End Semester: Essay on Stock Price Prediction using Extreme Gradient Boosting.

D Shanmukha Sai Krishna
Mechanical Engineering
IIT Madras
me18b135@smail.iitm.ac.in

***Abstract*—This document explains visualization of stock prices in brief and application of regression models in prediction of the stock prices in further days.**

I. Introduction

A. Broad Overview

Extreme Gradient Boost Regressor is one of the tools that helps us to perform regression analysis based on features. The Extreme Gradient Boost Regressor are not affected by the relation between the features as it has a base of tree. In the following paper we are going to learn about Extreme Gradient Boost Regressor in detail including the math behind it. We are given data set of training data of features containing the features such as opening price of stock, closing price of stock, highest and lowest price of stock in day as well as the volume of stock traded in that day.Visualization of data is done so as to get an idea of how the features are varying through the days perform time series analysis. We will be developing a **Extreme Gradient Regression** model to train on the given data set to learn different parameters and develop the statistical model to predict the stock prices.

B. Brief on Extreme Gradient Boost

Extreme Gradient Boosting in decision trees are a non-parametric supervised machine learning method that is widely used for classification and regression. In Decision trees we will be creating a model that predicts the target variable using the rules that are learnt from the feature variables. Extreme Gradient Boosting Regressors will build the trees on the basis of the amount of error they made in the previous tree and build new tree subsequently and the prediction is made using all these trees commonly known as ensemble method. Extreme Gradient Boosting can be used in:

- Credit Card Fraud Detection
- Loan Default Prediction
- Stock Market Analysis

The decision trees models the probable outcomes using a sequence of control statements that can be used to predict the outcomes and random forests form a set of decision trees and the output is calculated using the aggregate of all the trees formed. These can be used when there are multiple outputs in target variable which would be easily handled due to control statements.

C. Aim

Aim of this project is to explore the visual analysis of the all of the stock prices with the opening,closing prices, high and low prices in the day, volume of the stock traded in a day and the moving average of the stock for ten, twenty and fifty days. Using the above data we will take training data till a specific date and testing data from that date. We will develop a extreme gradient boosting regressor to train on the training data and predict the prices of the testing set.

II. Extreme Gradient Boosting Regressor

A. About Extreme Gradient Boosting

Extreme Gradient Boosting is a highly scalable additive based ensemble algorithm. This runs ten times faster compared to the traditional gradient boosting. This model combines a set of weak learners in addition to achieve a complete model. The quickness of this algorithm lies in the fact that this algorithm uses parallel processing of the data points and building trees in parallel which leads to the faster building times. This algorithm is also used in prediction of the missing values due to its robustness. These follow a similar approach to those of the random forests but build the trees sequentially and data is processed parallel and the error is estimated much quicker and the tree is penalized on where the error is made so as to learn the point properly by creating good decision boundaries. The XGBoost also penalizes the L1 and L2 norm so that the model do not overfit on the dataset. XGBoost on the other hand make splits upto the max depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain which follows a approximate greedy algorithm.

B. Math behind Extreme Gradient Boosting Regressor

The basic idea in forming a new tree is that we will be splitting the feature at node that maximizes the information gain.

Construction of Extreme Gradient Boosting Regressor:

- The main idea is to build the a model that is continuously appended by the weak trees multiplied by a constant known as a learning rate. These trees that are built have to be continuously approach or get better residuals than formed by the previous trees.
- A decision tree is built initially and the predictions on the data is performed and the residuals are taken together.

- We will be building a new tree on the residuals obtained from the previous tree and then the new tree is built.
- The above two steps are repeated until we have achieved the desired accuracy or the residuals reach below a certain level of uncertainty.
- During forming the tree, the tree is completely built till the maximum depth that is specified and the pruning is then done on the tree to cut the tree and get only positive gains from tree.

The Mathematical Intuition is shown below:

$$y_i = \sum_{i=1}^{K} f_k(x_i); f_k \in F$$

which f_k is k^{th} tree in the subspace F of all the trees and y_i is prediction from all the trees for the data point x_i. The objective function is defined as a combination of normal loss function along with a regularization term to prevent overfit on the data.

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f)$$

As the models are developed using the additive model, the $\hat{y}_i$ can be written in terms of $\hat{y}_{i-1}$ using the below:

$$\hat{y}_i^t = \hat{y}_{i-1}^t + f_t(x_i)$$

Converting the above three equations, we get:

$$obj(\theta)^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^t) + \sum_{i=1}^{t} \Omega(f) + Constant$$

The $\Omega(f)$ can be written as : $\gamma T + 0.5 * \lambda \|w\|^2$ where T is number of leaves in tree and w is the weight of leaves in tree and can be written as follows : Upon differentiating the above

$$\text{obj}^{(t)} = \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + \text{constant}$$

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$
$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$

equation with x to get the weight of each leaf and the value of objective.

III. Data Insights and Visualization

A. Features of Data set

There were six different data sets that makeup the stock prices of companies of Cognizant,HCL, HDFC Bank, ICICI Bank, Infosys, SBI. The different features that are given in the data sets to train model are:

- Date - Date on which the stock price is measured
- Open - Opening price of stock on that date
- High - Highest price of stock on that date
- Low - Lowest price of stock on that date
- Close - Closing price of stock on that date
- Volume - The amount of stocks traded in that day

We can see that all the data that is present in the continuous data and are positive floating values and a column of dates.

B. Visualizing in the Data set

We will be analyzing and plotting the amount of high, low, open price, close price and the volume of the stock traded for all the time in the data for each company individually.

1) Cognizant: We can see that the Opening price have risen initially but have taken two sharp dips in the opening prices during May 2019 and March 2020. The opening price have been gradually increasing till May 2021 can be seen in Fig 1.

We can see that the Highest price in day have risen initially

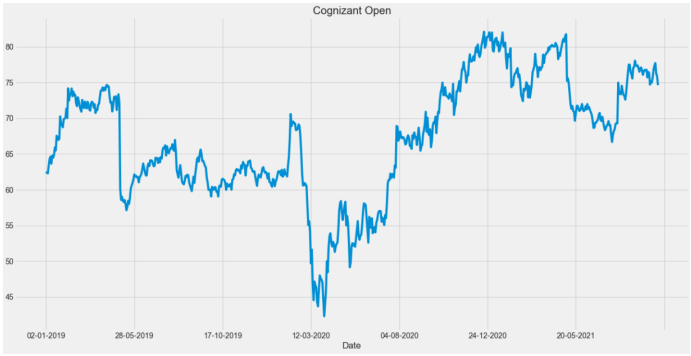

Fig. 1. Cognizant Opening Price over dates

but have taken two sharp dips in the highest prices during May 2019 and March 2020. The highest price have been gradually increasing till May 2021 can be seen in Fig 2. This is a similar trend observed in the opening price.

We can see that the Lowest price in day have risen initially

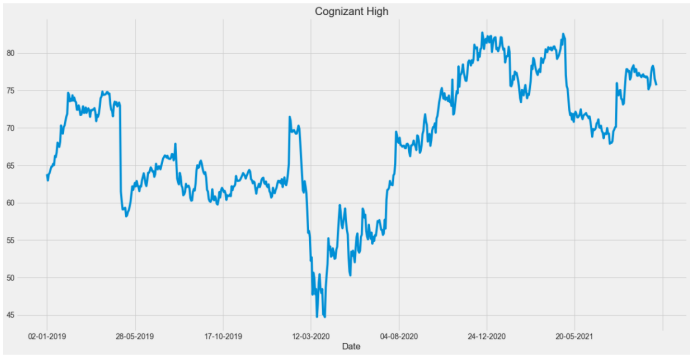

Fig. 2. Cognizant High Price over dates

but have taken two sharp dips in the lowest prices during May 2019 and March 2020. The highest price have been gradually increasing till May 2021 can be seen in Fig 3. This is a similar

trend observed in the opening price.

We can see that the Volume of stocks brought in day have

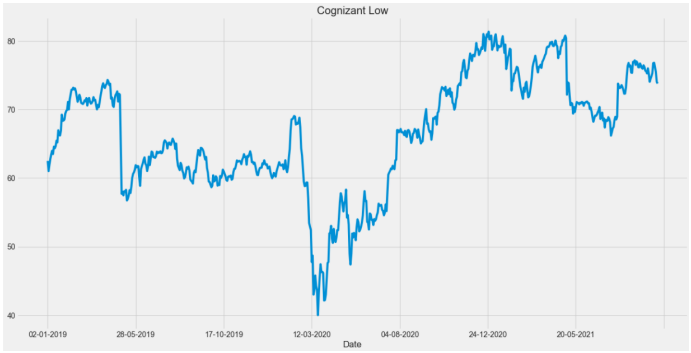

Fig. 3. Cognizant Low Price over dates

been constant initially but have seen a large increase during May 2019. The stock volume that has been brought have been constant afterwards can be seen in Fig 4.

We can see that the Lowest price in day have risen initially

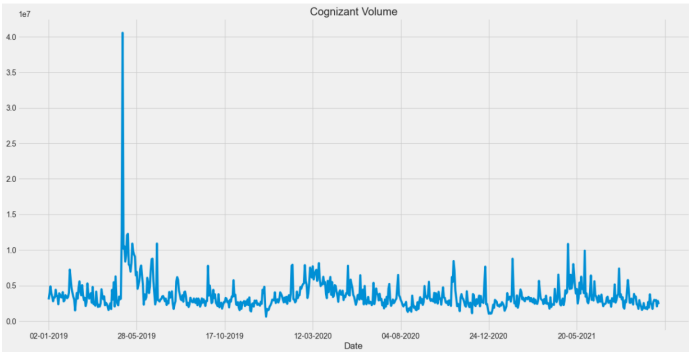

Fig. 4. Cognizant Volume over dates

but have taken two sharp dips in the lowest prices during May 2019 and March 2020. The highest price have been gradually increasing till May 2021 can be seen in Fig 5. This is a similar trend observed in the opening price.

The moving average of the price considering 10 days follows

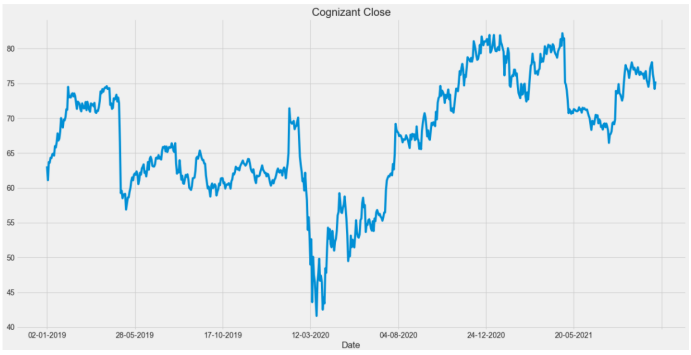

Fig. 5. Cognizant Closing Price over dates

the general trend line, for 20 days follows general trend line and for 50 days it also stays as a constant with a slight decrease and increase in the days as seen in Fig 6.

From the heatmap in Fig 7 we can see that all the

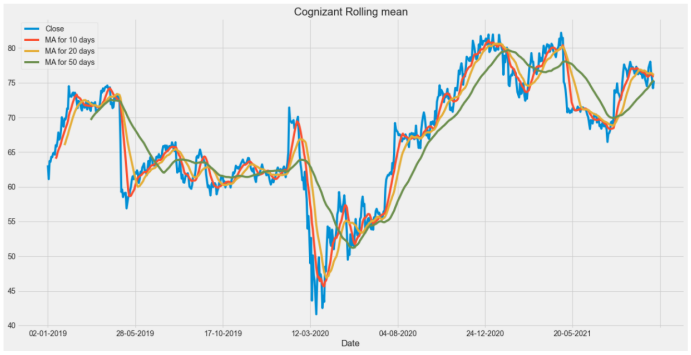

Fig. 6. Cognizant Moving Average Price over dates

high,low,open,close values are perfectly correlated which means that based on one plot we can say the movement of all the other features.

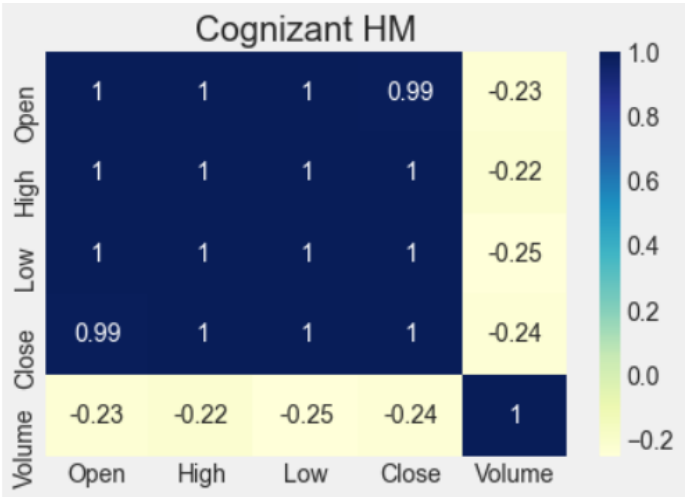

Fig. 7. Cognizant data Correlation

2) HDFC Bank: Initially the heatmap of the data was plotted in Fig 8 and it had been observed similar to the case of the Cognizant prices wherein the high,low,open and close values are perfectly correlated and hence one of the feature movement can provide insights on the remaining data.

The highest price of the stock has been constant and have taken a sharp dip in the March 2020 but the volume of stock have been increased in the month. Then in a due course of time the price have gradually increased and was constant which can be observed in Fig 9.

The volume of HDFC stocks that are brought have spiked in the months before May 2019 and also spiked during March 2020 where many companies started to lose volume. After that the volume has gradually been decreasing and can be observed

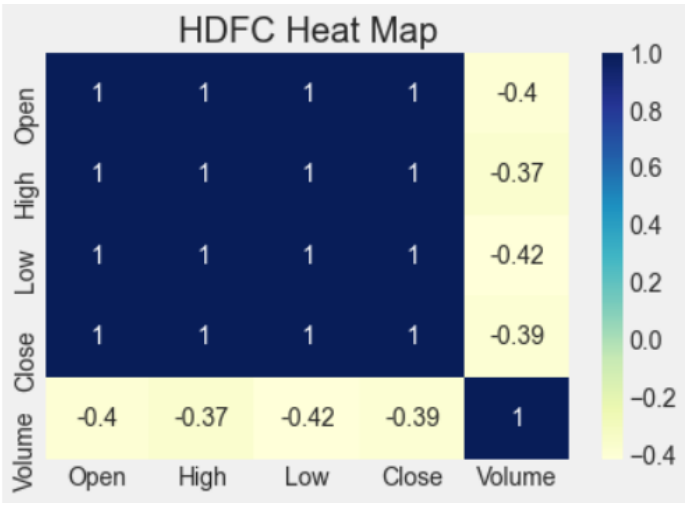

Fig. 8. HDFC data Correlation

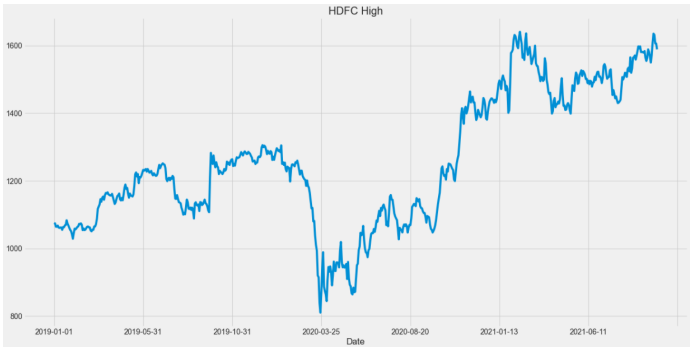

Fig. 9. HDFC High Price over dates

in Fig 10.

The closing price being correlated to highest price of the

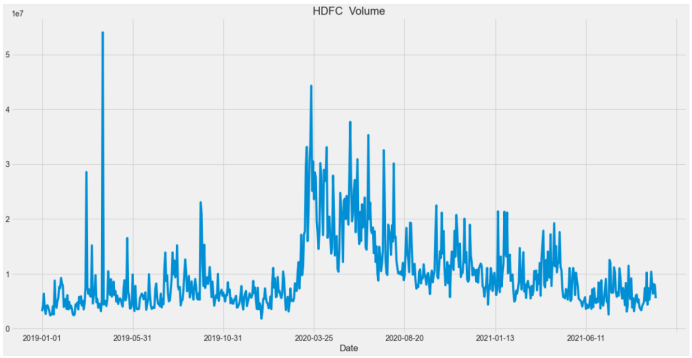

Fig. 10. HDFC Volume over dates

stock, it has been constant and have taken a sharp dip in the March 2020 but the volume of stock have been increased in the month. Then in a due course of time the price have gradually increased and was constant which can be observed in Fig 11.

The moving average of the price considering 10 days follows the general trend line, for 20 days follows general

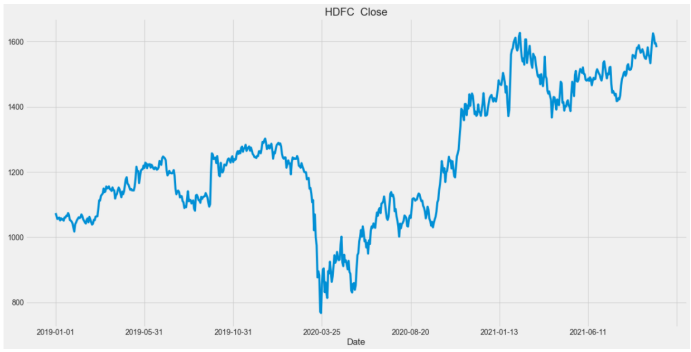

Fig. 11. HDFC Closing Price over dates

trend line and for 50 days it also stays as a constant with a slight decrease and increase in the days as seen in Fig 12.

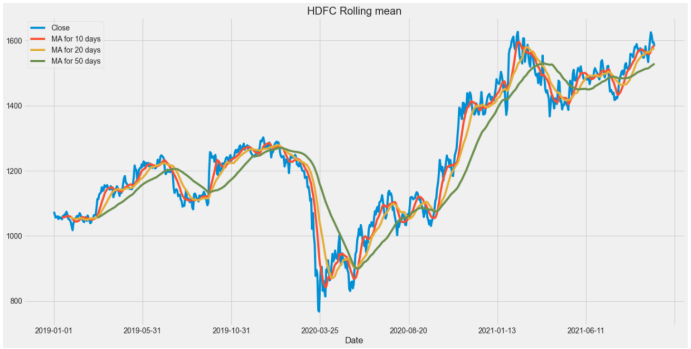

Fig. 12. HDFC Moving Average Price over dates

3) HCL Technologies: Initially the heatmap of the data was plotted in Fig 13 and it had been observed similar to the case of the Cognizant prices wherein the high,low,open and close values are perfectly correlated and hence one of the feature movement can provide insights on the remaining data.

The highest price of the stock has been constant and have been increasing continuously where in the other companies have faced losses. Then in a due course of time the price have gradually increased and was constant which can be observed in Fig 14.

The volumes of the stock that was brought initially very large and fell down and became constant till the month of January where it has risen. The volume of stocks that have been changing are constantly decreasing that can be observed in Fig 15.

The closing price being highly correlated with the highest price of the stock has been constant and have been increasing continuously where in the other companies have faced losses. Then in a due course of time the price have gradually increased and was constant which can be observed in Fig 16.

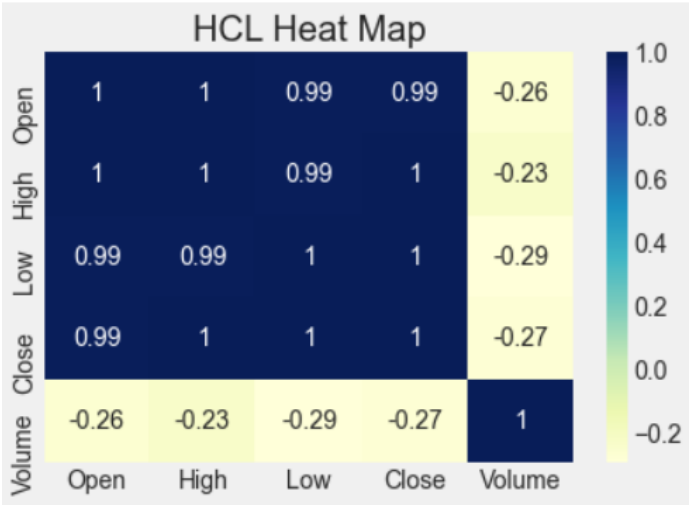

Fig. 13. HCL data Correlation

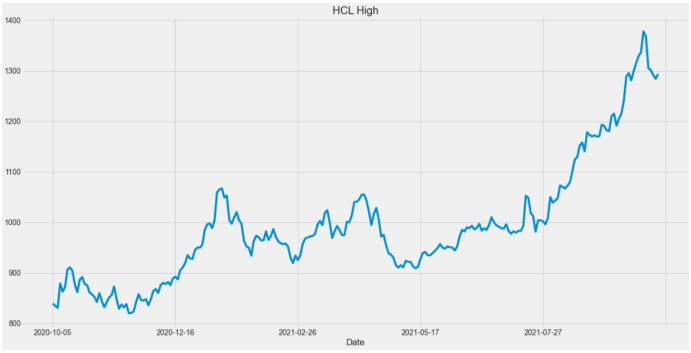

Fig. 14. HCL High Price over dates

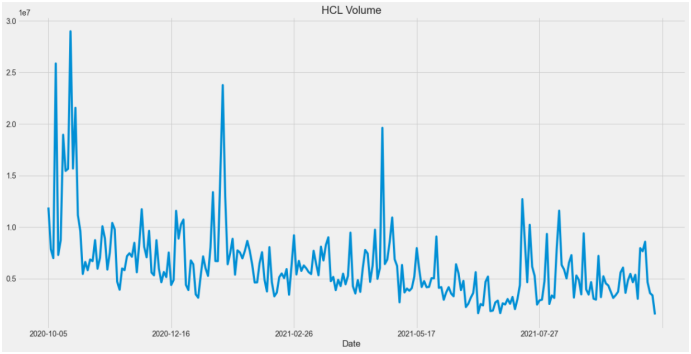

Fig. 15. HCL Volume over dates

Fig. 16. HCL Closing Price over dates

The moving average of the price considering 10 days follows the general trend line, for 20 days follows general trend line and for 50 days it also stays as a constant with a slight decrease and increase in the days as seen in Fig 17.

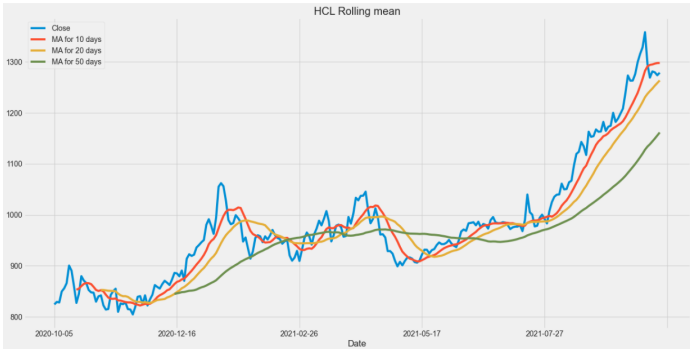

Fig. 17. HCL Moving Average Price over dates

4) ICICI Bank: Initially the heatmap of the data was plotted in Fig 18 and it had been observed similar to the case of the Cognizant prices wherein the high,low,open and close values are perfectly correlated and hence one of the feature movement can provide insights on the remaining data.

The volume of the stocks that are brought and held by the people have been remaining constant over the period of time.During the time of October 2019 the volume has suddenly increased and then fell down and became constant as shown in Fig 19.

As the data of high,low,close and open are highly correlated, only the plot for the closing price is plotted as the remaining prices will also have the same movements of up and down due to high correlation. The closing price plot can be observed in Fig 20.

The moving average of the price considering 10 days follows the general trend line, for 20 days follows general trend line and for 50 days it also stays as a constant with a

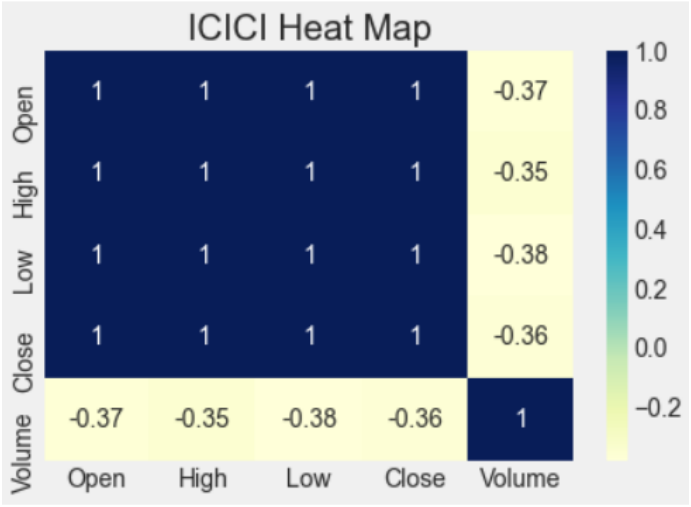

Fig. 18. ICICI data Correlation

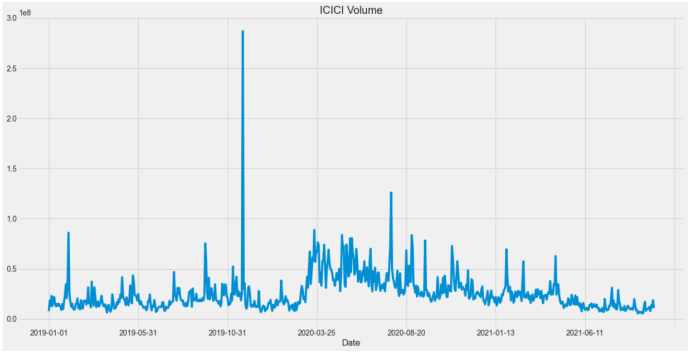

Fig. 19. ICICI Volume over dates

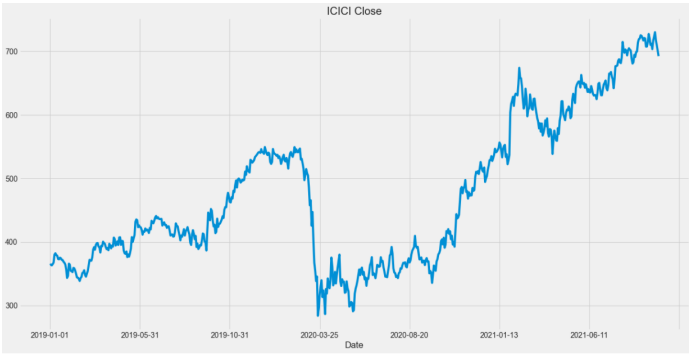

Fig. 20. ICICI Closing Price over dates

large decrease and increase in the days as seen in Fig 21.

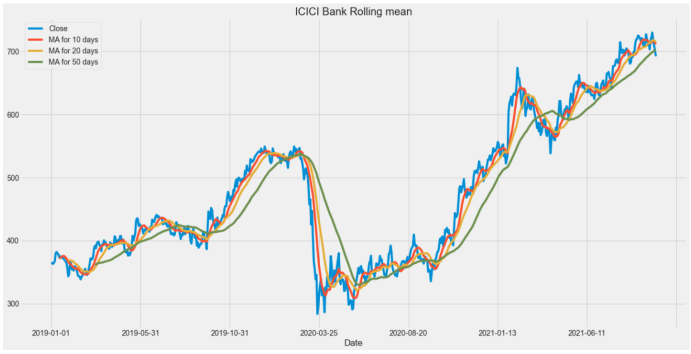

Fig. 21. ICICI Moving Average Price over dates

5) Infosys: Initially the heatmap of the data was plotted in Fig 22 and it had been observed similar to the case of the Cognizant prices wherein the high,low,open and close values are perfectly correlated and hence one of the feature movement can provide insights on the remaining data completely related to the ups and downs of the prices.

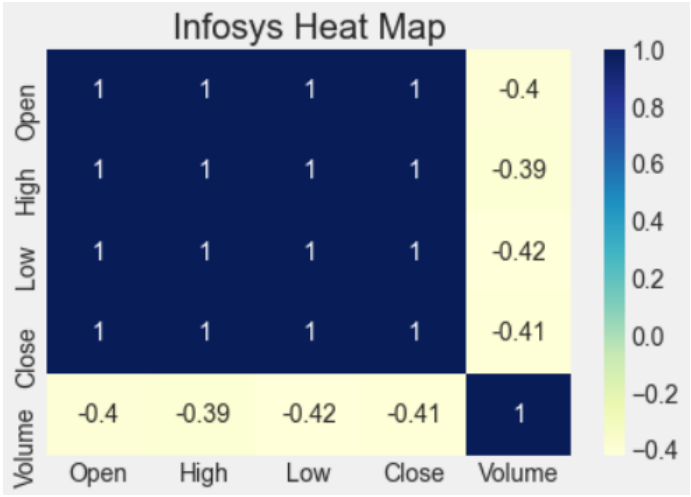

Fig. 22. Infosys data Correlation

As the data of high,low,close and open are highly correlated, only the plot for the closing price is plotted as the remaining prices will also have the same movements of up and down due to high correlation. The closing price plot can be observed in Fig 23. We can also observe that the closing price of Infosys have been continuously increasing the case with the remaining three features.

The moving average of the price considering 10 days follows the general trend line, for 20 days follows general trend line and for 50 days it also stays as a constant with a large decrease and increase in the days as seen in Fig 24.

Fig. 23. Infosys Closing Price over dates

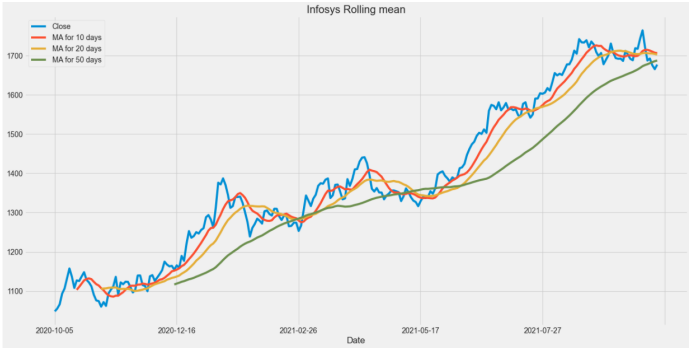

Fig. 24. Infosys Moving Average Price over dates

6) SBI Bank: Initially the heatmap of the data was plotted in Fig 25 and it had been observed similar to the case of the Cognizant prices wherein the high,low,open and close values are perfectly correlated and hence one of the feature movement can provide insights on the remaining data.

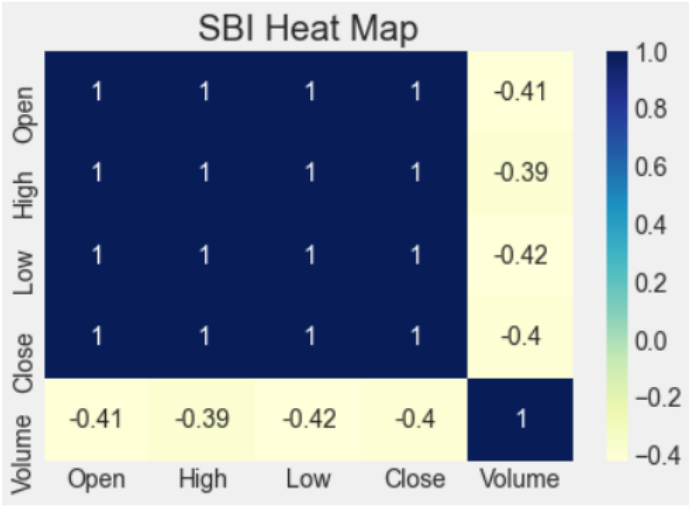

Fig. 25. SBI data Correlation

As the data of high,low,close and open are highly correlated, only the plot for the closing price is plotted as the remaining prices will also have the same movements of up and down due to high correlation. The closing price plot can be observed in Fig 26. We can see a large dip in price of stock during the months of March to may in 2020.

.

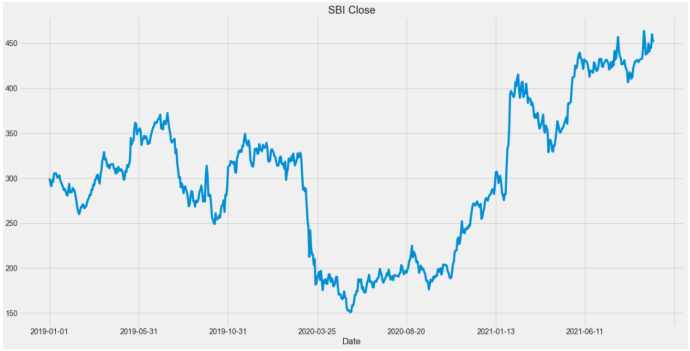

Fig. 26. SBI Closing Price over dates

The moving average of the price considering 10 days follows the general trend line, for 20 days follows general trend line and for 50 days it also stays as a constant with a slight increase in the days as seen in Fig 27.

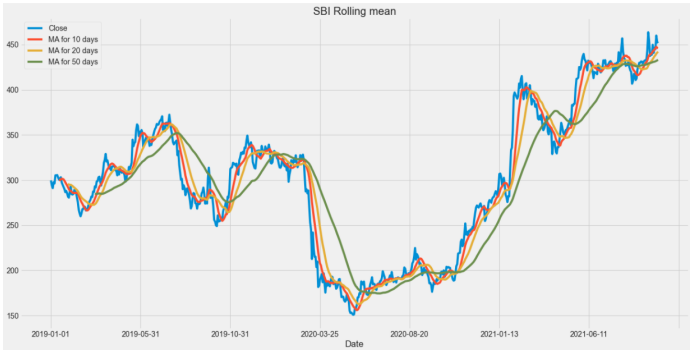

Fig. 27. SBI Moving Average Price over dates

7) Value of INR compared to USD: Initially the heatmap of the data was plotted in Fig 28 and it had been observed similar to the case of the Cognizant prices wherein the high,low,open and close values are perfectly correlated and hence one of the feature movement can provide insights on the remaining data. The adjusted closing price is also highly correlated with the closing price

As the data of high,low,close and open are highly correlated, only the plot for the closing price is plotted as the remaining prices will also have the same movements of up and down due to high correlation. The closing price plot can be observed in Fig 29. We can see that the exchange rate of USD to INR have increased significantly during the months of March 2020 to May 2020.

.

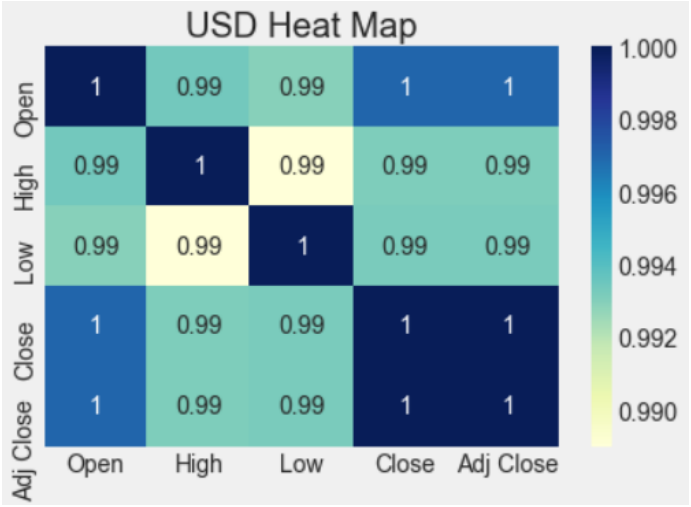

Fig. 28. USD data Correlation

Fig. 29. USD Closing Price over dates

The moving average of the price considering 10 days follows the general trend line, for 20 days follows general trend line and for 50 days it also stays as a constant with a large increase in the days of March 2020 as seen in Fig 30.

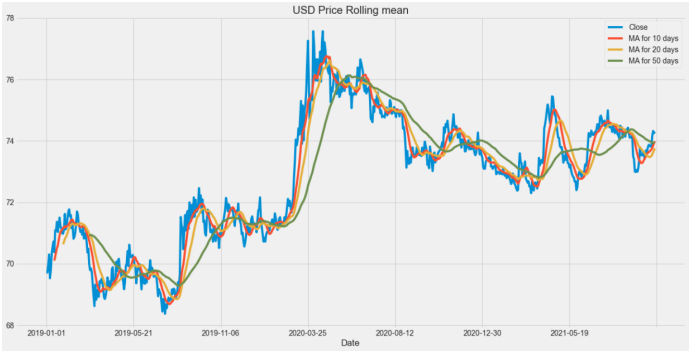

Fig. 30. USD Moving Average Price over dates

C. Types of Data and Preprocessing

We can observe from types of data all the features are of numerical data for the required model as it can work well with numbers only.

1) Missing Data Visualization: We can see that only 2 days of the data is missing from the data set . We will be removing the two dates from the data set and can be removed as it do not effect the predictions drastically.

IV. Model

After all the preprocessing of the data we are left with features open, high, low, volume, close which are only numeric values. We will be building a Extreme Gradient Boosting Regressor Model to predict the closing prices of the company.

A. XGBoost Regressor Model

The xgboost library in Python provides us with api's for directly creating XGBoosted regressor and classifiers. For developing the model we will be splitting the training data into train part and test part. The train part is the one on which the xgboost regressor model is going to be trained on and the testing set is used to check if any over fitting of the data is occurring on the model. We will be splitting the data as follows 80% to the training set and 20% to the testing set with no shuffling as the data is a time-series that needs to be predicted in order. We will then train on 80% data and predict on the 20% testing data. The scikit learn library provides us with the metrics that are required to validate our model such as Mean Squared error,R^2 score etc. A separate model is trained to predict the prices for different companies.

The model built for predicting the cognizant stock prices had a Mean Square Error of **0.005** on the training set and **0.4215** on the testing set of data which means that the model would be well able to predict into the future and no overfitting is present in it. The predictions can be seen in Fig 31.

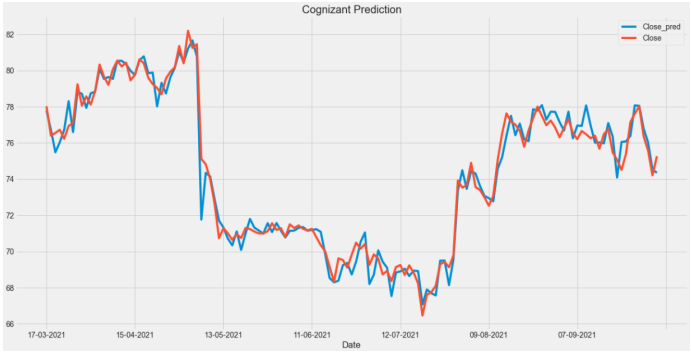

Fig. 31. Cognizant testing vs predicted prices

The model built for predicting the HDFC stock prices had a Mean Square Error of **26.91** on the training set and **220.55** on the testing set of data which means that the model would be well able to predict into the future and some overfitting is present in it. The predictions can be seen in Fig 32.

The model built for predicting the USD prices had a Mean Square Error of 10^-5 on the training set and **0.00061** on the testing set of data which means that the model would be well

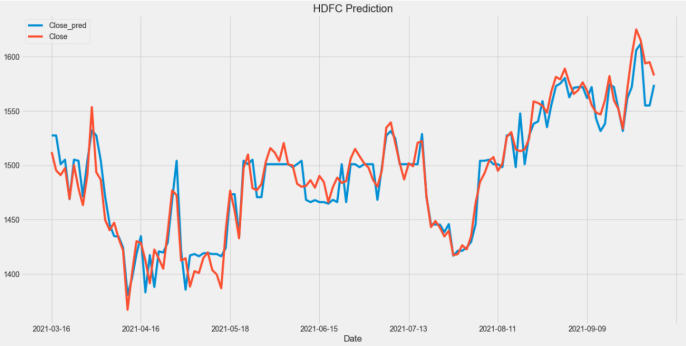

Fig. 32. HDFC testing vs predicted prices

able to predict into the future and very slight overfitting is present in it. The predictions can be seen in Fig 33. We can also observe that the predictions almost follow the same testing line.

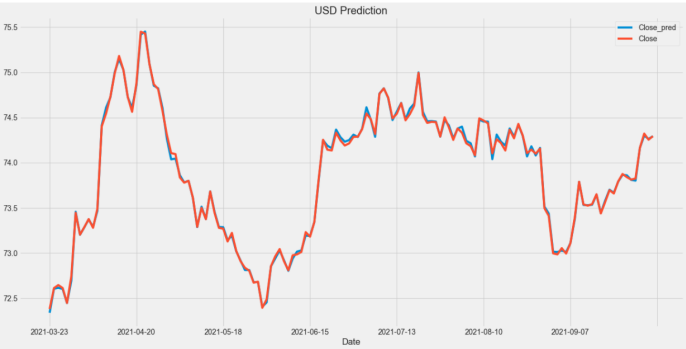

Fig. 33. USD testing vs predicted prices

The model built for predicting the ICICI prices had a Mean Square Error of **1.69** on the training set and **490** on the testing set of data which means that the model would be well able to predict into the future for some time and remains constant and large overfitting is present in it. The predictions can be seen in Fig 34. We can also observe that the predictions almost follow the same testing line for some of the days and have remained constant which can be avoided by having large amounts of data.

The model built for predicting the SBI prices had a Mean Square Error of **0.6060** on the training set and **360** on the testing set of data which means that the model would be well able to predict into the future for some time and remains constant and large overfitting is present in it. The predictions can be seen in Fig 35. We can also observe that the predictions almost follow the same testing line for some of the days and have remained constant and then predicted correctly for some days after, which can be avoided by having large amounts of data.

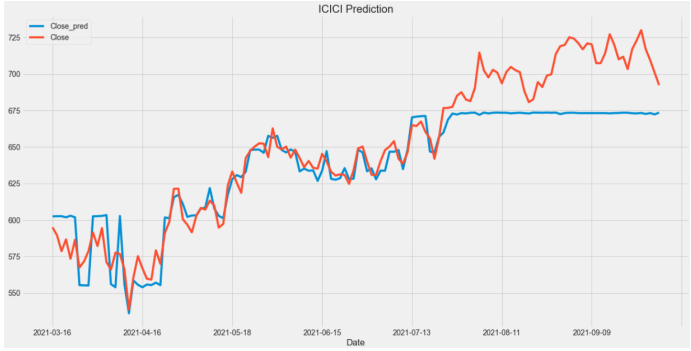

Fig. 34. ICICI testing vs predicted prices

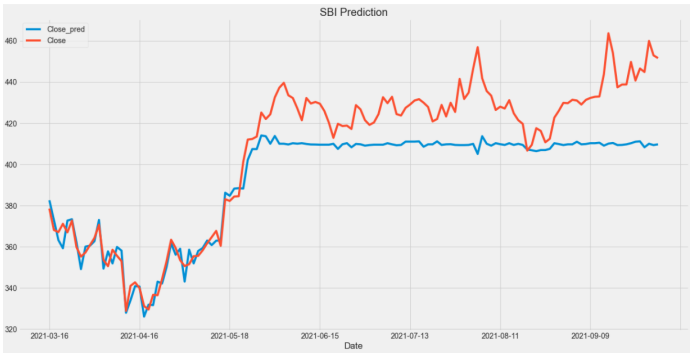

Fig. 35. SBI testing vs predicted prices

The model built for predicting the Infosys prices had a Mean Square Error of **0.043** on the training set and **164** on the testing set of data which means that the model would be well able to predict into the future for some time and remains constant and large overfitting is present in it. The predictions can be seen in Fig 36. We can also observe that the predictions almost follow the same testing line for some of the days and have remained constant and then predicted correctly for some days after, which can be avoided by having large amounts of data. This can also be observed due to fact that the number of data-points in the Infosys stock are less leading to split of 90:10 to training and testing respectively.

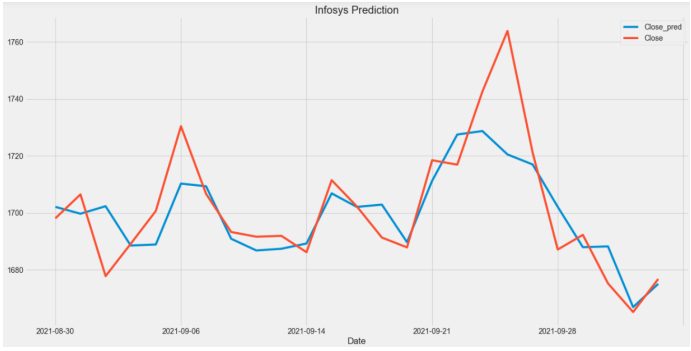

Fig. 36. Infosys testing vs predicted prices

The model built for predicting the HCL prices had a Mean Square Error of **0.0288** on the training set and **25,180** on the testing set of data which means that the model would not be well able to predict into the future a large overfitting is present in the model as the data in the training set is oscillating whereas in testing it is raising which shows that testing data cannot be predicted accurately with XGBoost model. The predictions can be seen in Fig 37. This can also be observed due to fact that the number of data-points in the HCL stock are less leading to split of 90:10 to training and testing respectively.

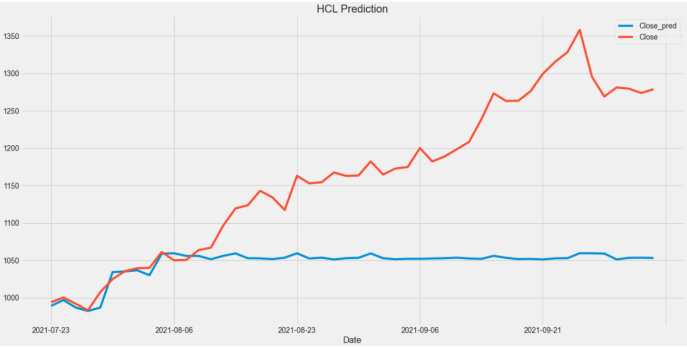

Fig. 37. HCL testing vs predicted prices

V. Conclusion

We can see that during the months of March 2020 the prices of almost every company have fallen and the exchange rate of USD to INR have been increased. This can be explained due to the fact that the lockdown was imposed during those time and saw a fall in the prices of stocks initially but began to rise shortly. We can see that the moving averages of the stocks closing prices of have been generally following the closing prices trend line. The moving average for a short number of days will approximate the closing price trend line whereas for the large number of days it approximates the average price of stock.The data in the training data set is split into ratio of 80:20 for 80% to the training of the model and 20% to the testing of model which can be used to validate if the model over fits. We have used the fastest and robust extreme gradient boosting algorithm that is trained on the data from the previous dates to predict the dates afterwards. We have used metrics of Mean Squared Error for predicting the fit of the model and the comparable errors that have been obtained in all the cases shows that the XGBoost model do not overfit on the data. We can also see that the prediction trend lines also follow the testing set trend lines which shows that out models will be predicting the future very well. The data we have also obtained is less approximately 700 datapoints which will not be completely sufficient to train any model completely and companies HCL and Infosys have further less data which lead to models being overfit on the training data and not able to generalize for the testing set.

References

[1] ”An Introduction to Statistical Learning with applications in R”
[2] Lingyu Zhang et al 2021 J. Phys.: Conf. Ser. 1873 012067
[3] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM.
[4] ”https://xgboost.readthedocs.io/en/stable/tutorials/model.html”