**Lebanese American University**

****

*COE599E: Tp. Programming for Data Science*

Final Project Report

*Dr. Dany Ishak*

Presented by:

* Richard Younes – 202004917
* Luidovic Zgheib – 202001451

Contents

[Abstract 4](#_Toc134962158)

[Introduction 5](#_Toc134962159)

[Literature Review 6](#_Toc134962160)

[Data Collection 6](#_Toc134962161)

[Models and Methods 9](#_Toc134962162)

[Linear Model 9](#_Toc134962163)

[Long Short-Term Memory (LSTM) 11](#_Toc134962164)

[Results 17](#_Toc134962165)

[Linear Model Results 17](#_Toc134962166)

[Long Short-Term Memory 19](#_Toc134962167)

[Discussion 20](#_Toc134962168)

[Conclusion and Future Work 22](#_Toc134962169)

[Reference List 23](#_Toc134962170)

Table of Figures

[Figure 1 shows an example of the currency used in the Dataset. 6](#_Toc134962089)

[Figure 2- shows the code used to combine the datasets into one. 7](#_Toc134962090)

[Figure 3- shows the code for cleaning the data. 8](#_Toc134962091)

[Figure 4- shows the libraries imported for Linear Model. 9](#_Toc134962092)

[Figure 5- shows the training and testing of the data. 10](#_Toc134962093)

[Figure 6- shows the Linear Regression model. 10](#_Toc134962094)

[Figure 7- shows the LSTM chain structure. 11](#_Toc134962095)

[Figure 8- shows the formula of the MinMaxScaler. 12](#_Toc134962096)

[Figure 9- Shows the scaling of the data. 13](#_Toc134962097)

[Figure 10- Shows the append of X and y train 13](#_Toc134962098)

[Figure 11- shows the reshaping process. 14](#_Toc134962099)

[Figure 12- shows the model creation and the layers added. 15](#_Toc134962100)

[Figure 13- shows fitting and plotting the loss function 16](#_Toc134962101)

[Figure 14- shows the libraries used to plot the graphs. 17](#_Toc134962102)

[Figure 15- shows the creation of a layout. 17](#_Toc134962103)

[Figure 16- shows the creation of the traces of the graph. 18](#_Toc134962104)

[Figure 17- shows the linear model. 19](#_Toc134962105)

[Figure 18- shows the creation of the 2 traces, layout and plot of the graph 19](#_Toc134962106)

[Figure 19- shows the graph of the LSTM model. 20](#_Toc134962107)

[Figure 20- shows the LSTM plot zoomed in 21](#_Toc134962108)

# Abstract

Stock prediction plays a major role in the market nowadays. It has a significant effect on investors, traders and analysts. Recently, as technology is booming and dominating the world through Machine Learning, stock prediction of elite companies such as Google, Apple and Meta have become a necessity for some investors to maximize their profits in these companies. Hence, to know when to buy and sell stocks.

# Introduction

In recent years, the application of Machine Learning (ML) in real life has gained a significant attention among people, especially in the market among elite companies and the business world. Machine Learning can be described as a subfield of artificial intelligence AI that focuses on building models, training data and constructing algorithms in order to find a pattern in the data. One main field of ML is prediction, specifically stock prediction which is a technical analysis of the pattern of a dataset provided. This paper will tackle the application of ML techniques to predict stocks using Linear Model and Neural Networks with Keras API. It will also show how the data was collected and prepared to be analyzed in order to predict the stock price of an elite company.

# Literature Review

## Data Collection

In order to have accurate values for the data so it can be predicted correctly, it has to be collected from a credible and well-trusted source: in this case finance.yahoo.com was used. A list consisting of a large companies’ stocks prices was used:

* Apple
* AMD Ryzen
* Amazon
* Google
* IBM
* Intel
* Meta
* Microsoft
* Nvidia
* Tesla

In addition to the list above, it was crucial to make sure that the currency used was the same among all the dataset (USD).

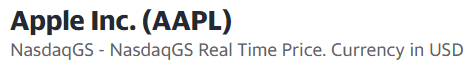


Figure shows an example of the currency used in the Dataset.

Each dataset was composed of 6 main columns labeled as:

1. Date
2. Open which represents the opening price of the stock price during a specific day.
3. High which represents the highest price a particular stock has reached during a day.
4. Low which represents the lowest price a particular stock has reached during a day.
5. Close which represents the closing price a particular stock has reached during a day.
6. Adj Close which represents the adjusted closing price of a stock during a day.
7. Volume which represents the total number of shares of a stock traded during that day.

To be able to predict that data thoroughly, combining all datasets into one vast dataset was necessary.

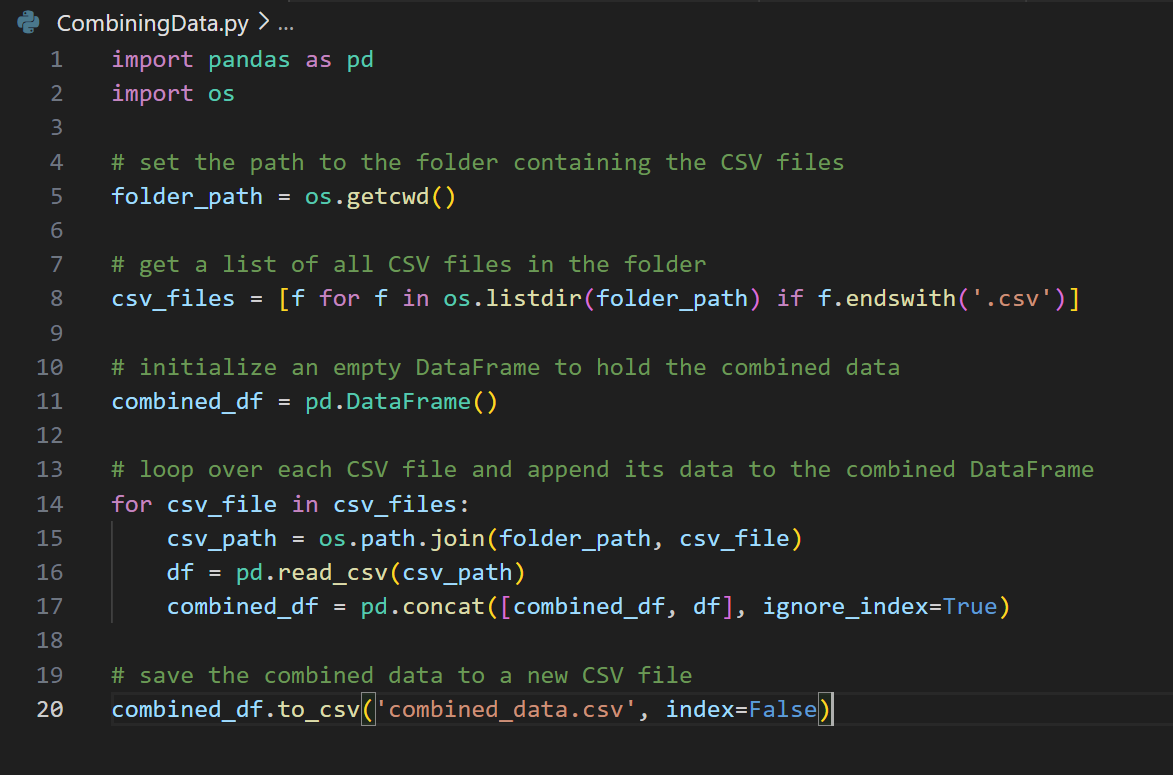


Figure - shows the code used to combine the datasets into one.

Importing OS (operating system) to help interact with the computer. What is common for all datasets is the fact that they are excel files and have an extension of “csv”. Thus, using this information all csv files were put inside a list. Then, using the for loop, each dataset is being read and put inside a “combined\_df” which was created to be empty initially, and every iteration what is inside combined\_df will be concatenated with the dataset in the list. At the end, all the datasets were combined into 1 dataset named “combined\_data.csv”.

Next, to have a clean data to be used for prediction, it is advisable to remove the row if it has a value of NAN (not a number) and most importantly, it should consist of less than or equal to 5% of all the rows. In case of having a NAN value higher than 5% it is recommended that the data must be replaced with the mean value of the column. The code below summarizes this theory.



Figure - shows the code for cleaning the data.

The first step was to print the percentage of missing data inside each column. Then proceeded to loop around the columns of the dataset. If the percentage missing is smaller of equal than 5% the column is dropped. Otherwise, it was filled with the mean value. And then, the result is all saved in a new dataset.

## Models and Methods

A model is referred to as an object that has been trained on data so that it is able to predict based on a pattern found in that data. Multiple ML models are present in Python such as scikit-learn, TensorFlow and PyTorch. To create a model, it is important to define its Architecture at the beginning along with the number of layers, types and parameters if necessary.

### Linear Model

One possible way to predict data is using Linear Model which uses scikit-learn library in Python. Linear Model uses a wide range of ML algorithms such as linear regression to help predict the data trained in a linear way.

In order to ensure a successful model, importing needed libraries are necessary.

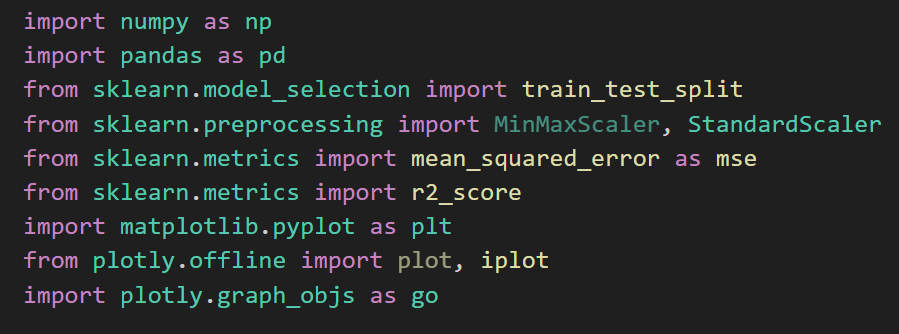


Figure - shows the libraries imported for Linear Model.

After reading the data and cleaning it, the feature data needed to be reshaped to a 2-dimensional array so that it can fit the model properly. The target column or the column that was intended to be predicted what the “Close” since it is a straightforward indicator of the stock’s value at the end of the day.

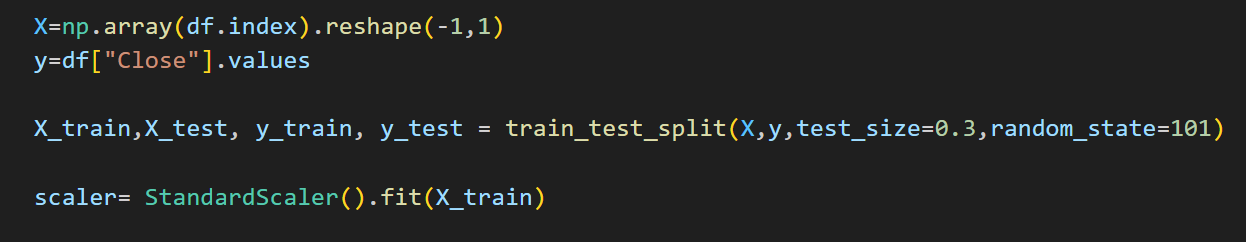


Figure - shows the training and testing of the data.

Next, the data was split among 4 variables (X\_train, X\_test, y\_train, y\_test) which were relative to X and y representing consecutively the features and the predictor. And then StanderScaler is used, which is a preprocessing Machine Learning concept that standardizes the input features (which is X\_train in this case) to have 0 mean and unit variance. It is used to prevent large values from dominating the model.

A picture containing text, font, screenshot, design

Description automatically generated

Figure - shows the Linear Regression model.

Next, Linear Regression model is used to predict a stock price.

### Long Short-Term Memory (LSTM)

Long Short-Term Memory is a model that relies on recurrent Neural Network (RNN) architecture. It is used in deep learning to make predictions on trained datasets. LSTM is designed to address data on long period of time based on timesteps used as milestones throughout the data and the patterns found but not in a linear way. It gives the shape of the graph or almost tries to predict it. LSTM uses various deep learning frameworks such as Keras and TensorFlow.

The LSTM was chosen for being the most suited model for time series, and stock prices are estimated as time series.

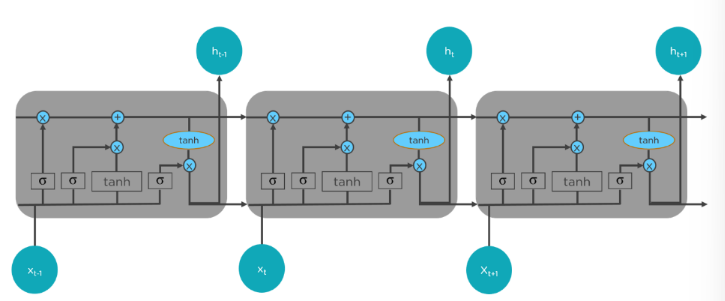


Figure - shows the LSTM chain structure.

As shown in the figure above LSTMs have a chain-like structure and possess four interacting layers communicating. The LSTM process needs 3-steps:

* The first step is deciding which information to be removed from the cell in a particular time step, with the help of the sigmoid function, which looks at previous state and the current input, the computes the function.
* The second layer contains two functions. The first function is the sigmoid, and the second function is the tanh. The sigmoid function is used to decide which data to let through (0 or 1). On the other hand, the tanh function decide the importance of the data that passed by giving them weights between -1 and 1 depending on their importance.
* The third step is deciding the final output. First, the sigmoid layer is used to determine what parts of the cell state makes it to the output. Then the cell state is passed through the tanh function to push the values between -1 and 1, then multiply it by the output of the sigmoid gate.

The first step was getting and cleaning the data, after which the feature selection took place. The feature chosen to train the model on was the ‘Close’ feature in stock prices. After getting the ‘Close’ column, the data was checked that it is all numeric and there were no strings, and if there were the data would be transformed into numeric data types. Then the MinMaxScaler was used to scale the data between 0 and 1. The MinMaxScaler is a of the sklearn preprocessing library which takes the minimum value and the maximum then assign to them the respective values of 0 and 1, and all other values get assigned values using the following formula:



Figure - shows the formula of the MinMaxScaler.

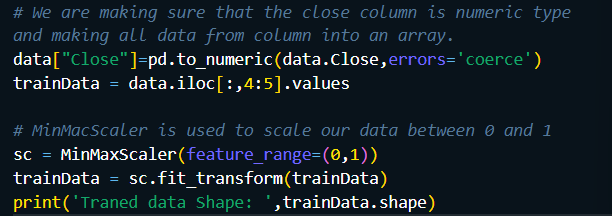


Figure - Shows the scaling of the data.

After scaling the data, two empty lists were created called ‘X\_train’ and ‘y\_train’ and a timestep variable of value 60. The two lists are filled using a for loop and taking into account the timestep. It appends the previous 60 data points to X\_train and the next data point to y\_train. This way, the neural network can learn to predict the next value based on the previous 60 values.

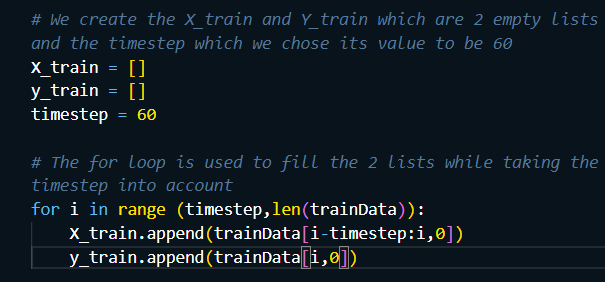


Figure - Shows the append of X and y train

The two lists are then converted to arrays, and the X\_train which is going to be used as input to the model. But the LSTM model takes a 3D array instead of 2D array, so the reshape method was used to make the X\_train into a 3D array.

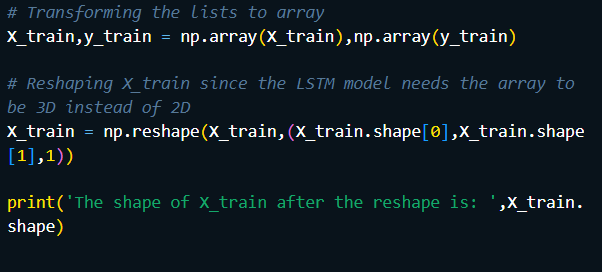


Figure - shows the reshaping process.

The second step is creating the model, when all of the necessary data has been prepared. The model is a sequential model built with Keras, a deep learning toolkit built on top of TensorFlow. Except for the third layer, which contains just 20 units, the model has numerous LSTM layers, each with 100 units. Following each LSTM layer is a dropout layer with a dropout rate of 0.2. Dropout is a regularization strategy that randomly removes a portion of the connections between neurons during training to prevent overfitting. The final LSTM layer has return\_sequences = false, meaning it outputs a single vector rather than a sequence of vectors. There is also a dense layer with one unit in the model, which is the output layer. This layer outputs a single value that represents the predicted value for the next time step.

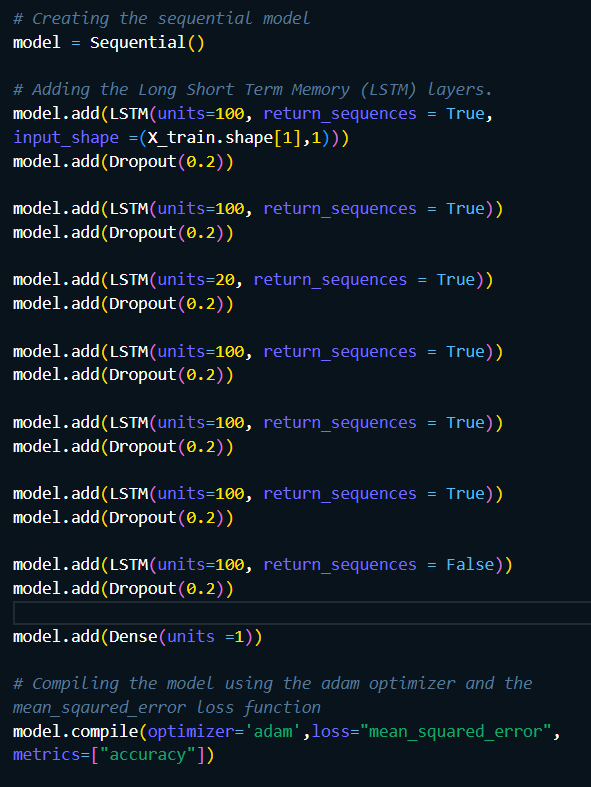


Figure - shows the model creation and the layers added.

The adam optimizer was used with the mean squared error loss function since they are the most compatible with stock prices and continuous data.

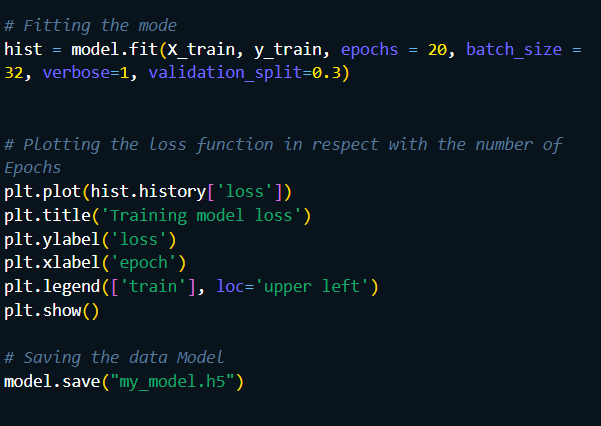


Figure - shows fitting and plotting the loss function

Finally, the mode got fitted using 20 epochs and a batch size of 32, and then plotted the loss function with respect to the number of epochs used.

# Results

The results will be displayed in graphs and explained in detail.

## Linear Model Results

Linear Model results are plotted using 2 main libraries which are:

* Matplotlib
* Plotly.

A picture containing font, text, screenshot, graphics

Description automatically generated

Figure - shows the libraries used to plot the graphs.

To use such libraries, it is advisable to implement a layout for the graph so that it is designed properly, and the plots are easily distinguished.



Figure - shows the creation of a layout.

Moving on to creating traces, which are actually the graphs observed, it is created through specifying the x axis which is in this case X\_train representing the days and the y axis which is in this case y\_train for the actual trained data and y (represents the predicted value) representing the stock price. Next, the traces are appended into a list that will be responsible for drawing the traces.

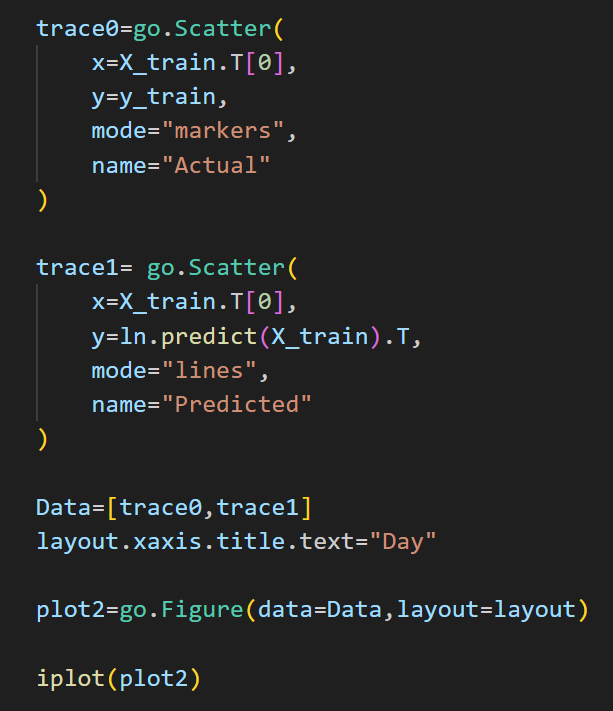


Figure - shows the creation of the traces of the graph.

The resulted graph:

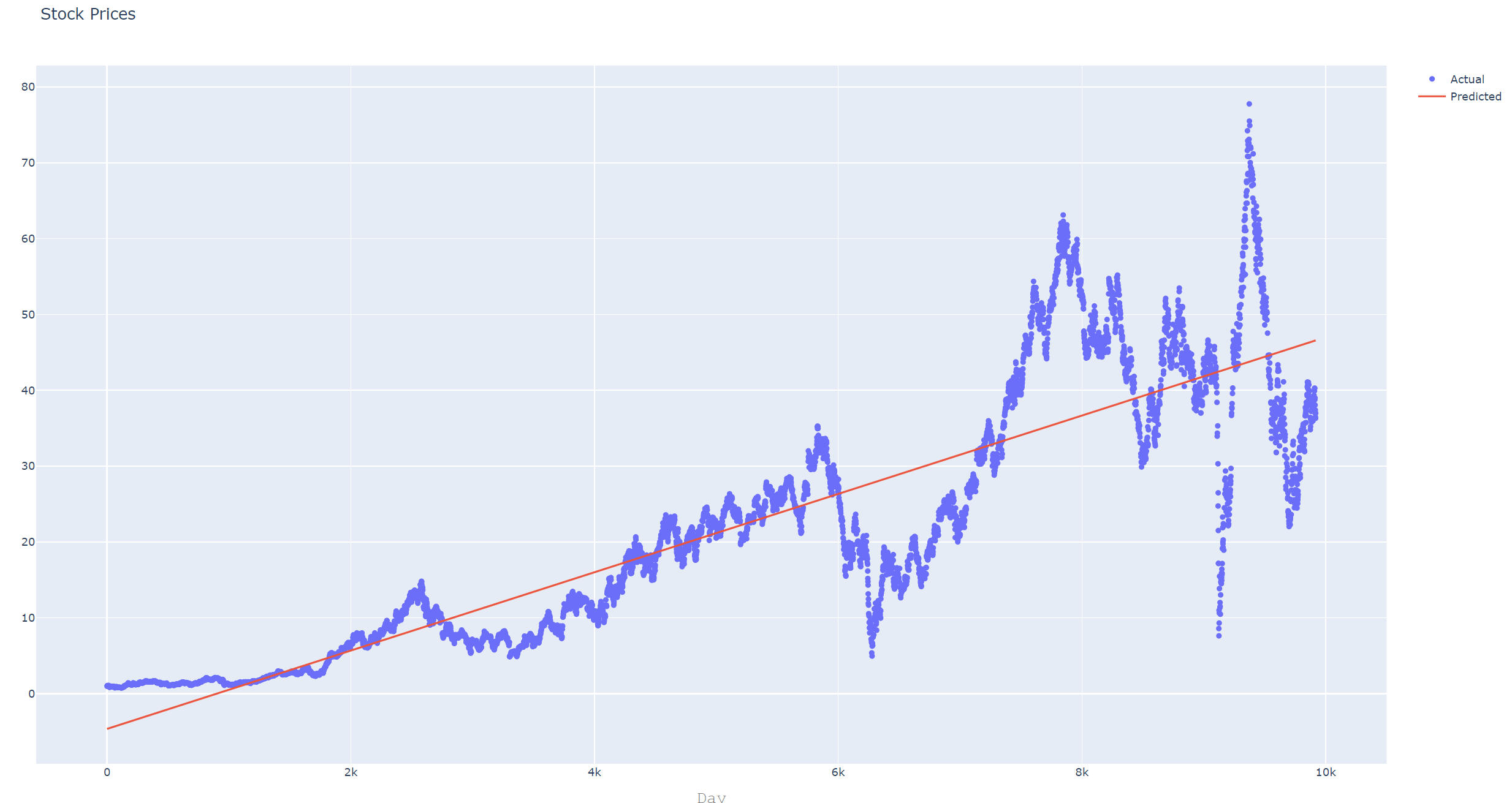


Figure - shows the linear model.

## Long Short-Term Memory

The same way the code works for the LSTM graph.

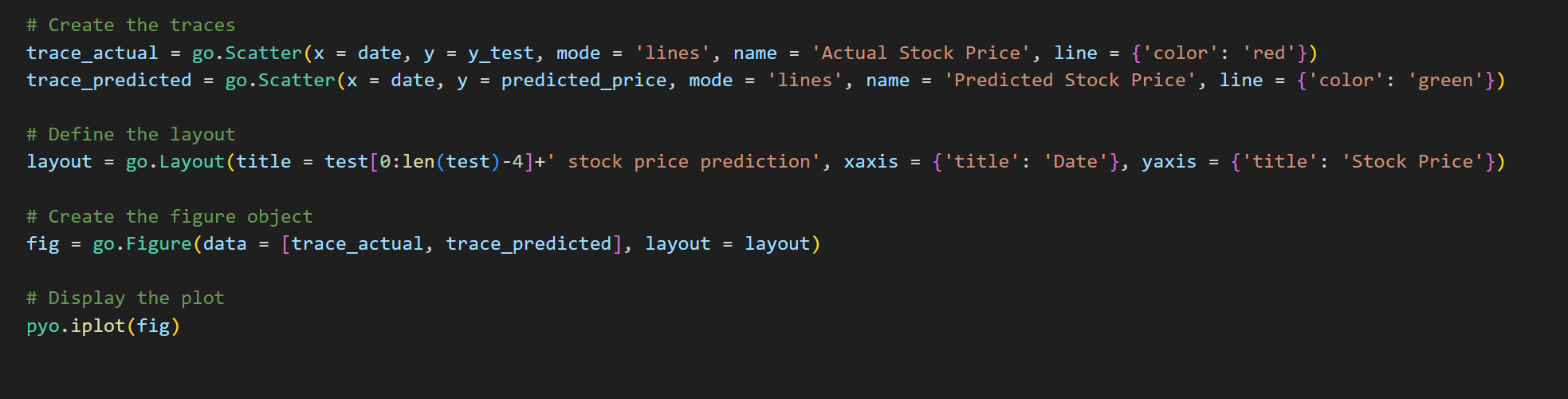


Figure - shows the creation of the 2 traces, layout and plot of the graph

Now, in order to properly compare the 2 models, the use of the same testing data is recommended. Thus, in the linear model, “EAT.csv” was used so it’ll also be tested here as well.

The result is the following graph:



Figure - shows the graph of the LSTM model.

# Discussion

After multiple tries, the LSTM model used have been found to be highly effective in predicting stock prices as it has the ability to find complex pattern through the data given. Additionally, it has shown to be flexible and can adapt to different horizons and input format. The data that was fed to the model was a combination of a list of big companies. The combination of the data of multiple companies together is used because all companies are related in a way or another. For example, the competition between companies such as Apple and Samsung contribute to either elevate or demote the stock prices. Another example is Tesla and Elon musk tweets through twitter as Tesla’s stock prices are being directly affected by Musk’s tweets. For example, in 2018, Musk tweeted about taking Tesla private, leading to a temporary increase in the stock price. However, the tweet was later found to be misleading, and the stock price fell. Going back to LSTM, even though it can predict accurately the stock prices, however, it has one major limitation which is the requirement of a large amount of data to be trained effectively. Also, the account for the number of epochs and batch size should also be considered. Not to forget that a data of almost 70,000 rows takes approximately 2 hours to save the model. So, all in all, it is time consuming. As for the Linear Model, it is simple, does not need a huge dataset along with the fact that it can be done instantly. One major limitation of Linear Model is that it is inaccurate when it comes to the complex world. For example, in the case of the graph plotted in figure 17, the predicted line does not match the actual stock prices at all. Compared to LSTM, after zooming in figure 19, it is obvious that the predicted plot follows the actual stock price.



Figure - shows the LSTM plot zoomed in

# Conclusion and Future Work

In conclusion, the LSTM and Linear model are both powerful tools for predicting stock prices. The LSTM model has shown to be far better and accurate when it comes to predicting complex and big datasets. It has proved to be able find the pattern in the datasets even if it takes too long to create the model. The Linear model is mainly used for simple stock prices not the complex ones and does not require a huge dataset. As the literature review has shown some of the advantages and disadvantages of the LSTM and Linear model, in addition to the fact that the market is a very complex place and especially when it comes to stock prices. As for future work, the use of more layers in the model is advisable. Additionally, adding more epochs so that the error decreases is a necessity. However, all of this cannot be done on simple computers, or it will take forever to run. Thus, the use of Quantum Computers or Super computers is advisable to ensure a high success rate and an accurate prediction for the stock prices.

# Reference List

[1] "Stock Price Prediction." Papers With Code, [Online]. Available: <https://paperswithcode.com/task/stock-price-prediction#:~:text=Stock%20Price%20Prediction%20is%20the,future%20performance%20of%20a%20stock>

[2] "Stock Price Prediction Using Machine Learning." Simplilearn, [Online]. Available: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning>

[3] "Stock Price Prediction Using Machine Learning Project." ProjectPro, [Online]. Available: <https://www.projectpro.io/article/stock-price-prediction-using-machine-learning-project/571>