Package 'misl'

May 8, 2026

Title Multiple Imputation by Super Learning

Version 2.0.0

Description Performs multiple imputation of missing data using an ensemble super learner built with the tidymodels framework. For each incomplete column, a stacked ensemble of candidate learners is trained on a bootstrap sample of the observed data and used to generate imputations via predictive mean matching (continuous), probability draws (binary), or cumulative probability draws (categorical). Supports parallelism across imputed datasets via the future framework.

License MIT + file LICENSE

URL https://github.com/JustinManjourides/misl

BugReports https://github.com/JustinManjourides/misl/issues

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.0), future.apply (>= 1.11.0), parsnip (>= 1.2.0), recipes (>= 1.0.0), rsample (>= 1.2.0), stacks (>= 1.0.0), stats, tibble (>= 3.2.0), tidyr (>= 1.3.0), tune (>= 1.2.0), utils, workflows (>= 1.1.0)

Suggests earth (>= 5.3.0), future (>= 1.33.0), ggforce, ggplot2, knitr, MASS, ranger (>= 0.16.0), rmarkdown, scales, testthat (>= 3.0.0), xgboost (>= 1.7.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Justin Manjourides [aut, cre] (ORCID: <https://orcid.org/0000-0002-2454-4489>), Thomas Carpenito [aut] (ORCID: <https://orcid.org/0000-0003-3591-0680>)

Maintainer Justin Manjourides <j.manjourides@northeastern.edu>

Repository CRAN

Date/Publication 2026-04-08 05:00:02 UTC

Contents

`list_learners` *List available learners for MISL imputation*

Description

Displays the built-in named learners available for use in `misl()`. Note that any parsnip-compatible model spec can also be passed directly via the `*_method` arguments.

Usage

```
list_learners(outcome_type = "all", installed_only = FALSE)
```

Arguments

`outcome_type`	One of `"continuous"`, `"binomial"`, `"categorical"`, `"ordinal"`, or `"all"` (default).
`installed_only`	If `TRUE`, only learners whose backend package is already installed are returned. Default `FALSE`.

Value

A tibble with columns `learner`, `description`, `package`, `installed`, and outcome-type support flags (when `outcome_type = "all"`).

Examples

```
list_learners()
list_learners("continuous")
list_learners("ordinal")
list_learners("categorical", installed_only = TRUE)
```

misl *MISL: Multiple Imputation by Super Learning (v2.0)*

Description

Imputes missing values using multiple imputation by super learning.

Usage

```
misl(
  dataset,
  m = 5,
  maxit = 5,
  seed = NA,
  con_method = c("glm", "rand_forest", "boost_tree"),
  bin_method = c("glm", "rand_forest", "boost_tree"),
  cat_method = c("rand_forest", "boost_tree"),
  ord_method = c("polr", "rand_forest", "boost_tree"),
  cv_folds = 5,
  ignore_predictors = NA,
  quiet = TRUE
)
```

Arguments

dataset	A dataframe or matrix containing the incomplete data. Missing values are represented with NA.
m	The number of multiply imputed datasets to create. Default 5.
maxit	The number of iterations per imputed dataset. Default 5.
seed	Integer seed for reproducibility, or NA to skip. Default NA.
con_method	Character vector of learner IDs, a list of parsnip model specs, or a mixed list of both, for continuous columns. Default c("glm", "rand_forest", "boost_tree").
bin_method	Character vector of learner IDs, a list of parsnip model specs, or a mixed list of both, for binary columns (values must be 0/1/NA or a two-level factor). Default c("glm", "rand_forest", "boost_tree").
cat_method	Character vector of learner IDs, a list of parsnip model specs, or a mixed list of both, for unordered categorical columns. Default c("rand_forest", "boost_tree").
ord_method	Character vector of learner IDs, a list of parsnip model specs, or a mixed list of both, for ordered categorical columns. Default c("polr", "rand_forest", "boost_tree").
cv_folds	Integer number of cross-validation folds used when stacking multiple learners. Reducing this (e.g. to 3) speeds up computation at a small cost to ensemble accuracy. Default 5. Ignored when only a single learner is supplied.
ignore_predictors	Character vector of column names to exclude as predictors. Default NA.
quiet	Suppress console progress messages. Default TRUE.

Details

Built-in named learners (see `list_learners()`):

- `"glm"` - base R (logistic for binary, linear for continuous)
- `"rand_forest"` - **ranger**
- `"boost_tree"` - **xgboost**
- `"mars"` - **earth**
- `"multinom_reg"` - **nnet** (unordered categorical only)
- `"polr"` - **MASS** (ordered categorical only)

Any parsnip-compatible model spec can also be passed directly via the `*_method` arguments. Named strings and parsnip specs can be mixed in the same list:

```
library(parsnip)
misl(data,
  con_method = list(
    "glm",
    rand_forest(trees = 500) |> set_engine("ranger")
  )
)
```

The mode (regression vs classification) is always enforced by `misl` regardless of what is set on the spec.

Value

A list of `m` named lists, each with:

`datasets` A fully imputed tibble.

`trace` A long-format tibble of mean/sd trace statistics per iteration, for convergence inspection.

Parallelism

Imputation across the `m` datasets is parallelised via **future.apply**. To enable parallel execution, set a **future** plan before calling `misl()`:

```
library(future)
plan(multisession, workers = 4)
result <- misl(data, m = 5)
plan(sequential)
```

Examples

```
# Using named learners (same as v1.0)
set.seed(1)
n <- 100
demo_data <- data.frame(x1 = rnorm(n), x2 = rnorm(n), y = rnorm(n))
demo_data[sample(n, 10), "y"] <- NA
```

```
misl_imp <- misl(demo_data, m = 2, maxit = 2, con_method = "glm")

# Using a custom parsnip spec
## Not run:
library(parsnip)
misl_imp <- misl(
  demo_data, m = 2, maxit = 2,
  con_method = list(
    "glm",
    rand_forest(trees = 500) |> set_engine("ranger")
  )
)

## End(Not run)
```

`plot_misl_trace` *Plot trace statistics from a MISL imputation*

Description

Plots the mean and standard deviation of imputed values across iterations for all incomplete variables, paginated in grids of up to 3 variables per page. Stable traces that mix well across datasets indicate convergence. Note that trace statistics are only computed for continuous and numeric binary columns – categorical and ordinal columns are excluded automatically.

Usage

```
plot_misl_trace(misl_result, ncol = 2, nrow = 3)
```

Arguments

`misl_result`	A list returned by `misl()`.
`ncol`	Number of columns per page. Default 2.
`nrow`	Number of rows per page. Default 3.

Value

Invisibly returns the long-format trace data frame used for plotting.

Examples

```
set.seed(1)
n <- 100
demo_data <- data.frame(x1 = rnorm(n), x2 = rnorm(n), y = rnorm(n))
demo_data[sample(n, 10), "y"] <- NA
misl_imp <- misl(demo_data, m = 3, maxit = 3, con_method = "glm")
plot_misl_trace(misl_imp)
```

Index