Stock Price Prediction

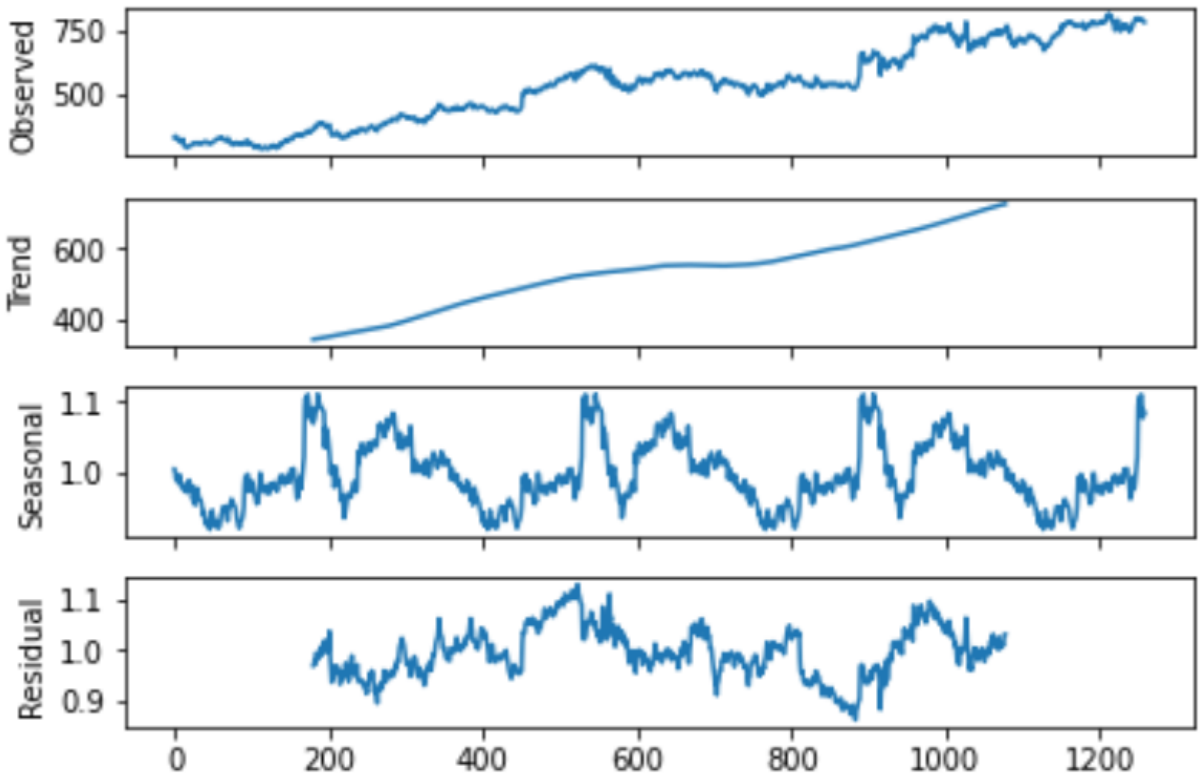
-Aman Bhansali

Abstract: In this report using Google Stock Price Dataset we try to observe the trends, train different models and compare their results with the test data provided. Also provided a pipeline that could help in the prediction of future events from the past observed data.

Introduction:
Stock market is where investors buy/sell shares of companies. The rates of the shares of the companies always keeps on varying. The trends and the ups and downs of the stock market are though influenced by many external factors which makes the prediction a difficult task. If we could built a nearly accurate model which could predict the future prices would be of a great use case for the world.

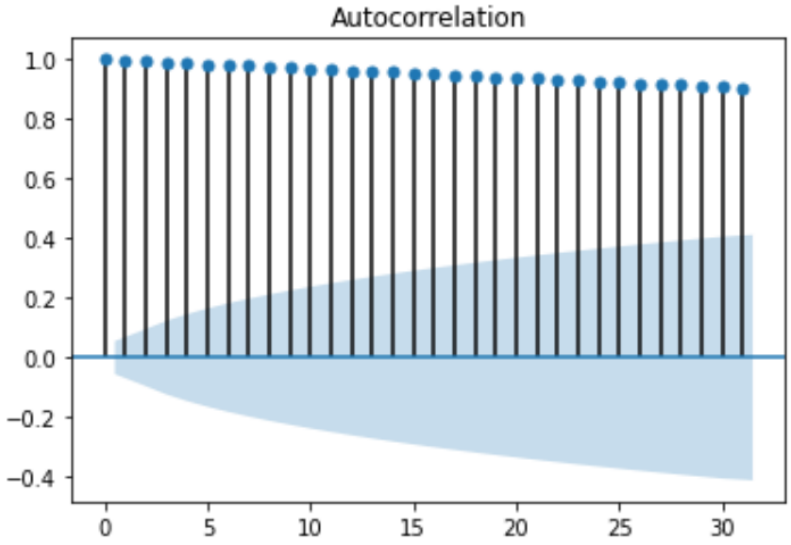
Datasets:
The train dataset contains 1258 rows and 6 columns. The columns are for the Date, Open, High, Low, Close and Volume. The rows contains the values of the each columns from 3rd Jan 2012 to 30th Dec 2016. The last columns of Close and Volume are of **string** type so a pre-processing step needs to be performed to further convert that to the **int** type as are the other datasets.
Dataset: https://www.kaggle.com/datasets/akram24/google-stock-price-train
The test dataset contains 20 rows and 6 columns. The prices are from 3rd Jan 2017 to 31st Jan 2017. All the prices are dated considering the days when the market remains close.
Dataset: https://www.kaggle.com/datasets/akram24/google-stock-price-test

Data Visualisation:



The above graph shows the opening price of the stock over 2012-2016. Also it shows the **Trend**, **Seasonality**, **Residual**. It can be visualised that the Trend shows an increasing curve which could be observes from the dataset also the prices increase over the interval. The seasonality curve shows the repeating cycles over time interval.

Autocorrelation Plot:



The spike that has a value that is significantly different from zero. If a spike is significantly different from zero, that is evidence of autocorrelation.

Observing this I tried to convert the problem to a supervised learning problem using the fact that the previous timesteps data are related to the one occurring next.

Open Price(Today) = f(Open Price(Today-1), Open Price(Today-2), ……..Open Price(Today-20))
Working on this I observed that the results were impressive.
Looking at the results now I tried to go for the more early cases and thus this time I worked with 40 and 60 previous timesteps data.

Open Price(Today) = f(Open Price(Today-1), Open Price(Today-2), ……..Open Price(Today-40))
Open Price(Today) = f(Open Price(Today-1), Open Price(Today-2), ……..Open Price(Today-60))

Dataset Creation:
To create this kind of dataset the first thing was to run a loop and store the top k i.e for the 1st case it would be from [0, 19] stored together and then [1, 20] stored in the same 2D array and then convert that to our x_train. Similarly we start a loop from 20th timestep of the train dataset and store them in one array i.e from [20, end-1]. Thus, again converting this to dataframe which is called as y_train. Now for the test dataset similar preprocessing steps are required.

For the 20 timesteps case we see that creating similar kind of processed dataset is easier for the test dats but for the other cases such as the 40 and the 60 timestep cases it can't be dine the same way. Here, say for 40 we need to add the last 20 timesteps of the train dataset to the test datset and then again follow the same procedure to get the required dataset.

Models Applied and Evaluation of Models:

1. Linear Regression
2. Random Forest Regression
3. Light GBM Regression
4. RNN/LSTM

The models are evaluated based on two parameters: Mean Squared Error, R2-Score.
The values to be observed are for the Opening Price as the rest of the columns performance was quite similar to the Opening so analyzing for one case appeared to be enough.

Results:
For the past 20 days the results for different models are as follows:

Models	MSE	R2-score
Linear regression	69.3553153187003	0.6808897316147363
Random forest	239.14878688450617	-0.10034585260040174
Light GBM	628.9537104752876	-1.893874628489507
RNN/LSTM	79.83507330259104	0.6326713886155764

For the 40 and 60 cases models used were Linear regression and RNN/LSTM.

40 Timesteps

Models	MSE	R2-score
Linear regression	62.96470672035705	0.7102935172595928
RNN/LSTM	89.04671147743036	0.590287783022649

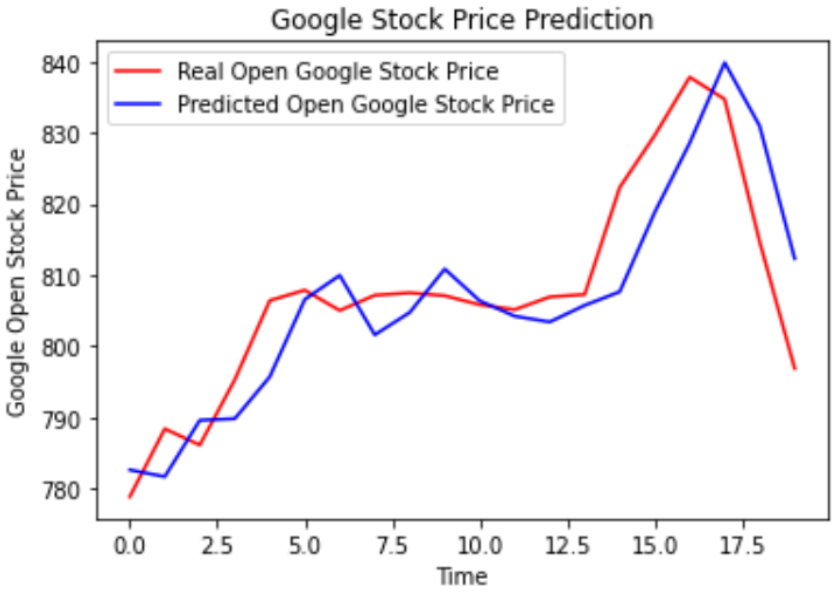
60 Timesteps

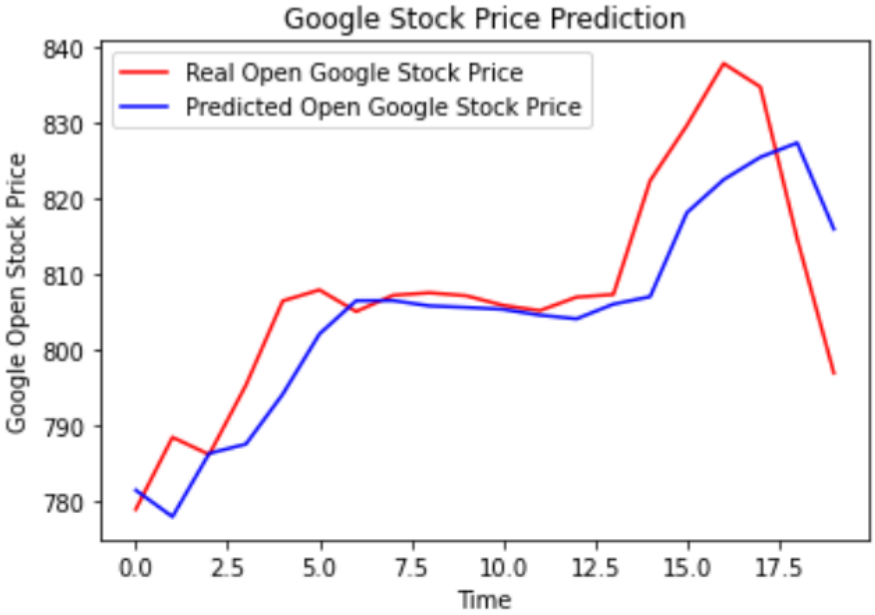
Models	MSE	R2-score
Linear regression	64.32344182853454	0.7040418504184488
RNN/LSTM	85.99185800090599	0.6043434485227163

Graph Visualisation:

For the 20 timesteps:

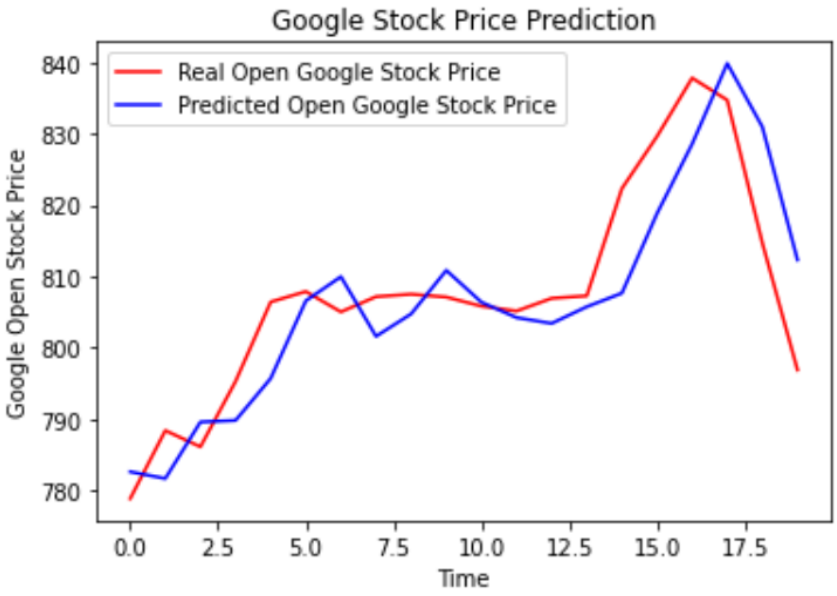
1. LR
2. RNN/LSTM

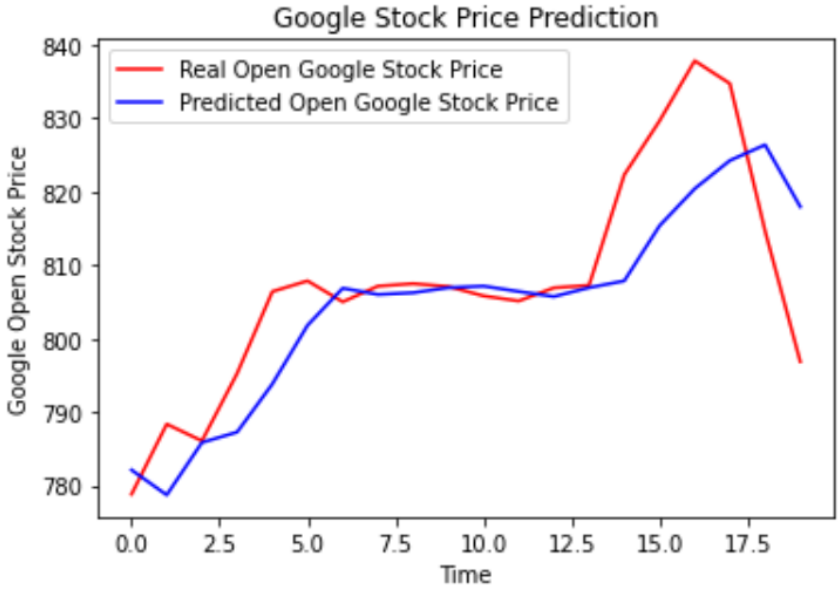




For the 40 timesteps

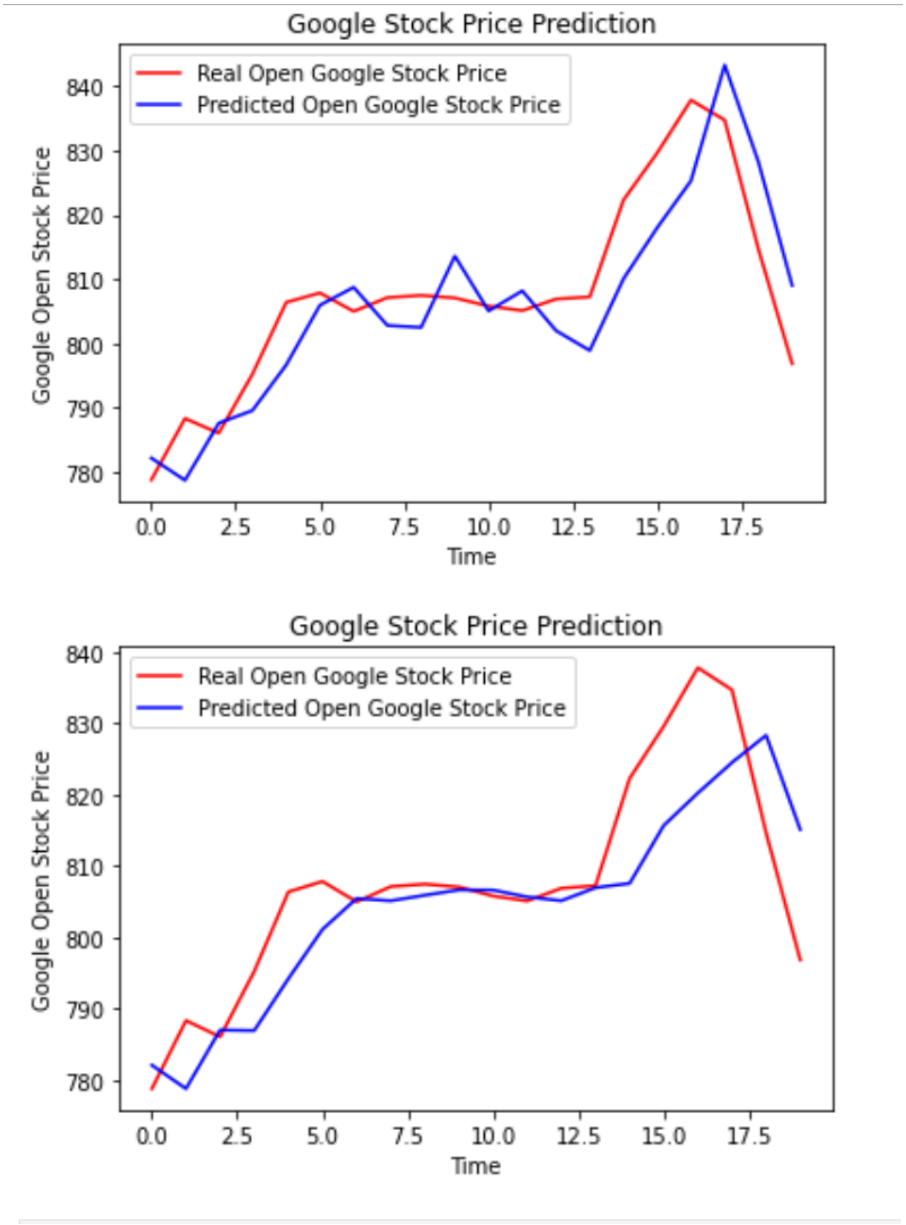
1. LR
2. RNN/LSTM



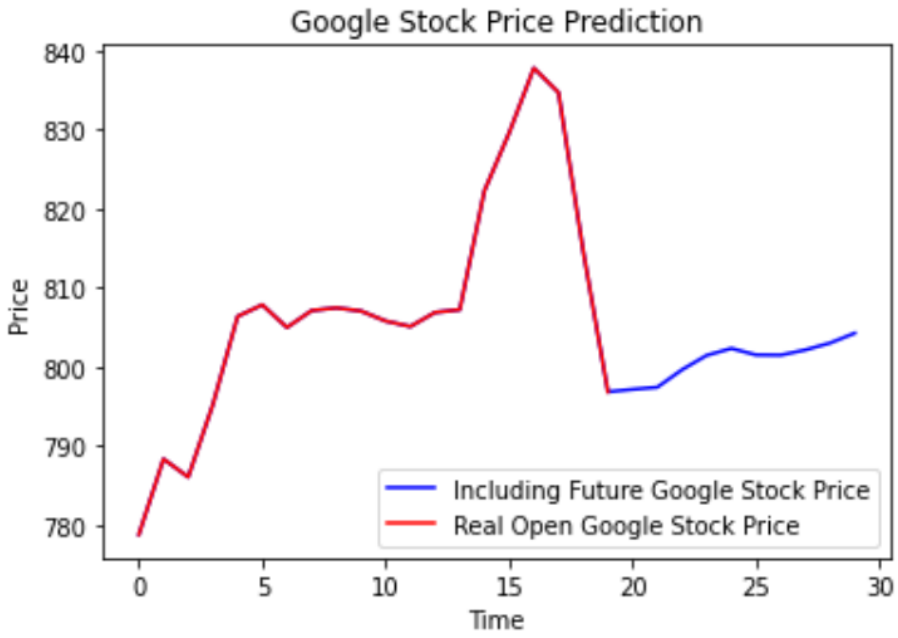


For the 60 timesteps:

1. LR
2. RNN/LSTM



For the future prediction I developed a similar pipeline through which future values could be predicted the predictyed values would be re-inserted inside the dataset to make it fot future predictions.
Future prediction curve shown here:



References:

1. https://keras.io/api/layers/recurrent_layers/lstm/
2. https://medium.com/swlh/time-series-analysis-7006ea1c3326
3. A Comprehensive guide to Time Series Analysis - Analytics Vidhya