Peter Minwegen

Homework 2 – Individual Project Proposal

**Algorithmic Trading Bot**

1)

I would like to develop an algorithmic trading bot. There are swaths of information available online covering this topic. According to the information that I have come across, there are three primary stages that I must implement. I would like to use Python over C++ for my individual project because C++ trading bots were designed for low latency high frequency trading. C++ trades speed for functionality. I would like to incorporate a variety of libraries from Python such as Pandas, NumPy, and SciPy to assist in the generation of my trading strategy formula (Heavy math base).

The first stage would be access to financial data so that I may perform data analysis on it (Yahoo Finance, Google Finance, etc…)(EDIT: Finnhubb.io will be the API used). The form of financial data could be received from numerous sources as many API’s are available. I will likely select the best (data fitting trading strategy) free one, but I might be willing to pay for an API that is affordable if it allows for a great decrease in latency or more calls / second..

The second stage would be to perform data analysis on the current portfolio (if it exists). The algorithm should also parse all stocks present in the imported data from stage 1. The algorithm will perform analysis on the stock data according to the preferred trading strategy (To be determined… ideas include: scalping and swing trades) and create an array of good potential investments name investIn. If time allows, I would also like to import data such as sentiment analysis ( ideally from Stocktwits but after some research I do not believe they allow it to the public. I will continue to research the possible options). While this data is not necessarily crucial; it could aid in the decision between stocks given an account with limited capital. I believe the algorithm needs a sort of ranking system that estimates the potential returns which will be accessed upon analyzing all stocks. Array investIn will be a sorted array (or other data structure that works better) with the best potential stock to invest in at index 0 and the least best stock to invest in would be at index len(investIn)-1. This bot will NOT be a High Frequency Trading bot because I do not believe that I will have the latency to be competitive. A sector identification function might be utilized if this information is not available in any API. Sector identification could be useful for a trading bot to limit the bots trading scope when certain sectors are in economic downturn (example: natural gas & oil) but this also depends on the decided trading strategy because as Warren Buffet has stated, “Be fearful when others are greedy and greedy when others are fearful”. TO CLARIFY. This is a theoretical algorithmic trading bot. Real trades will not be made (although it is possible to implement) and real money will not be used. This tool is designed for educational purposes only.

The third stage will account for a dummy trade. The portfolio will be theoretically impacted and theoretical trades will be implemented. Data such as ticker symbol, number of shares, average price, buy time, sell time will be accounted for. This information will be displayed in a theoretical portfolio history and kept track of in total returns.

2)

I plan on using Python as the programming language for (Libraries: mathplotlib, pandas, numpy, maybe scipy, and maybe others..) I am comfortable using Python (pandas and numpy) as I have not only used this programming language at CU but also in the industry at Two Internships. I have do however have limited knowledge with API’s (I have used Firebase API for swift in mobile application development but I do not have extensive experience using API’s in Python. However, I do believe that I can locate/learn/implement a solution regardless of that limited exposure. I am completely familiar with financial markets and I am an active trader. I have dabbled with Python trading bots through Quantopian but this platform does not implicitly require an API.

3)

**My project won’t work if it doesn’t have current financial data**

(I do not have backup plan for obtaining current financial data because there are numerous sources from which it is available free/paid). This data is essential to determine whether or not a trade should be made and how good of a trade it is.

**My project won’t work if it doesn’t have historical financial data.**

(I do not have backup plan for obtaining historical financial data because there are numerous sources from which it is available free/paid). This data is essential to determine whether or not a trade should be made and how good of a trade it is.

**My project won’t work if it doesn’t have access to a broker**.

If I am unable to access my Robinhood broker. I can pivot the application to other brokerage firms and/or crypto exchanges. Access to the broker/exchange is essential to commit trades.

**My project won’t work if it doesn’t have access to the internet**

Without internet I would be unable to access up-to-date information nor would I be able to trade. I plan to have an internet connection. No backup plan here.

**My project won’t work if it doesn’t have a trading strategy.**

Any trading strategy will do but that doesn’t necessarily ensure profits. I will be continueing research on this topic but some potential strategies might include (stock comparison trading: Stock A and B are directly related. When stock A goes up, stock B goes up. One strategy could account for differences in the relationship between stock A and stock B and invest in any discrepancies. Another strategy could include fundamental analysis (highs/lows, relative highs/relative lows, volume, direction/momentum). Another potential investment strategy could be scalping (Scalping is a trading strategy in which traders profit off small price changes for a stock. Maybe attempt to evaluate oversold positions on stocks to catch a 1-3% rebound day trade).

**My project won’t work if it doesn’t have any capital to trade.**

I currently possess capital in my Robinhood account so no backup plan is necessary.

4)

Outside resources include API calls to Finnhub ( <https://finnhub.io/>) which is FREE. Requests are limited to 1 per second. This information should be current enough. I can simplify requests pertaining to specific tickers if desired rather than all tickers.

5)

Backend will be Mysql. Connection could be established through potential method such as PyMySQL import. I believe that I will attempt to create a front end GUI through PyQT which runs on my computer. If I am unsuccessful in my attempts at creating a GUI, I can implement the program into an iOS application in which I am familiar building (x-code, mobile application development experience).

**User interface Options:**

Start Trading, Stop Trading, Show portfolio, Show history, (maybe avoid X investment sector, or focus on sector X), and possibly more

I plan on having a stocks class with objects relating to each stock. A portfolio class will inherit information from the stocks class. A Historical class will inherit from the portfolio class. Objects will be displayed underneath ‘History’ in the GUI/front end. Current portfolio could be listed above ‘History” maybe underneath “My Investments”.

**Design Pattern:**

**Factory Method,** creating an instance of multiple derived classes.

Main

Scan Portfolio History

B) If Firebase is used….

- Users

- User10234123’s Identifier

- Portfolio Value

- Owned

- (stock ticker symbol)

- Price

- Watching

- Price

If MySQL

User Table:

UniqueID as Primary key, Portfolio Value, Total Return

Porfolio Table: TradeID as Primary key, ticker, price return, percentage return, time

6)

GUI: could display a chart displaying your current portfolio. Next to it could be a display of earnings since launching the bot. Underneath it could be a trading history for the account with its buy in and sell out prices and profit gained or lost in both value (example: $3.42) and percentage (example: 1.45%).

GUI:

Circle is Pie chart that says (no investments at this time). Will adjust according to portfolio. When history is clicked, you can see every trade in a listed format. Displaying ticker, buy price, sell price, price difference, difference in %

A close up of text on a white background

Description automatically generated

7)

i) Homework 2, Part 3 - February

-Create generic Python program that is connected to Finnhubb API and display Finnhub API data when python program is ran. A user will see outputs from the API such as trade data, resistance/support levels, and pattern readings. This will be a demonstration of the API connection at this stage in the project.

ii) Homework 3 - March 13th

-Now that Finnhubb API has been implemented, it is time to prepare the data for analysis. At this stage I will develop a method to call upon the accessed data from the API (likely in a separate thread). I will generate functions which determine critical information that is not supported in the API such as SMA (Simple Moving Average) at specific time frames such as 1M, 5M, 10M, 30M, and 1H. A highestPrice and lowestPrice variable will also be introduced. The program will be rearranged to accurately represent the structure of the program (Classes will be created and organized appropriately) At this stage, the execution of the python program should produce 1M, 5M, 10M, 30M, and 1H Simple Moving Averages for any given financial instrument (most likely a crypto because crypto markets are open 24/7 which is helpful when testing). I will also focus on learning the best possible investment strategy. At the moment I am pursuing a strategy that is determined based on a daily performance rather than a historical or future one (a day trading bot.

iii) Homework 4 - April 3rd

-At this stage I would like to really hone in and improve the trading strategy (optimization). I would like to analyze data capture in the previous stage (Simple Moving Averages, highPrice, lowPrice) to determine relative lows and relative highs. I would also like to use the previously mentioned data to determine a direction of trend. I will determine entry/exit strategies for the investment strategy at this time as well. At this stage relative lows and highs and trend direction will be available to view once the program is executed.

iv) Homework 5, checkpoint - April 17th

At this stage, I would like to execute dummy trades and store them in a dummy portfolio. Theoretical gains and loss should be available. At this stage, the user should be able to execute the python program and view output that trades have been made. If no trades have been made, no output will be visible. The backend will become connected through PyMySQL and MariaDB which will store executed trades. Construction of the GUI through PyQT will begin.

v) Homework 5, final due date - April 29th

By this time, the GUI/front end will be correctly communicating with the program! The user will be able to execute the program to initiate theoretical trades which will be recorded in the MariaDB database.

Learn how to relate a GUI to a Python program

8)

I plan to stay engaged in the course through participation and activity. I still plan on attending lecture and keeping up with PE and lecture assignments. I plan to proactively meet project deadlines. Other forms of engagement include analyzing and executing code demonstrated in lecture on my own machine. I would still develop the PE and lecture assignments in C++ even though I am requesting to use Python for my individual project.

**Github Link:** <https://github.com/techLord-pm/PPWProject/tree/master>