintainer Tim Taylor <tim.taylor@hiddenelephants.co.uk>

Repository CRAN

Date/Publication 2020-11-20 10:50:02 UTC

R topics documented:

evaluate_resampling 2

Index 4

 evaluate_resampling *Tools for model evaluation*

Description

These functions provide tools for evaluating `trending::trending_models`, based on the goodness of fit or on predictive power. `evaluate_aic()` evaluates the goodness of fit of a single model using Akaike's information criterion, measuring the deviance of the model while penalising its complexity. `evaluate_resampling()` uses repeated K-fold cross-validation and the Root Mean Square Error (RMSE) of testing sets to measure the predictive power of a single model. `evaluate_aic()` is faster, but `evaluate_resampling()` is better-suited to select best predicting models. `evaluate_models()` uses either `evaluate_aic()` or `evaluate_resampling()` to compare a series of models.

Usage

```
evaluate_resampling(
  model,
  data,
  metrics = list(yardstick::rmse),
  v = nrow(data),
  repeats = 1
)

evaluate_aic(model, data)

evaluate_models(models, data, method = evaluate_resampling, ...)
```

Arguments

model	A <code>trending::trending_model</code> object.
data	a <code>data.frame</code> containing data (including the response variable and all predictors) used in model
metrics	a list of functions assessing model fit, with a similar interface to <code>yardstick::rmse()</code> ; see https://yardstick.tidymodels.org/ for more information
v	the number of equally sized data partitions to be used for K-fold cross-validation; v cross-validations will be performed, each using v - 1 partition as training set, and the remaining partition as testing set. Defaults to 1, so that the method uses leave-one-out cross validation, akin to Jackknife except that the testing set (and not the training set) is used to compute the fit statistics.
repeats	the number of times the random K-fold cross validation should be repeated for; defaults to 1; larger values are likely to yield more reliable / stable results, at the expense of computational time
models	a list of models specified as an <code>trending::trending_model()</code> objects.

method a function used to evaluate models: either `evaluate_resampling()` (default, better for selecting models with good predictive power) or `evaluate_aic()` (faster, focuses on goodness-of-fit rather than predictive power)

... further arguments passed to the underlying method (e.g. `metrics`, `v`, `repeats`).

Details

These functions wrap around existing functions from several packages. `stats::AIC()` is used in `evaluate_aic()`, and `evaluate_resampling()` uses `rsample::vfold_cv()` for cross-validation and `yardstick::rmse()` to calculate RMSE.

See Also

`stats::AIC()` for computing AIC; `rsample::vfold_cv()` for cross validation; `yardstick::rmse()` for calculating RMSE; `yardstick` also implements a range of other metrics for assessing model fit outlined at <https://yardstick.tidymodels.org/>; `trending::trending_model()` for the different ways to build the model objects.

Examples

```
x <- rnorm(100, mean = 0)
y <- rpois(n = 100, lambda = exp(x + 1))
dat <- data.frame(x = x, y = y)

model <- trending::glm_model(y ~ x, poisson)
evaluate_resampling(model, dat)
evaluate_aic(model, dat)

models <- list(
  poisson_model = trending::glm_model(y ~ x, poisson),
  linear_model = trending::lm_model(y ~ x)
)
evaluate_models(models, dat)
```

Index

`evaluate_aic (evaluate_resampling), 2`
`evaluate_aic(), 2, 3`
`evaluate_models (evaluate_resampling), 2`
`evaluate_models(), 2`
`evaluate_resampling, 2`
`evaluate_resampling(), 2, 3`

`rsample::vfold_cv(), 3`

`stats::AIC(), 3`

`trending::trending_model, 2`
`trending::trending_model(), 2, 3`

`yardstick::rmse(), 2, 3`