Predicting Stock Price Movements with Machine Learning

Name: Erik Buinevicius Email: embuinevicius@crimson.ua.edu

*Abstract*—

*1-2 sentences: How significant of the research questions you are proposing and any existing challenges related to your work? Pretty much like summarizing sections I-III*

*3-4 sentences: Any novelty in your work, or outline how you implement your work. Like, summarize section IV.*

*5-6 sentences: Summarize the most promising results in Section V and explain the implications or potential applications of the results discussed in Section VI.*

Keywords—component, formatting, style, styling, insert (key words)

# Introduction (*Heading 1*)

Trillions of dollars’ worth of assets are exchanged on financial markets around the world every day. The stock market has become an extremely a hot topic in the last couple of years, following a major interest boost in retail trading powered by so-called “meme stocks,” and more recently a bear market that has wiped out a majority of the gains assumed by investors who have entered the market in the last 5 years. In the last couple of decades, algorithmic trading has become its own force in the market, as firms such as Renaissance Technologies and Two Sigma, each with tens of billions of dollars under management, have funds dedicated purely to algorithmic trading. If we can develop an algorithm that can accurately predict the direction the market is moving, it can become a tool used to expand as well as preserve existing wealth.

# Research Questions

We will explore three distinct questions in this paper:

1. Can we accurately predict whether a stock will move up or down tomorrow?
2. Can we accurately forecast what the price of a stock should be based on historical moves using machine learning?
3. What machine learning models perform the best when working with historical financial data?

The first two questions above relate to the actual predictions we will be performing – that is, can we use a classification model to identify the direction that a stock will move tomorrow: “up” or “down,” based on the data that we have for today. Secondly, we will be performing an actual price prediction based on historical stock price data, and use a machine learning model to forecast this price into the future. The final question above relates to analysis of the work we will do, and finding the best method that we possibly can, which would in theory offer the best chance of outsized returns relative to the market’s historical average performance, while also influencing additional tweaking of our models and building of new models in the future.

# Related Work

Although the quantitative trading firms such as those mentioned in the introduction are careful about retaining the proprietary state of their work, the problem of forecasting financial market movements has been well-explored by researchers around the world, and plenty of amazing existing works on the topic are available.

Shen and Shafiq in [1]explore very similar problems to those that will be addressed in this paper. However, the predictions they performed were surrounding the Chinese stock market, and they had the capability to work with a much larger dataset and more complex models than will be explored here. Specifically, Shen and Shafiq use a dataset consisting of 3558 stocks from the Chinese stock market – a number which is simply not feasible for the computational power available for our research. Instead, we will focus one stock ticker specifically (SPY) – since it tracks the 500 largest companies in the United States (S&P 500) and is generally a good indicator of the overall market. Shen and Shafiq also spent a significant portion of their research on complex recursive feature engineering, which was important as they had to identify the relevant features from within their 3558-stock dataset. While in this paper we will indeed discuss feature engineering and relevancy, we will not be using a complex recursive approach to identify relevant features. The primary focus of our research will be building an LSTM model that can perform on the SPY historical data we have collected, and tuning hyperparameters to optimize performance. Furthermore, while Shen and Shafiq’s LSTM model simply returns a binary output of 0 or 1 representing whether a stock is going down or going up, the LSTM model developed here will output a price prediction, which can then be used to forecast even further into the future.

Bhandari, Rimal, Pokhrel, Rimal, Dahal, and Khatri in [8] also explore a very similar topic to that which is addressed in this paper. These researchers used an LSTM model to attempt to predict the next-day’s closing price of the S&P 500 index. These authors trained their model on 15 years of data, ranging from 2006 to 2020 [8]. These years were selected because they include times of bull markets as well as two major bear markets, being the 2008 financial crisis and the COVID-19 pandemic [8]. In this paper, we will also capture the most recent bear market in 2022 that has occurred primarily due to inflation and interest-rate hikes by the Federal Reserve. Bhandari, Rimal, Pokhrel, Rimal, Dahal, and Khatri also used indicators such as the CBOE Volatility Index (VIX), interest rate data, unemployment rates, consumer sentiment index, U.S. Dollar index, and various moving averages [8]. They trained various LSTM models, algorithmically tweaking hyperparameters based on past performance and training. When looking at this paper, it had already been decided that the VIX and 10-Year Treasury would be used as features. Therefore, it is great validation that this research paper also used these features for their model.

# Methods

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling, and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Data Acuqisition

All of the data used in my research can be easily accessed and gathered from Yahoo Finance’s historical databases [2]. Yahoo finance data can be downloaded directly by simply [visiting the website](https://finance.yahoo.com/), looking up the stock or other security ticker you wish to view data for, and visiting the historical tab. The time period of data that is desired can then be selected, and the dataset can be downloaded in CSV format. The sections of Yahoo Finance’s website described are highlighted in the below screenshot.

Graphical user interface

Description automatically generated

Alternatively, Yahoo Finance’s data can be accessed using a web-scraping technique. For our purposes, we will use the pandas-datareader module in Python, which allows us to automatically collect the data for a given ticker, which is essentially a unique identifier for a security in the U.S., and automatically dump the dataset into a dataframe. Sample code for gathering historical data for a ticker with the maximum historical range is shown below [3].

Graphical user interface, text

Description automatically generated

The pandas-datareader method will be more useful for our purposes than manually downloading the data, as it will allow us to easily experiment with various companies’ data without having to continually visit Yahoo’s website. It will also make it easy to implement company selection on the web-based application that we develop, so that users can test our model’s performance on a variety of company data.

As mentioned previously, we will primarily be using the historical price data of the SPDR S&P 500 ETF Trust (SPY) for our forecasting, due to the fact that it tracks the largest 500 companies in the United States. This means that our model will be a great way to track general stock market movements, and additionally be extendable to individual companies’ securities, the efficacy of which we will explore later in this paper.

In addition to historical SPY price data, we will be using two other significant indicators, being the CBOE Volatility Index (VIX) and the U.S. 10 Year Treasury yield (TNX). Like Apple’s historical data, historical values for these items can be gathered from Yahoo Finance in similar fashion. The VIX is a measure of volatility in the overall stock market and is typically accepted to be inversely correlated with stock market movements [8]. This means that if the VIX moves up over a period of time, it would be expected that the SPY would move down, and vice versa. The 10-Year Treasury will be a good feature to include in our model because it can essentially act as a proxy for interest rates set by the Federal Reserve. With the market’s current inflationary environment, interest rates have been a hot topic, and tracking this will be important for our models to perform.

## Data Cleaning and Labeling

We will primarily be focused on historical price data and alternative index data in our research, and it will be important to ensure that these datasets have no missing values when working with them. An example of what the dataframes for our respective datasets scraped from yahoo finance is shown below.

Text

Description automatically generated

The fields shown here are the same ones present in all three of our datsets, namely the SPY, VIX and 10-Year, meaning renaming of some columns must be performed. We will perform renaming such that company price data retains its original column names, while VIX data and Treasury data columns are renamed to VIX\_Name and TenYr\_Name respectively, where “Name” represents the original column name present in the dataset.

Fortunately, the historical data for SPY did not have any missing values, which is visualized below.

Graphical user interface

Description automatically generated with medium confidence

However, since we are dynamically reading in data with pandas-datareader, it is possible that there will be missing values in the future. Therefore, we should come up with a method for filling missing values in case it is necessary in the future. Since the value of SPY stock price generally increases throughout time (primarily due to inflation as well as other factors) we will use the mean of two days in the future and two days in the past to fill any missing values for price data. This will ensure that any recent data that is missing is not influenced by price data from 10 years ago, for example. It should be noted that the following code to perform this replacement will not work for the first two or last two rows in a dataframe.

Text

Description automatically generated

Since VIX data and treasury yield data fluctuate in more of a static range as shown, we will simply use the average of the entire column of data to fill empty values for these datasets. This can be performed with pandas’ dataframe.isna() combined with dataframe[column\_name].mean().

Ten-Year Treasury all time chart:

A screenshot of a computer

Description automatically generated with medium confidence

VIX all time chart:

Graphical user interface

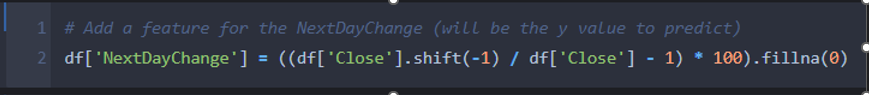
Description automatically generated

The three datasets that have been discussed and will be joined on the ‘Date’ column, as it will be the one identifying feature that can be used to link the datasets. Pandas’ merge( ) function can be used in a straightforward manner to perform an inner join.

## Feature Engineering

We will explore three different models, all of which will be trained on unique datasets. Hence, the feature engineering discussed below may be unique to one or two of our datasets, without being used in the other(s). For example, the LSTM model that we will use to forecast the future price of stocks will only use historical data for the SPY, VIX, and TNX and won’t require the “NextDayChange” or “Direction” features that will be essential to our other two models that explore classification. This is important to keep in mind as our three models and their respective datasets are different from one another, primarily because the goal of this research was to identify the best way to forecast stocks, so that further in-depth analysis and research can be executed from a solid bases of experimental results.

The majority of the data that we will be using is already in a state where it is ready to be normalized and trained on. However, for our classification models we will have to engineer a feature for the output, or y-values, of our training and test data. What this means is based on the data for a given day, we must identify if the stock moves up or down on the next day. To do so, we can engineer a feature titled “NextDayChange”. We can do this by taking the value of the “Close” column, and dividing the value of “Close” from one row in the future by the current value, doing so for each row in the dataframe. Pandas’ shift() method will come in handy here, as we can use it to look one row into the future. Code for creating this new feature is displayed below.



Once we have the value of NextDayChange, we must engineer another feature to simply identify whether this change value indicates an “up move” or a “down move” for tomorrow’s stock price. We can do this by identifying bins that range from [-100, 0] and (0, 100] respectively. We can then cut the NextDayChange column on these bins, creating a new binary feature labeled “Direction” as shown. After creating this feature, any row where the next day’s percent change is greater than zero will have a direction of “Up”, while a percent change less than or equal to zero will be identified as “Down”. In our model which uses Logistic Regression, we will further engineer this feature such that “Up” is represented by 1, and “Down” is represented by 0, which allows for scikit-learn’s LogisticRegression function to handle our data.

A screenshot of a computer

Description automatically generated

The NextDayChange and Direction values are the most fundamental features that we will engineer, and are the only modifications necessary for AutoML’s TPOTClassifier to work with our data, which is one of the models that we will explore. For our Logistic Regression model on the other hand, we will not just be using the VIX and 10-Year Treasury inputs along with our stock price data, and will additionally look at 30 trailing days of price data to see if we can predict the next day’s move as up or down. In order to get this trailing price data, we will again use the shift() method from pandas, but instead use it within a loop, storing 30 trailing days of price data along with a given day’s price in each row. The code for doing so is straightforward and is displayed below.

Graphical user interface, text, website

Description automatically generated

We will also generate a feature titled “Prev Change” in similar fashion to the way we generated “NextDayChange,” except we will shift the data in the other direction (backwards one day). This means that for our Logistic Regression dataset, each row will contain the day’s adjusted closing price, the VIX adjusted closing price, the TNX adjusted closing price, 30 trailing days of price data, and the previous day’s percent change, all of which will be used as inputs for the model.

## Model Development

We will develop three distinct models to predict stock prices, each of which have already been briefly mentioned and will have distinct datasets that are used as input.

The first two models that were developed are classification models that attempt to predict whether a stock will move up or down tomorrow based on the known data for a given day. For the first of these models, we used AutoML’s TPOTClassifier tool that exists as a Python package. The TPOTClassifier package’s fit() method takes in two datasets, one with input data than can exist as a dataframe with a variety of features, and one with output data that exists as a one-row dataframe containing classification outputs corresponding to the input rows (X and y data, respectively) [5]. The TPOTClassifier tool handles selection and optimization of the entire machine learning pipeline, and outputs a model with the optimized parameters based on its trial, modification, and error methodology. As the X-input for this model, we will use our three datasets joined together which contains historical data for AAPL, the VIX, and the 10-Year Treasury, with our y-output consisting of an “Up” or “Down” classifier.

The second model that was developed was a logistic regression model using scikit-learn’s LogisticRegression module. Logistic regression is a “statistical analysis method to predict a binary outcome, such as a yes or no, based on prior observations of a data set” [6]. Logistic regression allows the user to input unrelated features and attempts to use this data along with historical outcomes that it is trained on to predict whether the output falls into one of two categories. In our case, the model will once again be predicting either “Up” or “Down” regarding the next day’s price movement, however this time up will be classified as 1 and down will be classified as 0 in the data due to the limitations of sklearn’s LinearRegression implementation. We can input the dataset that we engineered that consists of a day’s closing price, 30 days of trailing price data, VIX adjusted close, and previous day’s percent change, with the output being either a 0 or 1 value that will be mapped to down and up respectively. This model was the most straightforward to develop out of the three.

The final model that will be develop will be a Long Short-Term Memory, or LSTM, model. This is the model that the most time was spent on, as it is well-known as a classical method of stock price prediction and forecasting. The reason that LSTM is often used for stock price prediction is that it is a variation of a Recurrent Neural Network which is better at retaining long-term information while still processing short-term memory effectively [7]. Classic RNNs suffer from the problems of vanishing gradients which results in them having difficulty retaining long-term patterns, and LSTM networks are designed to address this problem [8]. In our LSTM model we will use a subset of features used by Bhandari et al. in their LSTM model – namely SPY historical price data and VIX historical price data, as well as 10-Year Treasury historical yield data, an alternative to the EFFR interest rate tracker used in [8]. A single-layer LSTM will be used as the primary model, since in Bandari et al.’s research, they found that single-layer LSTMs outperform multi-layer LSTMs on average, when used for stock-price predictions. Various 3-layer LSTMs will also be developed with different number of neurons, dropout rates, and batch sizes.

## Model Training

Please describe how you train the model, especially any optimization strategies you adopt for hyperparameters and model parameters.

## Model Evaluation and Validation

Please describe the strategies you used for evaluation and validation of the trained models.

## Model Deployment

Please describe how you deploy the models to web.

# Results

Please show your results as below.

## Mode Performance in Evaluation

#### Please show the prediction performance of the models during evaluation and validation.

## Feature Importance

Please show how the features contribute to the model performance.

## Runtime Monitoring

Find other 2 classmates and let them use your web-based models. Each of them should create a testing data sample to test your web-based model performance. Please describe your web-based model’s testing performance here.

# Discussion

## Answers to Research Questions

Did your results answer the research questions. If yes, how much is being answered? If no, why?

## Limitations of your research.

Please explain any limitations of your research

## Lessons Learned

Please describe what kind of lessons you learned from the final project.

Overall, this project was an extremely valuable learning experience, and I feel that I gained more from knowledge from it than any other project throughout my college career. Coming into this class, I was extremely intimidated by the concepts of artificial intelligence and machine learning.

##### Acknowledgment

Please acknowledge your classmates and any others helped you during the research process..

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. Shen, J., Shafiq, M.O. “Short-term stock market price trend prediction using a comprehensive deep learning system.” *J Big Data* **7**, 66 (2020). <https://doi.org/10.1186/s40537-020-00333-6>
2. “Yahoo Finance - Stock Market Live, quotes, Business & Finance News,” *Yahoo! Finance*. [Online]. Available: https://finance.yahoo.com/. [Accessed: 05-Dec-2022].
3. “Remote Data access¶,” *pandas*. [Online]. Available: https://pandas-datareader.readthedocs.io/en/latest/remote\_data.html#remote-data-yahoo. [Accessed: 05-Dec-2022].
4. “Get started,” *Streamlit Docs*. [Online]. Available: https://docs.streamlit.io/streamlit-cloud/get-started. [Accessed: 05-Dec-2022].
5. R. S. Olson, “Using TPOT,” *Using TPOT - TPOT*. [Online]. Available: http://epistasislab.github.io/tpot/using/. [Accessed: 05-Dec-2022].
6. G. Lawton, E. Burns, and L. Rosencrance, “What is logistic regression? - definition from Searchbusinessanalytics,” *Business Analytics*, 20-Jan-2022. [Online]. Available: https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression#:~:text=Logistic%20regression%20is%20a%20statistical,or%20more%20existing%20independent%20variables. [Accessed: 05-Dec-2022].
7. M. Phi, “Illustrated guide to LSTM's and GRU's: A step by step explanation,” *Medium*, 28-Jun-2020. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. [Accessed: 05-Dec-2022].
8. Bhandari, H.N., Rimal, B., Pokhrel, N.R., Rimal, R., Dahal, K.R., Khatri, R.K.C. “Predicting stock market index using LSTM.” *Machine Learning With Applications*, 15 September 2022. <https://www.sciencedirect.com/science/article/pii/S2666827022000378>
9. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
10. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
11. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
12. K. Elissa, “Title of paper if known,” unpublished.
13. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
14. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
15. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**