Package 'flowml'

February 16, 2024

Type Package

Title A Backend for a 'nextflow' Pipeline that Performs Machine-Learning-Based Modeling of Biomedical Data

Version 0.1.3

Maintainer Sebastian Malkusch <sebastian.malkusch@boehringer-ingelheim.com>

Description

Provides functionality to perform machine-learning-based modeling in a computation pipeline. Its functions contain the basic steps of machine-learning-based knowledge discovery workflows, including model training and optimization, model evaluation, and model testing. To perform these tasks, the package builds heavily on existing machine-learning packages, such as 'caret' <https://github.com/topepo/caret/> and associated packages. The package can train multiple models, optimize model hyperparameters by performing a grid search or a random search, and evaluates model performance by different metrics. Models can be validated either on a test data set, or in case of a small sample size by k-fold cross validation or repeated bootstrapping. It also allows for 0-Hypotheses generation by performing permutation experiments. Additionally, it offers methods of model interpretation and item categorization to identify the most informative features from a high dimensional data space. The functions of this package can easily be integrated into computation pipelines (e.g. 'nextflow' <https://www.nextflow.io/>) and hereby improve scalability, standardization, and re-producibility in the context of machine-learning.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/Boehringer-Ingelheim/flowml

BugReports https://github.com/Boehringer-Ingelheim/flowml/issues

Imports ABCanalysis, caret, data.table, dplyr, fastshap, furrr, future, magrittr, optparse, parallel, purrr, R6, readr, rjson, rlang, rsample, stats, stringr, tibble, tidyr, utils, vip

RoxygenNote 7.2.3

Collate 'fml_resampler.R' 'fml_resample.R' 'fml_format_response.R' 'fml_parser.R' 'fml_bootstrap.R' 'fml_categorize.R'

'fml_example.R' 'fml_globals.R' 'fml_grids.R' 'fml_interpret.R'
'fml_train.R' 'fml_validate.R'

Suggests ada, adabag, arm, bartMachine, bst, C50, caTools, class, Cubist, e1071, earth, elasticnet, evtree, fastICA, foreach, frbs, gam, gbm, ggplot2, glmnet, h2o, hda, ipred, keras, kernlab, kknn, klaR, knitr, kohonen, lars, leaps, LiblineaR, LogicReg, MASS, Matrix, mboost, mda, mgcv, monomvn, neuralnet, nnet, nnls, pamr, partDSA, party, partykit, penalized, pls, plyr, proxy, quantregForest, randomForest, ranger, rFerns, rmarkdown, rpart, rrcov, rrcovHD, RSNNS, RWeka, sda, shapviz, spls, superpc, VGAM, xgboost

Depends R (>= 3.5.0)

NeedsCompilation no

Author Sebastian Malkusch [aut, cre] (<https://orcid.org/0000-0001-6766-140X>), Kolja Becker [aut] (<https://orcid.org/0000-0001-8282-5329>), Alexander Peltzer [ctb] (<https://orcid.org/0000-0002-6503-2180>), Neslihan Kaya [ctb] (<https://orcid.org/0000-0002-0213-3072>), Boehringer Ingelheim Ltd. [cph, fnd]

Repository CRAN

Date/Publication 2024-02-16 10:40:02 UTC

R topics documented:

`create_parser` *create_parser*

Description

Creates an object that defines and handles command line arguments.

Usage

```
create_parser()
```

Details

A parser that organizes the communication between the user and th function. It also provides a help message.

Value

An instance of type 'optparse::OptionParser'.

Author(s)

Sebastian Malkusch

`create_resample_experiment`
create_resample_experiment

Description

Creates an object of a resampling experiment.

Usage

```
create_resample_experiment(
  seed,
  data_df,
  parser_inst,
  model_inst,
  config_inst,
  n_features
)
```

Arguments

`seed`	sets the seed for the random number generator to guarantee reproducibility. (int)
`data_df`	data frame to be learned from. (tibble::tibble)
`parser_inst`	instance of parser object. (optparse::parse_args).
`model_inst`	instance of caret_train object (caret::train).
`config_inst`	list of config options (list).
`n_features`	number of features (int).

Details

Creates a resampling experiment. It uses user defined parameters to set up the experiment. It creates an instance of the Resampler object and runs the experiment according to the user-defined parameters.

Value

An instance of type 'Resampler'.

Author(s)

Sebastian Malkusch

`fml_bootstrap` *fml_bootstrap*

Description

Pipeline function that sets up and runs a resampling experiment.

Usage

```
fml_bootstrap(parser_inst)
```

Arguments

`parser_inst`	Instance of fml_parser class that comprises command line arguments.

Details

The experiment is run in parallel. All results are written to files.

Value

none

Author(s)

Sebastian Malkusch

Examples

```
## Not run:
parser_inst <-  flowml::create_parser()

parser_inst$pipeline_segment <- "bootstrap"
parser_inst$config <- flowml::fml_example(file = "reg_config.json")
parser_inst$data <- flowml::fml_example(file = "reg_data.csv")
parser_inst$samples_train <- flowml::fml_example(file = "reg_samples_train.txt")
parser_inst$samples_test <- flowml::fml_example(file = "reg_samples_test.txt")
parser_inst$features <- flowml::fml_example(file = "reg_features.txt")
parser_inst$extended_features <- flowml::fml_example(file = "reg_features_extended.txt")
parser_inst$trained <- flowml::fml_example(file = "reg_fit.rds")
parser_inst$permutation <- "none"
parser_inst$result_dir <- tempdir()
```

```
flowml::fml_bootstrap(parser_inst = parser_inst)

## End(Not run)
```

`fml_example` *fml_example*

Description

path to flowml examples data

Usage

```
fml_example(file = NULL)
```

Arguments

`file`	Name of file. If 'NULL', the example files will be listed.

Details

flowml comes bundled with a number of sample files in its 'inst/extdata' directory. This function allows to access them.

Value

The path of to an example file, if file is defined. Else, a list of example files.

Author(s)

Sebastian Malkusch

Examples

```
## Not run:
fml_example()
fml_example(file = "reg_config.json")

## End(Not run)
```

fml_interpret *fml_interpret*

Description

Pipeline function that sets up and runs a post-hoc interpretation of an ml experiment. All results are written to rds files.

Usage

```
fml_interpret(parser_inst)
```

Arguments

parser_inst	instance of fml_parser class that comprises command line arguments.

Value

none

Author(s)

Sebastian Malkusch

Examples

```
## Not run:
parser_inst <-  flowml::create_parser()

parser_inst$pipeline_segment <- "interpret"
parser_inst$config <- flowml::fml_example(file = "reg_config.json")
parser_inst$data <- flowml::fml_example(file = "reg_data.csv")
parser_inst$samples_train <- flowml::fml_example(file = "reg_samples_train.txt")
parser_inst$samples_test <- flowml::fml_example(file = "reg_samples_test.txt")
parser_inst$features <- flowml::fml_example(file = "reg_features.txt")
parser_inst$extended_features <- flowml::fml_example(file = "reg_features_extended.txt")
parser_inst$trained <- flowml::fml_example(file = "reg_fit.rds")
parser_inst$interpretation <- "shap"
parser_inst$result_dir <- tempdir()

flowml::fml_interpret(parser_inst = parser_inst)

## End(Not run)
```

fml_train *fml_train*

Description

Pipeline function that performs a hyper-parameter screeing experiment.

Usage

```
fml_train(parser_inst)
```

Arguments

parser_inst	instance of fml_parser class that comprises command line arguments.

Value

none

Author(s)

Kolja Becker

Examples

```
## Not run:
parser_inst <-  flowml::create_parser()

parser_inst$pipeline_segment <- "train"
parser_inst$config <- flowml::fml_example(file = "reg_config.json")
parser_inst$data <- flowml::fml_example(file = "reg_data.csv")
parser_inst$samples_train <- flowml::fml_example(file = "reg_samples_train.txt")
parser_inst$samples_test <- flowml::fml_example(file = "reg_samples_test.txt")
parser_inst$features <- flowml::fml_example(file = "reg_features.txt")
parser_inst$extended_features <- flowml::fml_example(file = "reg_features_extended.txt")
parser_inst$result_dir <- tempdir()

flowml::fml_train(parser_inst = parser_inst)

## End(Not run)
```

fml_validate *fml_validate*

Description

Pipeline function that performs a validation experiment on a a caret train object based on test samples.

Usage

```
fml_validate(parser_inst)
```

Arguments

parser_inst	instance of fml_parser class that comprises command line arguments.

Value

none

Author(s)

Kolja Becker

Examples

```
## Not run:
parser_inst <-  flowml::create_parser()

parser_inst$pipeline_segment <- "validate"
parser_inst$config <- flowml::fml_example(file = "reg_config.json")
parser_inst$data <- flowml::fml_example(file = "reg_data.csv")
parser_inst$samples_train <- flowml::fml_example(file = "reg_samples_train.txt")
parser_inst$samples_test <- flowml::fml_example(file = "reg_samples_test.txt")
parser_inst$features <- flowml::fml_example(file = "reg_features.txt")
parser_inst$extended_features <- flowml::fml_example(file = "reg_features_extended.txt")
parser_inst$trained <- flowml::fml_example(file = "reg_fit.rds")
parser_inst$permutation <- "none"
parser_inst$result_dir <- tempdir()

flowml::fml_validate(parser_inst = parser_inst)

## End(Not run)
```

format_y *format_y*

Description

Formats response variable based on the ml-type variable passed by the config file. For regression analyses the response variable will be explicitly transformed to type numeric. For Classification experiments the response variable will be explicitly transformed to a factor. Time-to-event models are to be implemented in the near future.

Usage

```
format_y(y, ml.type)
```

Arguments

y	vector of response varibale.
ml.type	type of experiment (chracater).

Value

a transformed version of the response variable y.

Author(s)

Kolja Becker

Resampler *Resampler*

Description

Model validation by repeated bootstrapping

Format

[R6::R6Class] object.

Details

Uses repeated bootstrapping to validate models without a test data set. For each experiment multiple metrics are measured. For classification experiments the confusion matrix is calculated additionally. In order to test hypotheses, either features or the response variable can be permuted.

Active bindings

`permute` returns the instance variable 'permute'. (character)

`permute_alphabet` returns the instance variable 'permute_alphabet'. (character)

`n_resample` returns the instance variable 'n_resample'. (integer)

`fml_method` returns the instance variable 'fml_method'. (character)

`fml_type` returns the instance variable 'fml_type'. (character)

`fml_type_alphabet` returns the instance variable 'fml_type_alphabet'. (character)

`pre_process_lst` returns the instance variable 'pre_process_lst'. (character)

`hyper_parameters` returns the instance variable 'hyper_parameters'. (list)

`response_var` returns the instance variable 'response_var'. (character)

`n_features` returns the instance variable 'n_features'. (integer)

`strata_var` returns the instance variable 'strata_var'. (character)

`metrics_df` returns the instance variable 'metrics_df'. (tibble::tibble)

`confusion_df` returns the instance variable 'confusion_df'. (tibble::tibble)

Methods

Public methods:

- `Resampler$new()`
- `Resampler$print()`
- `Resampler$fit()`
- `Resampler$clone()`

Method `new()`**:** checks, if permutation is requested. If true, performs the permutation task.
Checks if ml.type is classification. If true, calculates confusion matrix.
Creates and returns instance of Resampler class.

Usage:

```
Resampler$new(
  n_resample = 500,
  fml_method = "pcr",
  fml_type = "classification",
  hyper_parameters = "list",
  pre_process_lst = c("center", "scale"),
  permute = NULL,
  n_features = 0,
  response_var = "character",
  strata_var = NULL
)
```

Arguments:

`n_resample` number of bootstrap resamples. The default is 500 (integer)

`fml_method` ML model that is being used. The default is 'pcr' (character).

`fml_type` ML model type. Needs to be 'classification', 'regression' or 'censored'. Default is 'classification' (character).

`hyper_parameters` List of model hyper parameters. (list)

`pre_process_lst` Vector of pre-processing steps. Default is 'c("center", "scale")' (character).

`permute` Permutation method. Needs to be 'none', 'features' or 'response'. (character)

`n_features` Number of features to be chosen in the permutation experiment. Default is 0 (integer).

`response_var` Response variable of the model (character).

`strata_var` Stratification variable (character).

Returns: Resampler

Method `print()`**:** Print instance variables of Resampler class.

Usage:

```
Resampler$print()
```

Returns: character

Method `fit()`**:** Runs the bootstrap analysis based on the instance variables chosen under initialize.

Usage:

```
Resampler$fit(data_df = "tbl_df")
```

Arguments:

`data_df` data set to be analyzed (tibble::tibble).

Returns: None

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

Usage:

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

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Sebastian Malkusch

`run_abc_analysis` *Performs item categorization*

Description

Performs item categorization on permutation or shap analysis object

Usage

```
run_abc_analysis(data_obj, method)
```

Arguments

`data_obj`	Resuls of model interpretation experiment
`method`	Method used for model interpretation (permutatopn or shap)

Details

Interpretation results are passed to the function. Based on the type of interpretation experiment the data is transformed into a uniformly structured data frame. Item categorization is performed by computed ABC analysis. The result is returned in form of a tibble.

Value

A tibble with item categories

a tibble

Author(s)

Sebastian Malkusch

Index