New York University Computer Science Department Predictive Analytics

Final Project Report 2022

Dr. Anasse Bari

Portfolio Management Using Reinforcement Learning and Sentimental Analysis

Name	**NetID**
Anugya Shah	as15119
Arshdeep Singh Gill	ag7982
Lakshmi Sai Anmol Ramayanam	lr2947
Mohammed Aarem Barge	mpb9162
Mohit Kandelwal	mk7711

Abstract

One of the most challenging financial systems to model at the moment is the stock market. In order to efficiently maximize return while lowering risk, an optimal investment strategy must be developed for a carefully chosen collection of companies. This makes resolving the stock portfolio allocation problem challenging. By using an intelligent agent that has been trained on past stock prices, deep reinforcement learning techniques have demonstrated promising outcomes when applied to automate portfolio allocation. Modern investors, however, are actively using digital channels like social media and online news websites to better understand and assess portfolios. Current Predictive Analysis methods do not take into account market sentiment, which have been empirically proven to affect investment decisions. In our research, we suggest a cutting-edge deep reinforcement learning method to efficiently train an intelligent automated trader that not only analyzes historical stock price data but also discerns market sentiment for a stock portfolio.

Keywords: Portfolio allocation, Deep learning, Reinforcement learning, Sentiment analysis, Stock trading

Introduction

One of the trickiest and most intriguing issues in contemporary finance is portfolio allocation. This is due to the fact that the stock market is a complex system that is entangled in a web of interconnected return effects, making it difficult to separate and model return regularities. Predicting stock price changes is a difficult undertaking due to the ongoing evolution of the stock market. A trader must continually be able to diversify and re-allocate funds among the companies in their portfolio to maximize profit while also avoiding risk if they are to attain the best portfolio allocation solution.Finding a trading strategy that is both profitable and low-risk is undoubtedly one of the finest methods to secure financial success. Consequently, a number of investment management firms are consistently working to find better solutions for this issue by utilizing more advanced techniques.

The foundation of some of the earlier publications on portfolio allocation was a mathematical model that made use of methods like quadratic programming, stochastic calculus, numerical analysis, etc. The Newton-Raphson technique, for example, was employed to solve logistic regression. Initially, statistical learning approaches were utilized for simple answers that needed numerical analysis. But when supervised machine learning technologies like artificial neural networks gained popularity in the 1990s, multiple deep learning strategies were put forth for a range of stock market applications. The ability of neural networks to learn complex nonlinear functions is thought to be the reason why they are more effective at predicting stock returns than other machine learning techniques.

Studies that employed supervised learning to address portfolio allocation generally used a trading through forecasting strategy as their main method of operation.There are two steps in this method :

1. Utilizing historical asset price data as well as other pertinent parameters for training, create a predictive model to anticipate how the price of the assets will change over time.
2. Use the asset price projection to determine whether to purchase, hold, or sell while making trading decisions.

This straightforward and well-liked two-step method can appear perfect at first, but there are some significant limits that could result in less-than-ideal performance. The goal of a system that seeks to solve portfolio allocation is not to minimize forecasting error (as part of the first phase). Instead, the decision-making stage takes on a greater significance in the process as a whole. There is a "forecast bottleneck" caused by the widespread practice of solely using forecasts from the first step for decision-making, which results in less information being provided to the decision-making module than to the forecasting module.

In recent years, reinforcement learning has shown to be a potent technique for solving sequential decision-making problems and has been utilized to create intelligent agents that are capable of learning sophisticated tactics.The allocation of a portfolio can be modeled as a Markov Decision Process in reinforcement learning, which not only combines the two steps needed in traditional supervised learning approaches into a single integrated step that closely resembles the thought process of an actual investor but also overcomes their shortcomings. However, basic reinforcement learning finds it difficult to scale with the extensive information processing needed for the majority of real-life issues. In order to scale reinforcement learning to previously unsolvable issues, deep learning techniques are now being used in conjunction with it. Approximating functions and sophisticated feature extraction are the two main methods used to do this. We employ a deep reinforcement learning strategy for our work that is based on the latter.

Purely quantitative models have improved in performance over time, but they have always only taken into account a portion of the situation; they presuppose that an investor bases all of their decisions solely on quantitative inputs like stock prices, company performance metrics, and other technical aspects. Several qualitative elements, including the general market outlook, a company's reputation and brand, shareholder contentment, and others, have been found to have an impact on investors. "Market sentiment" or "investor sentiment," which is a gauge of the general view of investors on a stock or financial market, captures these elements.Through communication and information sharing, investors may now thoroughly examine and analyze their financial holdings to make smarter, more informed decisions thanks to the internet and social media. Massive volumes of unstructured data are now readily available, which enables models to accurately assess market sentiment and simulate genuine market dynamics.

In this essay, we provide a method in which our trader receives an external market stimulant from sources like social media and traditional news media, just like a trader would in reality.We take into account a portfolio of FANG companies from the American stock market, which are regarded as some of the most significant and reputable tech giants. We offer an approach to deep reinforcement learning that incorporates market sentiment and learns to dynamically use it, based on the adaptive deep deterministic policy gradients (DDPG) algorithm . A qualitative element called market sentiment will be added to a quantitative deep reinforcement learning model, and its impact will be examined.

Outline

The rest of this essay is structured as follows: in Section(Background), we provide the background information on portfolio allocation that is important, outline our problem, go over pertinent research on market sentiment analysis, and introduce deep reinforcement learning techniques. In Section (Methodology), we go into more depth about the specifics of our suggested strategy, and in Section (Data Gathering and Processing), we go into more detail about the methods applied to gather and analyze the data. We review the findings in Section (Results) and wrap up in Section (Conclusion) by outlining our major contributions, summarizing our work, and laying out the possibilities for further investigation.

Background

Stock Portfolio Allocation

One of the most popular ways that investors try to increase their financial standing is through stock trading. A stock portfolio is the group of equities that an investor selects to monitor, assess, allocate, and transfer capital. In general, a certain amount of risk is expected with investments on a stock portfolio due to the volatility and uncertainty associated with stock market price movements, i.e., along with the potential for great returns, there is a chance that the value of the investment may decline or not perform as expected. Therefore, the optimum investment strategy for an investor would be one that attempts to maximize profits while assuming the least amount of risk.Portfolio allocation is the process of putting such a strategy into practice while taking the time horizon, risk tolerance, and investment goals into account.

Sentiment analysis

Sentiment analysis is also known as a contextual mining of text. It identifies subjective information in source material and extracts it. This generally helps a business to understand their brand value and what is going on in the market through the online chats, tweets, social media, etc. However, this analysis of social media tweets, comments are generally restricted to just the basic sentiment analysis.

For the extraction of contents from social media, the social site that we used is Twitter and we also used Tweepy. Twitter, which is a popular social networking site where we generally share tweets in form of our opinions. Hence, for the data extraction part Twitter allows us to mine the data of any user using Twitter API. This Twitter API is also commonly knows as Tweepy.

Tweepy

The twitter RESTful API the data will be tweets extracted from the user. Here, the first step that we have to follow is to get the consumer key, consumer secret from the twitter developer. These keys are further used for the API authentication. With api.user_timeline command in use we can get the tweets of a particular user per se. Once the data is collected we can hence apply the machine learning algorithms to extract the most out of it.

The other technology that we used is known as Opengym ai:

Reinforcement Learning with OpenAI Gym

There are mainly three categories of learning og data models and Reinforcement learning is one of these. In most of the real world scenarios calculating the reward value function generally becomes tough and hence rather than learning from the data provided to it the learning takes place in real time with mostly the interactions. Hence this is the winning strategy of the Reinforcement learning. As this is mainly based with the interactions rather than the pre trained model or before hand data this learning is mostly known as the model free learning.
OpenAI Gym is known as the most common choice for implementing environment in which we can train our agents. This further provides an easy API to implement your own environments. This is a python library. It provides a huge number of test environments to work on the Reinforcement learning algorithms with the shared interfaces for test and writing the general algorithms. OpenAI Gym is compatible with both the Pytorch and also the Tensorflow.

Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is one of several varieties of Recurrent Neural Network (RNN), and it has the ability to collect data from earlier stages and utilize it to make predictions for the future.Three layers typically make up an artificial neural network (ANN): input layer, hidden layer, and output layer are in that order.

The number of nodes in the input layer of a NN with a single hidden layer always depends on the dimension of the data, and the nodes of the input layer are linked to the hidden layer by connections known as "synapses." Every two-node relationship from (input to the hidden layer) contains a coefficient called weight that determines how signals are processed. After the learning process is complete, the Artificial NN will have the best weights for each synapse. Learning is a natural process that involves ongoing weight adjustments.

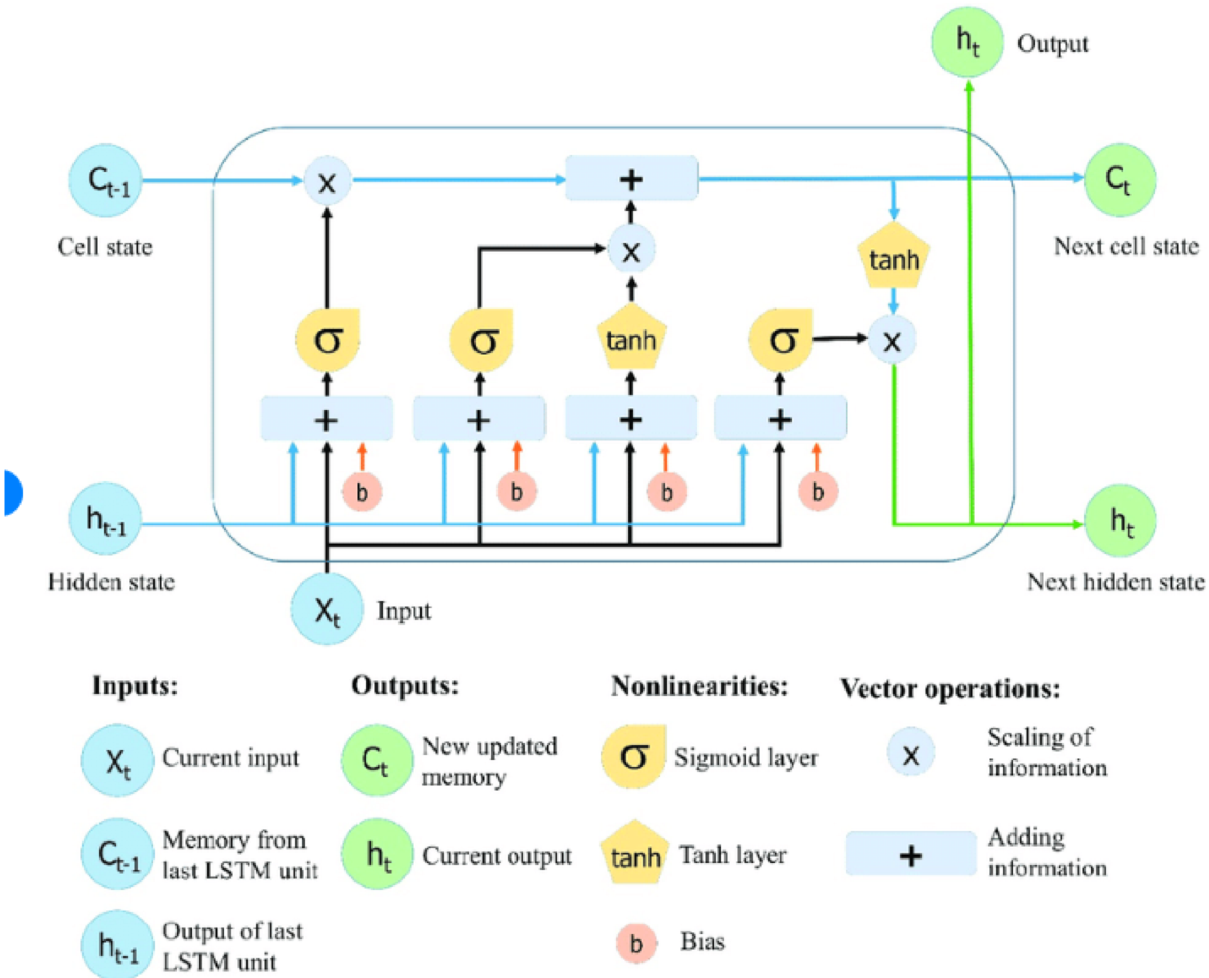
The activation function, which is a transformation that produces values with a reduced error rate between train and test data using the SoftMax function, is applied by the hidden layer nodes to the sum of weights coming from the input layer.

The values obtained following this transformation make up the output layer of our NN; however, these values might not be the best. In this case, a back propagation process will be used to target the optimal value of error; this process connects the output layer to the hidden layer and sends a signal conforming the best weight with the optimal error for the chosen number of epochs. We'll go through this process again in an effort to refine our predictions and reduce prediction error.

The model will then be trained once this process is finished. Recurrent Neural Networks (RNN) are a class of NN that use previous stages to learn from data and forecast future trends. They estimate future value based on previous sequences of observations.

To forecast and guess future values, it is important to keep in mind the earlier phases of the data; in this situation, the hidden layer serves as a repository for historical data from the sequential data. The method of using components of previous sequences to predict future data is referred to as recurrent.

Long Short-Term Memory (LSTM) based on "memory line" found to be highly helpful in forecasting scenarios with long time data because RNN cannot store long time memory. A LSTM's gates with an integrated memory line can be used to recall information from prior stages. The make-up of LSTM nodes is shown in diagram 1 below.



The LSTM is a unique class of RNNs since it has the capacity to memorize data sequences. The upper line in each cell connects the models as a transport line, handing over data from the past to the present ones. The independence of cells aids the model's disposal filter by adding values from one cell to another. Every LSTM node must consist of a set of cells responsible for storing passed data streams. By discarding or allowing data to flow, the sigmoidal neural network layer that makes up the gates ultimately drives the cell to an ideal value. Each sigmoid layer has a binary value of either 0 or 1, where 0 allows for "nothing to pass through" and 1 allows for "everything to pass through."

In order to control each cell's state, the gates are managed as follows:

- The Forget Gate generates a value between 0 and 1, with 1 denoting "totally keep this" and 0 denoting "completely ignore this."
- Memory Gate selects the fresh data that will be kept in the cell. The initial "input door layer" of a sigmoid layer selects the values to be modified. A vector of potential new candidate values that could be added to the state is then created by a tanh layer.
- Each cell's output is determined by the output gate. The filtered and most recent added data, along with the cell state, will be used to determine the output value.

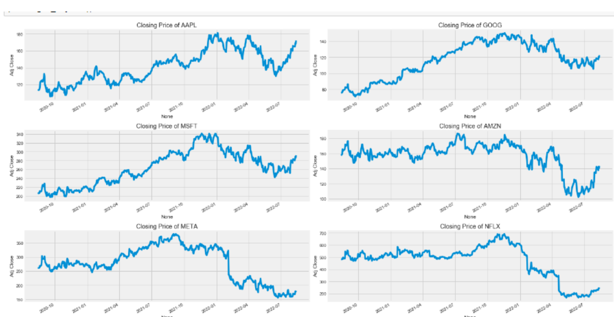
Data Processing

Financial Data

Market data are the temporal historical price-related numerical data of financial markets. This data is used to analyze the historical trend and the latest stock prices in the market, which helps in understanding market behavior.

The information about: is accessible through the Yahoo Finance API.
balance sheet and profits summaries in finance.
historical stock prices
stock trading (including splits and dividends).
All of this data was obtained from the official Yahoo Finance page.

This data is collected from Yahoo Finance using Yahoo Finance APIs. OHLC and Volume data are normalized. There are some limitations by making the call to Yahoo Finance API: Using the Public API (without authentication), you are limited to 2,000 requests per hour per IP (or up to a total of 48,000 requests a day).



Textual Data

Along with other search options like date range and many others, Twitter also offers the ability to search tweets based on keywords. Data collection from Twitter required three steps.Tweet retrieval, data extraction, and sentiment analysis.

To find tweets about companies on specific dates, we combined a keyword-based search with a date specification. To collect the most pertinent tweets, we scientifically framed our terms. We utilized search_tweets API available to retrieve the tweets for a corporation using the predefined keyword for all the days under consideration. The scraped tweets were saved into a data frame. This is the scraping phase for Twitter.

A tweet contains a variety of data variables, including the author, the author's handle, the tweet's text, the date and time it was posted, the number of retweets, replies, and likes, among other things. We refer to the number of retweets, replies, and likes as engagement data since they show how the community responded to the tweet. It's crucial to note that when a user retweets another user's tweet, they are effectively telling their followers the same thing. Similar to this, like a user's tweet typically signifies agreement with the idea or sentiments expressed therein. One of the most crucial fields in a tweet is the text because it expresses the user's concept or attitude about the business.We therefore focused only on the text and the date part of the tweet data extraction process because we were mainly interested in the date and text.

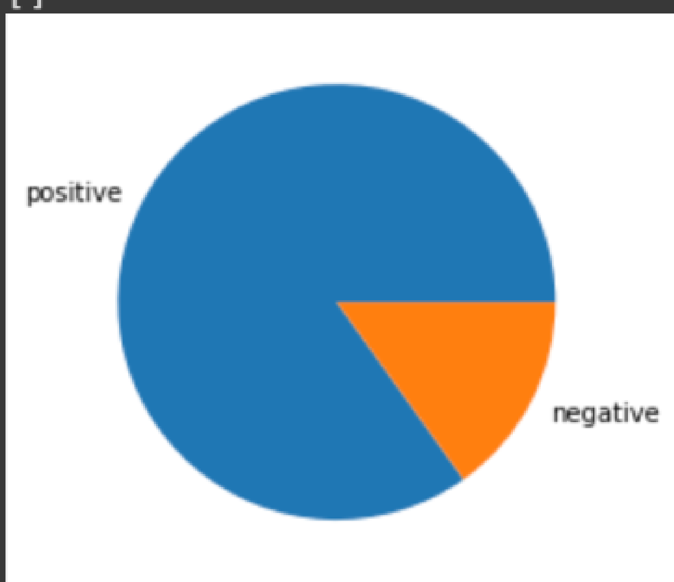
The texts in the tweets are processed to remove the content which is no longer relevant. We used textblob to do sentiment analysis on tweet texts to determine their polarity. The Twitter sentiment score (TS) for the company on that specific day was calculated using these polarity scores combined with the engagement information for each tweet. The tweets sentiment score for a given day is set to 0 if there are no available tweets for a certain company on that day, signifying neutral or unaltered sentiment. A feelings file specifically for the company houses the calculated TS for the given period. Such TS is computed over all dates and businesses. The period of sentiment analysis for tweets is now. We have also focused on computing the positive, negative and neutral sentiments. But considered TS polarity to be more relevant feature for our model.

```
cdata.head()
```

	Date	Tweets
0	2022-08-17	RT HonkaiImpact3rd New Theme Song TruE Release...
1	2022-08-17	RT appltrack The Apple Card is 3 years old tod...
2	2022-08-17	RT Hindus4HR We commend Apple and IBM for beco...
3	2022-08-17	RT AlbumTalksHQ After over 5 Weeks of release ...
4	2022-08-17	RT snohaalegra 3 years ago today I shared this...

	Date	Tweets	Comp	Negative	Neutral	Positive
0	2022-08-17	RT SpitfireVC Lizz Truss doesnt believe in han...	0.2584	0.071	0.779	0.15
1	2022-08-17	Stand Steady Premier ClampOn Keyboard Tray Ex...	0.0	0.0	1.0	0.0
2	2022-08-17	RT BookDuke THE PARTY HOUSE An engrossing Hig...	0.5319	0.0	0.823	0.177
3	2022-08-17	buoyantberries Join the Hunt as the Galloways ...	0.296	0.0	0.896	0.104
4	2022-08-17	RT ShannonVallor The saddest thing for me abou...	-0.7351	0.228	0.772	0.0

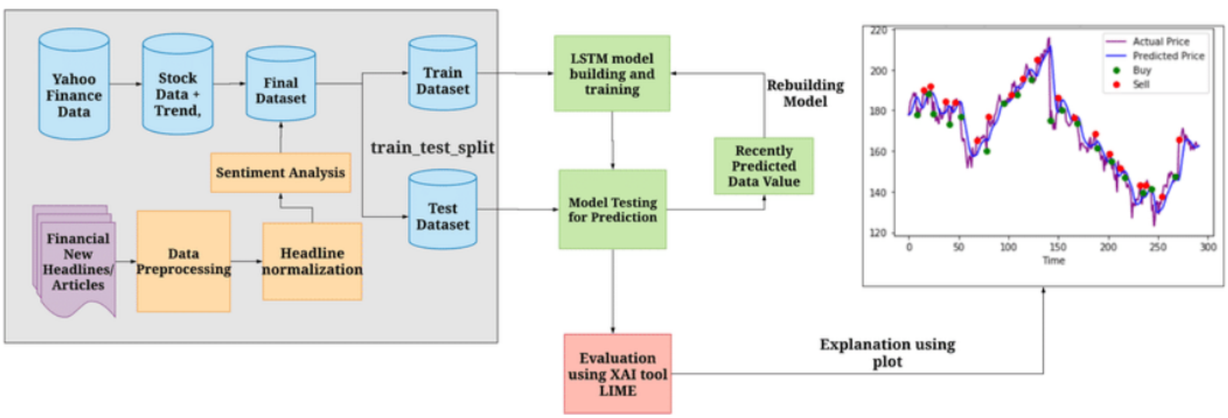
```
% of positive tweets=  56.62650602409639
% of negative tweets=  10.843373493975903
[]
```



Methodology

LSTM And Twitter Sentiment Analysis

Several factors have an impact on the stock market. It's crucial to have a firm understanding of the pertinent stock market information if you wish to correctly predict changes in the stock price of specific companies. In this study, we put forth a technique that looks at forum discussions and news articles for sentiment as the experience following fundamental research, and downloads historical stock prices for technical analysis. For the sake of this project we have used Tweets from Twitter as the source to predict the sentiment of the market.



To ascertain the likelihood of good or negative outcomes, retrieved tweets were fed into the BERT model. The opening price, closing price, highest price, lowest price, and volume of transactions during the last four years were integrated with our three-dimensional data on social media attitudes and our five-dimensional data on historical information. The LSTM neural network prediction model forecasts the stock prices for specific stocks using the eight dimensions of data as input characteristics.

Reinforcement Learning And Stocks

The environments where our trading bot will practice trading are first imported. The RL algorithm and helpers from stable baselines are the items we import next. The algorithm we will be employing is A2C. Advantage Actor-Critic is known by the acronym A2C. It is an RL method that incorporates approaches from Value-Based and Policy-Based RL. The MlpLstmPolicy, a deep neural network policy with an LSTM layer. It is a crucial policy as it enables a neural network to retain context and teach its neurons about the past.

In the modeling phase, we have first used only the historic data of the stock, such as Open, High, Low and Close, and Volume to the model. The results of this said model were not satisfactory

Reinforcement Learning with Technical Indicators

Here, we have created a custom environment for training our bot , where we included technical indicators, RSI(Relative Strength Index), OBV(On Balance Volume) and SMA(Simple Moving Averages). We have used a2c model and the input used was OHLC, volume and these technical indicators.

OBV - Using volume flow to forecast changes in stock price, on-balance volume (OBV) is a technical trading momentum indicator. The OBV metric was initially introduced by Joseph Granville in his 1963 book Granville's New Key to Stock Market Profits.

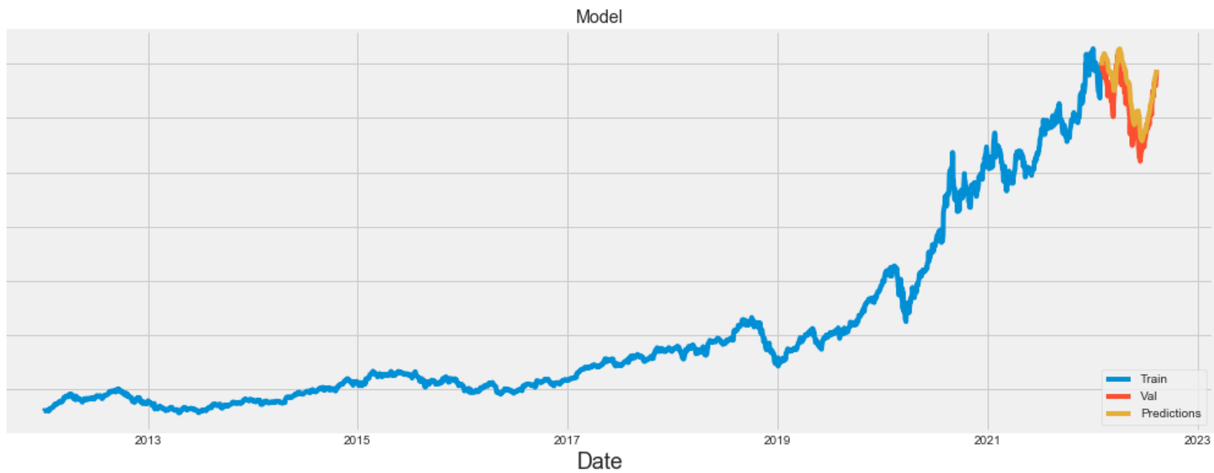
SMA - Adding current prices and dividing that total by the number of time periods used in the computation yields a simple moving average (SMA), which is an arithmetic moving average.

RSI - Technical analysis uses the relative strength index (RSI), a momentum indicator. To assess whether a security's price is overvalued or undervalued, RSI evaluates the speed and amplitude of recent price fluctuations. The relative strength index, which measures momentum, contrasts a security's strength on days when prices rise to its strength on days when prices fall. Trading professionals can predict how an asset will perform by relating the outcome of this comparison to price movement. The RSI can aid traders in making more informed trading decisions when used in conjunction with other technical indicators.

Reinforcement Learning with Technical Indicators and Sentimental Analysis

After evaluating the results of above models, we have concluded to add twitter polarity and volume as additional indicators to predict for better model performance. Hence our final dataset contains, Open, High, Low, Close, Adjusted Close, Volume of the stocks, Twitter Sentiment Polarity, Tweets volume, SMA, RSI and OBV. This data was fed to our final a2c model, giving us the optimal results we desired.

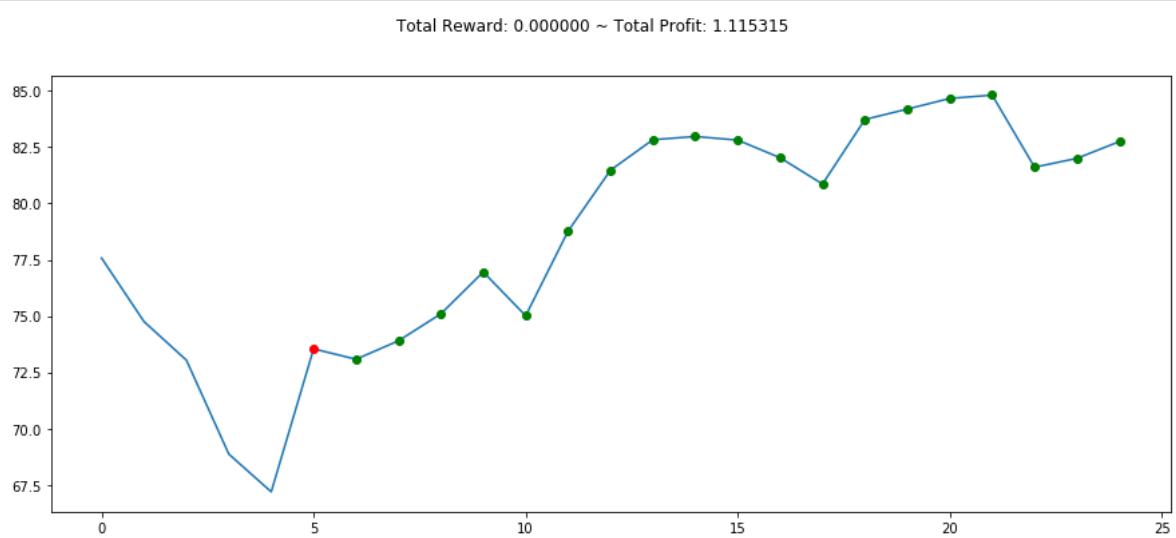
Results



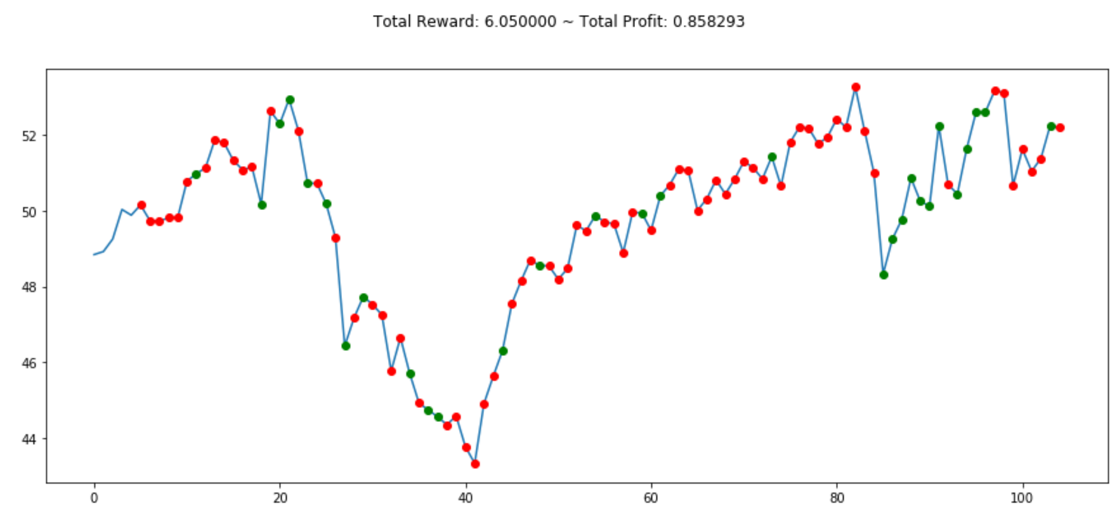

```
# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
rmse
```

```
]: 7.853716347781097
```

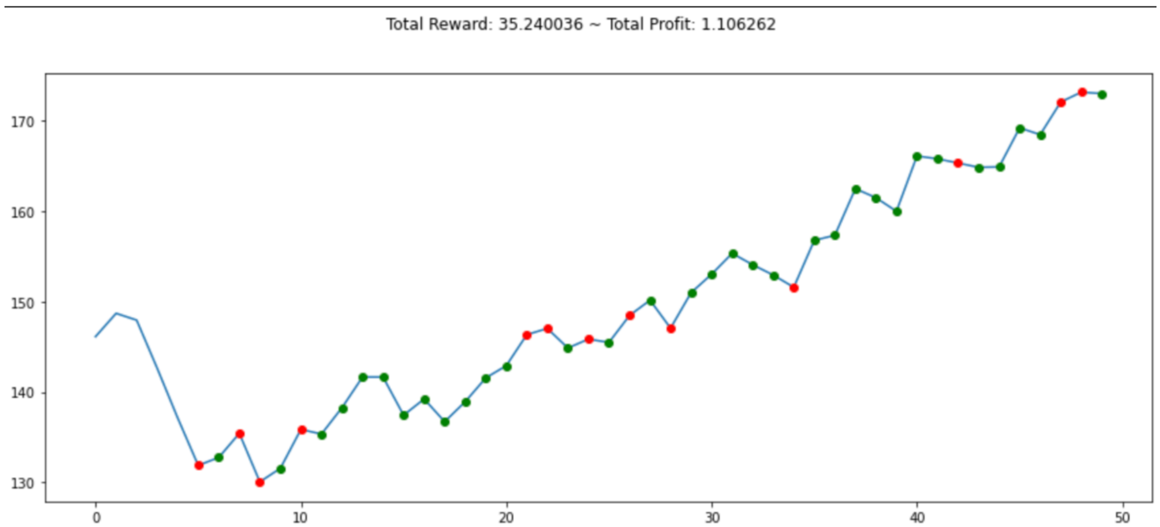
Here we are using the LSTM model to predict future stocks of Apple .Our Model is able to clearly identify the trend of apple stocks but RMSE value of this model is not good as it is quite high so we can not rely on this model to get predicted values of our model and we decided to modify our model by adding advanced financial indicators to our training dataset.



These are the results of our first RL bot trained just on basic financial indicators (open high close volume) . Here bot is clearly not able to analyze the trend of stock to increase the reward it is just buying the stocks but never selling them.



These are the results of our Second RL bot trained on basic financial indicators (open high close volume) plus advanced markers like OBV, SMA, RSI. This bot is clearly doing better then the previous one but here also it is sometimes selling on local minimas and buying on local maximas which is not a good strategy.



These are the results of our Final RL bot trained on basic financial indicators (open high close volume) plus advanced markers like OBV, SMA, RSI and powered with sentiment knowledge from twitter stocks .This bot is clearly able to distinguish all the local minima and selling at most of them and is able to hold the stocks if the price is really volatile and is selling at profitable local maximax.

Overall our final model is able to give 10 percent returns which are really good for industry standards.

Future Works

Future work could focus on increasing the number of tweets per day, expanding the sources from which insights are gathered, such as stock market-specific news websites (CNBC, BusinessStandard, etc.), and processing photos as well, as most tweets and online news are now shared as image snippets. For this situation, multi-agent reinforcement learning techniques can also be investigated.

We also anticipate addressing the numerous exogenous restrictions that institutional and retail traders are subject to, such as transaction costs, trading prohibitions, cash holding restrictions, and a lack of liquidity. Furthermore, because of the frequent use of metaphors, sarcasm, and domain-specific terms in everyday language, particularly in text that expresses an opinion, it can be said that natural language processing on financial data is a non-trivial task. Understanding such language would make it possible to make more accurate market sentiment estimates.

Conclusion

We've designed a simple and effective methodology for calculating market sentiment given a relevant corpus of unstructured textual data. Instead of restricting the representation of market sentiment into a specific number of categories, our approach defines it numerically, i.e., the polarity of the text is expressed as a real number, defined within a range of values. This has a twofold benefit – not only is the sentiment captured in a more nuanced way, but the utilization of this sentiment in the reinforcement learning process also becomes much more feasible.

We have developed a learning framework that enables an adaptable deep reinforcement learning algorithm to take into account the perceived market sentiment. An agent that has been educated on this framework can make trade decisions every day, significantly enhancing the outcomes of portfolio allocation. Financial literature has discussed the value of market mood in terms of forecasting stock movement and its capacity to affect investor choices. The outcomes show that market sentiment is justified in this situation and that it has the potential to be used in a variety of applications.

Acknowledgement

We are overwhelmed with gratitude and humility for everyone who has helped us translate these concepts into something concrete that is well above the level of simple.

We would like to express my sincere gratitude to:

- Our Professor Anasse Bari for helping us come up with this idea
- Special thanks to Chris Davidson for his constant support and prompt replies for our queries.

Also this project has been an enriching experience and helped us with clearing our concepts and helped us make us put our idea into reality.

References

https://towardsdatascience.com/free-stock-data-for-python-using-yahoo-finance-api-9dafd96cad2e
https://www.sciencedirect.com/science/article/pii/S2215098621000070
https://www.section.io/engineering-education/leveraging-openai-gym-and-the-anytrading-environment-for-trading/
https://github.com/openai/gym
https://python-yahoofinance.readthedocs.io/en/latest/api.html
https://docs.tweepy.org/en/stable/