Market Prediction Algorithm

ECS 171: Machine Learning
Group 7 Project Report

Group member:
Parminder Singh, Yuan Chang, Nathaniel Faxon, Cesar Guzman Avina, Kanchan Kaur, Will Colbert, Sergio Santoyo

Github Repository Link

1 Introduction & Background

The stock market has evolved into a major component of our lives, as well as a reflection of the national and global economy. A stock represents ownership, or equity stake, in the company, and has served as a way for companies to raise capital. Additionally, the stock market enables individuals to grow their wealth, while holding companies accountable and keeping an eye on corporate regulation. The stock market impacts the overall economy, and therefore, impacts everyone, including non-investors. Because stock markets span all industries and sectors, it is a key factor in determining the state and cycle of the economy.

When the stock market crashes, there is a sudden decline of stock prices, which results in a loss of wealth, and are further driven by other economic factors and current events. A crash is propelled with investors selling in panic, dropping prices even more. This provides an opportunity for investors, as they can potentially buy stocks back at a much lower price.

Machine learning can be used to identify stock market and financial trends, typically done using the LSTM (Long Short-term Memory) model, support vector machines (SVM), artificial neural networks (ANN), and back propagation neural networks (BPNN). While machine learning can be used to predict stock prices and market crashes, this is a challenging problem. The stock market can be volatile, influenced by many factors, ranging from politics, unexpected events, psychological factors, a company's performance, and so on. This dynamic nature of the stock market makes it difficult to accurately predict crashes and prices.

If a market prediction model is built to successfully predict a stock market crash, it can be beneficial to investors, while also giving a warning to companies and the general public. It can also be used in various fields/sectors. Here are a some examples:

- Communication and Media
  - Machine learning can process content on social media platforms from high-power individuals in the stock market, and predict the market in multiple scenarios.
- Finance
  - The finance sector is one of the most hit sectors during a crash. Predicting a crash could allow financial institutions to prepare for the consequences, like anticipating layoffs.
- Leisure and Hospitality
  - Leisure and hospitality companies face significant economic struggles after a crash, making it difficult for smaller companies to remain in business. Predicting a crash would allow these companies to prepare for such financial hardship, while also taking some action to keep customers, like offering discounted services that can be bought ahead of time, so customers will still participate after the crash.

2 Literature Review

Machine learning techniques have several applications to the problem of market prediction. This section includes examples and methods of related work done by researchers and analysts that have developed tools and techniques that predict stock market prices and assist in proper decision making.

Zhao et al. investigated the problem of financial time series prediction by utilizing Long Short-Term Memory (LSTM) neural network as a good predictor of static and dynamic prediction. Stock data are categorized as time-series multidimensional vectors. In many cases, the time-series problem can be tackled by the LSTM. Kai Chen al. presented that a LSTM based approach used for predicting stock movements would return a 13% better yield in accuracy compared to others.[1] Additionally more similar to the work presented in this paper is the work done by Luca Di al. indicated that using LSTM and dropout over a 5-day prediction interval with only price series data produced a 72% accuracy.[2] To test the effectiveness of a time-weighted LSTM model, Zhao et al. conduct further studies on stock indexes such as the S&P 500 and Dow Jones Industrial Index. It concluded that when altering the model to have the best parameters, the S&P 500 returns 80.24% accuracy while a single index such as Dow Jones Industrial Index returns an 83.21%.[3] The accuracy of the models in these works, though not the highest result, can still be an effective indicator in predicting the market.

3 Dataset Description and Exploratory Data Analysis

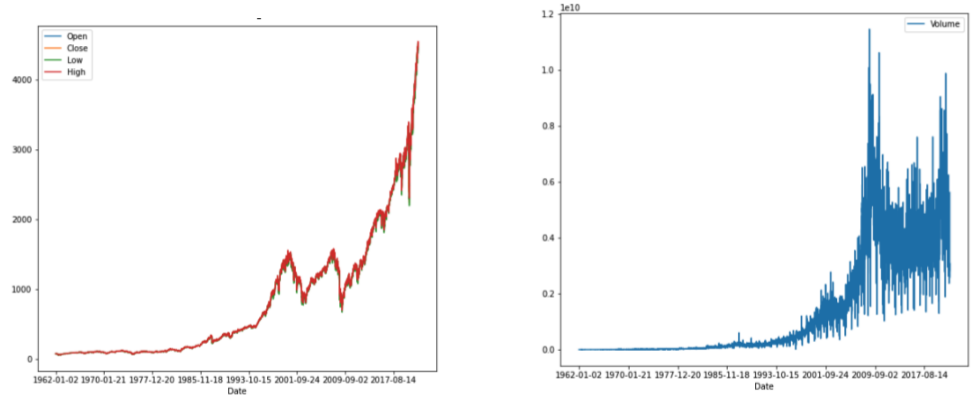
The dataset used throughout this project was supplied from Kaggle. It provided us with 15032 rows which included the columns: closing, opening, low, and high prices as well as the volume, P/E ratio, % gain and loss, and price variation for S&P 500. Data was taken from 1962 to 2021. Below is a description of each column:

The following are descriptions of the features for each observation in the data-set.

- Date - Trading date (YYYY-MM-DD)
- Open (#) - Market opening price
- High (#) - Highest price during the trading day.
- Low (#) - Lowest price during the
- Close (#) - Price when the market closed for the day.
- Adjusted Close (#) - Closing price after corporate actions are accounted for.

- Volume (#) - Number of shares traded during the trading day.

The dataset was selected because of the extensive range of dates it includes as well as the amount of details it has for each date. Overall there is a general trend upwards in the data relating to the Open, Close, High, Low, and Volume. As seen in the plots below:



(a) Plot for open, close, high and low

(b) Plot for volume

Figure 1: Example plots of the stock market's attributes

We thought that the PE ratio would be a stronger indicator of the market's trend, and we used an additional S&P 500 dataset that contains the monthly PE ratio. In order to facilitate the PE ratio data, we extracted the attributes of the stock market for the first day of each month, and added the PE ratio column. This becomes the condensed datasets, which contain 717 data points.

PE ratio was excluded from the input attributes at the later stage, because it didn't produce very accurate prediction models. After excluding it, we are free to use the daily stock dataset, which contains 15032 data points.

In order to process the input data set in the correct format for time series analysis, we chose 14 data points as our time step. Essentially, we are using the first 14 data points to predict the 15th data point, and slide the window along the entire dataframe.

4 Proposed Methodology

We will present here different models that we experimented with. They are polynomial regression, univariate and multivariate time series analysis models.

4.1 Polynomial and Linear Regression

As a basic starting step, we wanted to see if there were any relationships between a S&P 500's stock closing price and some other variable. After doing some research, we found that a good prediction of a stock's price comes from its P/E ratio multiplied by EPS. We made a dataset containing the value at the beginning of each month of the P/E ratio, EPS, P/E ratio multiplied by EPS, and the closing price for S&P 500. To confirm variable relationships, we created a heat map(figure 3) with the variables as well as a pairplot(figure 2):

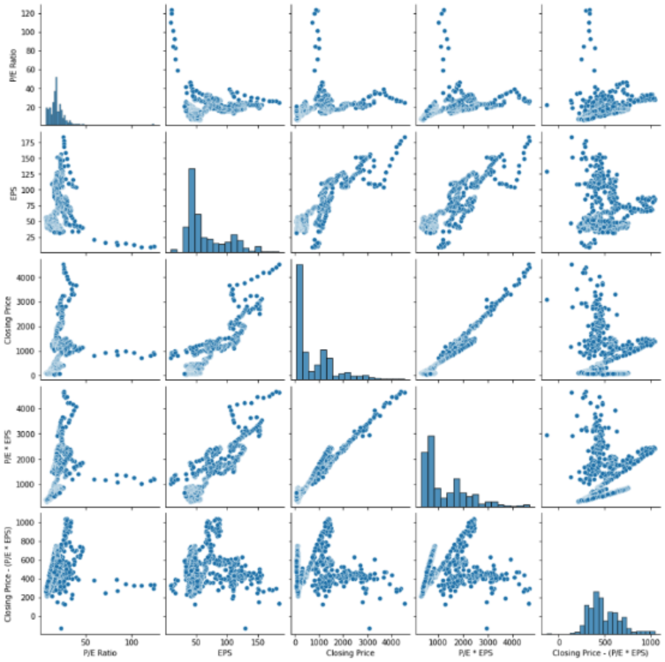


Figure 2: Heat map for the attributes

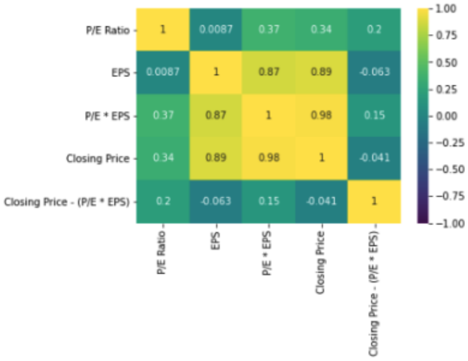
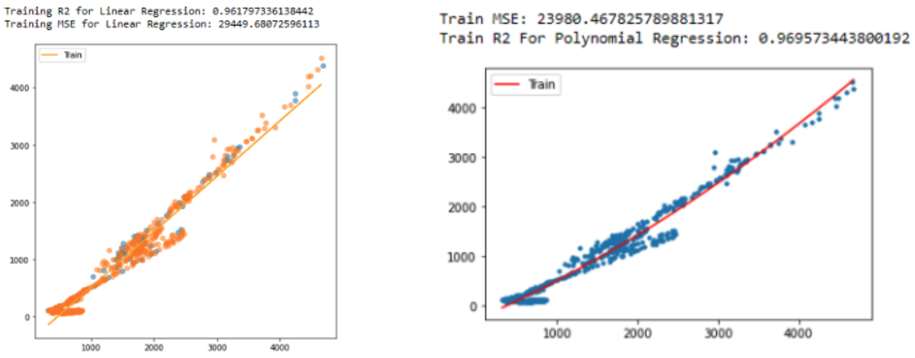


Figure 3: Heat map for the attributes

As seen from the figures 2 and 3, Closing price and P/E * EPS seems to have a linear relationship. To test this, we created a linear regression model first to see the training MSE (Mean Squared Error) and how well a line visually fits the data. We also did a polynomial regression model with a degree of 2 just to see if the data had a polynomial relationship. For both these models, Sklearn's Linear and Polynomial regression libraries were used and training and testing data was split by 80:20.



```
Training R2 for Linear Regression: 0.961797336138442
Training MSE for Linear Regression: 29449.68072596113
```

```
Train MSE: 23980.467825789881317
Train R2 For Polynomial Regression: 0.969573443800192
```

(a) Linear regression model plot

(b) Plot for volume

Figure 4: Plotted models

Based on the above results, there is a very high R2 score for both the linear and polynomial regressions which means that the line fits the data pretty well, thus high correlation between the two variables. There is a very high MSE, but this is due to how large the data values are and how many values there are in general. Though these models have helped show variable relationships and trends, we wanted to create more models that take into account more variables.

4.2 Univariate

In our most current implementation, we applied open, low, high, and volume features in addition to the closing price. In our second version of our implementation it came to notice that our dataset is linked to time. So we approached it now using a Long Short Term Memory neural network. We started with a univariate LSTM model where the model predicted the closing price of the next day using the previous 50 days (time step), but the model failed to predict anything. We then chose to change the time step to 14 days, as this was the most used machine learning approach on the stock market. That result was looking promising, but we then realized that using previous closing prices to predict the next closing price is not a good feature. We decided to proceed in using more than one feature in our dataset to strengthen our model.

4.3 Multivariate

In our most current implementation, we applied open, low, high, and volume features in addition to the closing price.

Since we were going to be predicting data points in the stock market, we knew that this would be a regression model problem. With this, we knew a neural network would need to be used to predict and find relationships between the data. Due to the fact that we are using multiple previous day values to predict the future stock price, we had to find a type of neural network that could utilize multiple sequences of data. After looking around we found that the Long Short Term Memory model was able to use multiple sequences, and then began to work with that type of model. We started with a univariate model of just using the Open price but realized that using more data (Open, Close, High, Low, and Volume) in a multivariate model was much better than the univariate.

While working on this model, we started to develop our website application so that our model could be interactive. We want to allow people to use our model and put in a date and get back an idea of the market price and whether or not there will be a crash. We use our pre-calculated data and upload it to a database to reference for when a date's data is requested.

4.4 Time Series Model Configuration

The exact configurations of the two time series models are shown below:

- Univariate model
  - Using closing price alone
  - 3 hidden layers
    - 2 LSTM layer(one with return sequence, one without), 1 regular dense layer
    - Units = [128,64,32,1]

    - ∗ All using tanh as activation function, as default
  - First 2 layers employed dropout rate of 20%
  - Mini batch size tested: 4, 12, 16, 24, where size 24 achieved best accuracy.

- Multivariate model
  - Using [Open, Low, High, Volume, Closing price] as input attributes
  - 3 hidden layers
    - ∗ 2 LSTM layer(one with return sequence, one without), 1 regular dense layer
    - ∗ Units = [64,64,32,1]
    - ∗ All using tanh as activation function, as default
  - First 2 layers employed dropout rate of 20%
  - Mini batch size tested: 24 and 36, where size 36 achieved best accuracy.

5 Experimental Results

Since the prediction model we are building is essentially a regression model, we are using MSE as the main evaluation metric to compare different performing models.

The initial model created was a LSTM Neural Network. This model had only a fraction of the data available in future models as it incorporated the PE Ratio, a measure only reported at the beginning of every month. As a result, the data fed into the network only included all quantitative data of every first day of the month since 1962. included 3 hidden layers, 128 nodes on the first, 64 nodes on the second, and 32 nodes on the final hidden layer node, all of which were using the tanh activation function.

6 Conclusion and Evaluation

References

[1] K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 2823-2824, doi: 10.1109/BigData.2015.7364089.

[2] Di Persio, Luca, and Oleksandr Honchar. "Recurrent neural networks approach to the financial forecast of Google assets." International journal of Mathematics and Computers in simulation 11 (2017): 7-13.

[3] Z. Zhao, R. Rao, S. Tu and J. Shi, "Time-Weighted LSTM Model with Redefined Labeling for Stock Trend Prediction," 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), 2017, pp. 1210-1217, doi: 10.1109/ICTAI.2017.00184.