Package 'mljar'

June 23, 2017

Title R API for MLJAR

Version 0.1.1

Author Dominik Krzemiński <raymon92@gmail.com>

Maintainer Piotr Płoński <contact@mljar.com>

Description Provides an R API wrapper for 'mljar.com', a web service allowing for online training for machine learning models (see <https://mljar.com> for more information).

License MIT + file LICENSE

URL http://mljar.com, https://github.com/mljar/mljar-api-R

BugReports https://github.com/mljar/mljar-api-R/issues

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests testthat, covr

Depends R (>= 3.1.2)

Imports httr, jsonlite, readr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-23 11:31:11 UTC

R topics documented:

`add_dataset_if_not_exists` *Add dataset if not exists*

Description

Checks parameters before adding new dataset and verifies if it doesn't exists already.

Usage

```
add_dataset_if_not_exists(project_hid, filename, title,
  prediction_only = FALSE)
```

Arguments

`project_hid`	character with project identifier
`filename`	character with filename containing data
`title`	title of dataset
`prediction_only`	boolean determining if data is used only for prediction

Value

parsed dataset details

add_experiment_if_not_exists *Add experiment if not exists*

Description

Check if experiment exists, verifies parameters, creates data to create_experiment function and finally starts creation of MLJAR experiment.

Usage

```
add_experiment_if_not_exists(project_hid, train_dataset, valid_dataset,
  experiment_title, project_task, validation_kfolds, validation_shuffle,
  validation_stratify, validation_train_split, algorithms, metric, tuning_mode,
  time_constraint, create_ensemble)
```

Arguments

project_hid	character with project identifier
train_dataset	character with path to training dataset
valid_dataset	character with path to validation dataset
experiment_title	character with experiment title
project_task	character with project task
validation_kfolds	number of folds to be used in validation
validation_shuffle	boolean which specify if shuffle samples before training
validation_stratify	boolean which decides whether samples will be divided into folds with the same class distribution
validation_train_split	ratio how to split training dataset into train and validation
algorithms	list of algorithms to use
metric	charcater with metric
tuning_mode	tuning mode
time_constraint	numeric with time limit to calculate algorithm
create_ensemble	whether or not to create ensemble

Value

experiment details structure

add_new_dataset *Adds new dataset*

Description

Adds new dataset

Usage

```
add_new_dataset(project_hid, filename, title, prediction_only = FALSE)
```

Arguments

project_hid	character with project identifier
filename	character with filename containing data
title	title of dataset
prediction_only	boolean determining if data is used only for prediction

Value

parsed by toJSON dataset details

create_experiment *Creates experiment from given parameters*

Description

Creates experiment from given parameters

Usage

```
create_experiment(data)
```

Arguments

data	list of experiment parameters

Value

experiment details parsed by fromJSON

create_project *Creates a new project*

Description

Creates a new project

Usage

```
create_project(title, task, description = "")
```

Arguments

title	character with project title
task	character with project task
description	optional description

Value

project details structure

delete_dataset *Deletes dataset*

Description

Deletes dataset

Usage

```
delete_dataset(dataset_hid)
```

Arguments

dataset_hid	character with dataset identifier

delete_project *Delete project*

Description

Delete project

Usage

```
delete_project(hid)
```

Arguments

hid	charceter with project identifier

get_all_models *Gives data.frame with basic data of all models*

Description

You can later get some specific model by calling e.g. `mod <- get_model(project_title, experiment_title, model_hi`

Usage

```
get_all_models(project_title, exp_title)
```

Arguments

project_title	character with project title
exp_title	character with experiment title

Value

data.frame with model's "hid", "model_type", "metric_value", "metric_type"

get_dataset *Gets dataset*

Description

Gets dataset

Usage

```
get_dataset(dataset_hid)
```

Arguments

dataset_hid	character with dataset identifier

Value

structure with parsed dataset and response

get_datasets *Gets list of available datasets*

Description

Gets list of available datasets

Usage

```
get_datasets(project_hid)
```

Arguments

project_hid	character with project identifier

Value

structure with parsed datasets and response

get_experiment *Gets experiment details*

Description

Gets experiment details

Usage

```
get_experiment(experiment_hid)
```

Arguments

experiment_hid	character with experiment identifier

Value

structure with parsed experiment and http response

get_experiments *Gets list of available experiments for given project*

Description

Gets list of available experiments for given project

Usage

```
get_experiments(project_hid)
```

Arguments

project_hid	character with project identifier

Value

structure with parsed experiments and http response

get_model *Get model*

Description

Gets model only if experiment finished and project with such a title and having such an experiment exists.

Usage

```
get_model(project_title, exp_title, model_hid)
```

Arguments

project_title	character with project title
exp_title	character with experiment title
model_hid	character with experiment identifier

Value

structure with model parameters

get_prediction *Gets MLJAR predictions*

Description

Gets MLJAR predictions

Usage

```
get_prediction(project_hid, dataset_hid, result_hid)
```

Arguments

project_hid	character with project identifier
dataset_hid	character with dataset identifier
result_hid	character with result identifier

Value

structure with parsed prediction and http response

get_project *Get project*

Description

Get data from a project of specified hid

Usage

```
get_project(hid)
```

Arguments

hid	character with project unique identifier

Value

structure with parsed project and http response

get_projects *Get projects*

Description

Gets list of available projects

Usage

```
get_projects()
```

Value

structure with parsed projects and http response

get_results *Gets results of MLJAR training*

Description

Gets results of MLJAR training

Usage

```
get_results(project_hid, experiment_hid)
```

Arguments

project_hid	character with project identifier
experiment_hid	character with experiment identifier

Value

structure with parsed results and http response

mljar_fit *MLJAR FIT*

Description

Verifies parameters and data and tries to run experiment.

Usage

```
mljar_fit(x, y, validx = NULL, validy = NULL, proj_title = NULL,
  exp_title = NULL, dataset_title = NULL, val_dataset_title = NULL,
  algorithms = c(), metric = "", wait_till_all_done = TRUE,
  validation_kfolds = MLJAR_DEFAULT_FOLDS,
  validation_shuffle = MLJAR_DEFAULT_SHUFFLE,
  validation_stratify = MLJAR_DEFAULT_STRATIFY,
  validation_train_split = MLJAR_DEFAULT_TRAIN_SPLIT,
  tuning_mode = MLJAR_DEFAULT_TUNING_MODE,
  create_ensemble = MLJAR_DEFAULT_ENSEMBLE,
  single_algorithm_time_limit = MLJAR_DEFAULT_TIME_CONSTRAINT)
```

Arguments

x	data.frame/matrix with training data
y	data.frame/matrix with training labels
validx	data.frame/matrix with validation data
validy	data.frame/matrix with validation labels
proj_title	charcater with project title
exp_title	charcater with experiment title
dataset_title	charcater with dataset name
val_dataset_title	charcater with validation dataset name
algorithms	list of algorithms to use For binary classification task available algorithm are: "xgb" which is for Xgboost, "lgb" which is for LightGBM "mlp" which is for Neural Network, "rfc" which is for Random Forest, "etc" which is for Extra Trees, "rgfc" which is for Regularized Greedy Forest, "knnc" which is for k-Nearest Neighbors, "logreg" which is for Logistic Regression. For regression task there are available algorithms: "xgbr" which is for Xgboost, "lgbr" which is for LightGBM, "rgfr" which is for Regularized Greedy Forest, "rfr" which is for Random Forest, "etr" which is for Extra Trees.
metric	charcater with metric For binary classification there are metrics: "auc" which is for Area Under ROC Curve, "logloss" which is for Logarithmic Loss. For regression tasks: "rmse" which is Root Mean Square Error, "mse" which is for Mean Square Error, "mase" which is for Mean Absolute Error.
wait_till_all_done	boolean saying whether function should wait till all models are done
validation_kfolds	number of folds to be used in validation
validation_shuffle	boolean which specify if shuffle samples before training
validation_stratify	boolean which decides whether samples will be divided into folds with the same class distribution
validation_train_split	ratio how to split training dataset into train and validation
tuning_mode	tuning mode
create_ensemble	whether or not to create ensemble
single_algorithm_time_limit	numeric with time limit to calculate algorithm

Value

structure with the best model

mljar_predict *MLJAR PREDICT*

Description

Makes prediction basing on trained model.

Usage

```
mljar_predict(model, x_pred, project_title)
```

Arguments

model	model or MLJAR result structure
x_pred	data.frame/matrix data to predict
project_title	character with project title

Value

data.frame with preditction

prediction_download *Function to get predictions from MLJAR.*

Description

Function to get predictions from MLJAR.

Usage

```
prediction_download(prediction_hid)
```

Arguments

prediction_hid	prediction identifier

Value

data.frame with prediction

print_all_projects *Print all projects*

Description

Gives data.frame with basic information about existing projects

Usage

```
print_all_projects()
```

Value

data.frame with projects

submit_predict_job *Submits dataset for MLJAR prediction*

Description

Submits dataset for MLJAR prediction

Usage

```
submit_predict_job(project_hid, dataset_hid, result_hid)
```

Arguments

project_hid	character with project identifier
dataset_hid	character with dataset identifier
result_hid	character with result identifier

`upload_file` *Uploads file into MLJAR*

Description

It uploads file into MLJAR and returns destination path.

Usage

```
upload_file(project_hid, filepath)
```

Arguments

`project_hid`	character with project identifier
`filepath`	character with path to file

Value

character with destination path

Index