Package: trendeval (via r-universe)

September 26, 2024

Title Evaluate Trending Models

Version 0.0.1.9001

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/reconverse/trendeval

BugReports https://github.com/reconverse/trendeval/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports trending, yardstick, rsample, stats, tibble, utils

Suggests testthat (>= 3.0.0), dplyr, outbreaks

Remotes reconverse/trending

Config/testthat/edition 3

Repository https://reconverse.r-universe.dev

RemoteUrl https://github.com/reconverse/trendeval

RemoteRef HEAD

RemoteSha 55e24fe84ac0eeef29462d7e38e64ba34c1708e5

Contents

`calculate_aic` *Generic for calculating the AIC*

Description

Generic `calculate_aic()` returns the Akaike's 'An Information Criterion' for the given input.

Usage

```
calculate_aic(x, ...)

## Default S3 method:
calculate_aic(x, ...)

## S3 method for class 'trending_model'
calculate_aic(x, data, as_tibble = FALSE, ...)

## S3 method for class 'list'
calculate_aic(x, data, ...)

## S3 method for class 'trending_fit'
calculate_aic(x, as_tibble = FALSE, ...)

## S3 method for class 'trending_fit_tbl'
calculate_aic(x, ...)
```

Arguments

`x`	An R object.
`...`	Not currently used.
`data`	a `data.frame` containing data (including the response variable and all predictors) used in the specified model.
`as_tibble`	Should the result be returned as tibble (`as_tibble = TRUE`) or a list (`as_tibble = FALSE`).

Details

Specific methods are given for `trending_fit` and `trending_fit_tbl` objects. The default method applies `stats::AIC()` directly.

Value

For a single `trending_fit` input, if `as_tibble = FALSE` the object returned will be a list with entries:

- metric: "AIC"

- result: the resulting AIC value fit (NULL if the calculation failed)
- warnings: any warnings generated during calculation
- errors: any errors generated during calculation

If `as_tibble = TRUE`, or the input is a `trending_fit_tbl`, then the output will be a tibble with one row for each fitted model columns corresponding to output generated with single model input.

Author(s)

Tim Taylor

#' @examples x = rnorm(100, mean = 0) y = rpois(n = 100, lambda = exp(1.5 + 0.5*x)) dat <- data.frame(x = x, y = y) poisson_model <- glm_model(y ~ x , family = "poisson") negbin_model <- glm_nb_model(y ~ x) fitted_model <- fit(poisson_model, dat) fitted_models <- fit(list(poisson_model, negbin_model), data = dat)

calculate_aic(poisson_model, dat) calculate_aic(fitted_model) calculate_aic(fitted_model, as_tibble = TRUE) calculate_aic(fitted_models)

`calculate_mae` *Generic for calculating the root mean squared error*

Description

Generic `calculate_mae()` returns the root mean square error for the given input.

Usage

```
calculate_mae(x, ...)

## Default S3 method:
calculate_mae(x, ...)

## S3 method for class 'trending_model'
calculate_mae(x, data, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'list'
calculate_mae(x, data, na.rm = TRUE, ...)

## S3 method for class 'trending_fit'
calculate_mae(x, new_data, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'trending_fit_tbl'
calculate_mae(x, new_data, na.rm = TRUE, ...)

## S3 method for class 'trending_predict'
calculate_mae(x, na.rm = TRUE, as_tibble = TRUE, ...)
```

```
## S3 method for class 'trending_predict_tbl'
calculate_mae(x, na.rm = TRUE, ...)

## S3 method for class 'trending_prediction'
calculate_mae(x, na.rm = TRUE, as_tibble = TRUE, ...)
```

Arguments

x	An R object.
...	Not currently used.
data	a data.frame containing data (including the response variable and all predictors) used in the specified model.
na.rm	Should NA values should be removed before calculation of metric (passed to the underlying function yardstick::mae_vec).
as_tibble	Should the result be returned as tibble (as_tibble = TRUE) or a list (as_tibble = FALSE).
new_data	a data.frame containing data (including the response variable and all predictors) on which to assess the model.

Details

Specific methods are given for trending_model (and lists of these), trending_fit, trending_fit_tbl, trending_predict_tbl, trending_predict_tbl and trending_prediction objects. Each of these are simply wrappers around the yardstick::mae_vec with the addition of explicit error handling.

Value

For a single trending_fit input, if as_tibble = FALSE the object returned will be a list with entries:

- metric: "mae"
- result: the resulting mae value (NULL if the calculation failed)
- warnings: any warnings generated during calculation
- errors: any errors generated during calculation

If as_tibble = TRUE, or for the other trending classes, then the output will be a tibble with one row for each fitted model columns corresponding to output generated with single model input.

Author(s)

Tim Taylor

#' @examples x = rnorm(100, mean = 0) y = rpois(n = 100, lambda = exp(1.5 + 0.5*x)) dat <- data.frame(x = x, y = y) poisson_model <- glm_model(y ~ x , family = "poisson") negbin_model <- glm_nb_model(y ~ x) fitted_model <- fit(poisson_model, dat) fitted_models <- fit(list(poisson_model, negbin_model), data = dat)

calculate_mae(poisson_model, dat) calculate_mae(fitted_model) calculate_mae(fitted_model, as_tibble = TRUE) calculate_mae(fitted_models)

calculate_rmse *Generic for calculating the root mean squared error*

Description

Generic `calculate_rmse()` returns the root mean square error for the given input.

Usage

```
calculate_rmse(x, ...)

## Default S3 method:
calculate_rmse(x, ...)

## S3 method for class 'trending_model'
calculate_rmse(x, data, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'list'
calculate_rmse(x, data, na.rm = TRUE, ...)

## S3 method for class 'trending_fit'
calculate_rmse(x, new_data, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'trending_fit_tbl'
calculate_rmse(x, new_data, na.rm = TRUE, ...)

## S3 method for class 'trending_predict'
calculate_rmse(x, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'trending_predict_tbl'
calculate_rmse(x, na.rm = TRUE, ...)

## S3 method for class 'trending_prediction'
calculate_rmse(x, na.rm = TRUE, as_tibble = TRUE, ...)
```

Arguments

`x`	An R object.
`...`	Not currently used.
`data`	a `data.frame` containing data (including the response variable and all predictors) used in the specified model.
`na.rm`	Should NA values should be removed before calculation of metric (passed to the underlying function yardstick::rmse_vec).
`as_tibble`	Should the result be returned as tibble (`as_tibble = TRUE`) or a list (`as_tibble = FALSE`).
`new_data`	a `data.frame` containing data (including the response variable and all predictors) on which to assess the model.

Details

Specific methods are given for `trending_model` (and lists of these), `trending_fit`, `trending_fit_tbl`, `trending_predict_tbl`, `trending_predict_tbl` and `trending_prediction` objects. Each of these are simply wrappers around the yardstick::rmse_vec with the addition of explicit error handling.

Value

For a single `trending_fit` input, if `as_tibble = FALSE` the object returned will be a list with entries:

- metric: "rmse"
- result: the resulting rmse value (NULL if the calculation failed)
- warnings: any warnings generated during calculation
- errors: any errors generated during calculation

If `as_tibble = TRUE`, or for the other `trending` classes, then the output will be a tibble with one row for each fitted model columns corresponding to output generated with single model input.

Author(s)

Tim Taylor

#' @examples x = rnorm(100, mean = 0) y = rpois(n = 100, lambda = exp(1.5 + 0.5*x)) dat <- data.frame(x = x, y = y) poisson_model <- glm_model(y ~ x , family = "poisson") negbin_model <- glm_nb_model(y ~ x) fitted_model <- fit(poisson_model, dat) fitted_models <- fit(list(poisson_model, negbin_model), data = dat)

calculate_rmse(poisson_model, dat) calculate_rmse(fitted_model) calculate_rmse(fitted_model, as_tibble = TRUE) calculate_rmse(fitted_models)

`calculate_rsq` *Generic for calculating the root mean squared error*

Description

Generic `calculate_rsq()` returns the root mean square error for the given input.

Usage

```
calculate_rsq(x, ...)

## Default S3 method:
calculate_rsq(x, ...)

## S3 method for class 'trending_model'
calculate_rsq(x, data, na.rm = TRUE, as_tibble = TRUE, ...)
```

```
## S3 method for class 'list'
calculate_rsq(x, data, na.rm = TRUE, ...)

## S3 method for class 'trending_fit'
calculate_rsq(x, new_data, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'trending_fit_tbl'
calculate_rsq(x, new_data, na.rm = TRUE, ...)

## S3 method for class 'trending_predict'
calculate_rsq(x, na.rm = TRUE, as_tibble = TRUE, ...)

## S3 method for class 'trending_predict_tbl'
calculate_rsq(x, na.rm = TRUE, ...)

## S3 method for class 'trending_prediction'
calculate_rsq(x, na.rm = TRUE, as_tibble = TRUE, ...)
```

Arguments

`x`	An R object.
`...`	Not currently used.
`data`	a `data.frame` containing data (including the response variable and all predictors) used in the specified model.
`na.rm`	Should NA values should be removed before calculation of metric (passed to the underlying function yardstick::rsq_vec).
`as_tibble`	Should the result be returned as tibble (`as_tibble = TRUE`) or a list (`as_tibble = FALSE`).
`new_data`	a `data.frame` containing data (including the response variable and all predictors) on which to assess the model.

Details

Specific methods are given for `trending_model` (and lists of these), `trending_fit`, `trending_fit_tbl`, `trending_predict_tbl`, `trending_predict_tbl` and `trending_prediction` objects. Each of these are simply wrappers around the yardstick::rsq_vec with the addition of explicit error handling.

Value

For a single `trending_fit` input, if `as_tibble = FALSE` the object returned will be a list with entries:

- metric: "rsq"
- result: the resulting rsq value (NULL if the calculation failed)
- warnings: any warnings generated during calculation
- errors: any errors generated during calculation

If `as_tibble = TRUE`, or for the other `trending` classes, then the output will be a tibble with one row for each fitted model columns corresponding to output generated with single model input.

Author(s)

Tim Taylor

#' @examples x = rnorm(100, mean = 0) y = rpois(n = 100, lambda = exp(1.5 + 0.5*x)) dat <- data.frame(x = x, y = y) poisson_model <- glm_model(y ~ x , family = "poisson") negbin_model <- glm_nb_model(y ~ x) fitted_model <- fit(poisson_model, dat) fitted_models <- fit(list(poisson_model, negbin_model), data = dat)

calculate_rsq(poisson_model, dat) calculate_rsq(fitted_model) calculate_rsq(fitted_model, as_tibble = TRUE) calculate_rsq(fitted_models)

`evaluate_aic` *Generic for calculating the AIC*

Description

`evaluate_aic()` is a a generic for evaluating the Akaike's 'An Information Criterion' for a given input

Usage

```
evaluate_aic(x, ...)

## Default S3 method:
evaluate_aic(x, ...)

## S3 method for class 'trending_model'
evaluate_aic(x, data, as_tibble = FALSE, ...)

## S3 method for class 'list'
evaluate_aic(x, data, ...)
```

Arguments

`x`	An R object.
`...`	Not currently used.
`data`	a `data.frame` containing data (including the response variable and all predictors) used in the specified model.
`as_tibble`	Should the result be returned as tibble (`as_tibble = TRUE`) or a list (`as_tibble = FALSE`).

Details

Specific methods are given for `trending_fit` and lists of these models.

Value

If `as_tibble = TRUE`, or the input is a list of models then the output will be a tibble with one row for each fitted model columns corresponding to output generated with single model input.

Author(s)

Tim Taylor

#' @examples x = rnorm(100, mean = 0) y = rpois(n = 100, lambda = exp(1.5 + 0.5*x)) dat <- data.frame(x = x, y = y) poisson_model <- glm_model(y ~ x , family = "poisson") negbin_model <- glm_nb_model(y ~ x)

evaluate_aic(poisson_model, dat) evaluate_aic(list(poisson_model, negbin_model), data = dat)

`evaluate_resampling` *Resampling approach for model evaluation*

Description

`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. Methods are provided for `trending::trending_model` (and lists of these) objects.

Usage

```
evaluate_resampling(x, ...)

## Default S3 method:
evaluate_resampling(x, ...)

## S3 method for class 'trending_model'
evaluate_resampling(
  x,
  data,
  metric = c("rmse", "rsq", "mae"),
  metric_arguments = list(na.rm = TRUE),
  v = nrow(data),
  repeats = 1,
  ...
)

## S3 method for class 'list'
evaluate_resampling(
  x,
  data,
  metric = c("rmse", "rsq", "mae"),
  metric_arguments = list(na.rm = TRUE),
  v = nrow(data),
```

```
  repeats = 1,
  ...
)
```

Arguments

`x`	An R object.
`...`	Not currently used.
`data`	a `data.frame` containing data (including the response variable and all predictors) used in the specified model.
`metric`	One of "rmse" (see calculate_rmse), "mae" (see calculate_mae) and "rsq" (see calculate_rsq).
`metric_arguments`	A named list of arguments passed to the underlying functions that calculate the metrics.
`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 the number of row in data, 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

Details

These functions wrap around existing functions from several packages. `evaluate_resampling.trending_model()` and `evaluate_resampling.list()` both use `rsample::vfold_cv()` for sampling and, for the calculating the different metrics, the yardstick package.

See Also

`calculate_aic()`, `calculate_rmse()`, `calculate_mae()` and `calculate_rsq()`.

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

evaluate_resampling(model, dat)
evaluate_resampling(models, dat)
```

Index