Package 'filtro'

July 18, 2025

Title Feature Selection Using Supervised Filter-Based Methods

Version 0.1.0

Description Tidy tools to apply filter-based supervised feature selection methods. These methods score and rank feature relevance using metrics such as p-values, correlation, and importance scores (Kuhn and Johnson (2019) <doi:10.1201/9781315108230>).

License MIT + file LICENSE

URL https://github.com/tidymodels/filtro

BugReports https://github.com/tidymodels/filtro/issues

Depends R (>= 4.1)

Imports purrr, rlang (>= 1.1.0), stats, tibble

Suggests aorsf, dplyr, FSelectorRcpp, modeldata, partykit, ranger, testthat (>= 3.0.0), titanic

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Frances Lin [aut, cre],
Max Kuhn [aut],
Emil Hvitfeldt [aut],
Posit Software, PBC [cph, fnd] (ROR: <https://ror.org/03wc8by49>)

Maintainer Frances Lin <franceslinyc@gmail.com>

Repository CRAN

Date/Publication 2025-07-18 15:20:24 UTC

Contents

`get_scores_aov` *Compute F-statistic and p-value scores using ANOVA F-test*

Description

Evaluate the relationship between a numeric outcome and a categorical predictor, or vice versa, by computing the ANOVA F-statistic or p-value. Output a tibble result with with one row per predictor, and four columns: `name`, `score`, `predictor`, and `outcome`.

Usage

```
get_scores_aov(score_obj, data, outcome)
```

Arguments

`score_obj`	A score object. See `score_aov()` for details.
`data`	A data frame or tibble containing the outcome and predictor variables.
`outcome`	A character string specifying the name of the outcome variable.

Details

The `score_obj` object may include the following components:

`neg_log10` A logical value indicating whether to apply a negative log10 transformation to p-values (default is `TRUE`).

- If `TRUE`, p-values are transformed as `-log10(pval)`. In this case:
  - The default `fallback_value` is `Inf`
  - The default `direction` is `"maximize"`
- If `FALSE`, raw p-values are used. In this case:
  - The `fallback_value` should be set to `0`
  - The `direction` should be set to `"minimize"`

Value

A tibble of result with one row per predictor, and four columns:

- `name`: the name of scoring metric.
- `score`: the score for the predictor-outcome pair.
- `predictor`: the name of the predictor.
- `outcome`: the name of the outcome.

Examples

```
data(ames, package = "modeldata")
data <- modeldata::ames |>
  dplyr::select(
    Sale_Price,
    MS_SubClass,
    MS_Zoning,
    Lot_Frontage,
    Lot_Area,
    Street
  )
# Define outcome
outcome <- "Sale_Price"
# Create a score object
score_obj <- score_aov()
score_res <- get_scores_aov(score_obj, data, outcome)
score_res
# Change score type
score_obj$score_type <- "pval"
score_res <- get_scores_aov(score_obj, data, outcome)
score_res
# Use raw p-values instead of -log10(p-values)
score_obj$score_type <- "pval"
score_obj$neg_log10 <- FALSE
score_obj$direction <- "minimize"
score_obj$fallback_value <- 0
score_res <- get_scores_aov(score_obj, data, outcome)
score_res
```

---

new_score_obj *Construct a new score object*

Description

Create a new score object that contains associated metadata, such as `range`, `fallback_value`, `score_type`, `direction`, and other relevant attributes.

Usage

```
new_score_obj(
  subclass = c("cat_num", "cat_cat", "num_num", "any"),
  outcome_type = c("numeric", "factor"),
  predictor_type = c("numeric", "factor"),
  case_weights = NULL,
  range = NULL,
  inclusive = NULL,
  fallback_value = NULL,
  score_type = NULL,
  trans = NULL,
```

```
  sorts = NULL,
  direction = NULL,
  deterministic = NULL,
  tuning = NULL,
  ties = NULL,
  calculating_fn = NULL,
  label = NULL,
  ...
)
```

Arguments

subclass	A character string indicating the type of predictor-outcome combination the scoring method supports. One of: • "cat_num" • "cat_cat" • "num_num" • "any"
outcome_type	A character string indicating the outcome type. One of: • "numeric" • "factor"
predictor_type	A character string indicating the predictor type. One of: • "numeric" • "factor"
case_weights	A logical value, indicating whether the model accepts case weights (TRUE) or not (FALSE).
range	A numeric vector of length two, specifying the minimum and maximum possible values, respectively.
inclusive	A logical vector of length two, indicating whether the lower and upper bounds of the range are inclusive (TRUE) or exclusive (FALSE).
fallback_value	A numeric scalar used as a fallback value. Typical values include: • 0 • 1 • Inf
score_type	A character string indicating the type of scoring metric to compute. Available options include: • ANOVA F-Test: "fstat", "pval" • Correlation: "pearson", "spearman" • Cross Tabulation: "pval_chisq", "pval_fisher" • Random Forest:"imp_rf", "imp_rf_conditional", "imp_rf_oblique" • Information Gain: "infogain", "gainratio", "symuncert" • ROC AUC: "roc_auc"
trans	Currently not used.

sorts	An optional function used to sort the scores. Common options include: • identity • abs • function(score) max(score, 1 - score)
direction	A character string indicating the optimization direction. One of: • "maximize" • "minimize" • "target"
deterministic	A logical value, indicating whether the score is deterministic (TRUE) or not (FALSE).
tuning	A logical value, indicating whether the model should be tuned (TRUE) or not (FALSE).
ties	An optional logical value indicating whether ties in score can occur (TRUE) or not (FALSE).
calculating_fn	An optional function used to compute the score. A default function is selected based on the score_type.
label	A named character string that can be used for printing and plotting.
...	Currently not used.

Value

A score object.

Examples

```
# Create a score object
new_score_obj()
```

---

score_aov *Create a score object for ANOVA F-test F-statistics and p-values*

Description

Construct a score object containing metadata for univariate feature scoring using the ANOVA F-test. Output a score object containing associated metadata such as range, fallback_value, score_type ("fstat" or "pval"), direction, and other relevant attributes.

Usage

```
score_aov(
  range = c(0, Inf),
  fallback_value = Inf,
  score_type = "fstat",
  direction = "maximize"
)
```

Arguments

`range`	A numeric vector of length two, specifying the minimum and maximum possible values, respectively.
`fallback_value`	A numeric scalar used as a fallback value. Typical values include: • `0` • `Inf` (default) For F-statistics, the `fallback_value` should be `"Inf"`. For p-values, since the default applies a negative log10 transformation to p-values, the `fallback_value` should be `"Inf"`.
`score_type`	A character string indicating the type of scoring metric to compute. Available options include: • `"fstat"` • `"pval"`
`direction`	A character string indicating the optimization direction. One of: • `"maximize"` (default) • `"minimize"` • `"target"` For F-statistics, the `direction` should be `"maximize"`. For p-values, since the default applies a negative log10 transformation to p-values, the `direction` should be `"maximize"`.

Value

A score object containing associated metadata such as `range`, `fallback_value`, `score_type` (`"fstat"` or `"pval"`), `direction`, and other relevant attributes.

Examples

```
# Create a score object
score_aov()
# Change score type
score_obj <- score_aov()
score_obj$score_type <- "pval"
```

Index