TECHNOCOLABS DATA SCIENCE INTERNSHIP

TEAM B – REPORT

TITLE : Predicting Stock Prices for Large Cap Technology Companies

ABSTRACT

Now a Days stock trading has become the most important activities. Stock market prediction is an act of trying to determine the future value of a stock. This paper explains the prediction of a stock using Machine Learning. 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. In this paper we propose a Machine Learning approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction.

INTRODUCTION :

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 analyse 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. In the recent years, increasing prominence of machine learning in various industries has enlightened many traders to apply machine learning techniques to the field, and some of them have produced quite promising results. This paper will develop a financial data predictor program in which there will be a dataset storing all historical stock prices and data will be treated as training sets for the program. The main purpose of the prediction is to reduce uncertainty associated to investment decision making. Stock Market follows the random walk, which implies that the best prediction you can have about tomorrow's value is today's value. Indisputably, the forecasting stock indices are very difficult because of the market volatility that needs accurate forecast model. The stock market indices are highly fluctuating and it effects the investor's belief. Stock prices are considered to be a very dynamic and susceptible to quick changes because of underlying nature of the financial domain and in part because of the mix of a known parameters (Previous day's closing price, P/E ratio etc.) and the unknown factors (like Election Results, Rumors etc.). There have been numerous attempts to predict stock price with Machine Learning. The focus of each research projects varies a lot in three ways.

The targeting price change can be near-term (less than a minute), short-term (tomorrow to a few days later), and a long-term (months later).

OVERVIEW:

- Data Segmentation and Data Cleaning
- Exploratory Data Analysis
- Training the model based on the data available

DATASETS:

The dataset consists of 9 years' worth of stock price data for 20 large-cap-technology companies and the NASDAQ-100 Technology Sector Index dating from 11/14/2012 to 11/14/2021. And other dataset consists of headlines of different companies from 01/01/2015 to 06/23/2021.

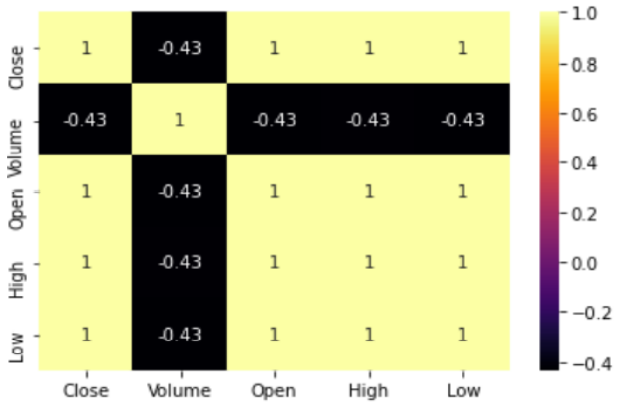
SL.NO	ATTRIBUTES	DESCRIPTION
1.	DATE	Date of the stock value
2.	OPEN	Opening price of stock
3.	CLOSE/LAST	Closing price of stock
4.	HIGH	Highest point of the price of stock at the exchange
5.	LOW	Lowest point of the price of stock at the exchange
6.	VOLUME	Volume of stock is average of total traded stocks at the exchange over a period of time.
7.	HEADLINES	companies headlines on a particular day

DATA CLEANING:

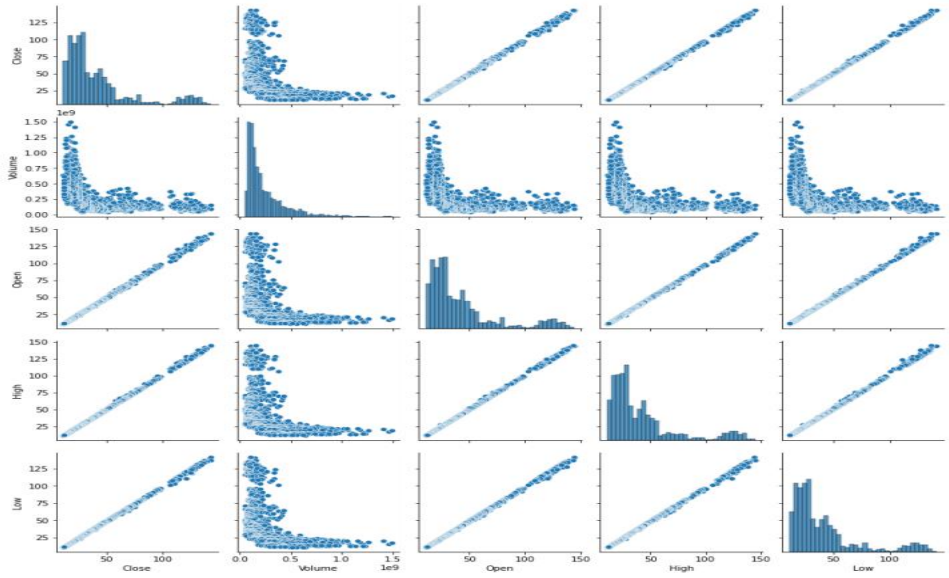
Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset that may negatively impact a predictive model. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. So, we need to clean the given dataset. In this project, we checked for null values and any duplicate values. We have done replacement of data and also data conversion. And also done the Data Visualization, This is helpful for exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more.

Exploratory Data Analysis

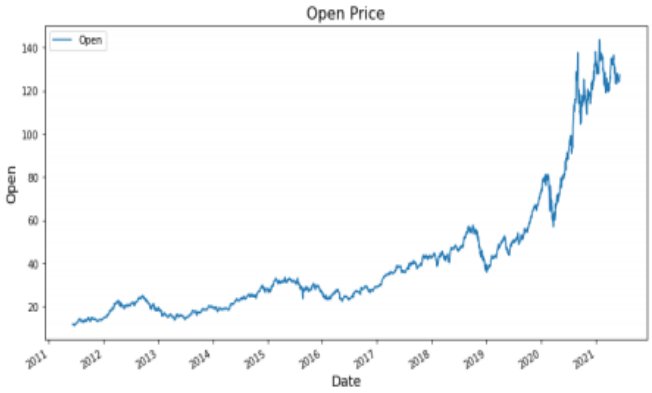
- **Plot 1:** This plot defines correlation between variables



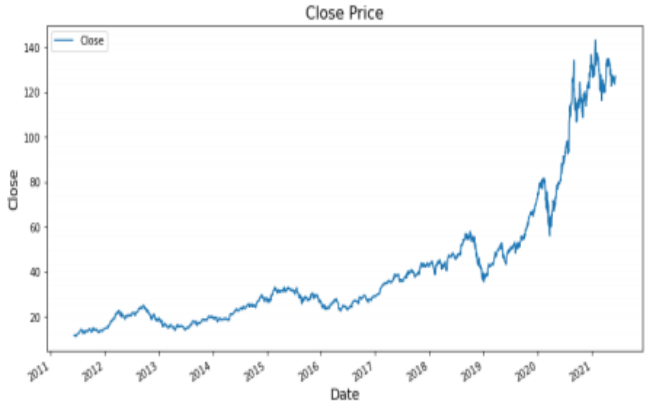
- **Plot 2:** Covariance of columns



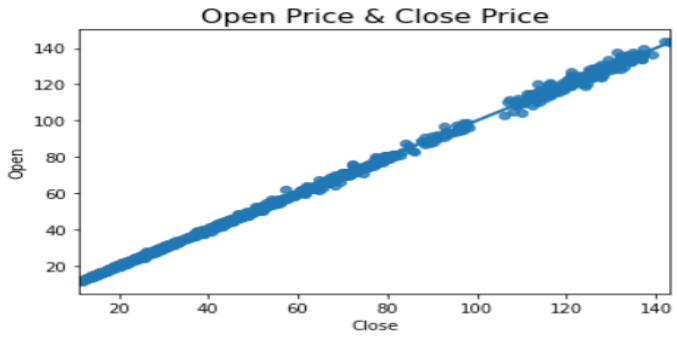
- **Plot 3:** Historical view of the Opening price



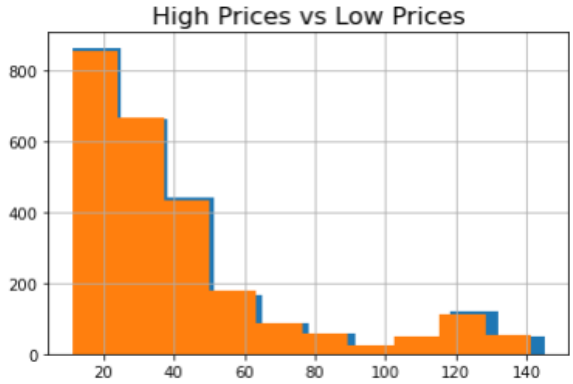
- **Plot 4:** Historical view of the closing price



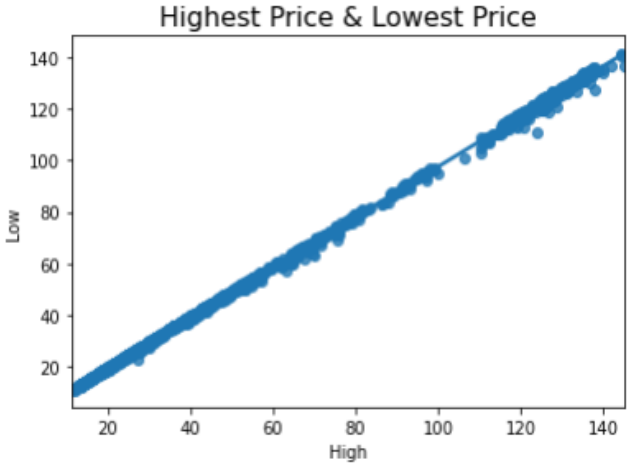
- **Plot 5:** regression line showing the relationship between 'Open Price' and 'Close Price'



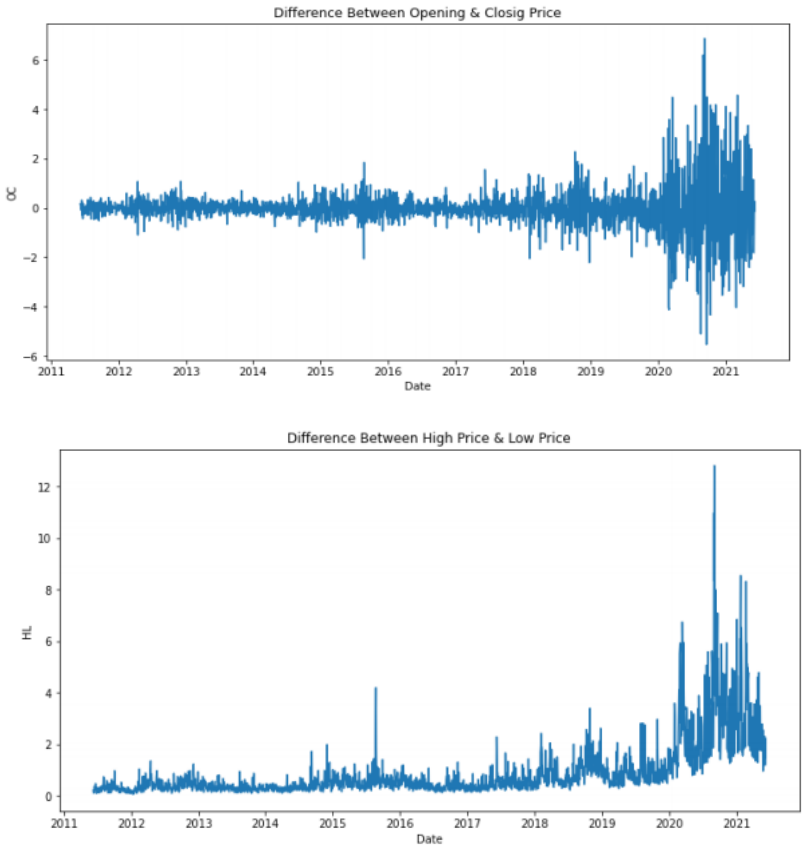
- **Plot 6:** Open Price & Close Price has a Positive Correlation



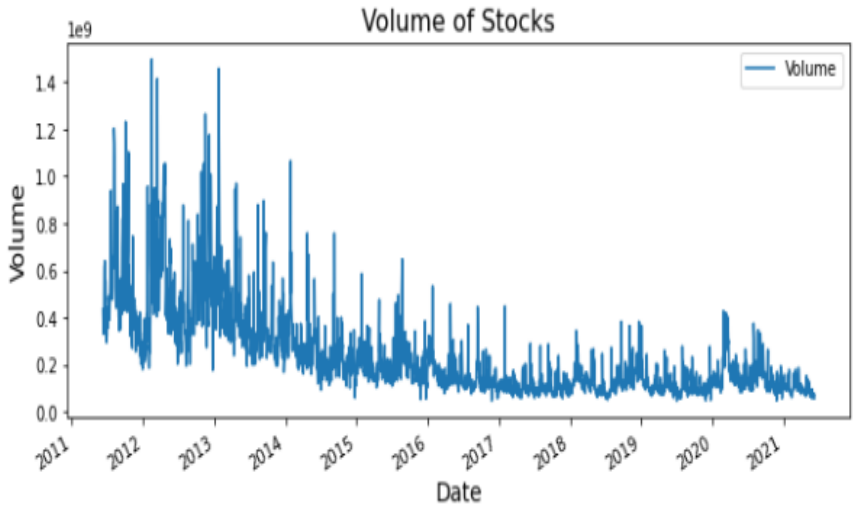
- **Plot 7:** Regression line showing the relationship between 'Highest Price' and 'Lowest Price'

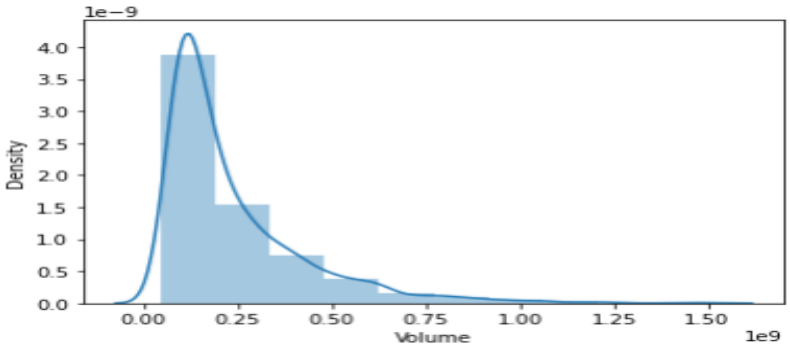


- **Plot 8:** Difference between Opening & Closing Price"

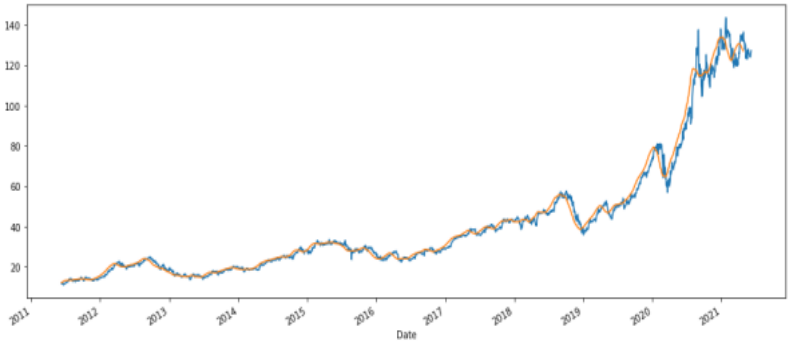


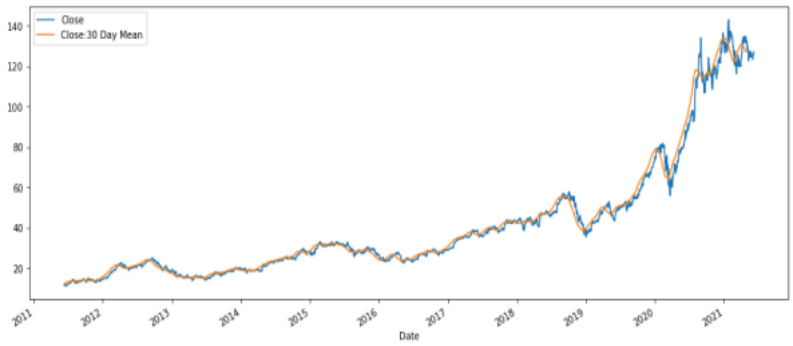
- **Plot 9:** Total volume of stock being traded each day over the years



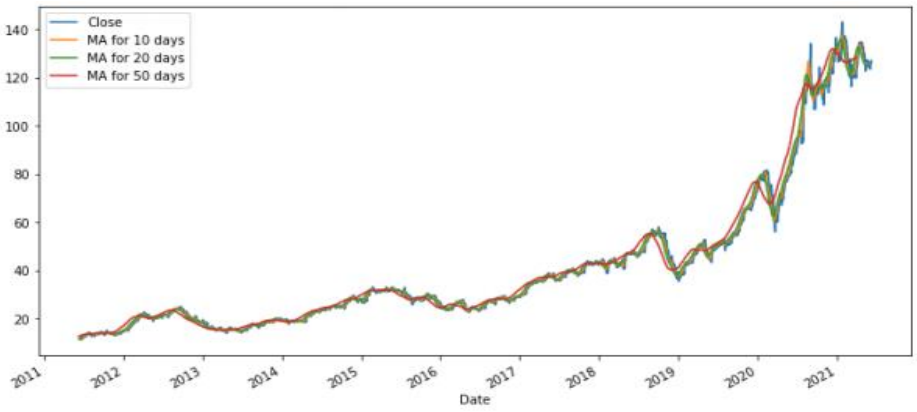


- **Plot 10:** Comparison using Average Values

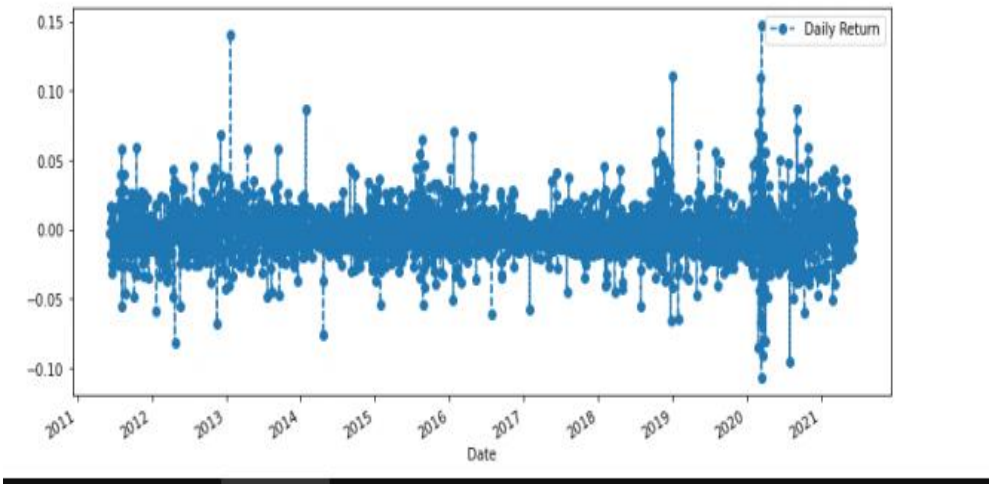




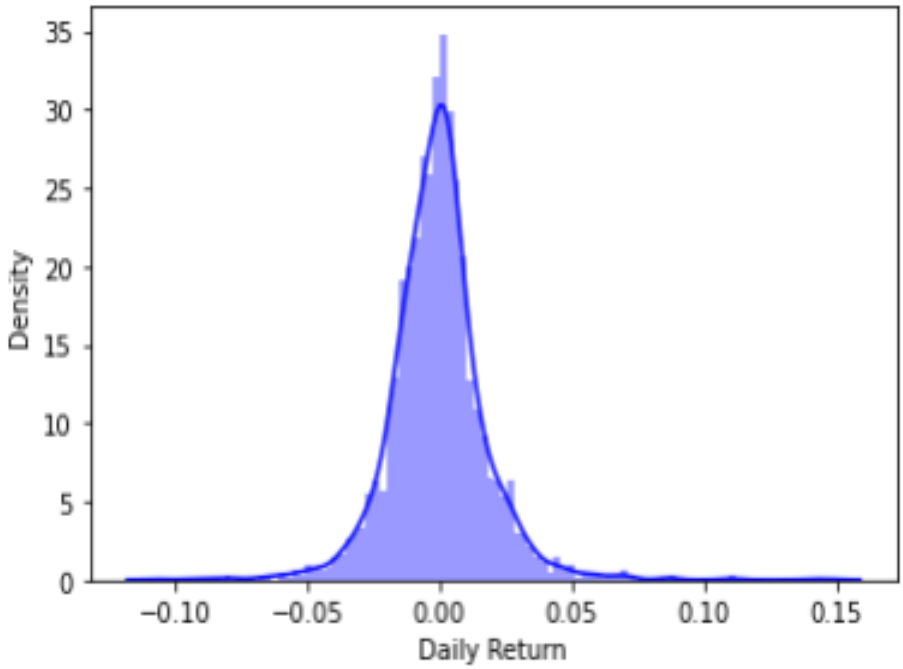
- **Plot 11:** Moving Average - MA: A trend-following, or lagging, indicator based on past prices.



- **Plot 12:** Daily Return Analysis: Daily changes of the stock



- **Plot 13:** Average daily return



TEXT PREPROCESSING

Text preprocessing is an important task and critical step in text analysis and Natural language processing (NLP). It transforms the text into a form that is predictable and analysable so that machine learning algorithms can perform better.

There are different ways to preprocess the text. Here are some of the approaches of preprocessing the text.

• Removing Punctuations

• Converting Headlines to lower case

• Tokenization

• Removing white space

• normalization

TRAINING AND TESTING THE MODEL:

The following models are trained and tested for this Project Models are:

• Linear Regression

• LSTM Neural network

Linear Regression :

Linear Regression is a linear model that assumes a linear relationship between input variables (independent variables 'x') and output variable (dependent variable-'y') such that 'y' can be calculated from a linear combination of input variables(x). For single input variable, method is referred to as Simple Linear Regression whereas for multiple input variables it is referred to as Multiple Linear Regression.

Mean squared error:

$$MSE = \frac{1}{n}\Sigma\left(y - \widehat{y}\right)^2$$

Root Mean squared error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Linear Regression Accuracy: 99.958896%

LSTM Neural Network

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at ht-1 and xt, and outputs a number between 0 and 1 for each number in the cell state Ct-1. A1 represents "completely keep this" while a0 represents "completely get rid of this

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

Next, we decide what new information we have to store in cell state. For that 'input gate layer' decides which value should be updated and tanh layer create a vector Ct, that could be added to the state. After that we work on old cell state for update that state. Which we usually represent by Ct-1, into the new cell state Ct. We multiply the old state by ft, forgetting the things we decided to forget earlier. Then we add it*Ct. This is the new candidate values, scaled by how much we decided to update each state value. Finally, for output we have to decide what output should be, and This output will be based on our cell state, but will be a filtered version. Firstly, we run sigmoid layer and multiply it by output of sigmoid function so that we get only that output which we have decided.

$$h_t = o_t \circ \sigma_h(c_t)$$

After the final output we got the accuracy for this model as

LSTM Neural network accuracy : 92.10369778655853%

MODEL EVALUATION:

Model evaluation aims to estimate the generalization accuracy of a model on future data. Methods for evaluating a model's performance are linear Regression. The term regression is used when you try to find the relationship between variables. In Machine Learning and in statistical modeling, that relationship is used to predict the outcome of events. Linear regression uses the least square method. The concept is to draw a line through all the plotted data points. The line is positioned in a way that it minimizes the distance to all of the data points. The distance is called "residuals" or "errors".

- In this module import the libraries you will need like : Pandas, matplotlib and Scipy
- Isolate Highest price as x. Isolate Lowest price as y
- Get important key values with: slope, intercept, r, p, std_err = stats.linregress(x, y)
- Create a function that uses the slope and intercept values to return a new value. This new value represents where on the y-axis the corresponding x value will be placed
- Run each value of the x array through the function. This will result in a new array with new values for the y-axis: mymodel = list(map(myfunc, x))
- Draw the original scatter plot: plt.scatter(x, y)
- Draw the line of linear regression: plt.plot(x, mymodel)
- Define maximum and minimum values of the axis

- Label the axis: "Highest price " and "Lowest price"

Table : Model Fitting :Finding the best algorithm

Accuracy	Linear Regression	LSTM(long short term memory)
Accuracy	99.822	93.338

From the table ,it is clear that “Linear Regression”outperforms the other algorithm .so we used linear regression model for the deployment.

DEPLOYMENT:

The project deployment is done using “Streamlit and Heroku”. Streamlit is an app framework to deploy machine learning apps built using Python. It is an open-source framework which is similar to the Shiny package in R. Heroku is a platform-as-a-service (PaaS) that enables deployment and managing applications built in several programming languages in the cloud.

- **Steps to deploy**
  Step 1: Run your Streamlit App Locally
  Step 2: Create a GitHub repository
  Step 3: Create a requirements.txt, setup.sh, and Procfile.
  Step 3: Connect to Heroku

Screenshot of the deployed project

Check the link for the deployed Application:

- https://apple-stock-price-predictor.herokuapp.com/

Conclusion:

Linear Regression algorithm performs well for this dataset compared with other algorithms (LSTM). The table shows the accuracy of the two algorithms.

Further we also found that to improve the model performance on the test set, the ensemble approach that is Stacking or Blending top performing algorithms can improve the accuracy of the model.

The stacking of model that is LSTM gave us a accuracy of 93.338

Github links

- GitHub - JORTINPAUL/Apple-Stock-Price-Prediction-
- GitHub - JORTINPAUL/Technocolabs-Internship-Stock-Price-Prediction-