# M05 : Open Science and Ethics

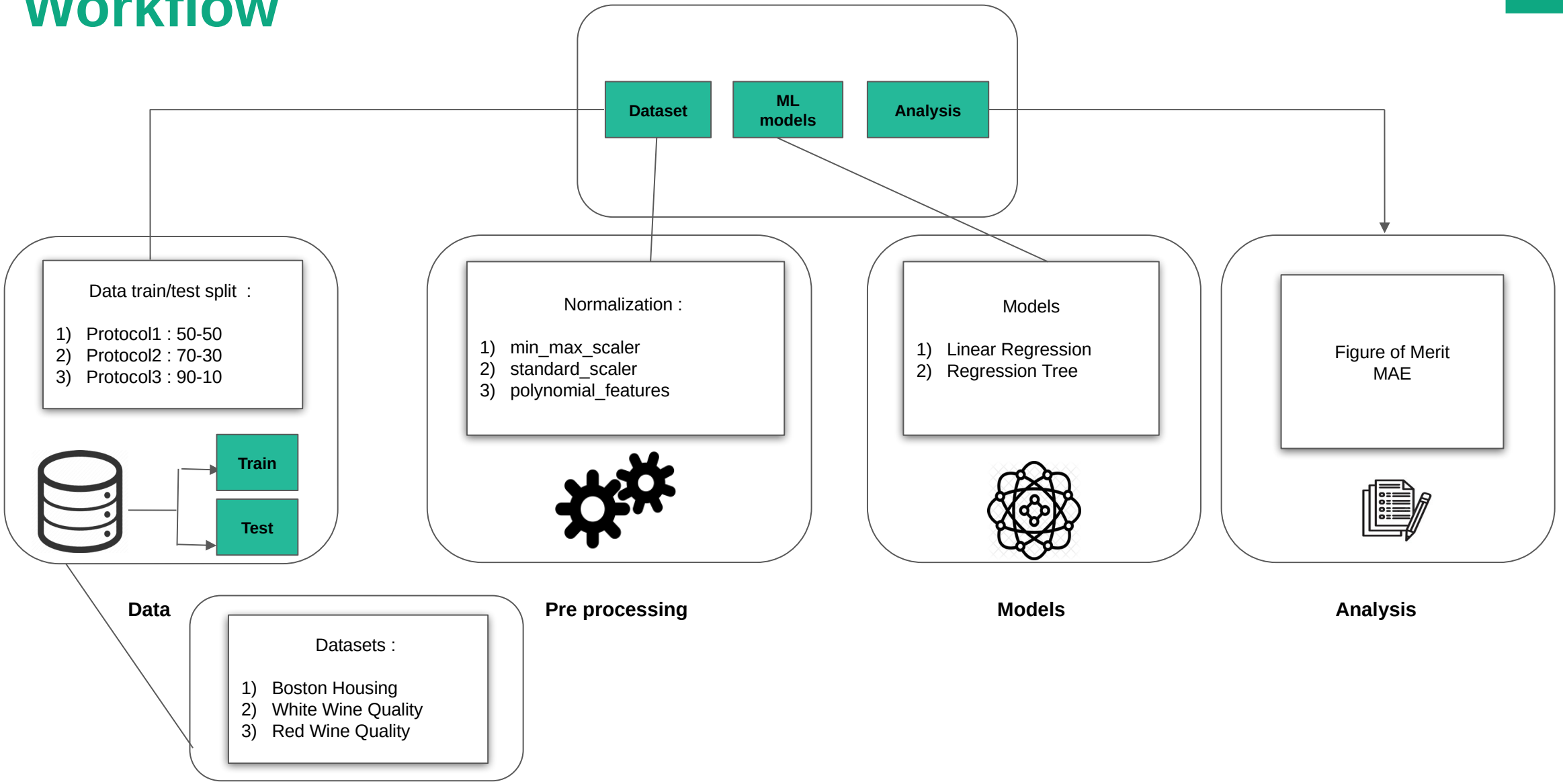
## Mini-Project : Wine Quality and Boston Housing Prices

**Cedric Lienhard**
**Mustapha Al-Dabboussi**
**André Anjos (supervisor)**
**Flavio Tarsetti (supervisor)**


**Master of Science in Artificial Intelligence**

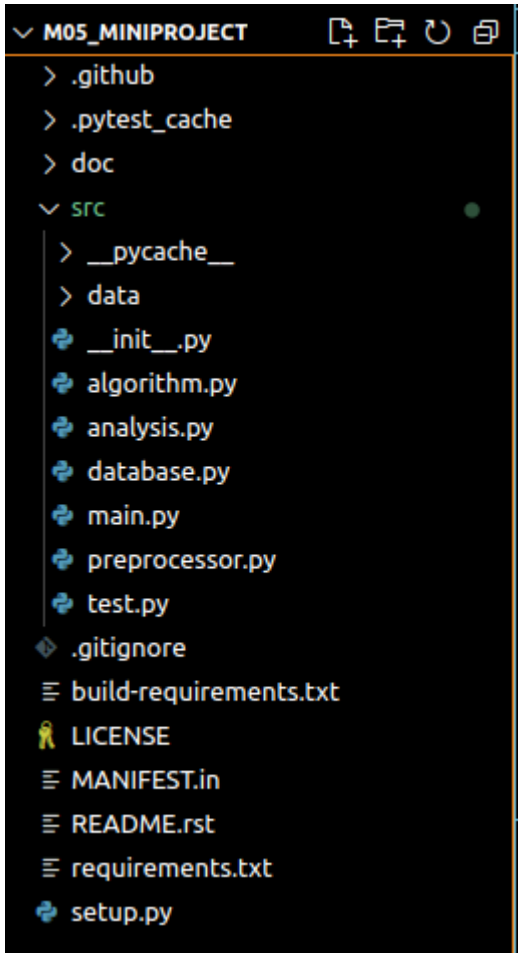
idiap RESEARCH INSTITUTE UniDistance.ch

# Workflow


Dataset
ML models
Analysis
Data train/test split :
1) Protocol1 : 50-50
2) Protocol2 : 70-30
3) Protocol3 : 90-10
Train
Test
Normalization :
1) min_max_scaler
2) standard_scaler
3) polynomial_features
Models
1) Linear Regression
2) Regression Tree
Figure of Merit
MAE
Data
Pre processing
Models
Analysis
Datasets :
1) Boston Housing
2) White Wine Quality
3) Red Wine Quality

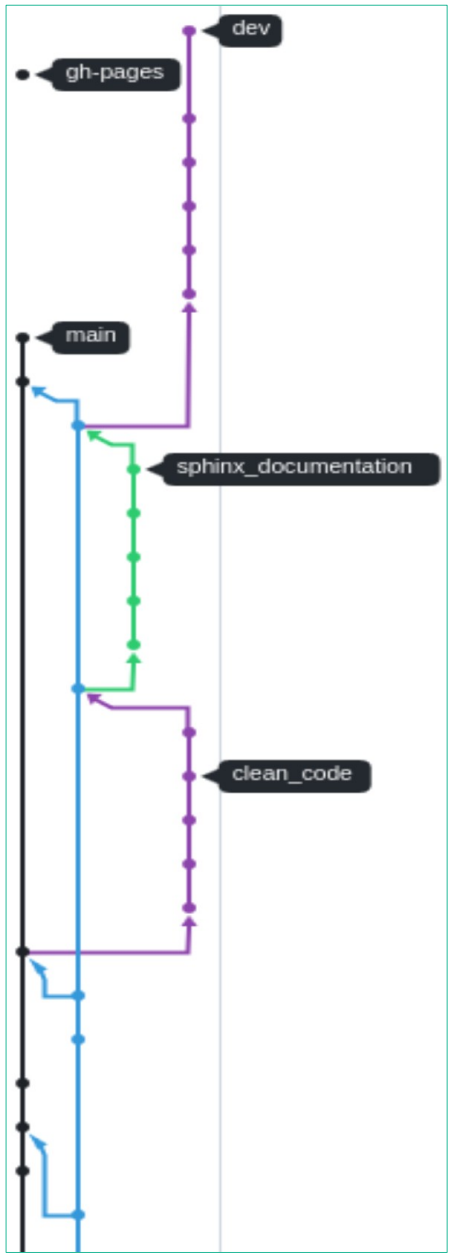
# Project Structure

- The workflow is in «src» folder with the «main.py» file in it
- We use GitHub for code sharing and version control (Git)
- Unit tests are in «test.py» and deployment on coveralls.io
- Continuous integration configuration in «ci-testing.yml»
- Documentation is in the «doc» folder
- Packaging with «setup.py»



# Git and GitHub

- Test and build our modifications on a local repo ⇒ Git
  - Then push them on remote repo ⇒ Github
- Using GitHub as a support to share the code and a remote repository for git
- Link : https://github.com/CedricLienhard/M05_miniProject

# Unit testing and Coverage

- Unit test are in «test.py» file to test functions and functionnalities in the project

```python
def test_MAE():
    Y_train = np.array([1, 2, 3])
    Y_train_predict = np.array([1.5, 2.5, 3.5])
    Y_test = np.array([4, 5, 6])
    Y_test_predict = np.array([4.5, 5.5, 6.5])

    mae_train, mae_test = analysis.compute_performance(
        Y_train, Y_train_predict, Y_test, Y_test_predict
    )
    assert mae_train == 0.5
    assert mae_test == 0.5
```

- The coverage is viewable on GitHub, through a badge
- The deployment is on coveralls.io

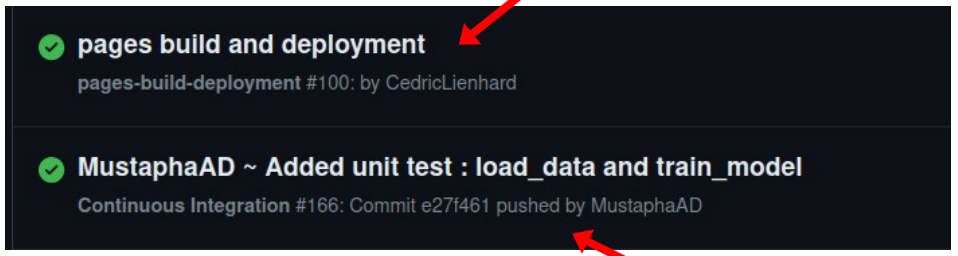

| COVERAGE | FILE | LINES | RELEVANT | COVERED |
|---|---|---|---|---|
| 0.0 | src/main.py | 121 | 46 | 0 |
| 62.5 | src/preprocessor.py | 106 | 24 | 15 |
| 87.5 | src/algorithm.py | 63 | 16 | 14 |
| 100.0 | src/test.py | 103 | 68 | 68 |
| 100.0 | src/__init__.py | 7 | 0 | 0 |
| 100.0 | src/analysis.py | 28 | 5 | 5 |
| 100.0 | src/database.py | 135 | 40 | 40 |

SHOW 10 ENTRIES Showing 1 to 7 of 7 entries

# Continuous Integration

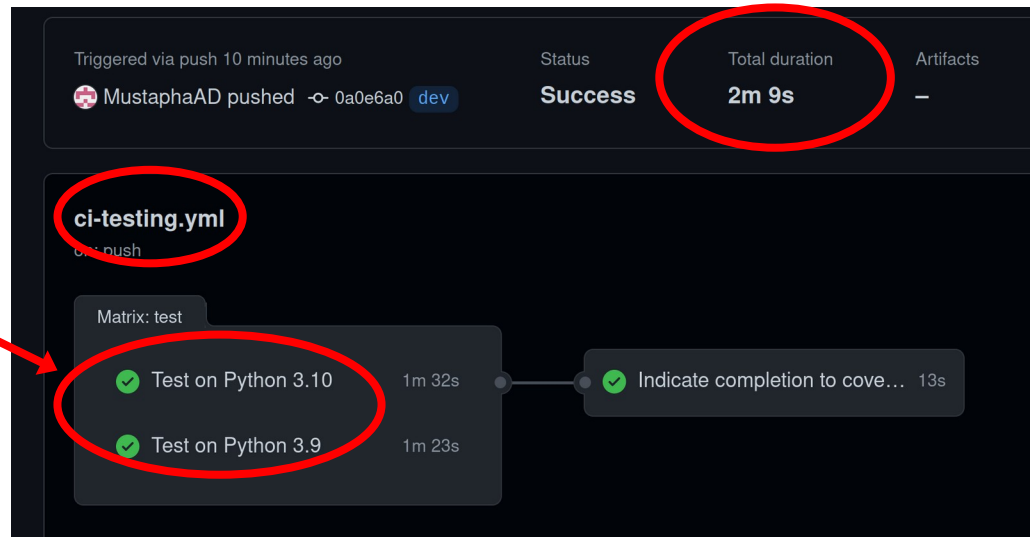
- Continous Integration with GitHub Actions
- Configuration in «ci-testing.yml»

Documentation Deployment




Continuous Integration

Tests run in parallel



# Documentation

- Black was used to format the text

```
-    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = PROTOCOLS[protocol]['test'], train_size = PROTOCOLS[protocol]['train'], random_state=5)
```

```
+    X_train, X_test, Y_train, Y_test = train_test_split(
+        X,
+        Y,
+        test_size=PROTOCOLS[protocol]["test"],
+        train_size=PROTOCOLS[protocol]["train"],
+        random_state=5,
+    )
```

- Docstrings in function

```python
def get(dataset_config, protocol_config):
    """
    Split the data for the dataset, according to the given protocol

    Parameters
    ----------
    dataset_config : str
        define which dataset to load (ex: boston_dataset)
    protocol_config : str
        define which protocol to apply (ex: protocol1 -> 50% / 50% )

    Returns
    ----------
    tuple
        X_train, X_test, Y_train, Y_test
        The value due the repartitions in the train and the test

    """
    X, Y = load_data(dataset_config)

    X_train, X_test, Y_train, Y_test = train_test_split(
        X,
        Y,
        test_size=PROTOCOLS[protocol_config]["test"],
        train_size=PROTOCOLS[protocol_config]["train"],
        random_state=5,
    )
    return X_train, X_test, Y_train, Y_test
```

```
(function) def get(
    dataset_config: Any,
    protocol_config: Any
) -> tuple[Any | list, Any | list, Any | list, Any | list]
```

Split the data for the dataset, according to the given protocol

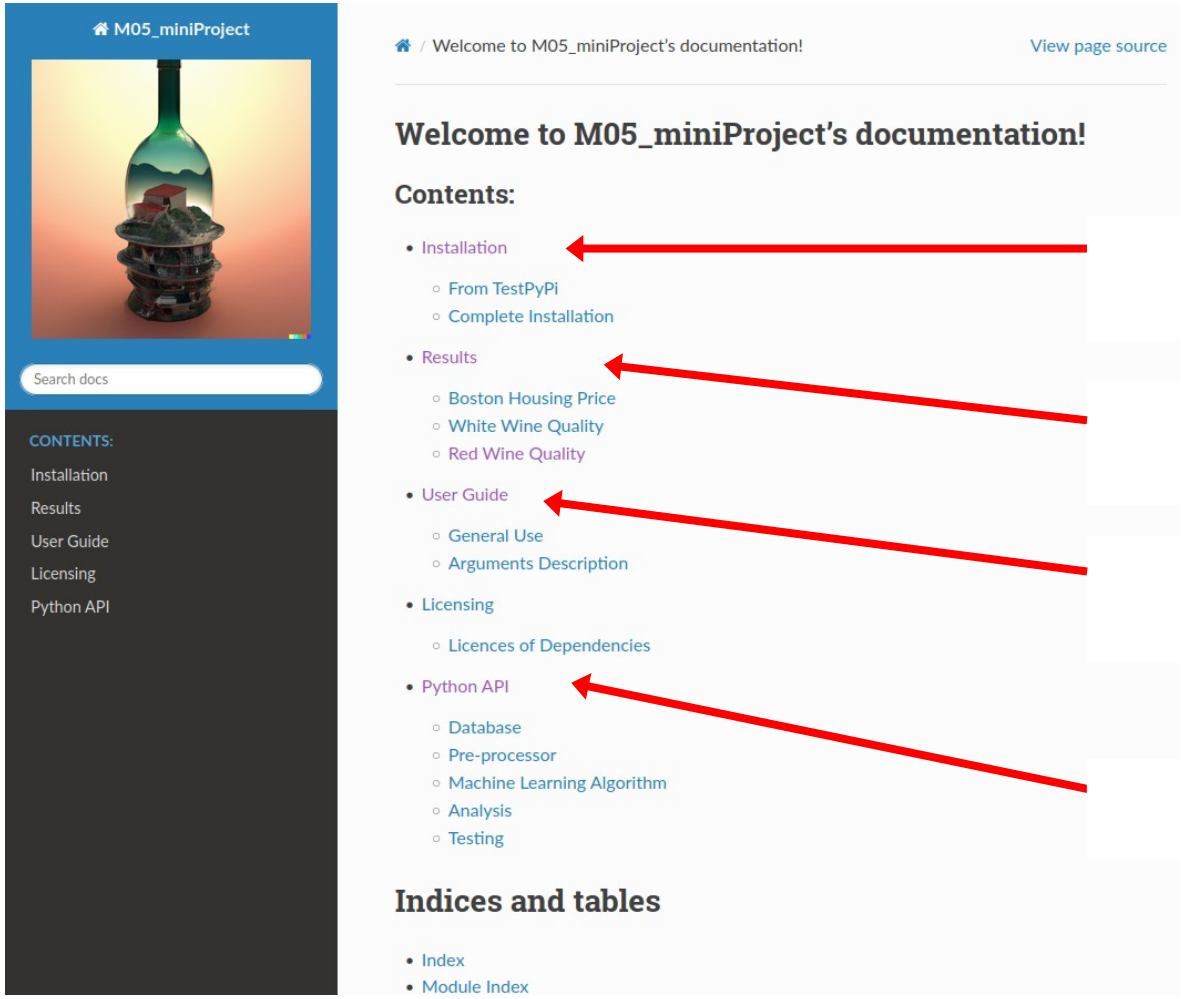
**Parameters**

dataset_config : str
  define which dataset to load (ex: boston_dataset)
protocol_config : str
  define which protocol to apply (ex: protocol1 -> 50% / 50% )

**Returns**

# Documentation

- The documentation page




How to install the package

Results obtained with this package

How to run the code properly

Auto-generated documentation

# Documentation

- Installation, Results and other pages

## Installation

This section shows how to install the package in order to reproduce our results.

### From TestPyPi

As this is a toy project, the package was only deployed on TestPyPi and not PyPi.

You can simply us 'pip' to install the full package.

1. First create a new virtual environment and activate it:

```
$ python -m venv ~/myenv
$ source ~/myenv/bin/activate
```

2. Install the package from TestPyPi with pip:

```
$ pip install --use-feature=2020-resolver --extra-index-url https://test.pypi.org/simple src-bc
```

3. Run the package:

```
$ src-main        # run the package with the standard configuration
$ src-main --help # show help to see how to run the package with a different configuration
```

## Results

This section contains the results of our analysis.

### Boston Housing Price

| Protocol | Preprocessing Algo | ML Algo | Mean Absolute Error (Training set / Test set) |
|---|---|---|---|
| protocol 1 | min-max | linear regression | 3.131 / 3.913 |
| protocol 1 | min-max | decision tree | 1.713 / 3.857 |
| protocol 1 | standard | linear regression | 3.131 / 3.632 |
| protocol 1 | standard | decision tree | 1.713 / 3.393 |
| protocol 1 | polynomial | linear regression | 2.119 / 3.492 |
| protocol 1 | polynomial | decision tree | 1.445 / 2.986 |

## User Guide

### General Use

This guide explains how to use this package and obtain the same results as shown in the Results section.

The main function can be called with different arguments, that can be shown by typing:

```
(project) $ python -m src.main --help
```

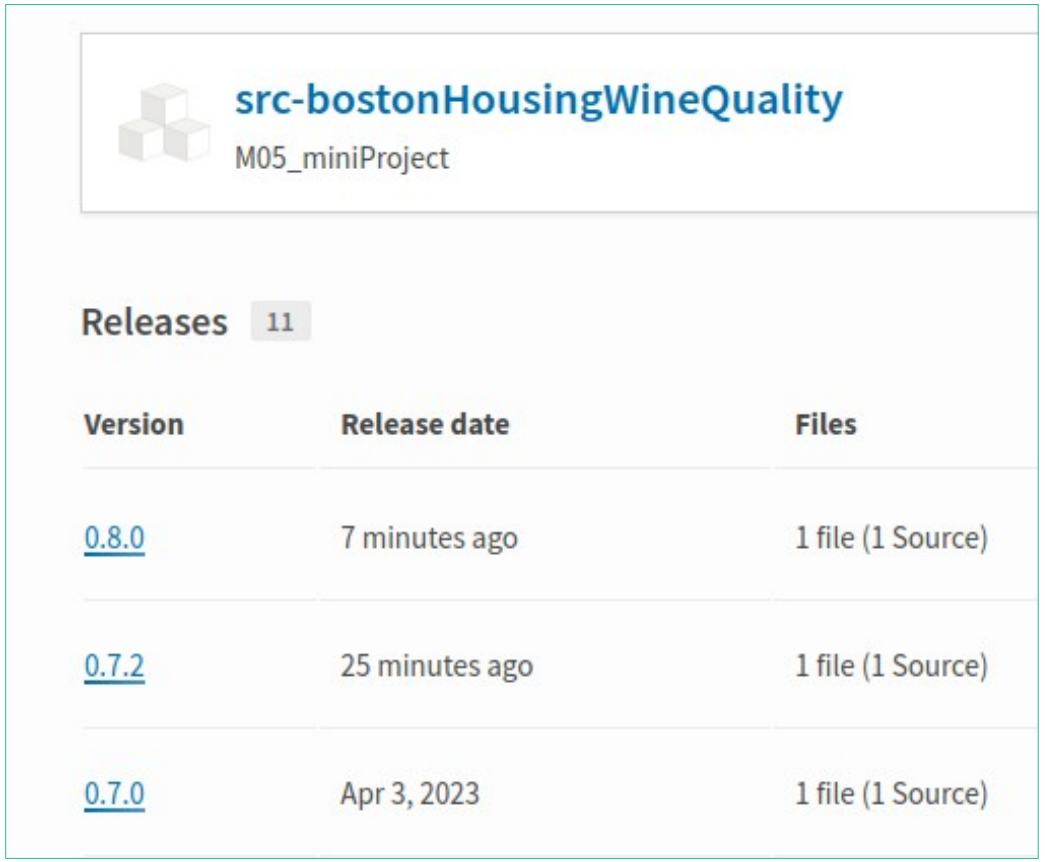
Which should output the following :

```
usage: main.py [-h] [--dataset {boston_dataset,wine_white_dataset,wine_red_dataset,all_datasets
            [--preprocessing {min_max_scaler,standard_scaler,polynomial_features,all_preprocess

optional arguments:
  -h, --help         show this help message and exit
  --dataset {boston_dataset,wine_white_dataset,wine_red_dataset,all_datasets}
                     Name of the dataset to use
  --protocol {p50_50,p70_30,p90_10,all_protocols}
                     Name of the protocol to use
  --preprocessing {min_max_scaler,standard_scaler,polynomial_features,all_preprocessings}
                     Name of the preprocessing algorithm to use
  --ml_model {linear_regression,regression_trees,all_models}
                     Name of the machine learning algorithm to use
```

# Packaging

- Packaging through TestPyPi (Toy project)
  - All required dependencies are included, with their versions
  - Ensures reproducibility of results
  - «Setup.py» file is executed in CI at each new github tag
- Installation with pip and venv
  - Python project (100 % python code)
  - Command ‘src-main’ allows to run the package easily



# Licensing

CedricLienhard/M05_miniProject is licensed under the
**MIT License**

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

**Permissions**

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

**Limitations**

- ✗ Liability
- ✗ Warranty

- Why MIT ?
  - Open Source
  - One of the most permissive free software licence
  - Respects all our dependencies' licences

- Use of pip-licenses to generate licences of dependencies
  - Usage described in documentation

```
(testenv1) [18:09] ~ $ pip-licenses
 Name                          Version  License
 joblib                        1.2.0    BSD License
 numpy                         1.24.1   BSD License
 pandas                        1.5.3    BSD License
 python-dateutil               2.8.2    Apache Software License; BSD License
 pytz                          2023.3   MIT License
 scikit-learn                  1.2.1    BSD License
 scipy                         1.10.1   BSD License
 six                           1.16.0   MIT License
 src-bostonHousingWineQuality  0.7.2    MIT License
 threadpoolctl                 3.1.0    BSD License
```

# Thank you for your attention!

**Reference : M05 Basic Science and Ethics course documentation - 2023**