

Package: tisthemachinelearner (via r-universe)

May 23, 2026

Type Package

Title Lightweight interface to sklearn, nnetsauce and unifiedbooster with conformal prediction

Version 0.10.0

Description Lightweight interface to Python packages sklearn, nnetsauce and unifiedbooster with conformal prediction.

Date 2026-04-23

License BSD_3_clause Clear + file LICENSE

Encoding UTF-8

LazyData true

Imports reticulate, R6, Rcpp, RcppProgress

Depends R (>= 3.5.0), reticulate (>= 1.18), Matrix

Suggests MASS, testthat, misc, knitr

LinkingTo Rcpp, RcppProgress

RoxygenNote 7.3.2

SystemRequirements Python (>= 3.6) with pip and virtualenv support

Config/reticulate list(packages = list(list(package = `` scipy"), list(package = `` numpy"), list(package = `` scikit-learn")))

Remotes thierrymoudiki/misc

VignetteBuilder knitr

Config/pak/sysreqs libpng-dev python3

Repository <https://techtonique.r-universe.dev>

Date/Publication 2026-04-23 16:14:43 UTC

RemoteUrl https://github.com/Techtonique/tisthemachinelearner_r

RemoteRef HEAD

RemoteSha d529183c4d6564beaacbd458038814f194fdac9a

Contents

Booster	2
boosterCpp	4
get_model_list	5
predict.booster	5
predict.regressor	6
predictBoosterCpp	6
Regressor	7
setup_sklern	8
simulate.regressor	9

Index	10
--------------	-----------

Booster	<i>R6 Class for Gradient Boosting</i>
---------	---------------------------------------

Description

An R6 Class that provides an interface to gradient boosting with neural network feature transformation.

Public fields

estimators List of fitted base models
 learning_rate Learning rate for boosting
 losses Vector of training losses
 n_estimators Number of estimators used

Methods

Public methods:

- [Booster\\$new\(\)](#)
- [Booster\\$fit\(\)](#)
- [Booster\\$predict\(\)](#)
- [Booster\\$clone\(\)](#)

Method `new()`: Create a new Booster object

Usage:

```
Booster$new(
  model_name = "ExtraTreeRegressor",
  n_estimators = 100L,
  learning_rate = 0.1,
  tolerance = 1e-04,
  calibration = FALSE,
  seed = 123L,
```

```
    show_progress = TRUE,  
    verbose = FALSE,  
    venv_path = "./venv"  
  )
```

Arguments:

`model_name` Name of the base model
`n_estimators` Number of boosting iterations
`learning_rate` Learning rate for boosting
`tolerance` Convergence tolerance
`calibration` Whether to calibrate the model
`seed` Random seed
`show_progress` Whether to show progress bar
`verbose` Whether to print detailed output
`venv_path` Path to the virtual environment

Method `fit()`: Fit the boosting model to training data

Usage:

```
Booster$fit(x, y)
```

Arguments:

`x` Feature matrix
`y` Target vector

Returns: The fitted object (invisible)

Method `predict()`: Make predictions on new data

Usage:

```
Booster$predict(newdata)
```

Arguments:

`newdata` New data to predict on

Returns: Vector of predictions

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
Booster$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`boosterCpp`*Fit a boosting model with neural network feature transformation*

Description

Fit a boosting model with neural network feature transformation

Usage

```
boosterCpp(  
  x,  
  y,  
  model_name,  
  n_estimators = 100L,  
  learning_rate = 0.1,  
  tolerance = 1e-04,  
  calibration = FALSE,  
  seed = 123L,  
  show_progress = TRUE,  
  verbose = FALSE,  
  venv_path = "./venv"  
)
```

Arguments

<code>x</code>	Input matrix
<code>y</code>	Target vector
<code>model_name</code>	Name of the base model
<code>n_estimators</code>	Number of boosting iterations
<code>learning_rate</code>	Learning rate for boosting
<code>tolerance</code>	Convergence tolerance
<code>calibration</code>	Whether to calibrate the model
<code>seed</code>	Random seed
<code>show_progress</code>	Whether to show progress bar
<code>verbose</code>	Whether to print detailed output
<code>venv_path</code>	Path to the virtual environment

get_model_list	<i>Get a list of all models in scikit-learn</i>
----------------	---

Description

This function retrieves a list of all models available in scikit-learn. It imports the necessary Python modules and retrieves all estimators, filtering them into classifiers and regressors.

Usage

```
get_model_list(venv_path = "./venv")
```

Value

A list with two elements: - 'classifiers': A character vector of all classifier models - 'regressors': A character vector of all regressor models

Examples

```
# model_list <- get_model_list()
# print(model_list$classifiers)
```

predict.booster	<i>Predict using a boosted model</i>
-----------------	--------------------------------------

Description

Predict using a boosted model

Usage

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

Arguments

object	A boosted model object
newdata	New data to predict on
...	Additional arguments

predict.regressor *Predict method for regressor objects*

Description

Predict method for regressor objects

Usage

```
## S3 method for class 'regressor'
predict(
  object,
  newdata,
  nsim = 250L,
  level = 95,
  method = c("none", "splitconformal", "surrogate", "bootstrap", "tsbootstrap",
             "bayesian"),
  seed = 123,
  ...
)
```

Arguments

object	A regressor object
newdata	New data to predict on
nsim	Number of simulations for bootstrap/tsbootstrap
level	Confidence level for prediction intervals
method	Method for computing prediction intervals
seed	Seed for the random number generator
...	Additional arguments

predictBoosterCpp *Predict using a boosted model*

Description

Predict using a boosted model

Usage

```
predictBoosterCpp(booster, x)
```

Arguments

booster	A boosted model object
x	New data to predict on

Regressor

R6 Class for Scikit-learn Regressors

Description

An R6 Class that provides an interface to scikit-learn regression models.

Public fields

`model` The underlying sklearn model

`residuals` Model residuals

`df.residual` Degrees of freedom of residuals

Methods

Public methods:

- [Regressor\\$new\(\)](#)
- [Regressor\\$fit\(\)](#)
- [Regressor\\$predict\(\)](#)
- [Regressor\\$clone\(\)](#)

Method `new()`: Create a new Regressor object

Usage:

```
Regressor$new(model_name, venv_path = "./venv", ...)
```

Arguments:

`model_name` Name of the sklearn model to use

`venv_path` Path to the virtual environment

`...` Additional parameters passed to the sklearn model

Method `fit()`: Fit the model to training data

Usage:

```
Regressor$fit(x, y, calibration = FALSE, seed = 42L, ...)
```

Arguments:

`x` Feature matrix

`y` Target vector

`calibration` Logical flag to indicate if calibration of residuals should be used

`seed` Seed for random number generator

Method `predict()`: Make predictions on new data

Usage:

```
Regressor$predict(
  newdata,
  method = c("none", "splitconformal", "surrogate", "bootstrap", "tsbootstrap",
    "bayesian"),
  nsim = 250L,
  level = 95,
  seed = 123
)
```

Arguments:

`newdata` New data to predict on
`method` Method for computing prediction intervals
`nsim` Number of simulations for bootstrap/tsbootstrap
`level` Confidence level for prediction intervals
`seed` Random seed

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
Regressor$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

 setup_sklearn

Setup Python environment using uv

Description

Setup Python environment using uv

Usage

```
setup_sklearn(venv_path = "venv")
```

Arguments

`venv_path` Path to virtual environment (default: `"/.venv"`)

Examples

```
## Not run:
# After creating venv with: uv venv venv
setup_sklearn()

## End(Not run)
```

simulate.regressor *Simulate method for regressor objects*

Description

Simulate method for regressor objects

Usage

```
## S3 method for class 'regressor'  
simulate(  
  object,  
  newdata,  
  nsim = 250L,  
  level = 95,  
  method = c("surrogate", "bootstrap", "tsbootstrap", "bayesian"),  
  seed = 123,  
  venv_path = "./venv",  
  ...  
)
```

Arguments

object	A regressor object
newdata	New data to predict on
nsim	Number of simulations for bootstrap/tsbootstrap
level	Confidence level for prediction intervals
method	Method for computing prediction intervals
seed	Seed for the random number generator
...	Additional arguments

Index

Booster, [2](#)

boosterCpp, [4](#)

get_model_list, [5](#)

predict.booster, [5](#)

predict.regressor, [6](#)

predictBoosterCpp, [6](#)

Regressor, [7](#)

setup_sklearn, [8](#)

simulate.regressor, [9](#)