

# Package: trendeval (via r-universe)

June 28, 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 . . . . .	2
calculate_mae . . . . .	3
calculate_rmse . . . . .	5
calculate_rsqr . . . . .	6
evaluate_aic . . . . .	8
evaluate_resampling . . . . .	9

---

calculate_aic	<i>Generic for calculating the AIC</i>
---------------	--

---

### 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

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

### 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 <code>data.frame</code> 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 <code>yardstick::mae_vec</code> ).
as_tibble	Should the result be returned as <code>tibble</code> ( <code>as_tibble = TRUE</code> ) or a list ( <code>as_tibble = FALSE</code> ).
new_data	a <code>data.frame</code> 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	<i>Generic for calculating the root mean squared error</i>
----------------	--

---

### 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

<code>x</code>	An R object.
<code>...</code>	Not currently used.
<code>data</code>	a <code>data.frame</code> containing data (including the response variable and all predictors) used in the specified model.
<code>na.rm</code>	Should NA values should be removed before calculation of metric (passed to the underlying function <code>yardstick::rmse_vec</code> ).
<code>as_tibble</code>	Should the result be returned as <code>tibble</code> ( <code>as_tibble = TRUE</code> ) or a list ( <code>as_tibble = FALSE</code> ).
<code>new_data</code>	a <code>data.frame</code> 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 <code>data.frame</code> 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 <code>yardstick::rsq_vec</code> ).
as_tibble	Should the result be returned as <code>tibble</code> ( <code>as_tibble = TRUE</code> ) or a list ( <code>as_tibble = FALSE</code> ).
new_data	a <code>data.frame</code> 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_rsqr(poisson_model, dat) calculate_rsqr(fitted_model) calculate_rsqr(fitted_model, as_tibble
= TRUE) calculate_rsqr(fitted_models)
```

---

evaluate\_aic

*Generic for calculating the AIC*

---

### Description

`evaluate_aic()` is 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

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

### 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

<code>x</code>	An R object.
<code>...</code>	Not currently used.
<code>data</code>	a <code>data.frame</code> containing data (including the response variable and all predictors) used in the specified model.
<code>metric</code>	One of "rmse" (see <a href="#">calculate_rmse</a> ), "mae" (see <a href="#">calculate_mae</a> ) and "rsq" (see <a href="#">calculate_rsqr</a> ).
<code>metric_arguments</code>	A named list of arguments passed to the underlying functions that calculate the metrics.
<code>v</code>	the number of equally sized data partitions to be used for K-fold cross-validation; <code>v</code> cross-validations will be performed, each using <code>v - 1</code> 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.
<code>repeats</code>	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\\_rsqr\(\)](#).

### 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

`calculate_aic`, [2](#)  
`calculate_aic()`, [10](#)  
`calculate_mae`, [3](#), [10](#)  
`calculate_mae()`, [10](#)  
`calculate_rmse`, [5](#), [10](#)  
`calculate_rmse()`, [10](#)  
`calculate_rsqr`, [6](#), [10](#)  
`calculate_rsqr()`, [10](#)

`evaluate_aic`, [8](#)  
`evaluate_resampling`, [9](#)  
`evaluate_resampling()`, [9](#)  
`evaluate_resampling.list()`, [10](#)  
`evaluate_resampling.trending_model()`,  
[10](#)

`rsample::vfold_cv()`, [10](#)

`stats::AIC()`, [2](#)

`tibble`, [2–9](#)  
`trending::trending_model`, [9](#)  
`trending_fit`, [2](#), [4](#), [6–8](#)  
`trending_fit_tbl`, [2–4](#), [6](#), [7](#)  
`trending_model`, [4](#), [6](#), [7](#)  
`trending_predict_tbl`, [4](#), [6](#), [7](#)

`yardstick::mae_vec`, [4](#)  
`yardstick::rmse_vec`, [5](#), [6](#)  
`yardstick::rsqr_vec`, [7](#)