preprocessor Documentation

Release 1.0.0

Harvey Bastidas

Apr 26, 2020

Contents

1 Contents **3**
- 1.1 License . . . 3
- 1.2 Contributors . . . 3
- 1.3 Changelog . . . 3
  - 1.3.1 Version 0.1 . . . 3
- 1.4 preprocessor . . . 4
  - 1.4.1 preprocessor package . . . 4
    - 1.4.1.1 Subpackages . . . 4
    - 1.4.1.2 Submodules . . . 7
    - 1.4.1.3 preprocessor.conftest module . . . 7
    - 1.4.1.4 preprocessor.preprocessor module . . . 7
    - 1.4.1.5 preprocessor.preprocessor_base module . . . 8
    - 1.4.1.6 Module contents . . . 8

2 Indices and tables **9**

Python Module Index **11**

Index **13**

Implements modular components for dataset preprocessing: a data-trimmer, a standardizer, a feature selector and a sliding window data generator.

CHAPTER 1

Contents

1.1 License

The MIT License (MIT)

Copyright (c) 2020 Harvey Bastidas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2 Contributors

- Harvey Bastidas <harveybc@ingeni-us.com>

1.3 Changelog

1.3.1 Version 0.1

- Feature A added

- FIX: nasty bug #1729 fixed
- add your changes here!

1.4 preprocessor

1.4.1 preprocessor package

1.4.1.1 Subpackages

preprocessor.data_trimmer package

Submodules

preprocessor.data_trimmer.data_trimmer module

This File contains the DataTrimmer class. To run this script uncomment or add the following lines in the [options.entry_points] section in setup.cfg:

console_scripts = data-trimmer = data_trimmer.__main__:main

Then run *python setup.py install* which will install the command *data-trimmer* inside your current environment.

class preprocessor.data_trimmer.data_trimmer.**DataTrimmer**(*conf*)

Bases: *preprocessor.preprocessor.Preprocessor*

The Data Trimmer preprocessor class

core()

Core preprocessor task after starting the instance with the main method. Decide from the arguments, what trimming method to call.

Args: args (obj): command line parameters as objects

load_from_config()

parse_args(*args*)

Parse command line parameters

Parameters **args** (*[str]*) – command line parameters as list of strings

Returns command line parameters namespace

Return type argparse.Namespace

store()

Save preprocessed data and the configuration of the preprocessor.

trim_auto()

Trims all the constant columns and trims all rows with consecutive zeroes from start and end of the input dataset

Returns: rows_t, cols_t (int,int): number of rows and columns trimmed

trim_columns()

Trims all the constant columns from the input dataset

Returns number of rows and columns trimmed

Return type rows_t, cols_t (int,int)

trim_fixed_rows (*from_start*, *from_end*)
Trims a configurable number of rows from the start or end of the input dataset

Parameters

- **from_start** (*int*) – number of rows to remove from start (ignored if auto_trim)
- **from_end** (*int*) – number of rows to remove from end (ignored if auto_trim)

Returns number of rows and columns trimmed

Return type rows_t, cols_t (int,int)

preprocessor.data_trimmer.data_trimmer.**run** (*args*)
Entry point for console_scripts

Module contents

preprocessor.feature_selector package

Submodules

preprocessor.feature_selector.feature_selector module

This File contains the FeatureSelector class. To run this script uncomment or add the following lines in the [options.entry_points] section in setup.cfg:

console_scripts = feature_selector = feature_selector.__main__:main

Then run *python setup.py install* which will install the command *feature_selector* inside your current environment.

class preprocessor.feature_selector.feature_selector.**FeatureSelector** (*conf*)
Bases: *preprocessor.preprocessor.Preprocessor*

The FeatureSelector preprocessor class

core ()

Core preprocessor task after starting the instance with the main method. Decide from the arguments, what method to call.

Args: args (obj): command line parameters as objects

feature_selection ()
Process the dataset.

load_from_config ()
Process the dataset from a config file.

parse_args (*args*)
Parse command line parameters

Parameters **args** (*[str]*) – command line parameters as list of strings

Returns command line parameters namespace

Return type argparse.Namespace

store ()
Save preprocessed data and the configuration of the preprocessor.

`preprocessor.feature_selector.feature_selector.run(args)`
Entry point for console_scripts

`preprocessor.feature_selector.feature_selector.score_func_classification(X, Y)`
Used to score the features for feature selection, for regression. To be used in the fFeatureSeclector.feature_selection() method.

`preprocessor.feature_selector.feature_selector.score_func_regression(X, Y)`
Used to score the features for feature selection, for regression. To be used in the fFeatureSeclector.feature_selection() method.

Module contents

preprocessor.sliding_window package

Submodules

preprocessor.sliding_window.sliding_window module

This File contains the SlidingWindow class. To run this script uncomment or add the following lines in the [options.entry_points] section in setup.cfg:

console_scripts = sliding_window = sliding_window.__main__:main

Then run *python setup.py install* which will install the command *sliding_window* inside your current environment.

class `preprocessor.sliding_window.sliding_window.SlidingWindow(conf)`
Bases: *preprocessor.preprocessor.Preprocessor*

The SlidingWindow preprocessor class

core()

Core preprocessor task after starting the instance with the main method. Decide from the arguments, what method to call.

Args: args (obj): command line parameters as objects

parse_args(args)
Parse command line parameters additional to the preprocessor class ones

Parameters **args** (*[str]*) – command line parameters as list of strings

Returns command line parameters namespace

Return type `argparse.Namespace`

sl_window()
Perform sliding window technique on the input the dataset.

store()
Save preprocessed data and the configuration of the preprocessor.

`preprocessor.sliding_window.sliding_window.run(args)`
Entry point for console_scripts

Module contents

preprocessor.standardizer package

Submodules

preprocessor.standardizer.standardizer module

This File contains the Standardizer class. To run this script uncomment or add the following lines in the [options.entry_points] section in setup.cfg:

console_scripts = standardizer = standardizer.__main__:main

Then run *python setup.py install* which will install the command *standardizer* inside your current environment.

class `preprocessor.standardizer.standardizer.`**`Standardizer`**(*conf*)

Bases: `preprocessor.preprocessor.Preprocessor`

The Standardizer preprocessor class

`core`()

Core preprocessor task after starting the instance with the main method. Decide from the arguments, what method to call.

Args: args (obj): command line parameters as objects

`load_from_config`()

Standardize the dataset from a config file.

`parse_args`(*args*)

Parse command line parameters

Parameters **`args`** (`[str]`) – command line parameters as list of strings

Returns command line parameters namespace

Return type `argparse.Namespace`

`standardize`()

Standardize the dataset.

`store`()

Save preprocessed data and the configuration of the preprocessor.

`preprocessor.standardizer.standardizer.`**`run`**(*args*)

Entry point for console_scripts

Module contents

1.4.1.2 Submodules

1.4.1.3 preprocessor.conftest module

1.4.1.4 preprocessor.preprocessor module

This File contains the Preprocessor class, it is the base class for DataTrimmer, FeatureSelector, Standardizer and SlidingWindow classes.

class preprocessor.preprocessor.**Preprocessor**(*conf*)

Bases: *preprocessor.preprocessor_base.PreprocessorBase*

Base class for DataTrimmer, FeatureSelector, Standardizer, SlidingWindow.

assign_arguments(*pargs*)

core()

Core preprocessor task after starting the instance with the main method. To be overriden by child classes depending on their preprocessor task.

main(*args*)

Starts an instance. Main entry point allowing external calls. Starts logging, parse command line arguments and start core.

Args: args ([str]): command line parameter list

parse_args(*args*)

Parse command line parameters, to be overriden by child classes depending on their command line parameters if they are console scripts.

Args: args ([str]): command line parameters as list of strings

Returns: `argparse.Namespace`: command line parameters namespace

parse_cmd(*parser*)

store()

Save preprocessed data and the configuration of the preprocessor.

1.4.1.5 preprocessor.preprocessor_base module

This File contains the Preprocessor class, it is the base class for DataTrimmer, FeatureSelector, Standardizer, SlidingWindow.

class preprocessor.preprocessor_base.**PreprocessorBase**(*conf*)

Bases: `object`

Base class for Preprocessor.

input_file = None

Path of the input dataset

load_ds()

Save preprocessed data and the configuration of the preprocessor.

output_file = None

Path of the output dataset

setup_logging(*loglevel*)

Setup basic logging.

Args: loglevel (int): minimum loglevel for emitting messages

1.4.1.6 Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

preprocessor, 8
preprocessor.conftest, 7
preprocessor.data_trimmer, 5
preprocessor.data_trimmer.data_trimmer, 4
preprocessor.feature_selector, 6
preprocessor.feature_selector.feature_selector, 5
preprocessor.preprocessor, 7
preprocessor.preprocessor_base, 8
preprocessor.sliding_window, 7
preprocessor.sliding_window.sliding_window, 6
preprocessor.standardizer, 7
preprocessor.standardizer.standardizer, 7

Index

A

assign_arguments() (*preprocessor.preprocessor.Preprocessor method*), 8

C

core() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 4
core() (*preprocessor.feature_selector.feature_selector.FeatureSelector method*), 5
core() (*preprocessor.preprocessor.Preprocessor method*), 8
core() (*preprocessor.sliding_window.sliding_window.SlidingWindow method*), 6
core() (*preprocessor.standardizer.standardizer.Standardizer method*), 7

D

DataTrimmer (*class in preprocessor.data_trimmer.data_trimmer*), 4

F

feature_selection() (*preprocessor.feature_selector.feature_selector.FeatureSelector method*), 5
FeatureSelector (*class in preprocessor.feature_selector.feature_selector*), 5

I

input_file (*preprocessor.preprocessor_base.PreprocessorBase attribute*), 8

L

load_ds() (*preprocessor.preprocessor_base.PreprocessorBase method*), 8
load_from_config() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 4
load_from_config() (*preprocessor.feature_selector.feature_selector.FeatureSelector method*), 5
load_from_config() (*preprocessor.standardizer.standardizer.Standardizer method*), 7

M

main() (*preprocessor.preprocessor.Preprocessor method*), 8

O

output_file (*preprocessor.preprocessor_base.PreprocessorBase attribute*), 8

P

parse_args() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 4
parse_args() (*preprocessor.feature_selector.feature_selector.FeatureSelector method*), 5
parse_args() (*preprocessor.preprocessor.Preprocessor method*), 8
parse_args() (*preprocessor.sliding_window.sliding_window.SlidingWindow method*), 6
parse_args() (*preprocessor.standardizer.standardizer.Standardizer method*), 7
parse_cmd() (*preprocessor.preprocessor.Preprocessor method*), 8
Preprocessor (*class in preprocessor.preprocessor*), 7
preprocessor (*module*), 8
preprocessor.conftest (*module*), 7
preprocessor.data_trimmer (*module*), 5

preprocessor.data_trimmer.data_trimmer (*module*), 4
preprocessor.feature_selector (*module*), 6
preprocessor.feature_selector.feature_selector (*module*), 5
preprocessor.preprocessor (*module*), 7
preprocessor.preprocessor_base (*module*), 8
preprocessor.sliding_window (*module*), 7
preprocessor.sliding_window.sliding_window (*module*), 6
preprocessor.standardizer (*module*), 7
preprocessor.standardizer.standardizer (*module*), 7
PreprocessorBase (*class in preprocessor.preprocessor_base*), 8

R

run() (*in module preprocessor.data_trimmer.data_trimmer*), 5
run() (*in module preprocessor.feature_selector.feature_selector*), 5
run() (*in module preprocessor.sliding_window.sliding_window*), 6
run() (*in module preprocessor.standardizer.standardizer*), 7

S

score_func_classification() (*in module preprocessor.feature_selector.feature_selector*), 6
score_func_regression() (*in module preprocessor.feature_selector.feature_selector*), 6
setup_logging() (*preprocessor.preprocessor_base.PreprocessorBase method*), 8
sl_window() (*preprocessor.sliding_window.sliding_window.SlidingWindow method*), 6
SlidingWindow (*class in preprocessor.sliding_window.sliding_window*), 6
standardize() (*preprocessor.standardizer.standardizer.Standardizer method*), 7
Standardizer (*class in preprocessor.standardizer.standardizer*), 7
store() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 4
store() (*preprocessor.feature_selector.feature_selector.FeatureSelector method*), 5
store() (*preprocessor.preprocessor.Preprocessor method*), 8
store() (*preprocessor.sliding_window.sliding_window.SlidingWindow method*), 6
store() (*preprocessor.standardizer.standardizer.Standardizer method*), 7

T

trim_auto() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 4
trim_columns() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 4
trim_fixed_rows() (*preprocessor.data_trimmer.data_trimmer.DataTrimmer method*), 5