Stock Market Prediction Using Machine Learning

Project Report for *Indian Institute of Technology, Bombay* - DS203: Programming for Data Science (2022)

Gautam Khona
Dept. of Mechanical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
210101001@iitb.ac.in

Jainam Shah
Dept. of Mechanical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
210100072@iitb.ac.in

Chandmal Kumavat
Dept. of Aerospace Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
190010017@iitb.ac.in

***Abstract*— In Stock Market Prediction, the aim is to predict the future value of the financial stocks of a company. The recent trend in stock market prediction technologies is the use of machine learning which makes predictions based on the values of current stock market indices by training on their previous values. Machine learning itself employs different models to make prediction easier and authentic. The paper focuses on the use of Regression, LSTM and ARIMA based Machine learning to predict stock values. Factors considered are open, close, low, high and volume.**

I. Introduction

Ever since it was automated in the 20th century, the barrier to stock market investing has significantly reduced and the common man with minimal knowledge of finance or economics is able to increase his returns by investing and benefit from the principles of supply and demand by simply performing technical analysis. However, the way the stock market operates is still a mystery, i.e by principles of macroeconomics it is impossible to beat the stock market. We intend to predict future stock prices using different methods of data analysis and interpretation taught to us.

In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python.

The vital part of machine learning is the dataset used. The dataset should be as concrete as possible because a little change in the data can perpetuate massive changes in the outcome [2]. In this project, supervised machine learning is employed on a dataset obtained from Google. This dataset comprises of following five variables: open, close, low, high and volume. Open, close, low and high are different bid prices for the stock at separate times with nearly direct names. The volume is the number of shares that passed from one owner to another during the time period. The model is then tested on the test data.

Basically, quantitative traders with a lot of money from stock markets buy stocks derivatives and equities at a cheap price and later on selling them at high price. The trend in a stock market prediction is not a new thing and yet this issue is kept being discussed by various organizations. There are two types to analyze stocks which investors perform before investing in a stock, first is the fundamental analysis, in this analysis investors look at the intrinsic value of stocks, and performance of the industry, economy, political climate etc. to decide that whether to invest or not. On the other hand, the technical analysis it is an evolution of stocks by the means of studying the statistics generated by market activity, such as past prices and volumes.

A correct prediction of stocks can lead to huge profits for the seller and the broker. Frequently, it is brought out that prediction is chaotic rather than random, which means it can be predicted by carefully analyzing the history of respective stock market. Machine learning is an efficient way to represent such processes. It predicts a market value close to the tangible value, thereby increasing the accuracy. Introduction of machine learning to the area of stock prediction has appealed to many researches because of its efficient and accurate measurements.

The probable stock market prediction target can be the future stock price or the volatility of the prices or market trend. In the prediction there are two types like dummy and a real time prediction which is used in stock market prediction system. In Dummy prediction they have define some set of rules and predict the future price of shares by calculating the average price. In the real time prediction compulsory used internet and saw current price of shares of the company.

Computational advances have led to introduction of machine learning techniques for the predictive systems in financial markets. In this paper we are using Machine Learning techniques in order to predict the stock market and we are using Python language for programming.

II. Prior Work

Big financial institutions like banks and insurance companies already employ a much more complex model of prediction compared to the one we give you in this project. Our model employs most of the things we learnt in class including EDA and implementation of Machine Learning principles.

III. Dataset and Methodology

We have used a variety of datasets from Kaggle such as Google stock data from June 2016 to June 2021 and Tesla stock data from July 2010 to March 2019. We shall explicitly analyze this data for various parameters.

A. Datasets

Tesla and Google: We have taken this data Kaggle. This data shows the Opening stock price on a particular day, the closing price for that day (when the bell rings), the highest price it reached, the lowest it reached on that day and the adj closing price. Adjusted closing price refers to the price of the stock after paying off the dividends. It also shows the volume of stock traded on that day. Volume refers to the total number of shares bought and sold on that day.

We have also separated Google data into testing and training data. Training data trains our ML model into understanding the trends in prices in past history for each parameter (Opening, Closing, High, Low, Volume, etc). This gives the model some training to be able to make predictions using this past data.

Data Pre-processing

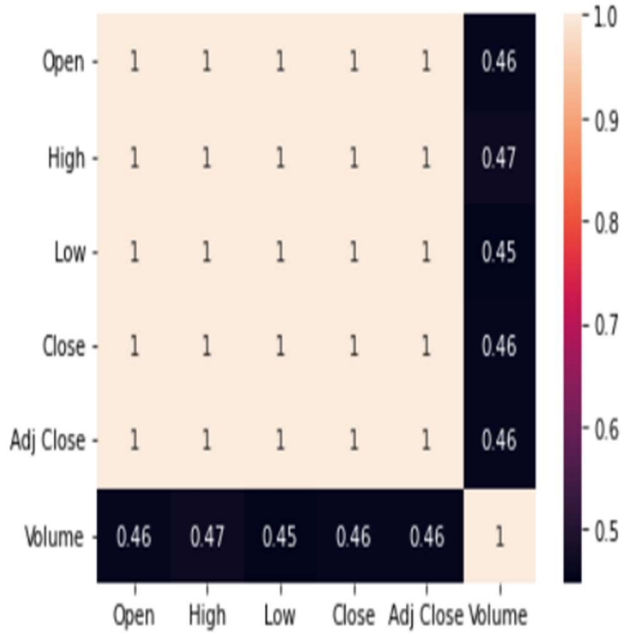
First, we started by getting initial superficial data for both the data sets. This means something like the shape, the basic information, number of null values, mean, median, quartiles, etc were found using Exploratory Data Analysis.

We also found statistical measures such as the correlation of the data for all parameters and plotted it on a heat map using matplotlib libraries. We also changed the index from normal integers to the date itself to have a time plot.

IV. Data Visualisation

A. Variable Correlation

We are faced with a large amount of data and thus, it is essential that we ask ourselves how we can make sense of this data. We found the correlation using library functions and hence found a few deductions.



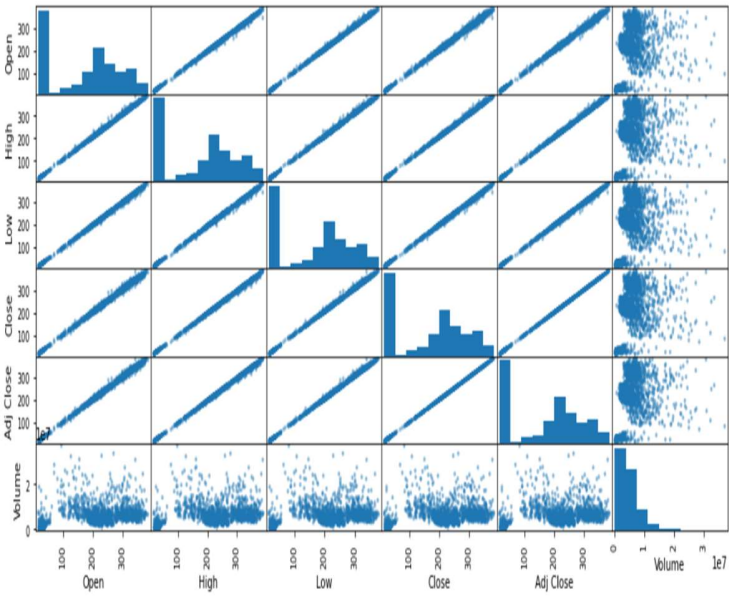
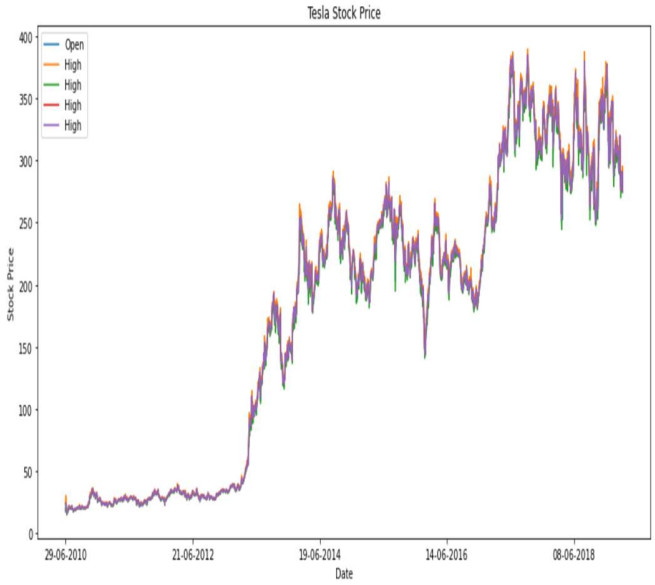
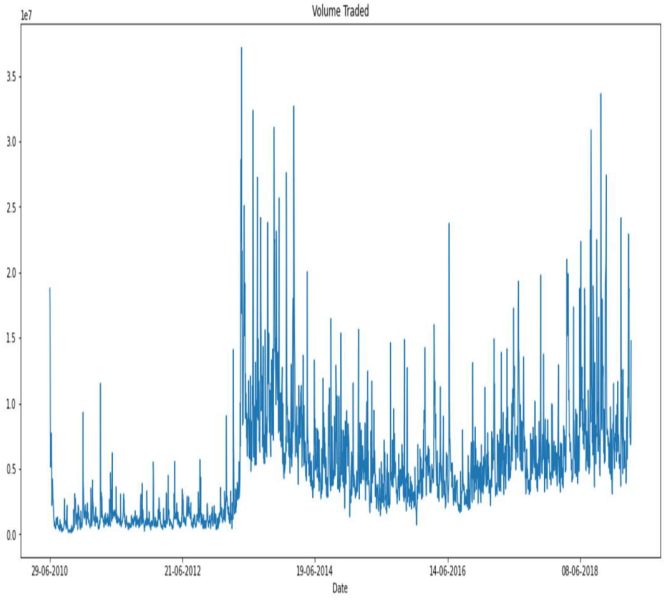


Fig. 1. Correlation between all metrics

Our inference from the correlations show that there is Linear variation among variables, but variable name 'Volume' is non uniformly scattered w. r. t. other variables.

Then we plotted the Tesla stock prices and subsequently volume traded.





An analysis of volume shows that good number of stocks were traded between the year 2013 and 2014 and then the pattern remains same but hikes after year 2018.

Then we did a bar plot of variables wrt date.

Linear Regression

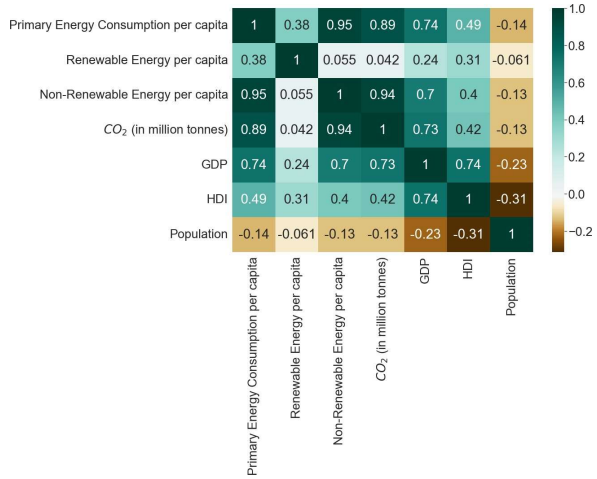
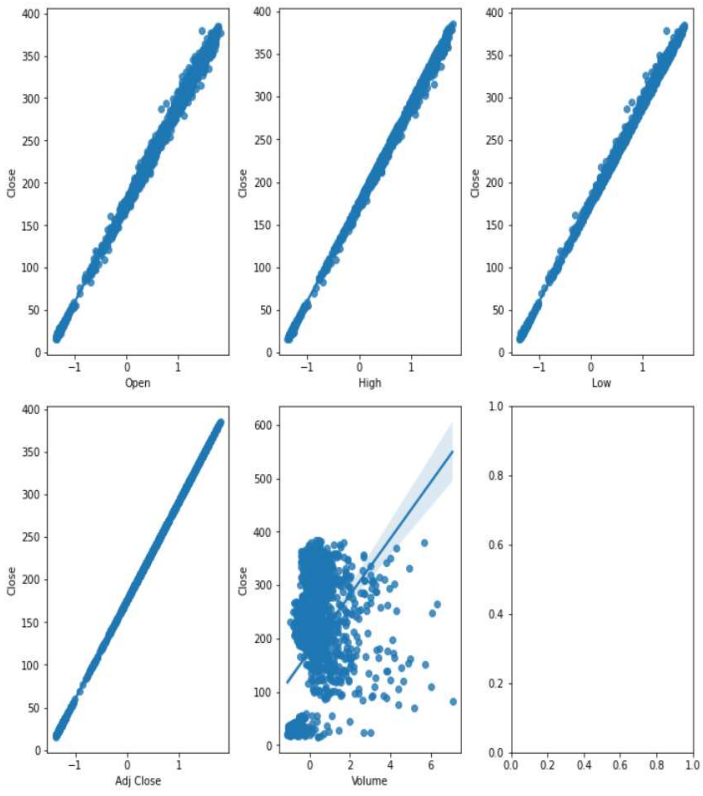


Fig. 4. Correlation Heatmap

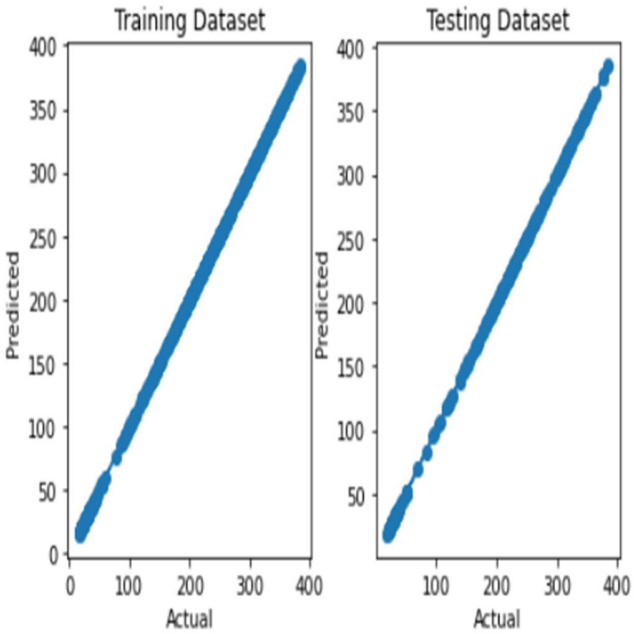
To write our feature variables we eliminated the variable ‘close’. Then we made the target variable to be ‘close’ and scaled our data using sklearn and other such library functions.

Hence we plotted our feature variables against ‘Closing stock price to find the relationship between the two.
Obviously, all variables except volume had linear dependence.

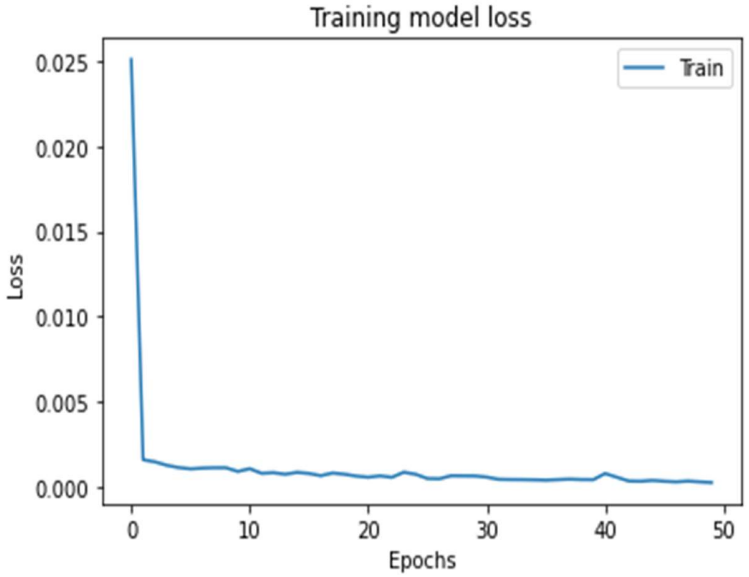


Next we split data into training and testing data and train and interpret the model using Linear Regression. Then we model evaluation for training and testing data.

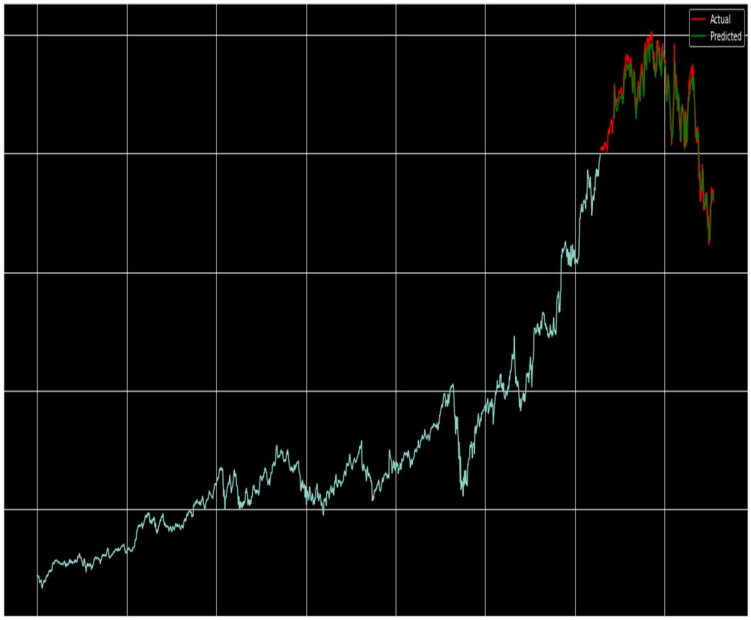
Finally we plot the actual and predicted values of data set.



Hence both of these match and our model works seamlessly! However, this is only for linear dependence data i.e data where correlations are linearly dependent on each other.
The effiency is determined by R2 score which is 1.0 i.e absolutely perfect.



Then we load our test data i.e our ‘close’ test data. Then we scale the test data. Then we reshape the data and predict the values. We also use inverse transform on predicted values. Then we plot actual and predicted values.



This means there is a near match between predicted and actual values.

LSTM Model

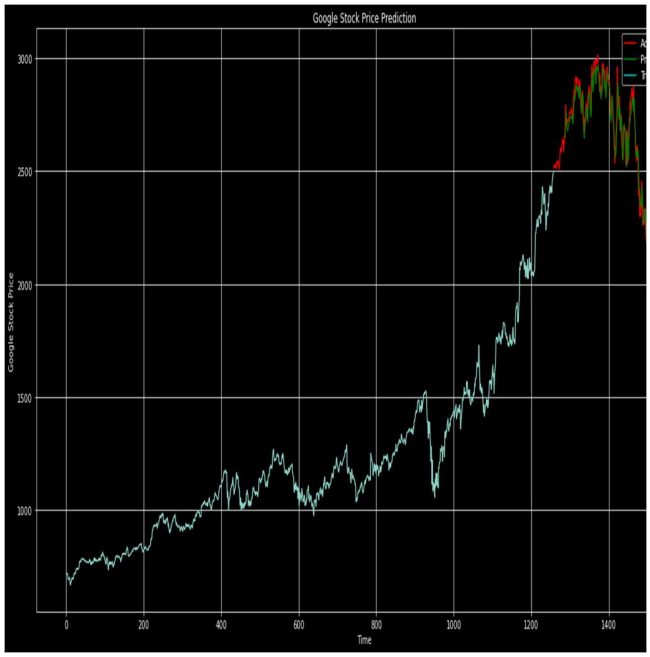
For this model, we used two different datasets both containing Google stock price data. After doing the same preliminaries to the data for basic info we did our analysis.

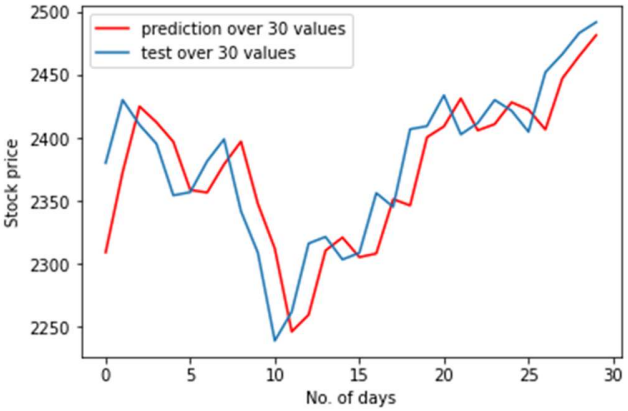
We scaled all values of Close using Minmax scaler. Creating x train and y train converted into a numpy array makes analysis easier. Reshaping the x train makes everything much easier that is we make it a 3 dimensional array not a 2 dimensional one.

Next we create and compile the LSTM model and fit data into the model (training dataset). The training loss is a metric used to assess how a deep learning model fits the training data**.** That is to say, it assesses the error of the model on the training set. Computationally, the training loss is calculated by taking the sum of errors for each example in the training set.

Finding R2 value gives us a **93.1% ACCURACY** which is insanely awesome.

Then we took a step further to predict stock prices that have not materialized, that is future stock prices, future in time. This means are model could with a high degree of precision calculate future stock values which could heavily benefit banks and the common man.





As you can see, our results do not match properly with actual results. This means our model ain't accurate enough. Observation is right, our R2 score turns out to be 67%. This proves that we need a better model that is more accurate. Hence, now we make 'Close' column as our training data. Again importing adfuller function from statsmodel gives us a p-value of 0.85 which is still >>> 0.05. So time series is still not stationary. Now we use auto_arima to get best model. Using training data for training the model over 1258 data and for testing using test data as 253. Hence, we created our ARIMA model and same warning as before applies. Now if we plot actual v/s predicted values, we get the following graph.

ARIMA Model :

The final model that we present to you is this using Google stock data again. After going through some preliminaries for showing superficial data properties such as number of null values, basic info, etc. we constructed the actual analysis of our data.

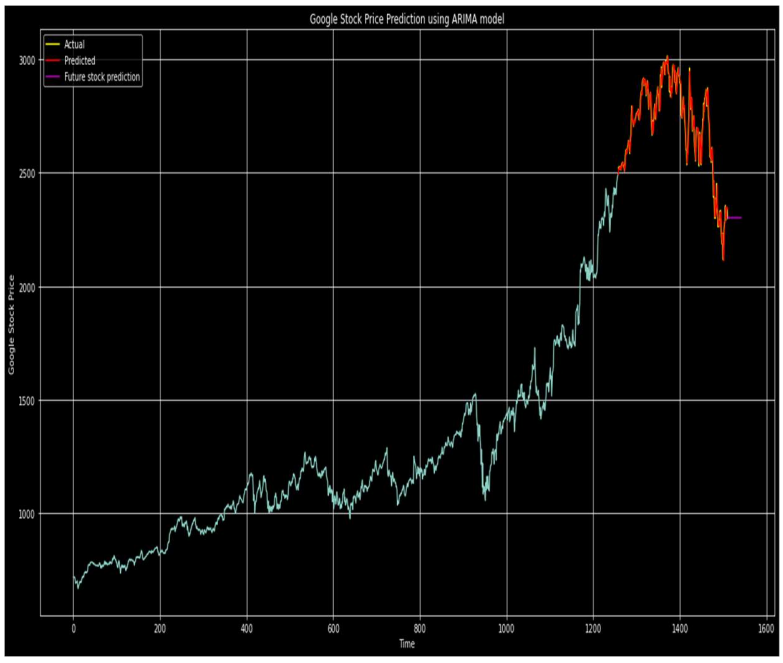
We use 'Close' as our target variable once again. We imported the adfuller function the statsmodel after that, and found the p-value from it. It was found to be 0.998 >> 0.05. So the time series is not stationary.

Then we imported auto_arima in order to find the most suitable model for our analysis. Also, our training data will consist of the entire stock data except last 30 entries which will serve as our test data to compare it to something.

Then we created the ARIMA model using our knowledge. However, a simple warning before proceeding, Covariance matrix calculated using the outer product of gradients. Then we fit the model and predict the results. And we plotted actual and predicted results on a graph.

Calculating mean square error and R2 value as before gives us an accuracy of R2 value of 94.66%. This is pretty good and even better than LSTM Model.

It shows ARIMA is best model.



V. Learning, Conclusions, and Future Work

A. Learning

This project exposed us to several aspects of finance and stock market concepts and techniques. We learnt about state of the art machine learning models and recognised the complexity of problems involving the stock market. We also learnt that the Machine Learning World offers tremendous diversity in choice for choosing the right model.

We significantly improved our grasp over the data analysis and visualisation libraries used with python, namely, matplotlib, seaborn and pandas. We also learnt the use of graphs to visualise data on the world map. We learnt the shortcomings of machine learning techniques in predicting complex phenomenon pertaining to supply and demand and recognised that certain events are inherently unpredictable even with a lot of information.

We learnt how to collaborate over code we people we don't know well enough. We also got a strong grasp of git and Github to manage code effectively. Working in a team taught us how to make the most of everyone's skills.

B. Conclusions

There is a strong correlation between closing stock price and several other metrics like volume, opening price, etc.

Two techniques have been utilized in this paper: Regression, LSTM and ARIMA, on the Google stock dataset. All the techniques have shown an improvement in the accuracy of predictions, thereby yielding positive results. Use of recently introduced machine learning techniques in the prediction of stocks have yielded promising results and thereby marked the use of them in profitable exchange schemes. It has led to the conclusion that it is possible to predict stock market with more accuracy and efficiency using machine learning techniques.

In the future, the stock market prediction system can be further improved by utilizing a much bigger dataset than the one being utilized currently. This would help to increase the accuracy of our prediction models. Furthermore, other models of Machine Learning could also be studied to check for the accuracy rate resulted by them.

C. Future Work

Hope to be able to convert real time news into predicting how much stock price will rise in real time. This could be the next break through in our study of ML. Big banks already do this and call it high frequency trading. They trade in huge volumes and analyze news faster automatically compared to traders or common men. So they buy and sell shares marginally faster than the ordinary individual and those small marginal profits multiply with large volumes to become a decent profit. Hence, our models could ultimately model the stock market itself days before those prices actually materialize and big banks will be able to literally predict the future and control people's money.

References

[1] Kaggle: for stock data