Package 'mlr3superlearner'

May 9, 2026

Type Package

Title Super Learner Fitting and Prediction

Version 0.1.2

Description An implementation of the Super Learner prediction algorithm from van der Laan, Polley, and Hubbard (2007) <doi:10.2202/1544-6115.1309 using the 'mlr3' framework.

License GPL (>= 3)

Encoding UTF-8

Imports checkmate, lgr, mlr3, data.table, purrr, cli, glmnet

RoxygenNote 7.3.2

Depends mlr3learners

Suggests ranger, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Nicholas Williams [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-1378-4831>)

Maintainer Nicholas Williams <ntwilliams.personal@gmail.com>

Repository CRAN

Date/Publication 2024-09-17 16:30:06 UTC

Contents

available_learners *Learners Available for Use*

Description

Learners Available for Use

Usage

```
available_learners(outcome_type = c("binomial", "continuous"))
```

Arguments

outcome_type	The outcome variable type.

Value

A data.table of available learners.

Examples

```
available_learners("binomial")
```

mlr3superlearner *Super Learner Algorithm*

Description

Implementation of the Super Learner algorithm using the 'mlr3' framework. By default, returning the discrete Super Learner. If using the ensemble Super Learner, The LASSO with an alpha value of 0 and a restriction on the lower limit of the coefficients is used as the meta-learner.

Usage

```
mlr3superlearner(
  data,
  target,
  library,
  outcome_type = c("binomial", "continuous"),
  folds = NULL,
  discrete = TRUE,
  newdata = NULL,
  group = NULL,
  info = FALSE
)
```

Arguments

data	[data.frame] A data.frame containing predictors and target variable.
target	[character(1)] The name of the target variable in data.
library	[character] or [list] A vector or list of algorithms to be used for prediction.
outcome_type	[character(1)] The outcome variable type. Options are "binomial" and "continuous".
folds	[numeric(1)] The number of cross-validation folds, or if NULL will be dynamically determined.
discrete	[logical(1)] Return the discrete Super Learner, or the ensemble Super Learner?
newdata	[list] A list of data.frames to generate predictions from.
group	[character(1)] Name of a grouping variable in data. Assumed to be discrete; observations in the same group are treated like a "block" of observations kept together during sample splitting.
info	[logical(1)] Print learner fitting information to the console.

Value

A list of class mlr3superlearner.

Examples

```
if (requireNamespace("ranger", quietly = TRUE)) {
  n <- 1e3
  W <- matrix(rnorm(n*3), ncol = 3)
  A <- rbinom(n, 1, 1 / (1 + exp(-(.2*W[,1] - .1*W[,2] + .4*W[,3]))))
  Y <- rbinom(n,1, plogis(A + 0.2*W[,1] + 0.1*W[,2] + 0.2*W[,3]^2 ))
  tmp <- data.frame(W, A, Y)
  mlr3superlearner(tmp, "Y", c("glm", "ranger"), "binomial")
}
```

---

predict.mlr3superlearner

Predict method for mlr3superlearner *object*

Description

Predict method for mlr3superlearner object

Usage

```
## S3 method for class 'mlr3superlearner'
predict(object, newdata, ...)
```

Arguments

object	[mlr3superlearner] An object returned from mlr3superlearner().
newdata	data [data.frame] A data.frame containing predictors.
...	Unused.

Value

A vector of the predicted values.

See Also

mlr3superlearner

Examples

```
if (requireNamespace("ranger", quietly = TRUE)) {
  n <- 1e3
  W <- matrix(rnorm(n*3), ncol = 3)
  A <- rbinom(n, 1, 1 / (1 + exp(-(.2*W[,1] - .1*W[,2] + .4*W[,3]))))
  Y <- rbinom(n,1, plogis(A + 0.2*W[,1] + 0.1*W[,2] + 0.2*W[,3]^2 ))
  tmp <- data.frame(W, A, Y)
  fit <- mlr3superlearner(tmp, "Y", c("glm", "ranger"), "binomial")
  predict(fit, tmp)
}
```

Index