Applied Time Series Analysis (CS575)

Project Report

on

Analysis of Financial Time Series

Submitted by

2011MT20 Shukla Aditya Manishbhai

Under the guidance of

Associate Professor Dr. Jimson Mathew

Department of Computer Science and Engineering

Indian Institute of Technology Patna

Bihta, Patna, Bihar, India – 801106

Contents

List of Figures

Chapter 1

Objective and Methodology

1.1 Objective

Objective of the project is to find an appropriate model for the stock prediction and Estimate the parameters of the chosen time series model. Accurate prediction of stock market returns is a very challenging task due to volatile and non-linear nature of the financial stock markets. With the introduction of artificial intelligence and increased computational capabilities, programmed methods of prediction have proved to be more efficient in predicting stock prices. In this, Auto Regressive Integrated Moving Average(ARIMA), Exponential Smoothing and Long Short-Term Memory (LSTM) techniques have been utilized for predicting the next day closing price for four companies. The models are evaluated using standard strategic indicator RMSE. The low values of RMSE show that the models are efficient in predicting stock closing price.

1.2 Methodology

The historical data for the five companies has been collected from Yahoo Finance [23]. The dataset includes 2 year data from 1/8/2014 to 30/11/2016 of IBM, Apple, Facebook and Google. The data contains information about the stock such as High, Low, Open, Close, Adjacent close and Volume. Only the day-wise closing price of the stock has been extracted.

1.2.1 ARIMA

ARIMA stands for Auto Regressive Integrated Moving Average. ARIMA is a simple stochastic time series model that we can use to train and then forecast future time points. ARIMA can capture complex relationships as it takes error terms and observations of lagged terms. These models rely on regressing a variable on past values.

Auto Regressive (AR) property of ARIMA is referred to as P.Past time points of time series data can impact current and future time points. ARIMA models take this concept into account when forecasting current and future values. ARIMA uses a number of lagged observations of time series to forecast observations. A weight is applied to each of the past term and the weights can vary based on how recent they are.AR(x) means x lagged error terms are going to be used in the ARIMA model. ARIMA relies on AutoRegression. Autoregression is a process of regressing a variable on past values of itself. Autocorrelations gradually decay and estimate the degree to which white noise characterizes a series of data.

If a trend exists then time series is considered non stationary and shows seasonality. Integrated is a property that reduces seasonality from a time series. ARIMA models have a degree of differencing which eliminates seasonality. D property of ARIMA represents degree of differencing.

Error terms of previous time points are used to predict current and future point's observation. Moving average (MA) removes non-determinism or random movements from a time series. The property Q represents Moving Average in ARIMA. It is expressed as MA(x) where x represents previous observations that are used to calculate current observation. Moving average models have a fixed window and weights are relative to the time. This implies that the MA models are more responsive to current event and are more volatile.

1.2.2 Exponential Smoothing

Triple exponential smoothing applies exponential smoothing three times, which is commonly used when there are three high frequency signals to be removed from a time series under study. There are different types of seasonality: 'multiplicative' and 'additive' in nature, much like addition and multiplication are basic operations in mathematics.

The method calculates a trend line for the data as well as seasonal indices that weight the values in the trend line based on where that time point falls in the cycle. Let s_t represent the smoothed value of the constant part for time t, b_t is the sequence of best estimates of the linear trend that are superimposed on the seasonal changes, and c_t is the sequence of seasonal correction factors. We wish to estimate c_t at every time t, L in the cycle that the observations take on. As a rule of thumb, a minimum of two full seasons (or 2L periods) of historical data is needed to initialize a set of seasonal factors.

The output of the algorithm is again written as F_{t+m}, an estimate of the value of x_{t+m} at time t+m > 0 based on the raw data up to time t. Triple exponential smoothing with multiplicative seasonality is given by the formulas

$$\begin{aligned}
s_0 &= x_0 \\
s_t &= \alpha \frac{x_t}{c_{t-L}} + (1-\alpha)(s_{t-1} + b_{t-1}) \\
b_t &= \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1} \\
c_t &= \gamma \frac{x_t}{s_t} + (1-\gamma)c_{t-L} \\
F_{t+m} &= (s_t + mb_t)c_{t-L+1+(m-1) \bmod L},
\end{aligned}$$

1.2.3 Long Short-Term Memory(LSTM)

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). 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.

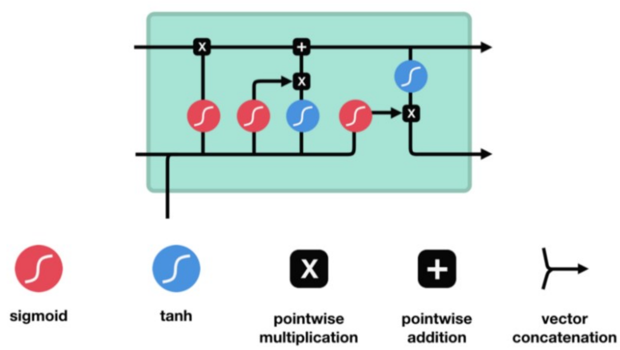

Figure 1.1: LSTM cell and it's operation

Chapter 2

Result and Analysis

2.1 Statistical Analysis

2.1.1 Stationarity of Time Series Data

Stationarity Shows the mean value of the series that remains constant over a time period; if past effects accumulate and the values increase toward infinity, then stationarity is not met.

Differencing Used to make the series stationary, to De-trend, and to control the auto-correlations; however, some time series analyses do not require differencing and over-differenced series can produce inaccurate estimates.

Augmented Dickey-Fuller Test (ADF)

This test is used to assess whether or not a time-series is stationary. This test will give test-statistic, based on which we can say, with different levels (or percentage) of confidence, if the time-series is stationary or not.
Results of Dickey-Fuller Test for 'Close' on IBM dataset:
Test Statistic : -2.279273
p-value : 0.178740
Critical Value (1%): -3.441520
Critical Value (5%) : -2.866468
Critical Value (10%): -2.569394

In all four data series, The test statistic of ADF is greater than the critical value, so we fail to reject the null hypothesis. So it is non-stationary series. Also P value is greater than 0.05 so, from that also we can say it is non-stationary.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

Results of KPSS Test for 'Close' on IBM dataset:
Test Statistic : 1.268862
p-value : 0.010000
Critical Value (1%): 0.739000
Critical Value (5%) : 0.463000
Critical Value (10%): 0.347000

In all four data series, The test statistic of KPSS is greater than the critical value so we can reject the null hypothesis. So it is non-stationary series.

2.1.2 ACF and PACF

An autocorrelation (ACF) plot represents the autocorrelation of the series with lags of itself. A partial autocorrelation (PACF) plot represents the amount of correlation between a series and a lag of itself that is not explained by correlations at all lower-order lags. Ideally, we want no correlation between the series and lags of itself. subcaption

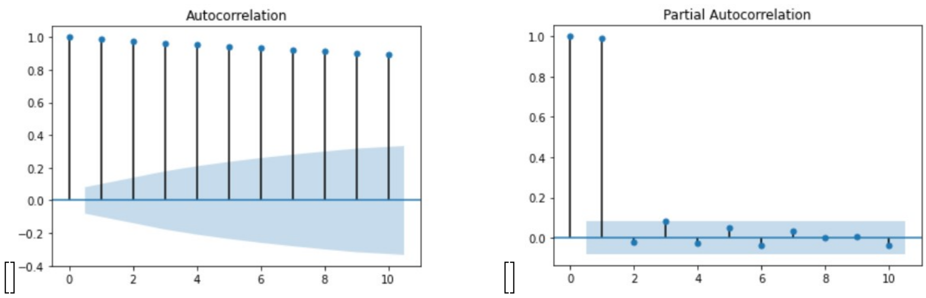

Figure 2.1: ACF and PACF plot

From ACF plot, we can see that at each lag autocorrelation decreasing slowly. So data is non stationary and to make it stationary we can use differencing method.

Here, I have used simple differencing and plotted the acf,pacf again to see status of stationarity. The above and below shown plots of ACF and PACF are common (i.e. getting almost similar plots) for all four dataset.

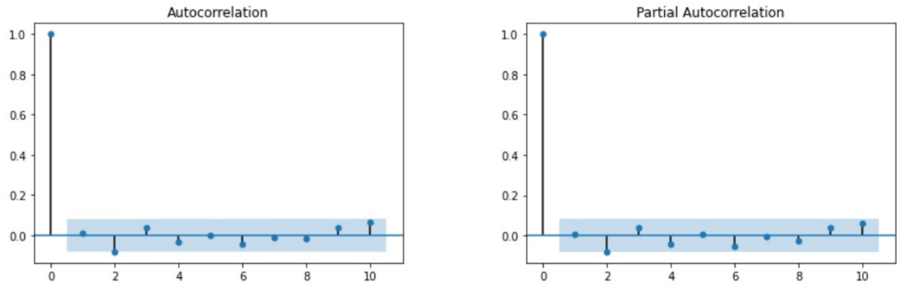

Figure 2.2: ACF and PACF plot after simple difference

2.2 Plots of data analysis design and comparison

This section contains original data plots,analysis of data and predicted data plots using three different models of all four dataset.

2.2.1 IBM dataset

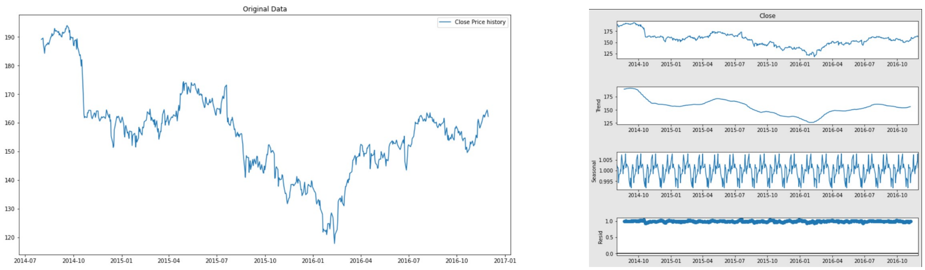

Figure 2.3: IBM Dataset (a)Original data plot (b)Decomposition of data

From visual observation, we can comment on the stationarity of the data. Also, Decomposition data plot is showing trend, seasonality and reside of original data.

Comparision of models on IBM dataset

Comparison of ARIMA, Exponential Smoothing and LSTM models from RMSE value and prediction plot visualization.

For a taken parameter, RMSE value on test data using
(1) ARIMA Model: 1.6630943070076387
(2) Exponential Smoothing: 1.8485760646136666
(3) LSTM: 2.7803292896838854

Prediction plots of all three models

:

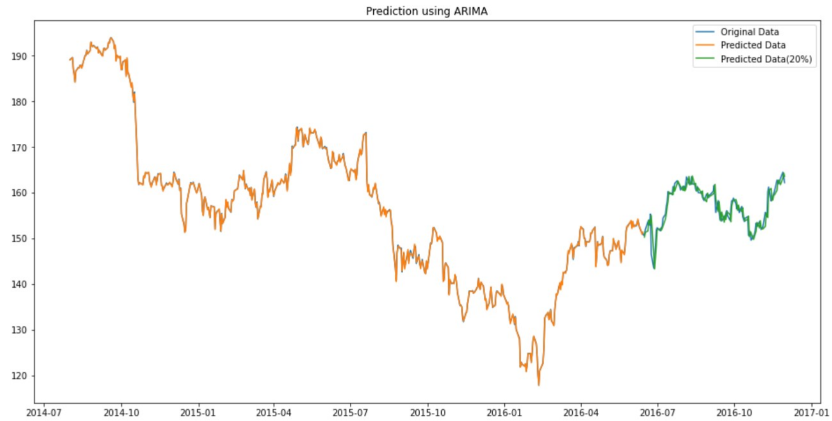

Figure 2.4: Prediction using ARIMA model on IBM data

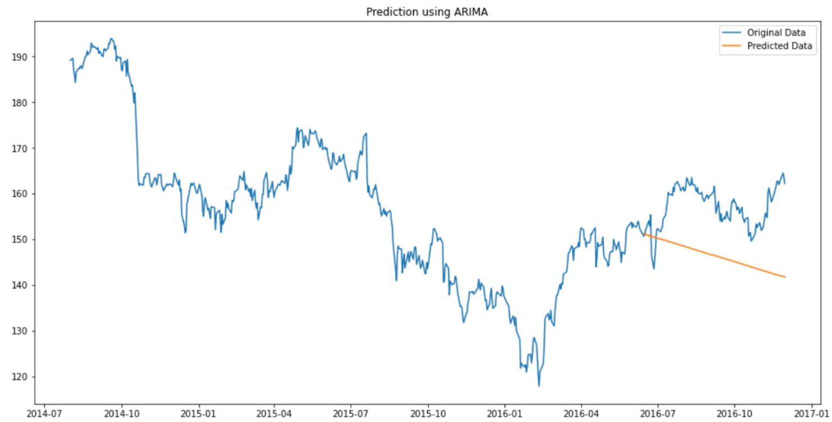

Figure 2.5: Prediction using ARIMA model(fitted train data only) on IBM data

So from RMSE value and prediction plots, we can say that ARIMA predicts better for a given parameter. In actual, above sentence can not be said. Statistical models like ARIMA , Exponential smoothing are not training the model like DL,ML model. If we want to predict future days stock price (i.e. future data is not available) using ARIMA or Exponential Smoothing, It will not give better result as you can see in the figure 2.5. Also, if we give high time-step in LSTM and train data is sufficient enough, It can predict much more accurate.

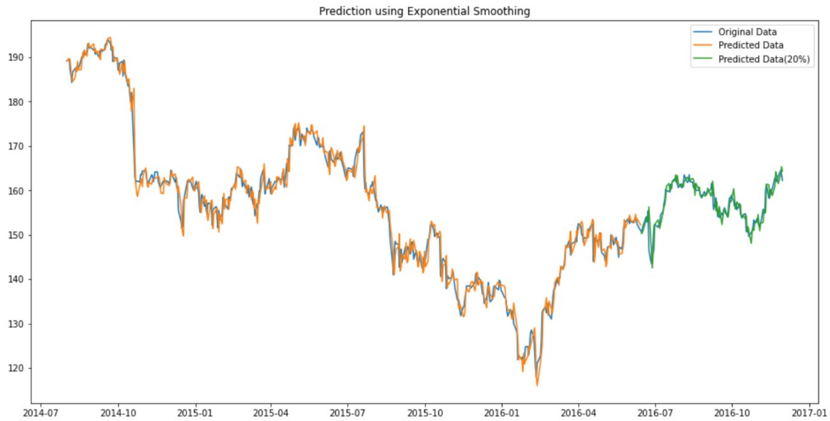

Figure 2.6: Prediction using Exponential Smoothing model on IBM data

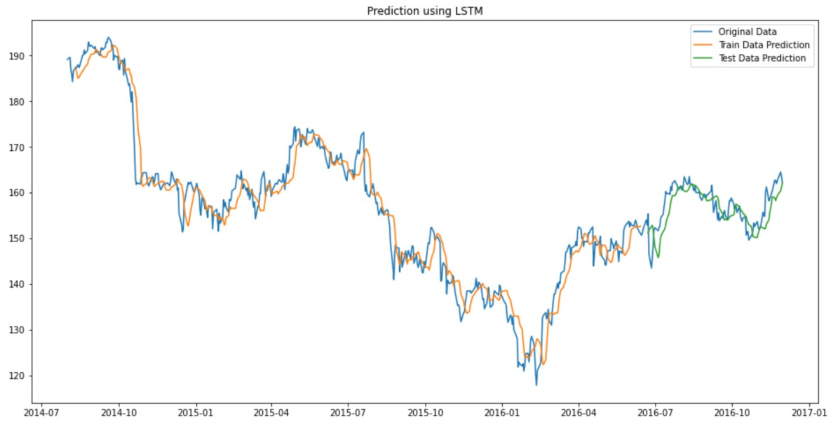

Figure 2.7: Prediction using LSTM model on IBM data

2.2.2 Apple dataset

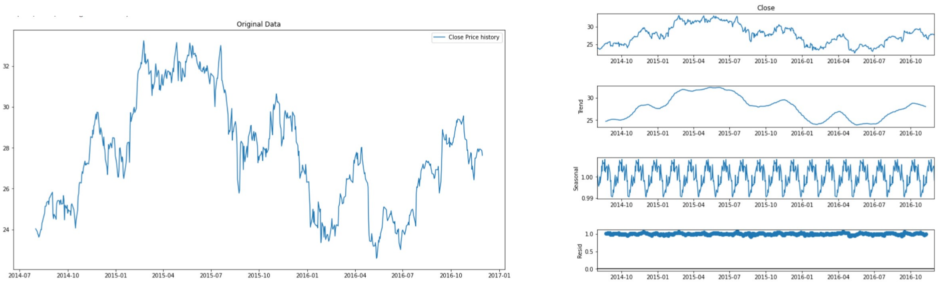

Figure 2.8: Apple Dataset (a)Original data plot (b)Decomposition of data

Comparision of models on Apple dataset

Comparison of ARIMA, Exponential Smoothing and LSTM models from RMSE value and prediction plot visualization.

For a taken parameter, RMSE value on test data using
(1) ARIMA Model: 0.339520427471949
(2) Exponential Smoothing: 0.3859302041134889
(3) LSTM: 0.6171199441917113
So from RMSE value and prediction plots, we can say that ARIMA predicts better for a given parameter.

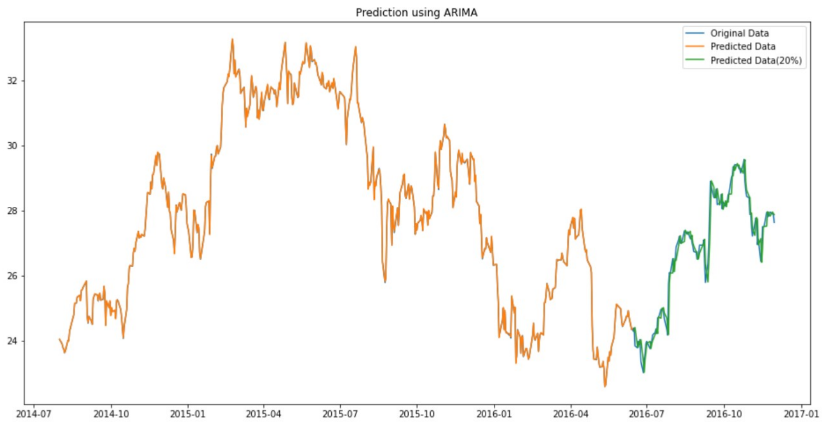

Figure 2.9: Prediction using ARIMA model on Apple data

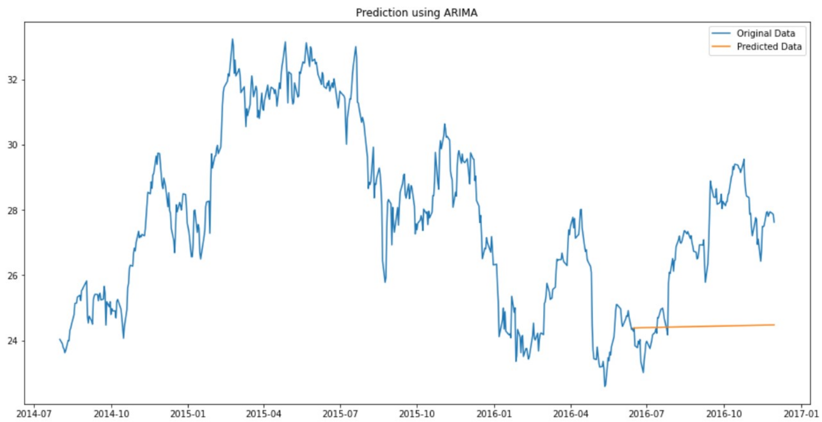

Figure 2.10: Prediction using ARIMA model(fitted train data only) on Apple data

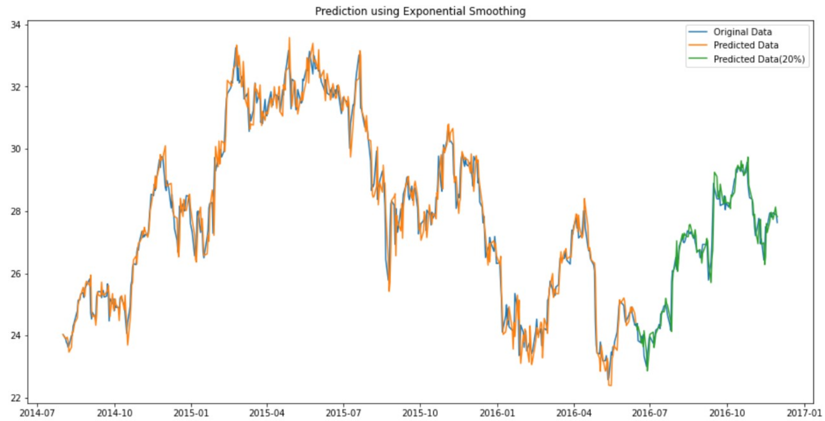

Figure 2.11: Prediction using Exponential Smoothing model on Apple data

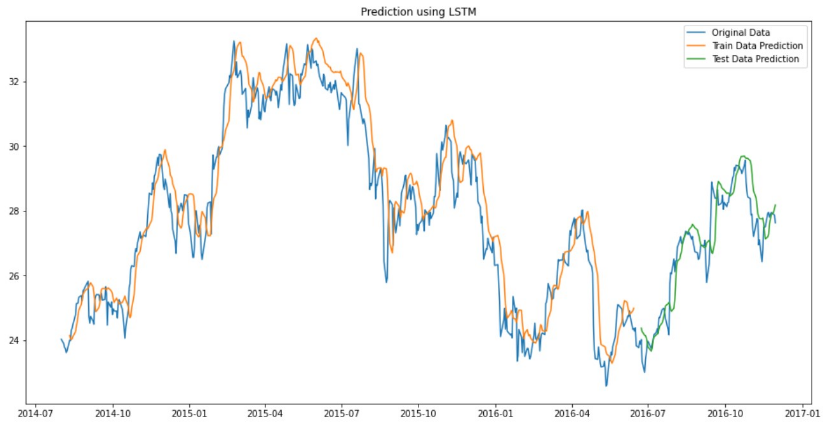

Figure 2.12: Prediction using LSTM model on Apple data

2.2.3 Facebook dataset

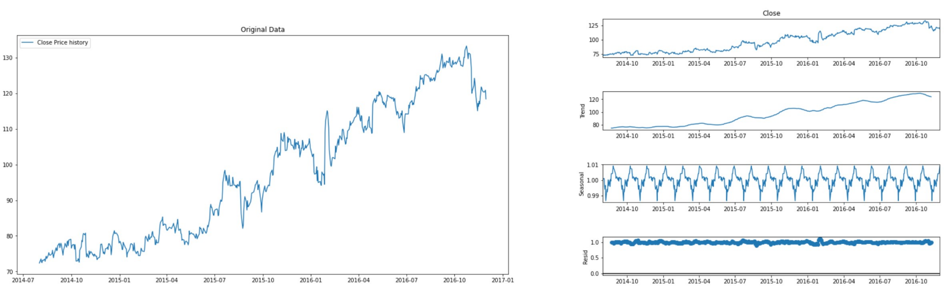

Figure 2.13: Facebook Dataset (a)Original data plot (b)Decomposition of data

Comparision of models on facebook dataset

Comparison of ARIMA, Exponential Smoothing and LSTM models from RMSE value and prediction plot visualization.

For a taken parameter, RMSE value on test data using
(1) ARIMA Model: 1.532645012443441
(2) Exponential Smoothing: 1.7682731892817989
(3) LSTM: 2.7084390945699O4
So from RMSE value and prediction plots, we can say that ARIMA predicts better for a given parameter.

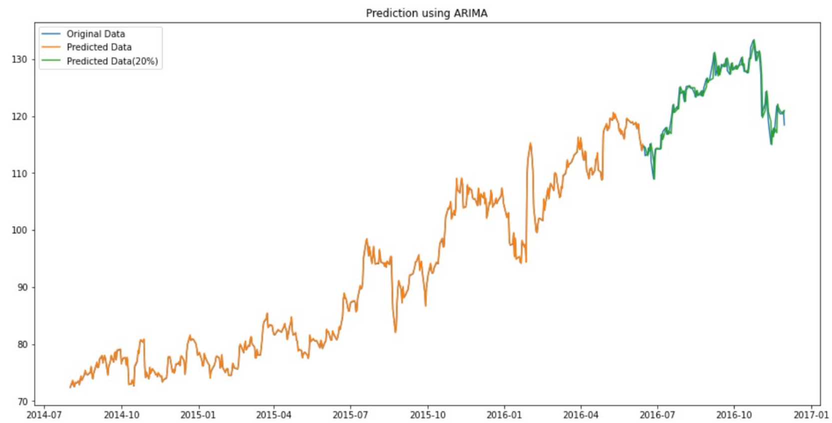

Figure 2.14: Prediction using ARIMA model on Facebook data

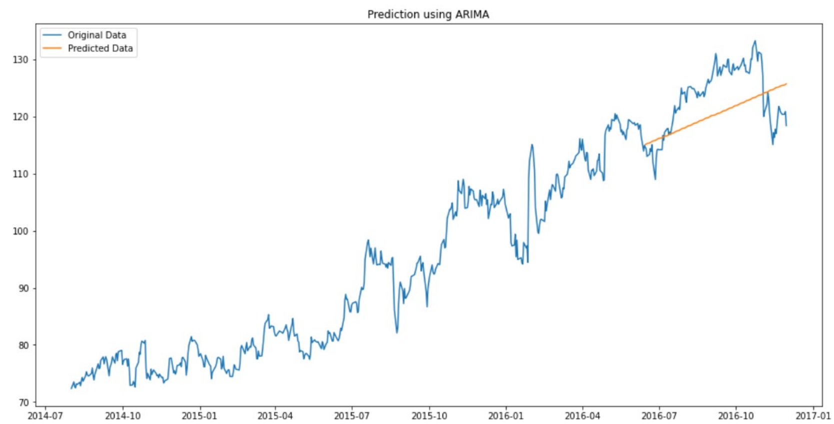

Figure 2.15: Prediction using ARIMA model(fitted train data only) on Facebook data

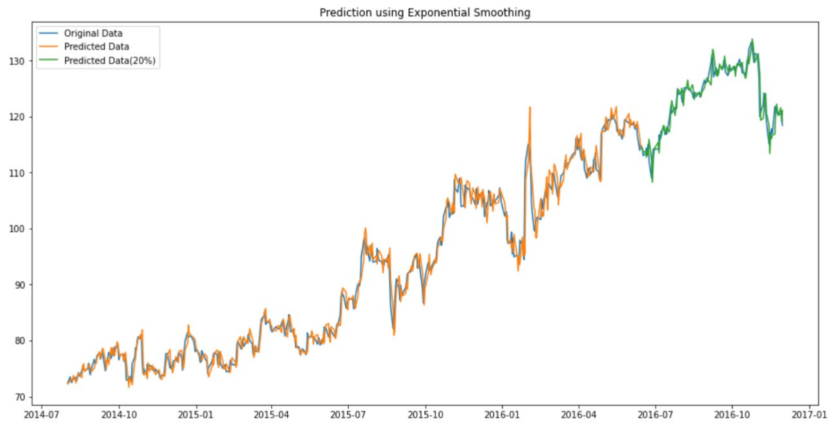

Figure 2.16: Prediction using Exponential Smoothing model on Facebook data

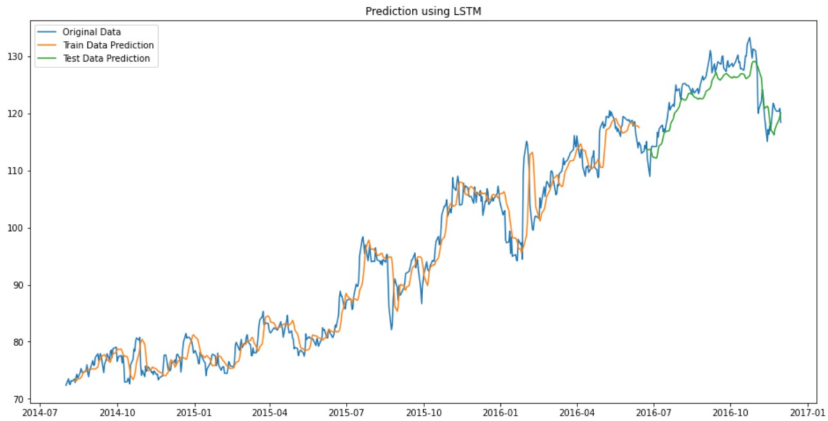

Figure 2.17: Prediction using LSTM model on Facebook data

2.2.4 Google dataset

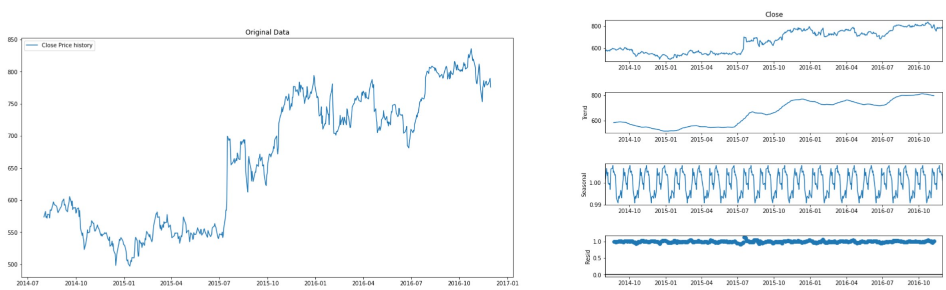

Figure 2.18: Google Dataset (a)Original data plot (b)Decomposition of data

Comparision of models on Google dataset

Comparison of ARIMA, Exponential Smoothing and LSTM models from RMSE value and prediction plot visualization.

For a taken parameter, RMSE value on test data using
(1) ARIMA Model: 8.428799407516538
(2) Exponential Smoothing: 9.584771173799444
(3) LSTM: 11.13909482757507
So from RMSE value and prediction plots, we can say that ARIMA predicts better for a given parameter.

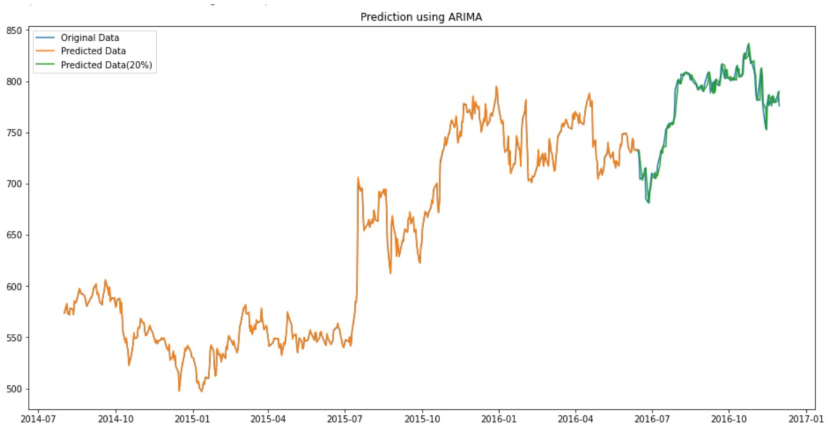

Figure 2.19: Prediction using ARIMA model on Google data

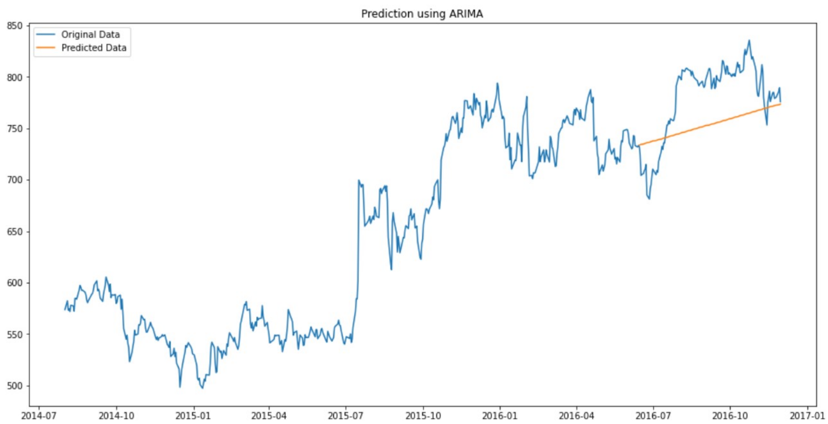

Figure 2.20: Prediction using ARIMA model(fitted train data only) on Google data

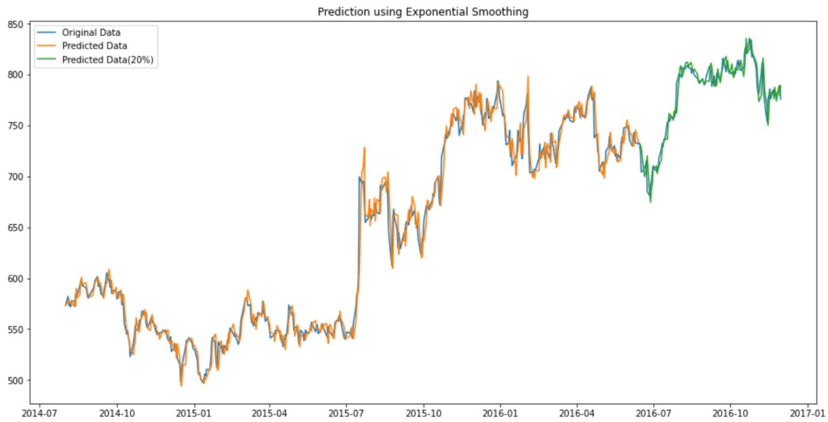

Figure 2.21: Prediction using Exponential Smoothing model on Google data

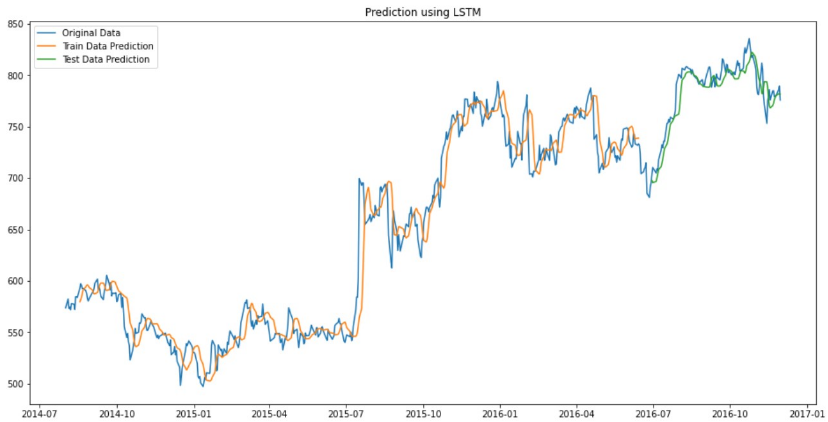

Figure 2.22: Prediction using LSTM model on Google data

Implementation code links

To see implementation code on IBM dataset click here.

To see implementation code on Apple dataset click here.

To see implementation code on Facebook dataset click here.

To see implementation code on Google dataset click here.

Chapter 3

Conclusion

From RMSE values and prediction plots of all four datasets, we think that ARIMA predicts better for a given parameter and it is predicting better if we fit the whole dataset into the model. But in actuality, It can not be said. Statistical models like ARIMA, Exponential smoothing are not training the model like DL, ML model. If we want to predict future days stock price (i.e. future data is not available) using ARIMA or Exponential Smoothing, It will not give better results as you can see the prediction plot in figures 2.5, 2.10, 2.15, and 2.20. Also, if we give a high time-step in LSTM and train data is sufficient enough, It can predict much more accurately. So from this, conclusion is that deep learning models (LSTM) can be improved by tuning the hyper-parameters and can predict efficiently compare to the stats model if we have large data to train.