Package ‘important’

August 20, 2025

Title Supervised Feature Selection

Version 0.0.1

Description Interfaces for choosing important predictors in supervised regression, classification, and censored regression models. Permuted importance scores (Biecek and Burzykowski (2021) <doi:10.1201/9780429027192>) can be computed for 'tidymodels' model fits.

License MIT + file LICENSE

URL `https://important.tidymodels.org/`, `https://github.com/tidymodels/important`

BugReports `https://github.com/tidymodels/important/issues`

Depends R (>= 4.1.0)

Imports cli, dplyr, generics, ggplot2, hardhat (>= 1.4.1), purrr, rlang, tibble, tidyr, tune, vctrs, withr, workflows

Suggests censored, future, future.apply, mirai, modeldata, parsnip, recipes, spelling, survival, testthat (>= 3.0.0), yardstick

Config/Needs/website tidyverse/tidytemplate, tidymodels

Config/testthat/edition 3

Config/usethis/last-upkeep 2025-06-09

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author Max Kuhn [aut, cre] (ORCID: <`https://orcid.org/0000-0003-2402-136X`>), Posit Software, PBC [cph, fnd] (ROR: <`https://ror.org/03wc8by49`>)

Maintainer Max Kuhn <`max@posit.co`>

Repository CRAN

Date/Publication 2025-08-20 17:10:10 UTC

Contents

---

autoplot.importance_perm *Visualize importance scores*

Description

Visualize importance scores

Usage

```
## S3 method for class 'importance_perm'
autoplot(
  object,
  top = Inf,
  metric = NULL,
  eval_time = NULL,
  type = "importance",
  std_errs = stats::qnorm(0.95),
  ...
)
```

Arguments

`object`	A tibble of results from `importance_perm()`.
`top`	An integer for how many terms to show. To define importance when there are multiple metrics, the rankings of predictors are computed across metrics and the average rank is used. In the case of tied rankings, all the ties are included.
`metric`	A character vector or `NULL` for which metric to plot. By default, all metrics will be shown via facets. Possible options are the entries in `.metric` column of the object.
`eval_time`	For censored regression models, a vector of time points at which the survival probability is estimated.
`type`	A character value. The default is `"importance"` which shows the overall signal-to-noise ration (i.e., mean divided by standard error). Alternatively, `"difference"` shows the mean difference value with standard error bounds.
`std_errs`	The number of standard errors to plot (when `type = "difference"`).
`...`	Not used.

Value

A `ggplot2` object.

Examples

```
# Pre-computed results. See code at
system.file("make_imp_example.R", package = "important")

# Load the results
load(system.file("imp_examples.RData", package = "important"))

# A classification model with two classes and highly correlated predictors.
# To preprocess them, PCA feature extraction is used.
#
# Let's first view the importance in terms of the original predictor set
# using 50 permutations:

imp_orig

autoplot(imp_orig, top = 10)

# Now assess the importance in terms of the PCA components

imp_derv

autoplot(imp_derv)
autoplot(imp_derv, metric = "brier_class", type = "difference")
```

---

importance_perm *Compute permutation-based predictor importance*

Description

`importance_perm()` computes model-agnostic variable importance scores by permuting individual predictors (one at a time) and measuring how worse model performance becomes.

Usage

```
importance_perm(
  wflow,
  data,
  metrics = NULL,
  type = "original",
  size = 500,
  times = 10,
  eval_time = NULL,
  event_level = "first"
)
```

Arguments

`wflow`	A fitted `workflows::workflow()`.
`data`	A data frame of the data passed to `workflows::fit.workflow()`, including the outcome and case weights (if any).
`metrics`	A `yardstick::metric_set()` or `NULL`.
`type`	A character string for which *level* of predictors to compute. A value of `"original"` (default) will return values in the same representation of `data`. Using `"derived"` will compute them for any derived features/predictors, such as dummy indicator columns, etc.
`size`	How many data points to predict for each permutation iteration.
`times`	How many iterations to repeat the calculations.
`eval_time`	For censored regression models, a vector of time points at which the survival probability is estimated. This is only needed if a dynamic metric is used, such as the Brier score or the area under the ROC curve.
`event_level`	A single string. Either `"first"` or `"second"` to specify which level of `truth` to consider as the "event". This argument is only applicable when `estimator = "binary"`.

Details

The function can compute importance at two different levels.

- The "original" predictors are the unaltered columns in the source data set. For example, for a categorical predictor used with linear regression, the original predictor is the factor column.
- "Derived" predictors are the final versions given to the model. For the categorical predictor example, the derived versions are the binary indicator variables produced from the factor version.

This can make a difference when pre-processing/feature engineering is used. This can help us understand *how* a predictor can be important

Importance scores are computed for each predictor (at the specified level) and each performance metric. If no metric is specified, defaults are used:

- Classification: `yardstick::brier_class()`, `yardstick::roc_auc()`, and `yardstick::accuracy()`.
- Regression: `yardstick::rmse()` and `yardstick::rsq()`.
- Censored regression: `yardstick::brier_survival()`

For censored data, importance is computed for each evaluation time (when a dynamic metric is specified).

By default, no parallelism is used to process models in **tune**; you have to opt-in.

Using future to parallel process:

You should install the package and choose your flavor of parallelism using the plan function. This allows you to specify the number of worker processes and the specific technology to use.

For example, you can use:

```
library(future)
plan(multisession, workers = 4)
```

and work will be conducted simultaneously (unless there is an exception; see the section below).

See `future::plan()` for possible options other than `multisession`.

Using mirai to parallel process:

To configure parallel processing with **mirai**, use the `mirai::daemons()` function. The first argument, `n`, determines the number of parallel workers. Using `daemons(0)` reverts to sequential processing.

The arguments `url` and `remote` are used to set up and launch parallel processes over the network for distributed computing. See `mirai::daemons()` documentation for more details.

Value

A tibble with extra classes `"importance_perm"` and either `"original_importance_perm"` or `"derived_importance_perm'`
The columns are:

- `.metric` the name of the performance metric:
- `predictor`: the predictor
- `n`: the number of usable results (should be the same as `times`)
- `mean`: the average of the differences in performance. For each metric, larger values indicate worse performance (i.e., higher importance).
- `std_err`: the standard error of the differences.
- `importance`: the mean divided by the standard error.
- For censored regression models, an additional `.eval_time` column may also be included (depending on the metric requested).

Examples

```
if (rlang::is_installed(c("modeldata", "recipes", "workflows", "parsnip"))) {
  library(modeldata)
  library(recipes)
  library(workflows)
  library(dplyr)
  library(parsnip)

  set.seed(12)
  dat_tr <-
    sim_logistic(250, ~ .1 + 2 * A - 3 * B + 1 * A *B, corr = .7) |>
    dplyr::bind_cols(sim_noise(250, num_vars = 10))

  rec <-
    recipe(class ~ ., data = dat_tr) |>
    step_interact(~ A:B) |>
    step_normalize(all_numeric_predictors()) |>
    step_pca(contains("noise"), num_comp = 5)

  lr_wflow <- workflow(rec, logistic_reg())
  lr_fit <- fit(lr_wflow, dat_tr)

  set.seed(39)
  orig_res <- importance_perm(lr_fit, data = dat_tr, type = "original",
                              size = 100, times = 3)
  orig_res

  set.seed(39)
  deriv_res <- importance_perm(lr_fit, data = dat_tr, type = "derived",
                               size = 100, times = 3)
  deriv_res
}
```

Index