

MLDS HW1 - AutoTrading

Overview

In this HW, we will implement a very aged prediction problem from the financial field. Given a series of stock prices, including daily open, high, low, and close prices, decide your daily action and make your best profit for the future trading. Can you beat the simple “buy-and-hold” strategy?

Evaluation Goal

Maximize the profit you gain.

We will use one month’s data as the test data set. Please aim to maximize revenue in 20 days.

Data

Please check the sample data. You will see each line contains four tuples: **open-high-low-close**. The sample data is **Taiwan Top 50 ETF**. The data, called `training_data.csv`, contains more-than-five-year daily prices, whose line number corresponds to the time sequence. Another data, called `testing_data.csv`, contains one-year daily prices, which corresponds to the time period next to the final date in `training_data.csv`.

[[StockHistoryReference](#)]



Input

- training_data:

<https://drive.google.com/file/d/1Zc2M3JFbNP8v-tSvUAn-zCm0fH4DEDbG/view?usp=sharing>

- testing_data:

<https://drive.google.com/file/d/1XCQFod6lv7veWEydJ8TnMyJ8EsoDMAkV/view?usp=sharing>

Output

1. The output file should be named as "output.csv"
2. Each line contains the action type which will be executed in the opening of the next day.
3. If the testing data contains 20 lines, the output should include 19 lines. The last day will be settled without executing the specified action, and we will use the close price of the last day as the settled price.

Action Type:

The action should be one of these three types:

1 → means to "Buy" the stock. If you short 1 unit, you will return to 0 as the open price in the next day. If you did not have any unit, you will have 1 unit as the open price in the next day. **"If you already have 1 unit, your code will be terminated due to the invalid status."**

0 → means to "NoAction". If you have 1-unit now, hold it. If your slot is available, the status continues. If you short 1 unit, the status continues.

-1 → means to "Sell" the stock. If you hold 1 unit, you will return to 0 as the open price in the next day. If you did not have any unit, we will short 1 unit as the open price in the next day. **"If you already short 1 unit, your code will be terminated due to the invalid status."**

In the final day, if you hold/short the stock, we will force your slot empty as the close price of the final day in the testing period. Finally, your account will be settled and your profit will be calculated.

Note

In this project, we ignore the transaction cost, meaning that you can do an action every day if you want without extra expense (at most one action can be executed within one day, as the open price)

You can hold 1 unit at most. But of course, you can consider "sell short", meaning that you can have "-1 unit".

So that in any time, your slot status should be:

1 → means you hold 1 unit.

0 → means you don't hold any unit.

-1 → means you short for 1 unit.

Grade

We will use two stocks, including the one **Taiwan Top 50 ETF** for your preparation. Another one is TSMC stock, and we believe their behavior are similar.

Data will be the same format: one is training_data.csv, another is testing_data.csv

Your rank is averaged from two results:

1. Top 10% will get Rank "A+".
2. Beat the simple "buy-and-hold" strategy, you will get Rank "A". (If <10% beat the simple strategy, they all get "A+")
3. The worst 10% will get "C-", generally mapping to a score < 70.
4. Others will get "B" or "B+", depending on your idea and profit value.
5. If your code cannot be legally executed (for example, you hold more than 1 unit, or your code cannot be executed), you get a score "D+" (or "D"), which will get a final score <65 (or < 60).
6. If you do not send your code, you will get "0".

Note

- If your code cannot be legally executed or cannot output the output.csv correctly, you will get some deduction.
- **You will get "0" point if one of these situations happens:**
 - Your code is the same as other classmate's.
 - Your Github repo does not contain your code.
 - You do not submit the homework before the deadline.
 - The training dataset you used is not provided by TA.
 - Your code can get data through the Internet during training or testing.
 - Your code terminates during testing.
- 以下中文補充，並無改變原先評分規則
- 每讀完一天的資料，就必須輸出隔天是否要購買股票，輸出後才能再讀下一天的資料，以此類推。
 - 以下列出常見的錯誤，成績將為 0 分

§ 一次讀取多天資料才輸出是否要購買

例子：

D = 新的一天股票資料產生

A = 模型執行的動作 (-1, 0, 1)

合法的順序：DADADADADA

不允許的順序：DDDDADDA

§ 一次讀完全部股票資料才輸出是否要購買

§ 修改輸出結果

§ 輸出結果數量不正確

Requirements

TA will test whether your code can run successfully and output the output.csv or not.
Here are something you need to do:

1. **Upload your code to Github and fill out the google form.**
 - [MLDS 2022 HW1- Github Repo Submission](#)
2. Please use python 3.9 or the higher version.
3. Please name your main python code trader.py, we will test the code by running
 - `python trader.py --training "Training Data" --testing "Testing Data" --output output.csv`
4. Please **clearly describe your idea** in README.md .
5. Please put all library and packages in requirements.txt.
 - a. We will import them by running `pip install -r requirements.txt`.
 - b. [Tutorial of requirements files](#)
 - c. requirements.txt example:

```
scikit-learn==0.24.1  
numpy==1.20.0  
pandas==1.2.3
```

A better way is to use Poetry. [Tutorial](#)

Code for TA trace

For TA traces your code, please write your main code as:

```
# You can write code above the if-main block.  
  
if __name__ == '__main__':  
    # You should not modify this part.  
    import argparse  
  
    parser = argparse.ArgumentParser()  
    parser.add_argument('--training',  
                        default='training_data.csv',  
                        help='input training data file name')  
    parser.add_argument('--testing',  
                        default='testing_data.csv',  
                        help='input testing data file name')  
    parser.add_argument('--output',  
                        default='output.csv',  
                        help='output file name')  
    args = parser.parse_args()  
  
    # The following part is an example.  
    # You can modify it at will.  
    training_data = load_data(args.training)
```

```

trader = Trader()
trader.train(training_data)

testing_data = load_data(args.testing)
with open(args.output, 'w') as output_file:
    for row in testing_data:
        # We will perform your action as the open price in the next
day.

        action = trader.predict_action(row)
        output_file.write(action)

        # this is your option, you can leave it empty.
        trader.re_training(i)

```

- In your if name == '__main__' block, you should call your defined functions to load the data, train your model, write your file into "output.csv" and etc.
- Other functions should be written before the main function or in other modules.
- Structuring your code well is encouraged but not mandatory.
- We will pass argument to your script for testing different datasets. **Make sure** you do not modify the argparse part. If we're not able to pass arguments to your script, your homework will not be graded.
- Note that we will trace your code. You should loop the testing data, and **check one row at a time without looking forward**. We suppose that you will not determine each day action using the whole testing file. If you do so, your homework will **not be graded**.
- [Example Github repo](#)

Test Environment

- Ubuntu 18.04.5 LTS
- Python 3.6

Testing Corrector

You can use the following calculator to check your result.

<https://github.com/NCKU-CCS/StockProfitCalculator>

Deadline

- **2022/10/20 23:59 pm**
- No Exception.

- TA will clone the code committed before the deadline. Any further change will be ignored.

Question

- If you have any questions, feel free to contact the TA or use the “Comment” in the right hand side of this line.
 - TA: 孫毅夫 ifusun@netdb.csie.ncku.edu.tw
 - TA: 曹鶴騰 htcao@netdb.csie.ncku.edu.tw
- Every question asked by students will be listed below

Paper Reference

- [Artificial Intelligence Applied to Stock Market Trading: A Review](#) [2021 IEEE Access]
- [StockAssIstant: A Stock AI Assistant for Reliability Modeling of Stock Comments](#) [2018 KDD]
- [High-order Hidden Markov Model for trend prediction in financial time series](#) [2019 Physica A-statistical Mechanics and Its Applications]
- [Practical Deep Reinforcement Learning Approach for Stock Trading](#) [2018 ArXiv]
- [Adaptive stock trading strategies with deep reinforcement learning methods](#) [2020 Inf. Sci.]

FAQ

Q: training model 資料來源？

- A: TA 批改作業時，將統一使用一組 Training Data 來訓練各位的模型

Q: 可以給 output file 範例嗎？

- A: 一行代表 testing data 一天要採取的動作，若 testing data 有 20 天，則 output.csv 須包含前 19 天的動作(即 19 行)，最後一天批改時會自動採取動作將你的 slot 變為 0
- (E.g., 若你的 slot 為-1，則最後一天會自動幫你決定買股票)

```
1
0
-1
1
```

Q: output file action 的數量是 testing data 的資料數減一嗎？

- A: 是

Q: 最後一天若是持有賣空的股票，會強制買進嗎？

- A: 如果最後一天還有賣空的股票，會以最後一天的收盤價計回現金

Q: 利潤計算方式？

- A: 非賣空利潤：賣價 - 買價；賣空利潤：賣空價 - 買價。
- 可以參照 [+MLDS HW1 - AutoTrading: Testing-Corrector](#)

Q: 那收盤價、最高點跟最低點分別代表甚麼資訊，會引響利潤的計算嗎？

- A: 收盤價僅用做最後一天結算餘款時使用。開高收低四項資料供同學在訓練模型時使用，可以自行決定是否要使用

Q: testing data 資料有缺失

- A: 此資料是取真實資料，應該是當天這筆資料沒有被記錄下來，已更新資料集移除該筆資料。
- 評分使用的 testing data 會跳過沒有完整 feature 的資料，同學不用另外處理資料缺失問題。

Q: 請問這段話的意思是什麼？

- A:
- 這段的意思是你決定隔天動作時，不能去看隔天之後(包括隔天)的股票資訊，
- E.g., 若是 training data 8/1~8/31, testing data 9/1~9/20,
- output.csv 則是 9/1~9/19 每一天是要買股、賣股或是持股的動作（利潤用“開盤價”計算），9/20 我們批改時會自動用“收盤價”計算。
- 而決定 9/1 動作，只能參考 8/1~8/31 股市資訊; 決定 9/2 動作時，只能參考 8/1~8/31 以及 9/1 股市資訊，以此類推。

Q: 可以自行添加 open high low close 到 excel 裡嗎？

- A: TA 做 testing 的時候會使用和目前提供的 testing data 相同格式，請讀取目前的格式後自行處理哦！

Q: training/testing data 沒有時間是正常的嗎？他們是否是連續的資料呢？

- A: 是哦！他們是連續開市的資料。

Q: 只會有 20 筆 testing data 嗎？

- A: testing data 將不限於 20 筆，本次作業為降低難度，會使用一個月(約 20 筆左右的資料)

