

Portfolio Selection – Project

In the final project, you will implement your own portfolio strategy based on course material, ideas from the paper presentation and your own knowledge of ML. Your selected portfolio will be tested over September's data.

Framework:

Formulate a portfolio selection strategy. Use stock data from the previous five years to train.

After training, for each day you will receive the updated data and may change your allocation. However, you **cannot** retrain (will be enforced). You **can** make a few decisions based on the newly received data: this includes simple decision rules or mathematical calculations (we have seen several theory grounded calculations). Yet, such measures won't necessarily improve your performance – test your framework!

Evaluation

The grading of the project will be mostly in the form of competition based on your performance in terms of *sharp ratio*. However, you will also be compared to two baselines: *Market* and *Min Variance Portfolio* (without regularization). In addition, your strategy will be evaluated based on your report.

Submission

Your submission must include report and code (30% and 70% of your grade respectively):

- **Project report** (one pager -pdf):
 - Short background of the methods used.
 - Methods – explain your strategy.
 - Experiments – how did you test your strategy?

- **Code:**

A *portfolio.py* module which contains *Portfolio* class.

- `__init__ (self, weights=np.Nan)` –

A constructor can be called with no parameters.

Otherwise, it **may** load a pre-saved weights vector.

Note: If you use a pre-saved weights, then your submission **must** include this file.

- `train (self, train_data: pd.DataFrame)–`

A function that performs the training process. We will not run this function.

Input: *train_data*: a dataframe as downloaded from yahoo finance, containing about 5 years of history, with all the training data. The following day (the first that does not appear in the index) is the test day.

Output (optional): weights vector.

- *get_portfolio(self, train_data: pd.DataFrame)* –

The function generates model's portfolio for the next day.

Input: *train_data*: a dataframe as downloaded from yahoo finance, containing about 5 years of history, with all the training data. The following day (the first that does not appear in the index) is the test day.

Output: numpy vector of stocks allocations.

Note: Entries must match original order in the of the input dataframe!

The model you submit should be trained, if needed, submit a weights' file. In the main.py file you can find functions that will be used to test your code. The submission should contain a .zip file with the name ID1_ID2.zip.

- Submission date: 30.08.22.
- Project must be done in pairs.

Good Luck!