

Stock Trading Bot

SP21 Capstone Design: *Final Design Report*

Team Number SP23-37

Team Members: Beaman Negash, Joseph Barr, David Valverde, Avery Turnbull, Adideb Nag

Adviser(s): Narayan Mandayam

Abstract — Money is one of the primary factors that dictate much of our lives and its quality. Many citizens spend their entire lives trying to figure out how to conserve and make more of it. People are often taught to save their money and budget which are valuable skills but studies show the best financial return is seen within the stock market. The average return from the stock market can be around 10%, far greater than any interest granted to you by a bank and in usual in shorter intervals of time as well. The issue is that the stock market is a very niche related topic with very few people having the necessary knowledge to maximize their profit through investments. That and the fact that the stock market can be extremely volatile and unpredictable creates an environment where people may not reap the full benefits of the stock market. So while it is true that utilizing the stock market is an amazing resource it can work against you without extensive knowledge, research and even some luck. A method to navigate through the perplexities of the stock market would be of great interest to nearly every individual interested in becoming better financially well off, which as mentioned before is everyone. This is the implementation of an algorithm that comes into play, to do the hard hard work for us. By using technology, historical data, and predictive methods into an algorithm, there is potential to do some real good for the economy and simply increase the quality of life of so many.

I. INTRODUCTION

II.

The stock market is an extremely complex mechanism that allows companies to sell shares of their ownership to investors. The value of each individual stock is determined by factors such as the companies financial performance, trends in the market, and overall financial conditions on a global scale. When almost anything seems to change the price of the market on a whim, it can be hard to grasp the facets that come with it. If someone uneducated were to try to dip their toes into the stock market, they could lose dangerous amounts of money before really realizing the damage. Even experienced traders with terabytes of data and statistical analysis are losing money over enough time. When you also account for the fact that in our world today, it is almost impossible to support a family with the median American wage, people start to look for new solutions. Along with the recent eruption in interest of stock market trading surrounding the GameStop incident,

many new resources regarding trading information became a lot more available to the public. Our group realized that this presents an opportunity to build a bot that could sift through magnitudes of more information than any human to try and predict the behavior of the stock market. Accomplishing such a thing, in the right hands, would help a lot of people. Given that the wealth difference in the country is larger than it has ever been, the average American needs to find new ways to earn income. Our group reasoned that a bot that can help people build profits, albeit small, from the stock market would be a good first step to helping close the wealth gap.

Since our group members are almost exclusively software developers, that is the kind of solution we built. Our approach was to try and first build a bot that could confidently generate profits before we tried to tackle any User Interface (UI) development. With some research, it is possible to find multiple similar solutions that claim to make very attractive returns on investments. However, some of these have been proven to be scams, while others don't return nearly as much as promised. However, the one thing that our group wanted to focus on was transparency. That the bot would log trades as well as the "reasoning" that it used. If fully developed, this feature would allow people to learn from the bot. With time, the factors that really carried weight, the news to pay attention to, and all other factors affecting the market would be truly understood. This understanding would give equal footing to anyone with the information. This was the motivation behind building the bot, to introduce ways for people without excess of resources to build success.

III. CONCEPTUAL DESIGN

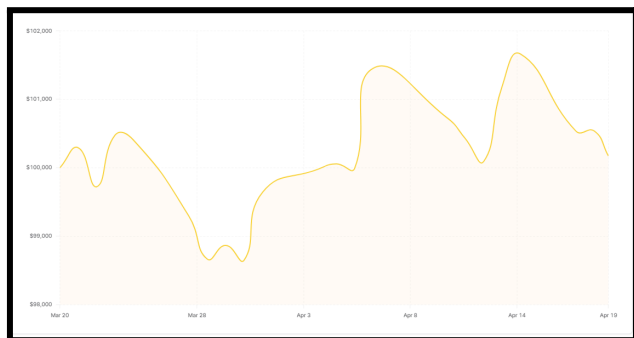
The basic framework of the project was to use a machine learning model with reinforcement learning and feed it historical data of the stock market across many different sectors. This would build a strong algorithm that could predict the behavior of the stock market based on trends that could span a day, a month, or even longer. Once this algorithm was developed, we would attach it to a python program that could feed it real time data from the stock market. After taking past and current data from the stock market and its polished algorithm, it would be allowed to execute trades. Doing so, we

could simulate what it would be like to allow this bot to trade on the open market.

IV. METHODS / RESULTS (ANY RELEVANT) / APPROACH

To handle this kind of project, we understood that we need to break it up into smaller parts. The first part of which was research to ensure that we begin the project with solid foundations. Research was broken up into 2 separate pieces, machine learning (and its implementation) and the stock market itself. From there, we used the information that we gathered to start building our conceptual design.

Regarding our results we started off with a starting balancing of 100,000 dollars in paper money. We used this balance to trade for approximately one month. From March 20 to April 19 we had the stock trading bot trade the Alphabet Inc. Class C Capital Stock. We used the Google stock because it was relatively stable and we could ensure some level of return for the sake of verifying that our bot could successfully operate. At the end of our one month operating we made a profit of \$202.29. The google stock is considered stable so in future experiments with more volatile stocks with higher peaks and would prove to be more fruitful such as penny stock. Even still a 20% increase is a considerable increase given the time from, as well the type of stock that we had the algorithm buy and sell. The Alpaca API allowed us access to a graph of the stock.



Above we can see a visual representation of the trend of the stock. The stock trading bot will consistently check that the stock value is higher than what it was bought for. As you can see by the chart there are essentially two major peaks and one major dip in the stock's value. This drop in value for a typically stable stock can be attributed to the technology industry being on an overall downtrend. Tech stocks fell more than 30% in 2022. This steep decline is as a result of higher interest rates, unstable economic conditions, and heightened inflation. This dip ultimately decreases the total potential profit than what we anticipated. The stock's starting price was \$101.06 when open (March 20) and this value increased to a stock price of \$104.11 at Close (April 19). The ending result was a 3.02% percentage increase in the cost of the stock that the algorithm initially bought.

V. COST AND SUSTAINABILITY ANALYSIS

Economics:

Employing a software implementation means that the cost was simply the man hours required to build the project. If we were to consider ourselves entry level software developers, the median wage in New Jersey is \$70 000, that equates to roughly \$35 / hour. Given that the project took roughly 30-40 hours to build, that comes to a total of \$1050 - 1400. If this compensation was applied to all 5 of our group, then that comes to the range of \$5250 - 7000. This price may seem high but considering the amount of research and development that brought us to this point is also considered within the price as well. In terms of reducing the price, it would be hard to do so since we worked all free software and implementation.

If the product were to be fully developed, the cost of production would be hard to estimate. However, given that it would be a one time expense, with time it would become negligible. In the case that it does take the form that we are hoping, it would be widely accessible to use. This would mean that expenses would shift from production to maintenance of the software and ensuring that the systems used can keep up with the demand of the growing user base. This would also include costs to keep servers up and running to allow the bots to continue analysis for better results when the markets open again.

Our built product does not use too many resources to begin with, so conservation of resources is a bit difficult. Considering that our final product would be quite resource-heavy in terms of electricity and computing power, efforts would need to go into developing algorithms to make it more energy and resource-efficient. In the case where the product were fully scaled up, we would want to make sure that efforts towards the program were actually making progress. Instead of focusing on building a company or brand, we would be trying to make a product that helped as many people as possible.

According to our research, there are no viable options when it comes to tax incentives. While we are trying to help people, there are no plans on donating money directly that could be used in a tax write off.

The availability of the bots would be solely dependent on that of the servers that we would use. That does not leave too much room for market vulnerabilities to sway the product cost and price.

Environmental Impact:

Our group determined that a purely software based solution would have negligible impacts on the environment.

Social Impact:

Our goal building the project was to design something that could eventually be the stepping stone people take to become financially independent. Investment, at this point, requires years of research and dedication to become more accustomed

to the nuances of the stock market. However, we wanted to pull back the curtain so that people can understand what it is that they need to know to properly invest. We believe that, if done correctly, this could begin bridging the gap between the wealthiest and the poorest in the country by teaching people how to build generational wealth.

This would hopefully start to attack many problems that plague lower income neighborhoods such as the national debt epidemic. By starting small and working incrementally, people will be able to create income that can grow passively. In the best case scenario, this would be a good way for people to start tackling both personal and community based issues.

If the product were to hit the markets, our group predicted that there would be little/ no change in employment in any one sector. The sector in question would be investment and brokerage, however, this product would be catered to those of lower income households that are most likely not using portfolio management at this point in time.

VI. CONCLUSION/SUMMARY

To summarize our project, we created an automatic stock trading bot to aid the common person in trading stocks to make a profit. We decided to use Python to implement this because this was the common skill set amongst ourselves and it provides the most useful libraries, modules, and functionalities involving data while cutting back on production time. Our algorithm can be adjusted by changing the refreshing period to make decisions more quickly when connected to the Alpaca application or any stock broker application. We tested our bot through a simulation using paper money (\$100,000) towards Google stocks to minimize project costs. Google was chosen because the stock prices tend to be more stable and predictable. We ended up earning about \$200 after investing for a month. If given more time to learn more about the stock market, machine learning, and applying different data mining techniques and data analytical methods, we could have created a more competitive model compared to what's available today. Trade Ideas is regarded as the best stock trading bot that executes dozens of specific investment algorithms, constantly compares past stock data with current data, offers practice sessions via simulations for user learning, and a simple and customizable layout to optimize user experience.

To further improve our project, we believe that in applying machine learning, additional security, web scraping, and a user-friendly user interface through mobile application (given our skillset, our app would be uploaded on the Apple Store).

Machine learning could be applied to our project by utilizing previous data from the stock market by passing it

through a predictive algorithm and learning algorithm to automate buying and selling stocks based on trends of the recent past through backtesting. First we would need to collect data (stock prices) from different sources to train the model. The raw data collected would then be preprocessed to be inputted into our machine learning model. Analyzing what features are most relevant to the change and predictiveness of trends would allow us to create a more reliable and accurate model. As for selecting what model would best be used, decision trees, random forests, and deep learning models would be appropriate (we would first need to learn these models in depth). After validating our models through simulating paper money, we could then deploy this using actual money (starting off small of course) to create a profit. Our algorithm is more of an iterative "greedy" algorithm that would check to see if the previous stock price were to drop by a certain amount and sell to prevent huge loss.

When we connect the API to Alpaca and from Alpaca to the stock market, it is important to have secured pipelines throughout transferring and transmitting sensitive data reliably and securely using TCP data packets. Although we could use UDP packets since that is what is most common in Internet applications, there is no guarantee in datagram delivery, less secure, and less reliable due to lack of error bit checking. On the other hand UDP could be easier to implement when creating the sockets through socket programming in Python which would be converted to C/C++ to have less execution time for this real-time application. Assuming that we are going to use TCP, we believe that a many-to-many network would need to be implemented so that it can handle multiple hosts (including users, servers, and the individual entities in our model). Using this many-to-many network we can create multiple pipelines between hosts and appropriate groups to regulate data traffic and bandwidth to uphold quality of service, prevent denial of services due to network overloading, and increase security that acts as a personal key to personal door(s). However this may be harder to implement because each link would need to be secured and issues like autoconfiguration to build these links would be tricky to deal with and coming up with routing algorithms that can scale well. A simpler first step would be to create a one-to-many network where there is one "huge" pipeline with a high bandwidth that would lead said hosts and groups. Similar issues would be faced like autoconfiguration, network resource consumption, and secured communication to prevent hackers from stealing private information through the main pipeline.

To our knowledge and as we see in social media, the stock price of any company can change drastically off of one tweet, post, or message by famous celebrities (i.e Elon Musk

skyrocketing GameStop's stock value). By creating a web scraping algorithm that collects companies' financial data, stock market data, and social media feedback from various reliable sources, we can utilize and analyze these different data sets to determine appropriate weights that have a strong correlation in the rise and fall of stock trends. This could be a long and tricky process because many interpretations can be acceptable based on the (data, weight) pairings and have multiple different predicted outcomes.

To make our project more common, accessible, and marketable in today's market is to create an app. Having an easy-to-use and easy-on-the-eyes user interface can generate popularity to an audience new to trading. However our app must also be informative and complex enough to gain popularity from those already familiar with trading or professionals themselves. To meet the needs of our two biggest audiences, we believe that creating a tutorial on how to use our app and providing adequate resources, videos, and extensive documentation to inform and educate our user to make educated decisions instead of blindly following a trend, someone else's advice, and things of that nature. Adding a set level to determine how much money the user is willing to invest with can be implemented to have the user start out small and get comfortable with trading before getting into the more

high risk and high reward territory (to avoid negative socioeconomic impacts in minority communities, where low income homes and poverty are common). We would also like to take in user experiences and feedback to further improve our app by creating widgets and features to aid in making decisions easier and more informative.

ACKNOWLEDGMENT

REFERENCES

- [1] Academic.oup.com. [Online]. Available: <https://academic.oup.com/rfs/article/32/3/992/5061375>. [Accessed: 03-May-2023].
- [2] M. Mathur, S. Mhadalekar, S. Mhatre, and V. Mane, "Algorithmic trading bot," ITM Web of Conferences, 09-Aug-2021. [Online]. Available: https://www.itm-conferences.org/articles/itmconf/abs/2021/05/itmconf_icacc2021_03041/itmconf_icacc2021_03041.html. [Accessed: 03-May-2023].
- [3] R. Whaley, "Trading volatility: At what cost?," SSRN, 07-May-2013. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2261387. [Accessed: 03-May-2023].
- [4] T. Parker, "4 behavioral biases and how to avoid them," Investopedia, 05-Apr-2023. [Online]. Available: <https://www.investopedia.com/articles/investing/050813/4-behavioral-biases-and-how-avoid-them.asp>. [Accessed: 03-May-2023].
- [5] "Were Tech Stocks Down In 2022—And How Long Will The Slump Last?" 19-Jan-2023 [Online]. Available: <https://www.forbes.com/sites/qai/2023/01/19/why-were-tech-stocks-down-in-2022-and-how-long-will-the-slump-last/?sh=716fb8a7f160>