Stock Price Prediction Using Time Series Data

Author
Mashtura Mazed
ID: 14201037

Supervisor
Dr. Mahabub Alam Majumdar
Chairperson
Department Of Computer Science and Engineering
BRAC University

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
August 2019

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mashtura Mazed
14201037
BRAC University

Approval

The thesis/project titled "Stock Price Prediction Using Time Series Data" submitted by

1. Mashtura Mazed (14201037)

Of Summer, 2015 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on August 28, 2019.

Examining Committee:

Supervisor & Head of The Department:

Dr. Mahabub Alam Majumdar
Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

Researchers has taken a lot of years to make algorithms fast and accurate enough to make stock price predictions accurately. Investors are looking for smarter techniques to forecast stock prices for investments and this has made this topic one of the most worked out researches in data science field. One of the trendy ways of forecasting is time series analysis. In this thesis, I have compared recent 3 most common time series forecasting algorithms that are- Autoregressive Integrated Moving Average, Facebook prophet and Long Short Term Memory, using company data (LMT and NOC) from yahoo finance. Firstly, I used K-Means clustering to choose a cluster with least number of companies and then used processed data to compare the accuracy of the algorithms.

Keywords: stock price, time series, ARIMA, LSTM, FB Prophet.

Dedication

I dedicate this work to my parents and my teachers....

Acknowledgement

My heartiest gratitude to the Almighty and my parents for enabling me to complete my undergraduate studies. I am thankful to my supervisor Mahbub sir for his teaching in Machine Learning course and giving me the flexibility to carry out my thesis without any pressure.This thesis would not have been possible without Iftekhar sir's guidance and I am grateful to him for his selfless help. Last but not the least, I thank my Brac University peers who helped me with information whenever I needed them.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ACF Auto-correlation Plot

AI Artificial Intelligence

$ARIMA$ Auto-regressive Integrated Moving Average

FBP Facebook Prophet

LMT Lockheed Martin Corporation

$LSTM$ Long Short term Memory

ML Machine learning

NLP Natural Language processing

NN Neural Network

NOC Northrop Grumman Corporation

$PACF$ Partial Auto-correlation Plot

$RMSE$ Root mean squared error

RNN Recurrent NN

SVM Support Vector Machine

TS Time series

TSA Time series analysis

Chapter 1

Introduction

1.1 Background Analysis

For many years, financial people as well as researchers have been working over profit maximization in stock business using historical data. Generally,they use fundamental analysis and technical analysis on this purpose. A third type of analysis, called sentiment analysis, that is also used to make profit out of people's sentiment or thought on a specific stock market and using vast social tweets and news headlines [1].

Time series analysis as forecasting technique is nothing very new, but it has now become a trend due to the presence of huge amount data both available online and offline. It can not only be used in forecasting price but also predicting future trend of consumerism.

To do a time series analysis on stock price, the first and the foremost important step is to fetch correct data. Firsthand data are always best for any analysis, but for this research purpose, I used data from yahoo finance. Next, to do forecasting, data needs to be generalized and outliers must be removed from the data. In time series techniques, data has to be stationary and so different stationary methods like- differentiation, logarithms, decomposition etc. techniques are used. Stock price data may have missing values of any date so null has to be removed to make data fit for TS algorithms.Finally algorithms has to be chosen that is fit for the data. Different machine learning algorithms like- clustering, regressions and deep learning i.e: Natural language processing, Neural Networks etc are mostly used to do stock price forecasting.

1.2 Motivation

Ever since the rise of machine learning and AI, future forecasting in financial field is a very demanding work as trillions of investments are done each day in every market around the world. Before, investors had been hiring financial people to do the task but now people use tools to do the basic analysis due to the existence of tons of data. Consequently, researchers have been thinking to come up with reliable

predictive models [1]. Financial solutions are a great business now and it will be in future. Since I am an AI and business enthusiast, I thought of learning different ML and NN algorithms to do financial analysis. As stock price data is the most available one in internet, I chose this topic for my undergrad thesis.

1.3 Objective

The foremost purpose of my research is to find a optimum and easy solution to forecast price. On this purpose I compared algorithms and came up with best and comparatively easy solution for forecasting and implementing on a system. Going through this research I also got a experience of working in a non major domain with my existing skills. Finally I came up with a hybrid system that can be easy to implement in modern systems.

1.4 Brief Methodology

To conduct the thesis, I first downloaded 29 random companies' closed price data from January 1 ,2017 to December 2018 using python's data-reader library and put then into clustering algorithm. For this section, I have used K-means clustering that is a very well known clustering algorithm and the python's k-means library is also rich. There are other clustering algorithms and libraries as well, but as I previously have the theoretical knowledge of this clustering technique, I chose it.After that I did some data processing as in TSA data needs to be stationarized if it is not. After that I used the data in every algorithm to make the research even in all terms. A flow chart of the whole process is given in Fig 1.1.

1.5 Thesis Outline

The thesis report has been structured in the following way:

Chapter 2 contains the literature review that is the previous works done on this topic.

Chapter 3 discusses about TSA that anyone has to do irrespective of the case specially with the data set.

Chapter 4 has the forecasting algorithms used in my thesis in an elaborate way.

Chapter 5 discusses implementation of the algorithms with data tables and graphs .

Chapter 6 contains the final analysis and the outcome of the research. It is wrapped up by conclusion and future work and APA citations

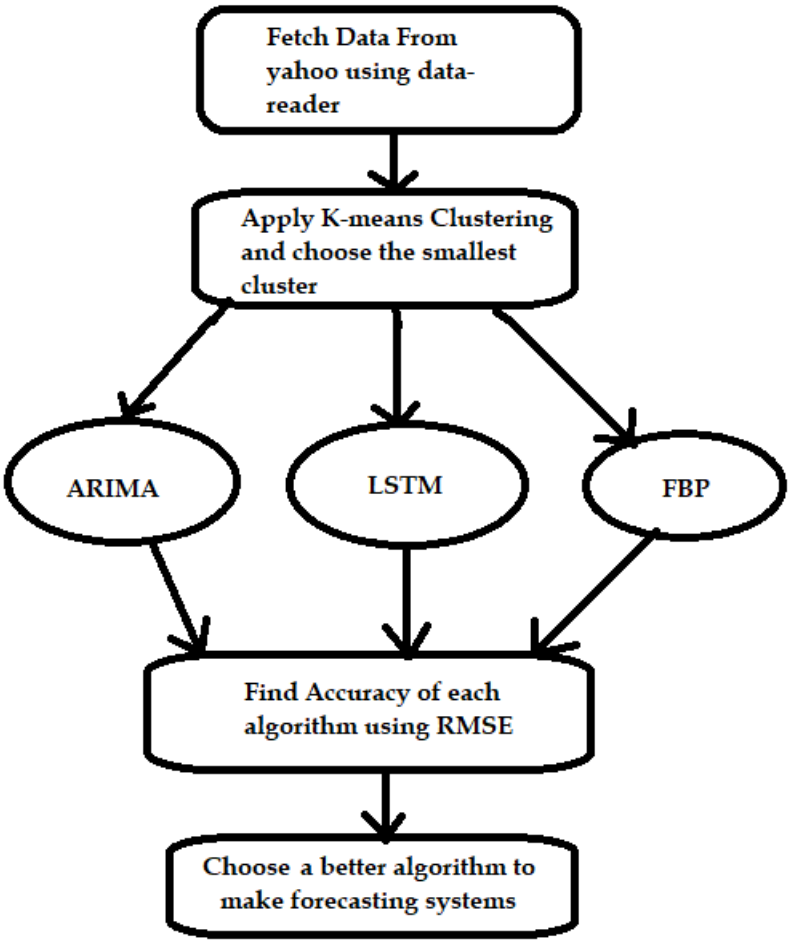

Figure 1.1: Work Flow

Chapter 2

Related Work

This chapter will discuss the previous work done on stock price forecasting using machine learning, deep learning and NN domain.

Stock analysis is a method for investors to determine buying and selling price using past and current data. There are two basic types of analysis and these are fundamental analysis and technical analysis. Technical analysis follows the trend as it forms through market action whereas fundamental analysis measures a security's intrinsic value by observing related economic and financial factors. Technical analysis says that the price moves in trend and history repeats itself. Machine learning (ML) and deep learning is uses the technical aspect of stock analysis [2]. There have been many algorithms used and optimized to such analysis. The most famous ones are – ARIMA, LSTM, KNN, CNN, NN, ANN, NLP, GRU, and SVM.

Phua, Ming and Lin have constructed forecasting by NNs using Singapore's stock market index showing a forecasting accuracy of 81% [3]. Pekkaya and Hamzacebi have conduct a comparative study on the results from using a linear regression in comparison to a a NN model. The objective of their research was to show that the NN gives much better results compared to the linear regression [4]. According to Wong, Selvi [5], the most benefit of NNs applications is in their ability to deal with fuzzy data. The only problem is NNs require huge number of data [6]. In some cases, like Yoon, Guimaraes,and Swales's paper where due to complex and unpredictable networks the estimated result become questionable.[7].

In [8] the forecasting was carried out on Google data and the result shows that there is a correlation in the weekly trend of stock prices and the news articles on the company.Sentiment analysis is used in this paper to the relation between business and news article.

Another technique used for prediction is Support Vector Machines (SVM). In a research, SVM was compared with random Walk, linear discriminant analysis, and quadratic discriminant analysis and Elman back propagation neural networks and SVM worked best in weekly prediction [9]. In another research, Kim compared SVM with NN and case-based reasoning where SVM performed better in forecasting the

daily change in the Korea composite stock price index (KOSPI) [10]. Amongst regression analysis ARIMA has done well and has become one of the most used one. In their research, Yilin, Du found that the prediction accuracy of ARIMA-BP neural network is better than the BP neural network, BP neural network is better than linear model ARIMA.[12]. In another paper [13], Jai and his group showed that ARIMA and Holt Winter are equally effective on short time TS data.Yang et al. used a margin-varying Support Vector Regression model and resulted in showing empirical results that have good predictive value for the Hang Seng Index [14].

Deep learning has been used in different markets due to the presence of huge amount of data in the market. In their paper Wei, Jian compared 6 algorithms and found that the performance of deep learning models MLP, RNN, LSTM is better than other ML models [15]. In another research, a hybrid deep learning models using LSTM and GRU has been used to predict stock price. [16]

In [17], stock market forecasting was done using technical analysis to improve the accuracy of the prediction for decision making and profit maximization. They classified the stocks into long-term and short-term investment categories. Adaptive stock market indicator selection was used to analyze past price data and predict future trends eectively. By doing stock market trading signal forecasting that is technical indicators were normalized and prepared for the forecasting.

From above analysis of previous work, we can see that Neural network domain is more accurate in predictions than other machine learning techniques in general.

Chapter 3

Time Series Analysis

3.1 Definition

TSA is a statistical method that analyzes time series data in order to extract meaningful observations on the characteristics of the data set. TSA are applied to real-valued, continuous data, discrete numerical data, or discrete symbolic data. Furthermore, TSA techniques can be parametric and non-parametric. The parametric approaches assume that the underlying stationary stochastic process has a certain structure.In this process, the task is to find the stochastic parameters. However, non-parametric methods explicitly estimate the co-variance or the spectrum of the process without the assumption of any particular structure. In this paper, I have used the former method.

3.2 Time Series Components

There are various reasons or the forces which affect the values of an observation in a time series are the components of a time series. The four categories of the components of time series are - seasonal, cyclical, trend and irregular movements. Seasonal and cyclical are short term movements of time series (fig: 3.1)

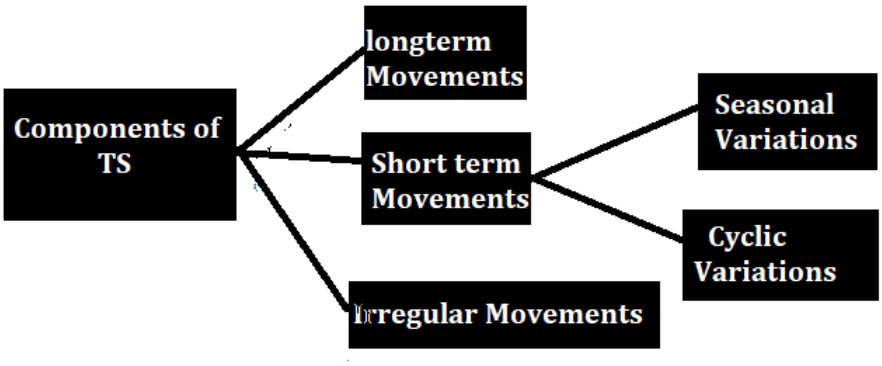

Figure 3.1: Components

Trend

The trend shows the tendency of the data to increase or decrease during a long period of time. Trend can fluctuate over time. But, overall it should be upward, downward or stable.

Linear and Non-Linear Trend:

The trend plot that goes around straight line is linear trend and otherwise non-linear. [Fig 3.2]

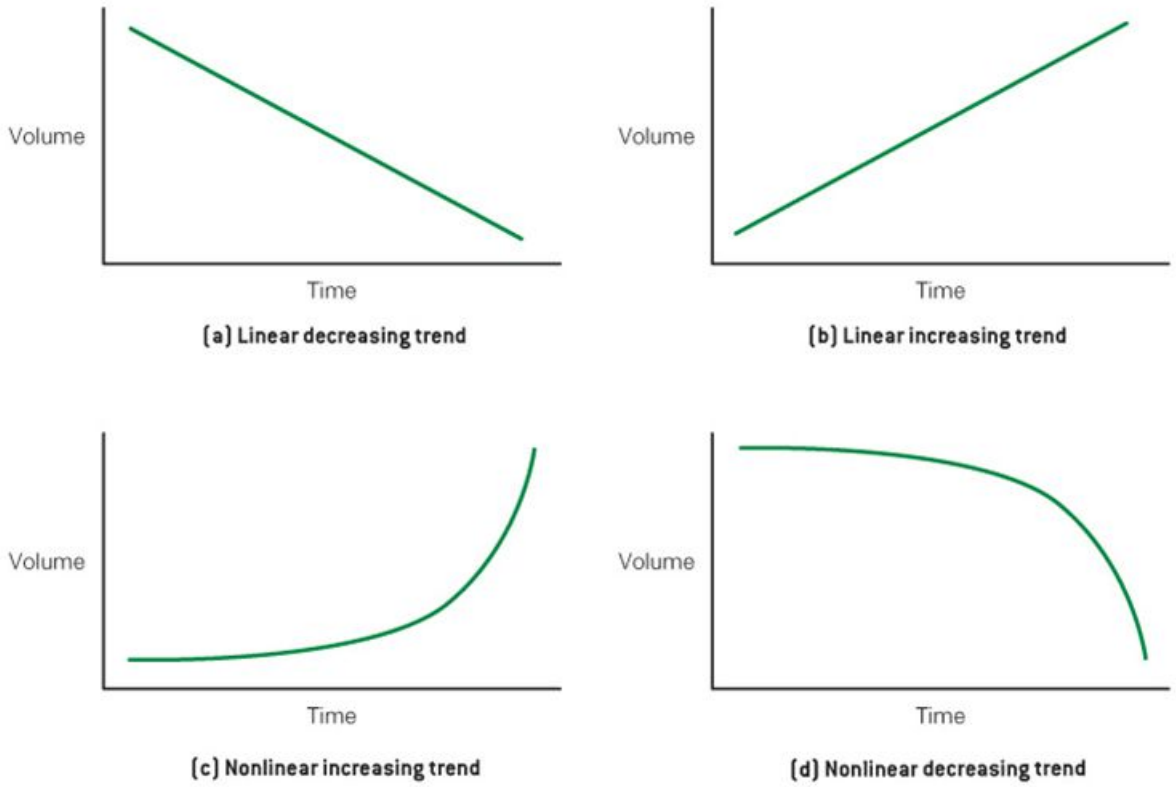

Figure 3.2: Trends

Seasonal and Cyclical variations:

Seasonal variations describes any regular variation with a period of less than one year. For example, traffic on roads in the morning and evening hours, sales during festivals, change in the number of passengers at weekends etc. by contrast, cyclical variations shows the variation with a period more than a year. It is a four-phase cycle: prosperity, recession, depression, and recovery. [19]. (Fig:3.3).

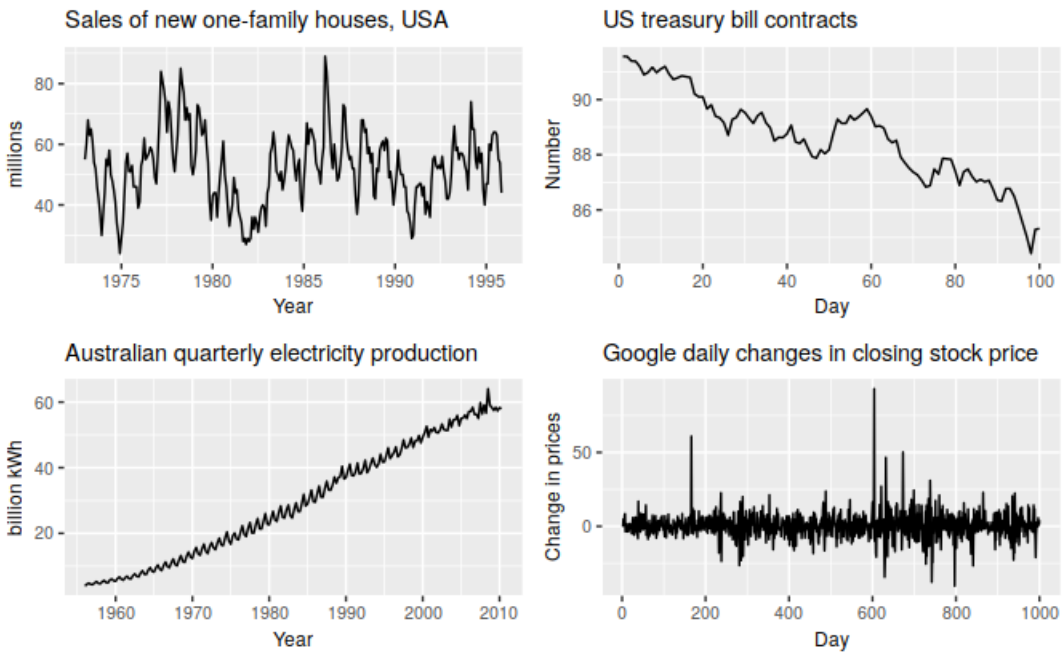

Figure 3.3: Seasons and Cycles

3.3 Processing Data for TSA:

a. Stationarity Test

Before choosing any forecasting model we have to make data prepare for time series analysis. To do this data has to go through stationarity test. Stationary data means that the variance and auto-correlation do not change over time [20].

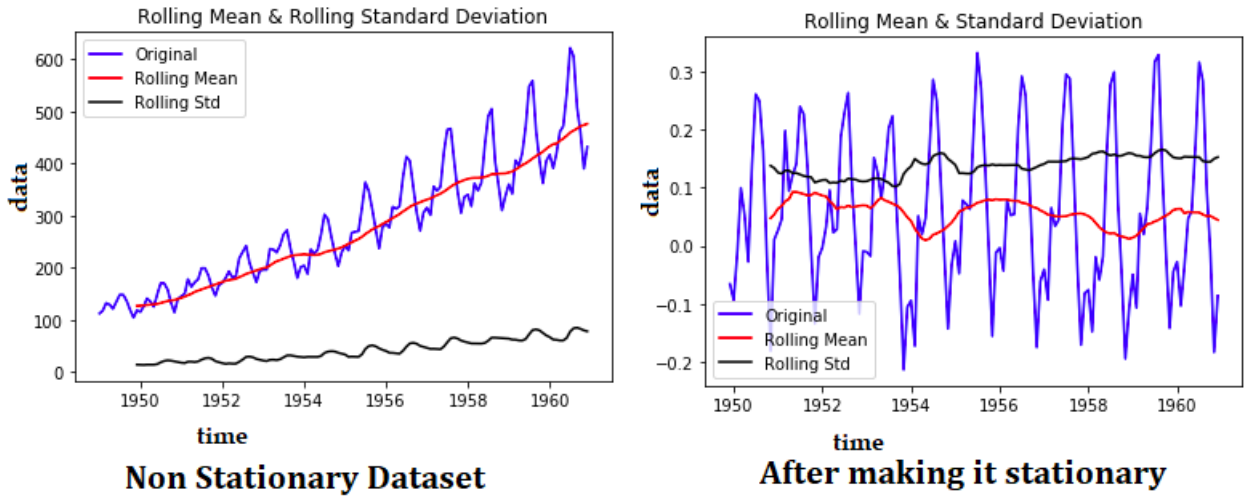

Figure 3.4: Stationarity of data

To test the stationarity of data we have to do the following:

Augmented Dicky-fuller test;

The Dickey Fuller test is the trendiest and used method of stationarity test. It is used to find the root of the time series data to check whether the data set passed the null hypothesis.[20] The null and alternative hypothesis of this test are:

Null Hypothesis: The series has a unit root (value of a =1)
Alternate Hypothesis: The series has no unit root.

A non-stationary data fails to reject the null hypothesis this means that the series can be linear or difference stationary.

Test for stationarity: If the test statistic (p) is less than the critical value, data is stationary. When p is greater than the critical value, data is non-stationary. Critical value for this test is .05 [20].

b. Making Data Stationary:

After doing stationarity test, if we found that data is stationary then it is alright to go for the forecasting models otherwise we have to make data stationary. In this research, I will be using ARIMA, FBP and LSTM models for forecasting. LSTM does not require stationarity whereas for ARIMA it is a must. Though for FBP making dataset stationary is optional according to documentation, I am going to use stationary dataset.
To do data stationary first one has to plot the real data and try to guess the properties of the data i.e.: random walk, linear trend, seasonality etc.

1. Differencing

Differencing helps stabilizing the mean of a time series by eliminating (or reducing) trend and seasonality. Also, looking at the time plot of the data, the ACF plot helps identifying non-stationary time series. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly. [21]

Pseudo code for differencing:
Box. Test (diff (goog200), lag=10, type="Ljung-Box")
- Box-Ljung test
- Data: diff (goog200)
- X-squared = 11, DF = 10, p-value = 0.4

Seasonal differencing:

A seasonal difference is the difference between an observation and the previous observation from the same season. The equation is–

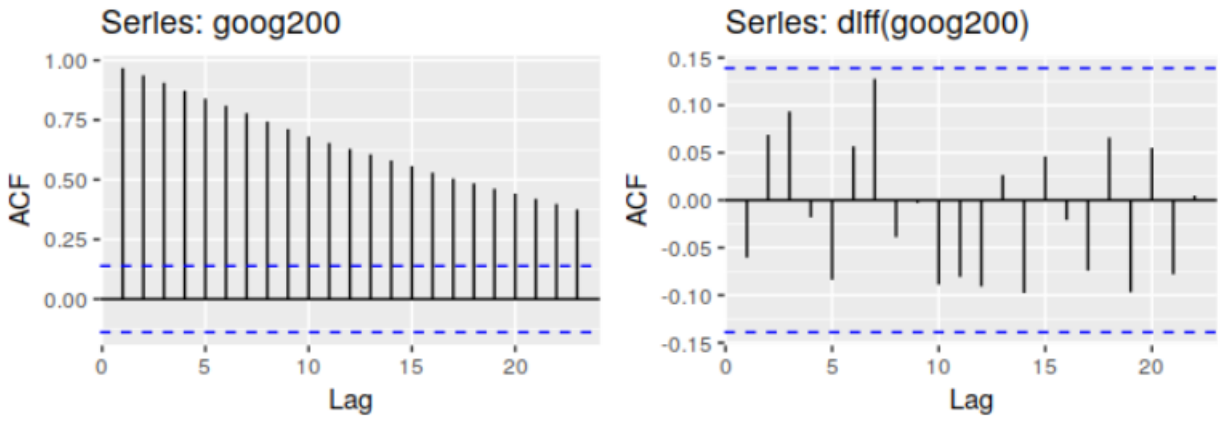

Figure 3.5: ACF plot before and after differencing

$$y'_t = y_t - y_{t-m},$$

Where m=the number of seasons. These are also called "lag-m differences".
If the data has a strong seasonality, seasonal differencing works well. But if there is less seasonality, it makes no difference after first differencing.

2. Log Transformation

Time series analysis uses log transformation to stabilize the variance of a series, it makes highly skewed distributions less skewed. It is typically used when properties are multiplicative related and data distribution is positive and highly skewed, for example, log transformation is used in series that are greater than zero and grows exponentially. Fig. 3.12 shows a plot of airlines passenger miles that has exponential growth and the variability of the series increases with time.
The following pseudo code computes the logarithms of the airline series:

Data lair;
Set sashelp.air;
Logair = log (air);
Run;

3. Decomposition

Time series data can exhibit a variety of patterns, and it is often helpful to split a time series into several components, each representing an underlying pattern category. Decomposing time series means thinking it in trend, seasonality and noise level. There are two types of decomposition additive and multiplicative. [23]

a. Additive Decomposition:

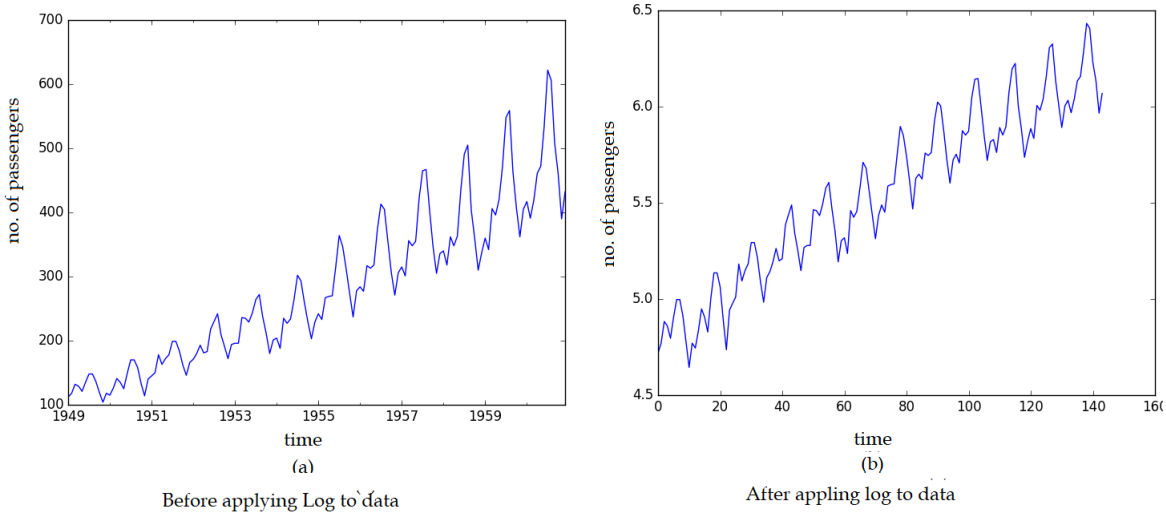

Figure 3.6: Airline Passengers miles

Additive decomposition has 4 steps:

Step 1: Let, m be an even number, for trend-cycle component T by 2m-MA. If m is an odd number, the trend-cycle component T_t uses an m-MA.

Step 2: For detrended series: y_t–T_t

Step 3: The seasonal component is computed by stringing together these monthly values, and replicating the sequence for each year of data. This gives S_t

Step 4: For the remainder component: $R_t = y_t$–T_t–S_t [23]

b. Multiplicative decomposition:

It is similar to additive model only difference is subtractions are replaced by division. [23]

Step 1: Let, m be an even number, for trend-cycle component T by 2m-MA. If m is an odd number,the trend-cycle component T_t is calculated using an m-MA.

Step 2: For detrended series: y_t/T_t

Step 3: The seasonal component is computed by stringing together these monthly values, and replicating the sequence for each year of data. This gives S_t

Step 4: For the remainder component: $R_t = y_t/(T_t * S_t)$

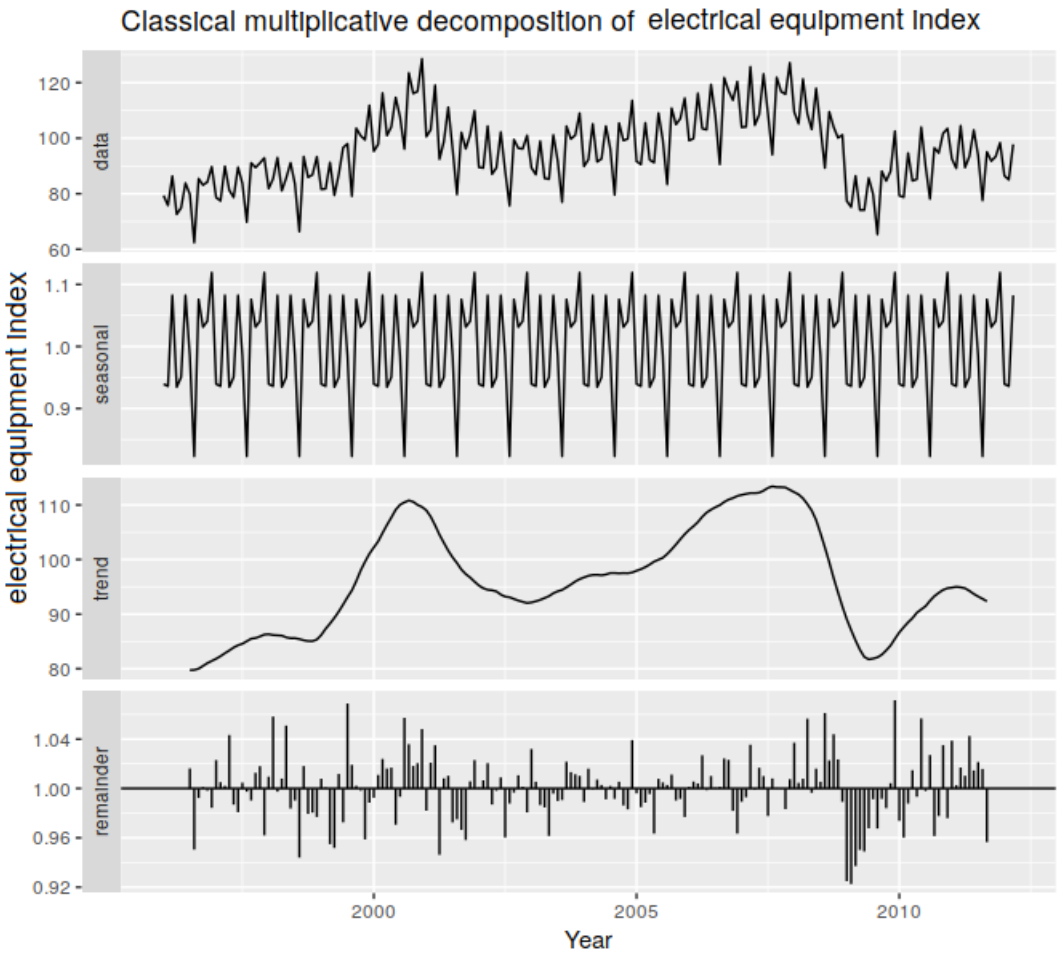

Figure 3.7: multiplicative decomposition for electrical equipment.

Chapter 4

Forecasting Algorithms

In this chapter I will discuss the algorithms used in my thesis with all terms and conditions. I will first discuss K-Means clustering , followed by ARIMA and LSTM and lastly FBP.

4.1 K-Means Clustering

Clustering is a very popular technique in ML field for feature extraction and grouping the data sets. K-Means cluster is one of the most common unsupervised algorithm in ML.[24] K means has K centroids that is used to define clusters. K-Means finds the best centroids by alternating between
(1) assigning data points to clusters based on the current centroids
(2) choosing centroids based on the current assignment of data points to clusters.
The algorithm is given below –

K-MEANS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
1 $(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
2 **for** $k \leftarrow 1$ **to** K
3 **do** $\vec{\mu}_k \leftarrow \vec{s}_k$
4 **while** stopping criterion has not been met
5 **do for** $k \leftarrow 1$ **to** K
6 **do** $\omega_k \leftarrow \{\}$
7 **for** $n \leftarrow 1$ **to** N
8 **do** $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
10 **for** $k \leftarrow 1$ **to** K
11 **do** $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

Figure 4.1: KMC Algorithm

Determining K

To make the algorithms more successful finding an optimum K is very important . To find it I have used elbow curve method.(Fig 4.2)

```
var sse = {};
for (var k = 1; k <= maxK; ++k) {
   sse[k] = 0;
   clusters = kmeans(dataset, k);
   clusters.forEach(function(cluster) {
      mean = clusterMean(cluster);
      cluster.forEach(function(datapoint) {
         sse[k] += Math.pow(datapoint - mean, 2);
      });
   });
}
```

Figure 4.2: Elbow curve Algorithm

Elbow curve method is to give KMC a range of k and find sum mean squared error over the clusters. Then plot the SSEs over K values and it will become like a elbow shape like fig 4.3. The value of K at the elbow point is the optimum k. According

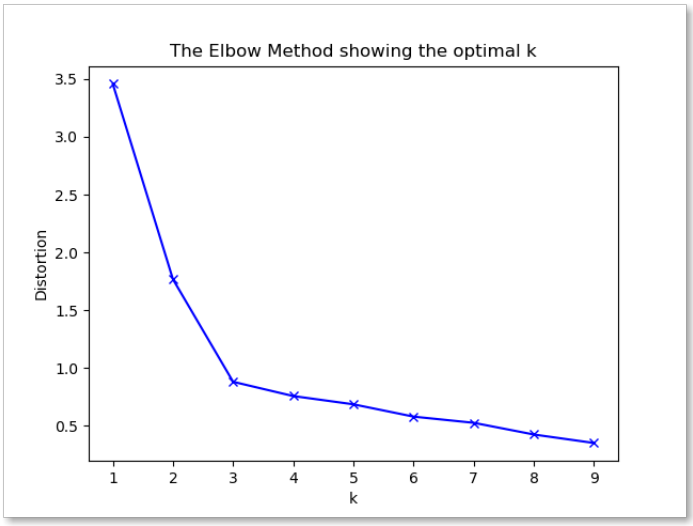

Figure 4.3: Elbow curve

to this graph the value of K is 3.

K-means can be visualized in different ways. I have used mesh graph in this paper from python's sciPy, numpy mesh grid library like fig 4.4.

4.2 ARIMA Model

Now, I am going to discuss about my first forecasting algorithm that is ARIMA. ARIMA is a regression model that is hugely used in TSA for forecasting prices and business digits.[24]. To use ARIMA we have to find it 3 parameters and those are p,q and d.

Finding Parameters:

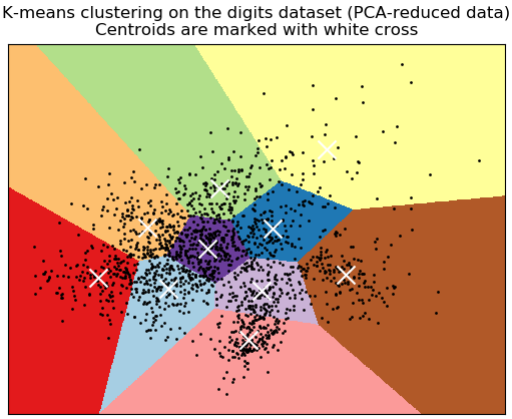

Figure 4.4: Mesh Grid

AR (p): The parameter p comes from AR part of the model that is auto-regressive. To find p we have to plot ACF graph on the stationary data and find the number of time the data point intersect the critical point. If p=0 then it means there is no auto correlation among the data.

MA(q): The q parameter comes from moving average. To find the q value we use PACF graph described in chapter 3.

I(d): d is simply the number of time we needed to difference the data to make it stationary. If data is linear the d is generally 0.

The flowchart of the model is given in fig 4.5.

4.3 LSTM

LSTM is the extension of RNN that can store memory for a long time. It is widely used in pattern recognition, forecasting , sentiment analysis etc. [25] LSTM has 3 gates input, forget and output. Input gate takes the input , forget gate deletes the memory if not necessary and output gate outputs the prediction. An illustration of LSTM is given in Fig 4.6.

4.4 Facebook Prophet

Facebook prophet is the library of Facebook inc. launched in 2017. It is based on additive regressive model that can do time series analysis and find weekly, yearly and daily trend. Though it is very new in the game, it has become popular among the researchers due to its sophisticated features and ease of use. It can be implemented in both python and R. Since it has started its journey , it is very hard to find out

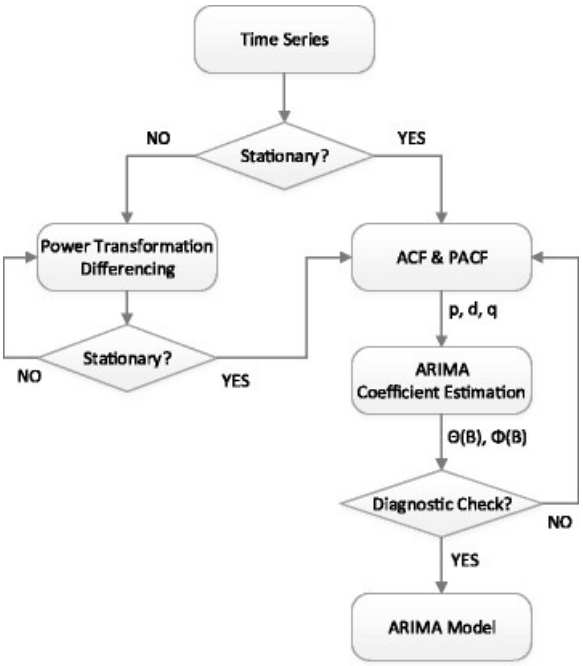

Figure 4.5: ARIMA Determining P, Q, D

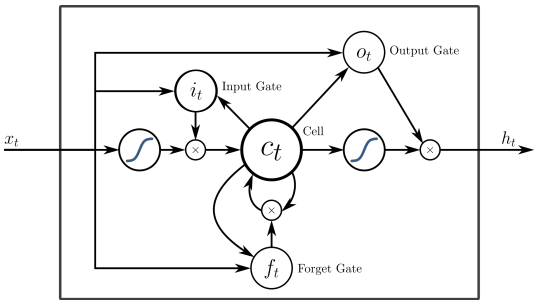

Figure 4.6: LSTM

its limitations and expertise. Fig 4.7 diagram illustrates the [26] forecasting process that have been found to work at a large scale.

4.5 Accuracy meter

I used RMSE to find the error in the algorithms' output. Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). It tells you how concentrated the data is around the line of best fit.

4.6 Data Set

I used yahoo finance to fetch all data used in this research. The data set comprises of open price, closed price, volume, change etc.I used only closed price for my research as it is the last price of a day.Fig 4.8 show the panel data information from the code.

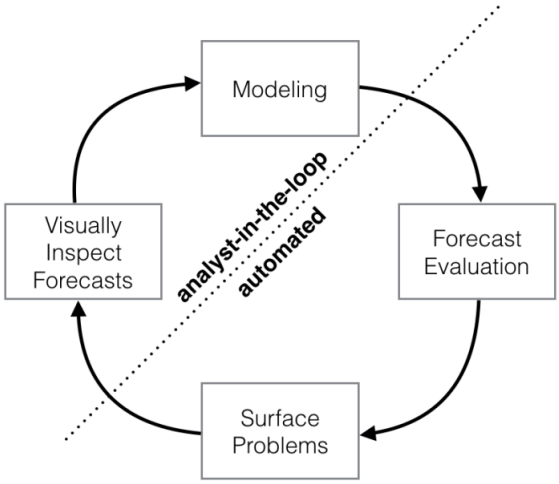

Figure 4.7: Facebook Prophet's Forecasting Process

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

```
[DatetimeIndex(['2017-01-03', '2017-01-04', '2017-01-05', '2017-01-06',
                '2017-01-09', '2017-01-10', '2017-01-11', '2017-01-12',
                '2017-01-13', '2017-01-17',
                ...
                '2018-12-17', '2018-12-18', '2018-12-19', '2018-12-20',
                '2018-12-21', '2018-12-24', '2018-12-26', '2018-12-27',
                '2018-12-28', '2018-12-31'],
               dtype='datetime64[ns]', name='Date', length=502, freq=None), MultiIndex(level
           codes=[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
           names=['Attributes', 'Symbols'])]
```

Figure 4.8: Panel data information

Chapter 5

Implementation

5.1 Programming Tools and IDEs

I have used different python libraries for difference algorithms. For K-means, I used sklearn K-mean library, for ARIMA, I used statsmodels.tsa.arima, for FBP it was prophet, for LSTM I used tensor flow. Lastly for all visualization purpose I used matplotlib library. I used Google's Colaboratory as Jupyter notebook to get rid of python's library installation in PC.

5.2 Clustering

The very first thing I did for the analysis is clustering . I used K-means clustering for the research and came up with 29 companies clustered in 5 clusters.

Data Normalization:

I used sklearn normalizer for data normalization to remove the out-liars in the data. The following is small snippet of my work.

```
[ ]
# Import libraries
from sklearn.pipeline import make_pipeline
from sklearn.cluster import KMeans
from sklearn.preprocessing import Normalizer

normalizer = Normalizer()

# Create 10 clusters
kmeans = KMeans(n_clusters = 10, max_iter = 1000)

# Make a pipeline to comine normalizer and KMeans
pipeline = make_pipeline(normalizer, kmeans)
```

Figure 5.1: Normalization

Finding Number of Clusters:

To find the optimum number of clusters I plotted elbow curve using sklearn's cluster library. [Fig : 5.2] From the curve we can see the curve has started to break near 4

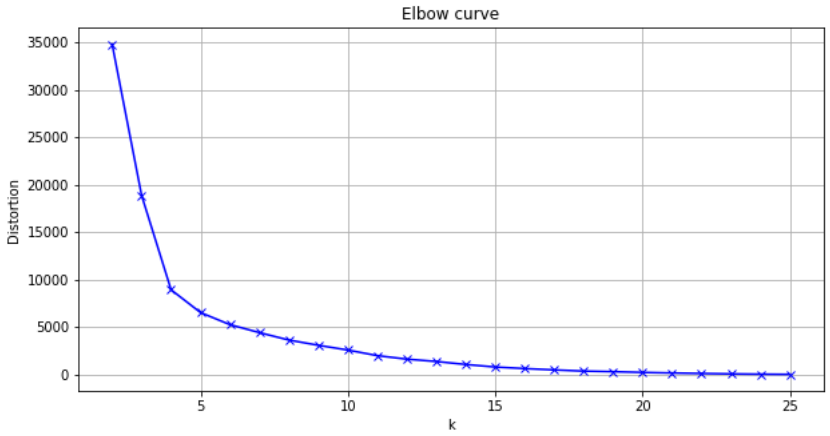

Figure 5.2: Elbow Curve

and had a full bend on 5. For my analysis's simplicity, I chose 5 as the number of clusters.

Cluster List:
After fitting 5 in the number of K-means functions , the final clusters as list is fig 5.3.

labels	companies
0	('IBM', 'IBM')
0	('American Equity Investment Life Holding Company', 'AEL')
0	('Toyota', 'TM')
0	('American Express', 'AXP')
0	('Bank of America', 'BAC')
0	('Navistar', 'NAV')
0	('Mitsubishi', 'MSBHY')
0	('Honda', 'HMC')
0	('Walgreen', 'WBA')
1	('Texas Instruments', 'TXN')
1	('Symantec', 'SYMC')
1	('Sony', 'SNE')
1	('Cisco Systems, Inc.', 'CSCO')
1	('Intel', 'INTC')
2	('Lockheed Martin', 'LMT')
2	('Northrop Grumman', 'NOC')
3	('General Electrics', 'GE')
3	('Chevron', 'CVX')
3	('Valero Energy', 'VLO')
3	('Exxon', 'XOM')
4	('Apple', 'AAPL')
4	('Microsoft', 'MSFT')
4	('Amazon', 'AMZN')
4	('Netflix, Inc.', 'NFLX')
4	('MasterCard', 'MA')

Figure 5.3: Company List

I used mesh graph to see the final clusters[Fig 5.4] . I chose the smallest cluster that is cluster 2 that comprises of LMT and NOC for my research.

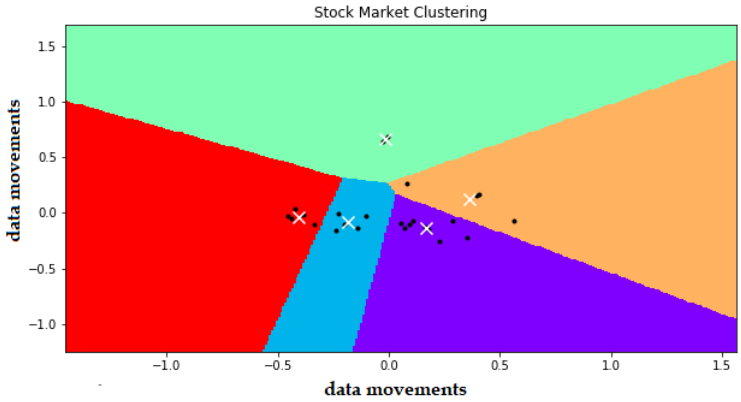

Figure 5.4: Clusters in Mesh

5.3 Implementing Forecasting Algorithms

1. ARIMA Model:

I will begin my forecasting with the most common one that is ARIMA model. Before doing any analysis , we should plot the raw data to try to understand it's type and methods that canbe applied to it. So, I plotted the raw data of both the companies and found the below–

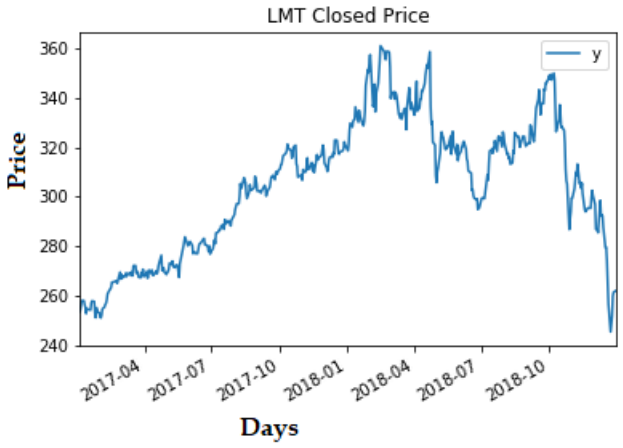

Figure 5.5: Original closed price of LMT stock

If we look into the data, the both the data sets are not linear as we have discussed in chapter 3. From here I assumed that it might need log differencing to make it

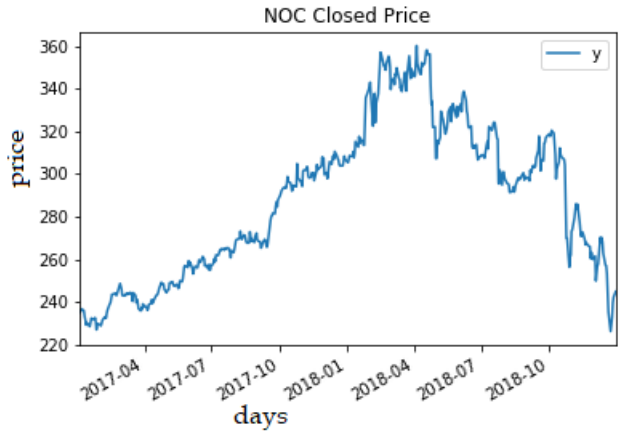

Figure 5.6: Original closed price of NOC stock

stationary.

Check Stationarity Of Data:

I plotted the rolling means to find the stationarity of the data set (both) and found fig 5.7 and fig 5.8.

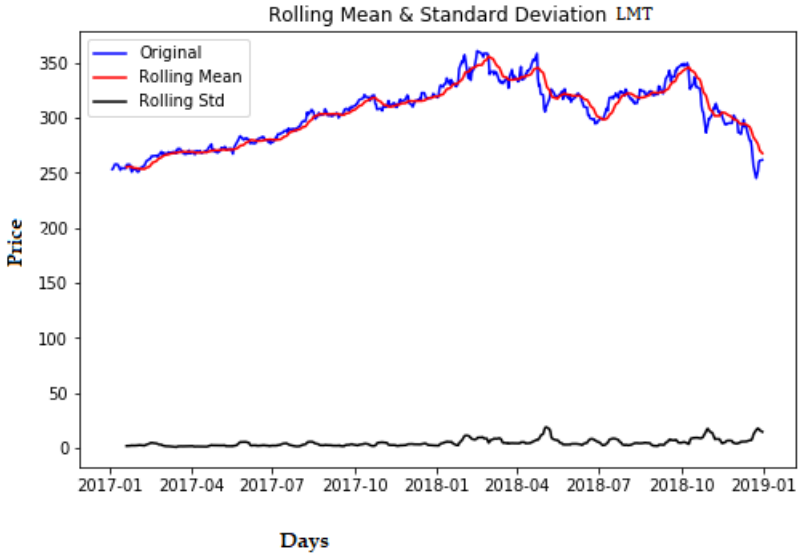

Figure 5.7: Rolling Means of LMT stock

As we can see the rolling statistics and STD is showing non stationary for both the data. Then I did ADF test to check the p value along with the other values and p was greater than critical value (.05) for both the companies.
Since both the test shows that data set is not stationary and the data has much noise, I did log difference on LMT first to test my assumptions and it came out right. (Fig 5.11).We can see p value became 0.0 for lmt, so the data became stationary.

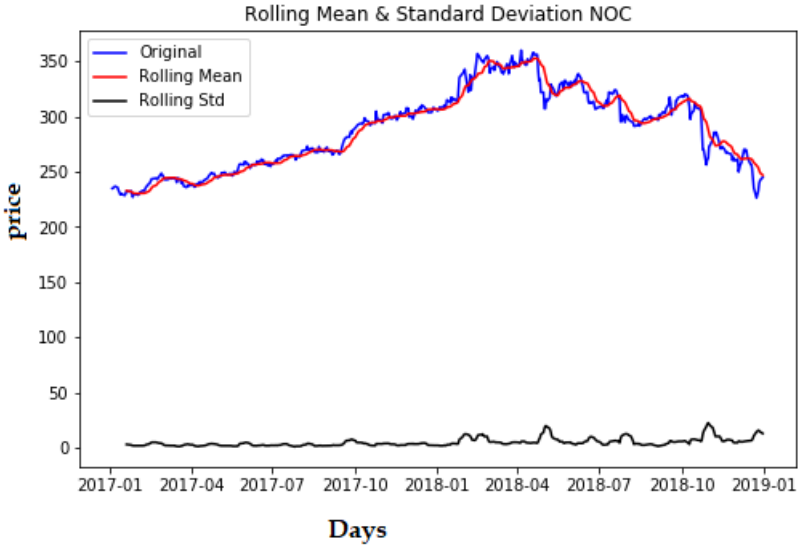

Figure 5.8: Rolling means of NOC stock

Results of Dickey-Fuller Test:
p-value = 0.393. The lmt is likely non-stationary.

Test Statistic	-1.775758
p-value	0.392562
#Lags Used	1.000000
Number of Observations Used	500.000000
Critical Value (1%)	-3.443496
Critical Value (5%)	-2.867338
Critical Value (10%)	-2.569858

dtype: float64

Figure 5.9: ADF test of LMT stock

Results of Dickey-Fuller Test:
p-value = 0.582. The NOC is likely non-stationary.

Test Statistic	-1.400410
p-value	0.582080
#Lags Used	0.000000
Number of Observations Used	501.000000
Critical Value (1%)	-3.443470
Critical Value (5%)	-2.867326
Critical Value (10%)	-2.569852

dtype: float64
After doing diff

Figure 5.10: ADF test of NOC stock

```
Results of Dickey-Fuller Test:
p-value = 0.000. The lmt is likely stationary.
Test Statistic                -20.771744
p-value                         0.000000
#Lags Used                      0.000000
Number of Observations Used   500.000000
Critical Value (1%)            -3.443496
Critical Value (5%)            -2.867338
Critical Value (10%)           -2.569858
dtype: float64
```

Figure 5.11: ADF test values after log-diff(LMT)

With this confidence, I did log differencing on NOC but it did not become stationary. So, I did only differencing as the raw data is not linear and found p=0.(Fig 5.12).

I also checked the rolling mean and STD for both the stationary data set and found

```
[ ]  Results of Dickey-Fuller Test:
     p-value = 0.000.  The NOC is likely stationary.
     Test Statistic                -21.216813
     p-value                         0.000000
     #Lags Used                      0.000000
     Number of Observations Used   500.000000
     Critical Value (1%)            -3.443496
     Critical Value (5%)            -2.867338
     Critical Value (10%)           -2.569858
     dtype: float64
```

Figure 5.12: ADF test values after Diff (NOC)

fig 5.13, fig 5.14.

Finding P, Q, D value:

After making data stationary, it is time to find the p, q, d value for ARIMA model. Since I did 1st differencing on both of the data-sets d value is 1 for both the sets. To find p and q, I plotted ACF and PACF for both of my data sets and found fig 5.15, 5.16.

And we can see the p value of LMT is quite clear and it is 1 but the q value is ambiguous. To reduce my ambiguity, I choose 0 and 1 as the value of q compared the RMSE value for both and ARIMA (1,1,0) gave better result and it was 100 % accurate.

Now let's come to NOC stock and there we can see it is also ambiguous to find the p,q value accurately from ACF and PACF plot. I did the same thing with NOC too and compared with different value of p and q . ARIMA (1,1,1) did the best result with accuracy of 95 %. The ARIMA information is given in fig 5.17 and fig 5.18.

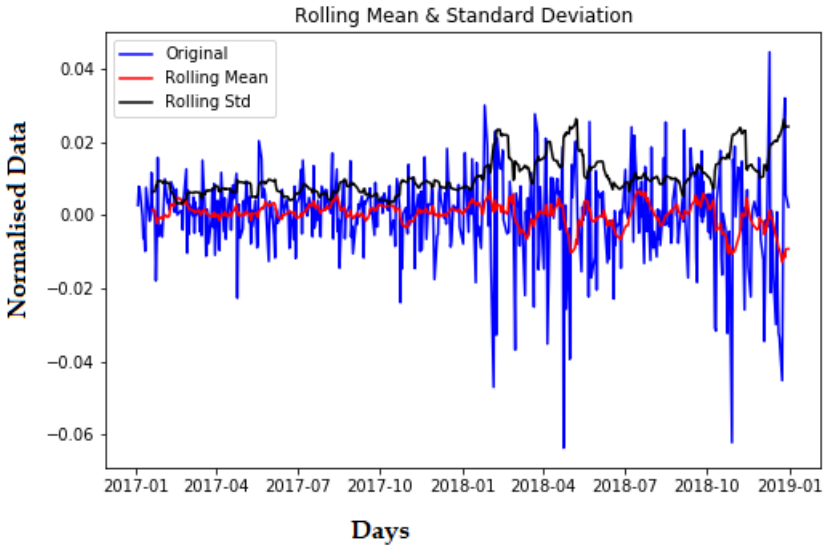

Figure 5.13: Rolling means of LMT Data after log.diff()

Plotting the final result:

I split the data into 2 parts that is testing and training and the portion was 4:1 that is 80% was on training and 20 % for testing. Using matplotlib I plotted the final predictions of both the data set. (Fig 5.19, Fig 5.20).

I will do the result analysis and the inferences on later chapter.For now, LMT prediction was more accurate on test data than NOC as we can see here.

2. FBP

For FBP, making data stationary is not necessary but as it might lead to ambiguity and confusion over the research , I have used the same data for FBP as well. I split the whole data set into 4:1 proportion like earlier.

Here I am going to do 30 days cycle as a the forecast input of FBP. The result for this implementation is in Fig:5.21 and Fig 5.22.
Here red dots are the predictions. The components that are weekly, mothly and yearly trend are in fig 23, fig 24.
Both LMT and NOC has similar components except daily cycle that is quite opposite to each other.

LSTM Model:

For LSTM as well, I used the same stationary data and split it into 4:1 proportion. LSTM's accuracy was well enough that is 95 % for both the companies.
I took epochs 1500 , batch size 1 and 4 neuron layer to do the prediction.Fig 5.25,5.26

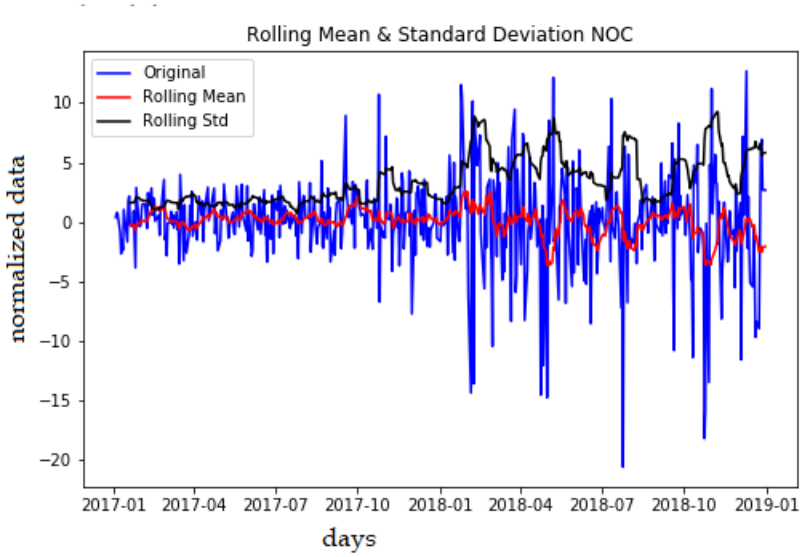

Figure 5.14: Rolling Means of NOC stock after diff()

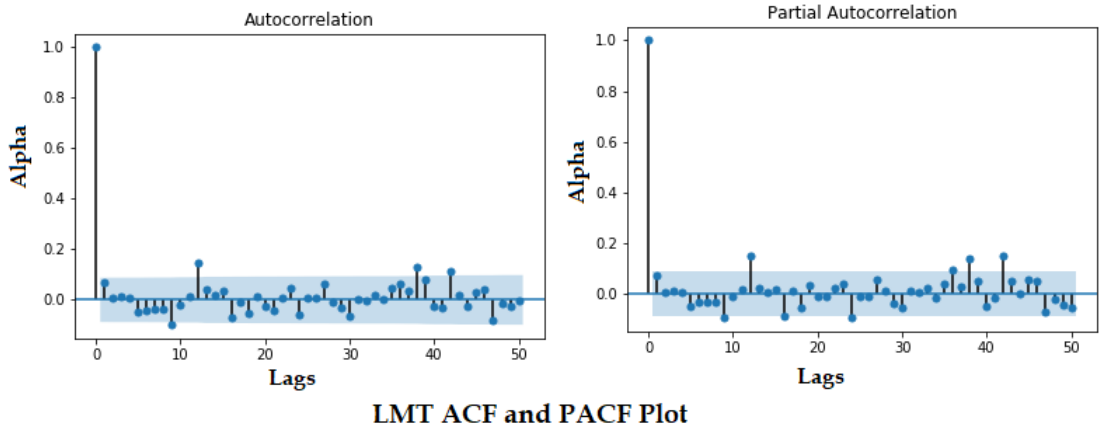

Figure 5.15: ACF and PACF of LMT Data

show the final result for both the companies.

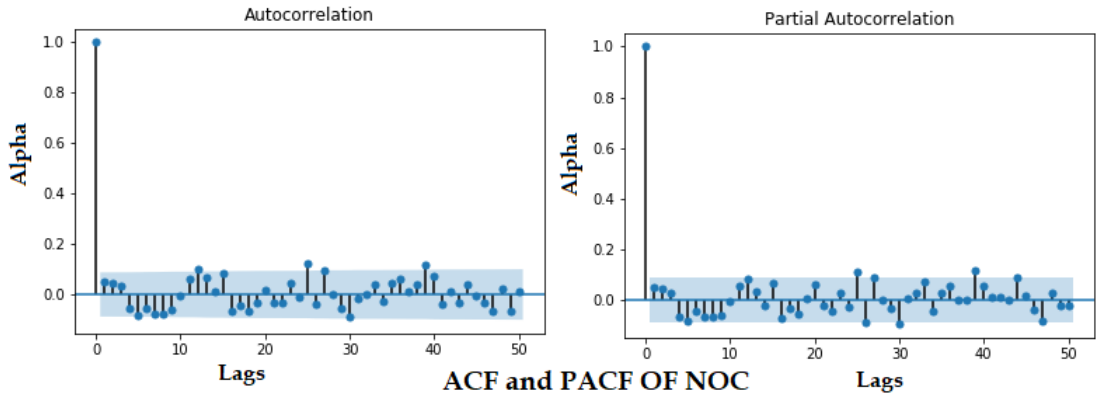

Figure 5.16: ACF and PACF of NOC stock

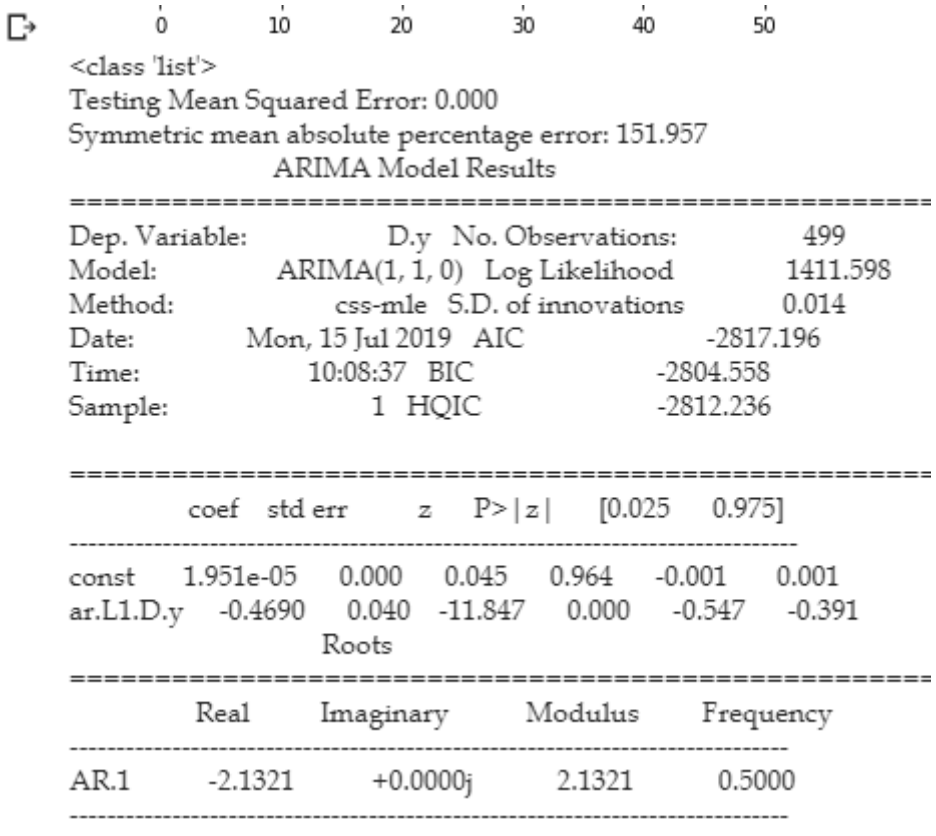

Figure 5.17: ARIMA of LMT stock

ARIMA Model Results

Dep. Variable:	D.y	No. Observations:	499
Model:	ARIMA(1, 1, 1)	Log Likelihood	-1399.517
Method:	css-mle	S.D. of innovations	3.973
Date:	Mon, 15 Jul 2019	AIC	2807.035
Time:	10:08:58	BIC	2823.885
Sample:	1	HQIC	2813.647

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0024	0.001	-1.846	0.065	-0.005	0.000
ar.L1.D.y	0.0439	0.045	0.980	0.327	-0.044	0.132
ma.L1.D.y	-1.0000	0.007	-151.548	0.000	-1.013	-0.987

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	22.7899	+0.0000j	22.7899	0.0000
MA.1	1.0000	+0.0000j	1.0000	0.0000

Figure 5.18: ARIMA of NOC stock

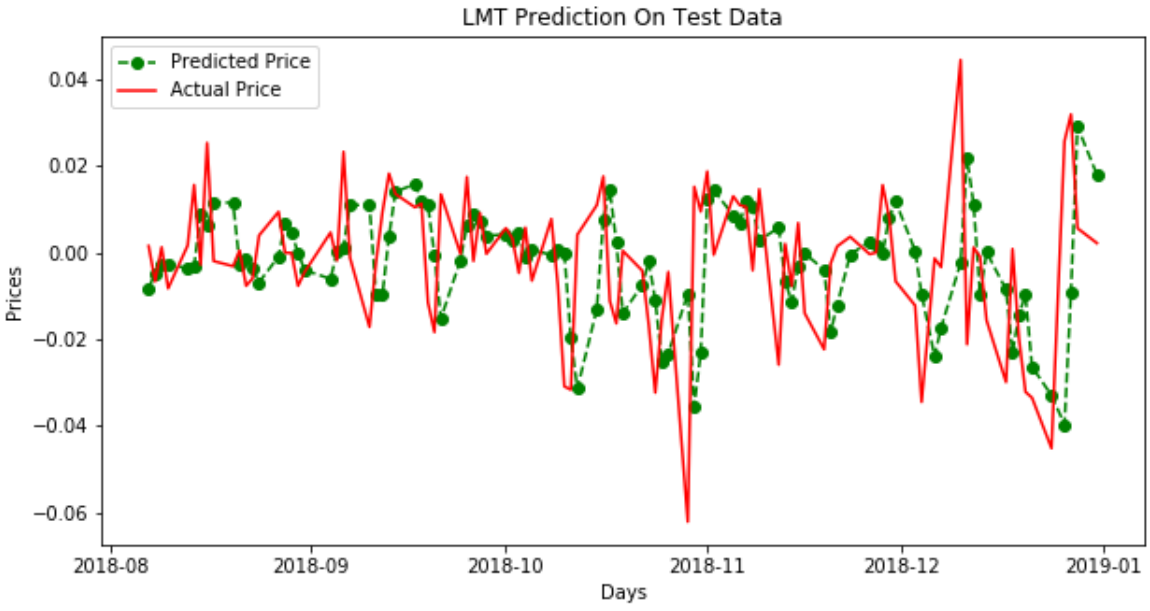

Figure 5.19: Prediction on test data of LMT

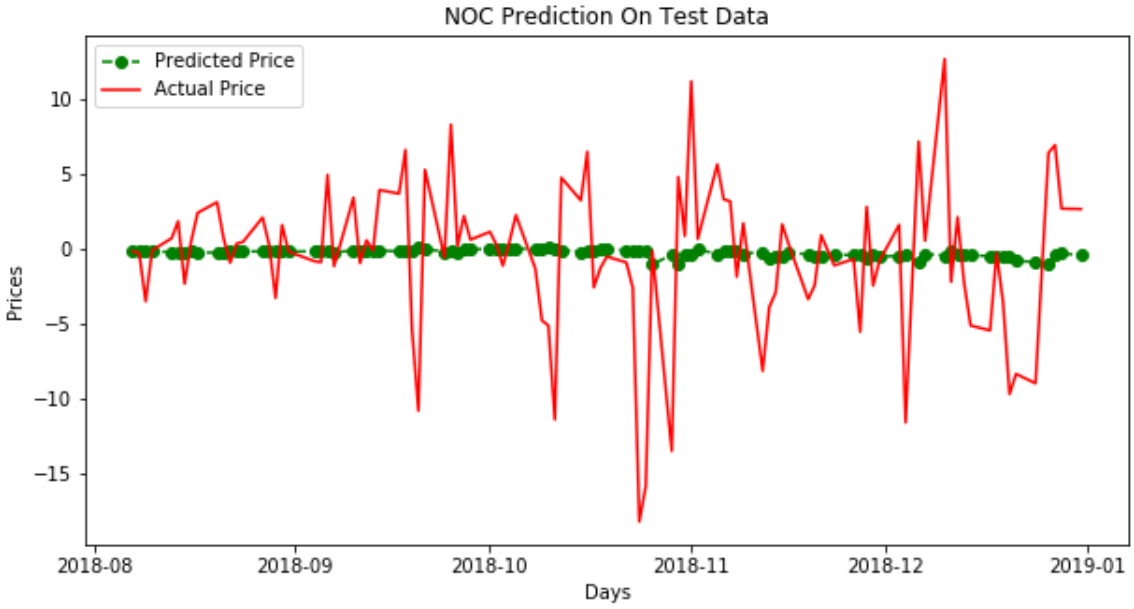

Figure 5.20: Prediction on test data of NOC

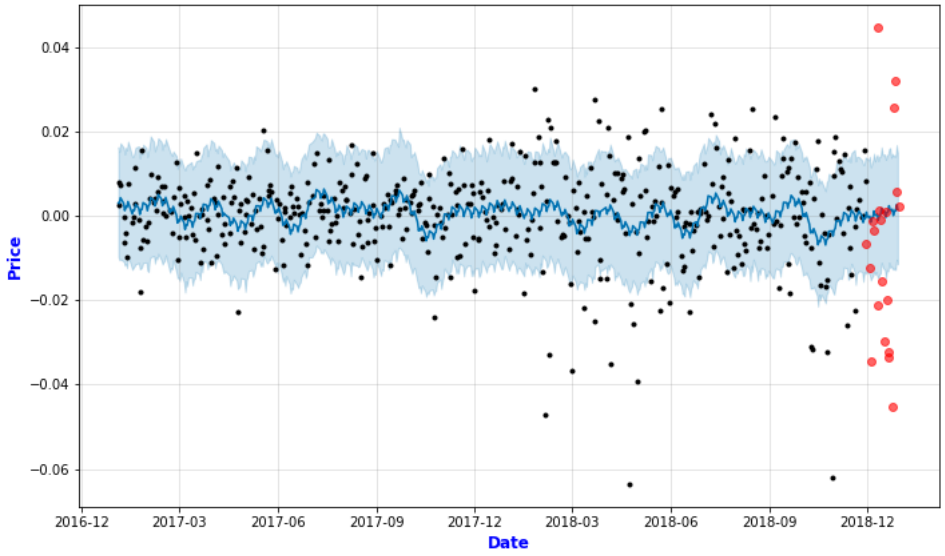

Figure 5.21: Prediction on LMT

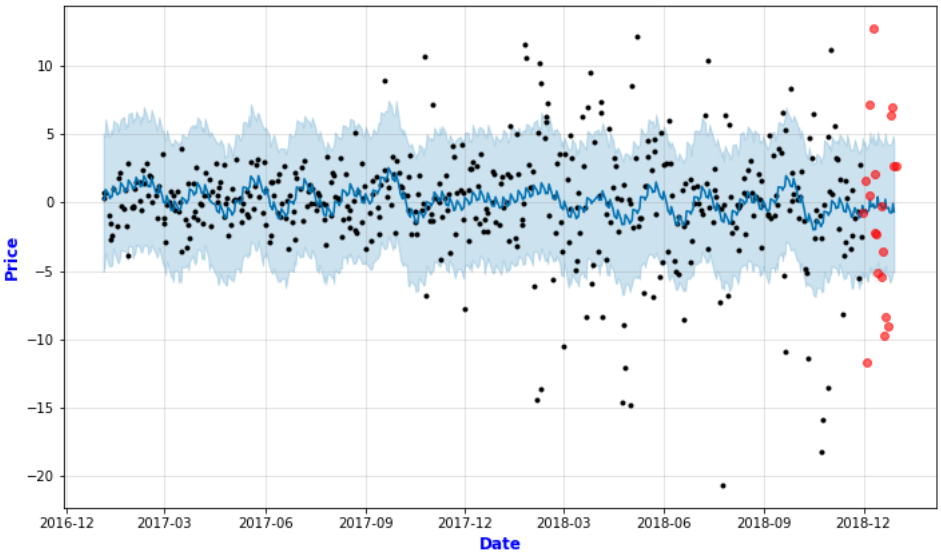

Figure 5.22: Prediction on NOC

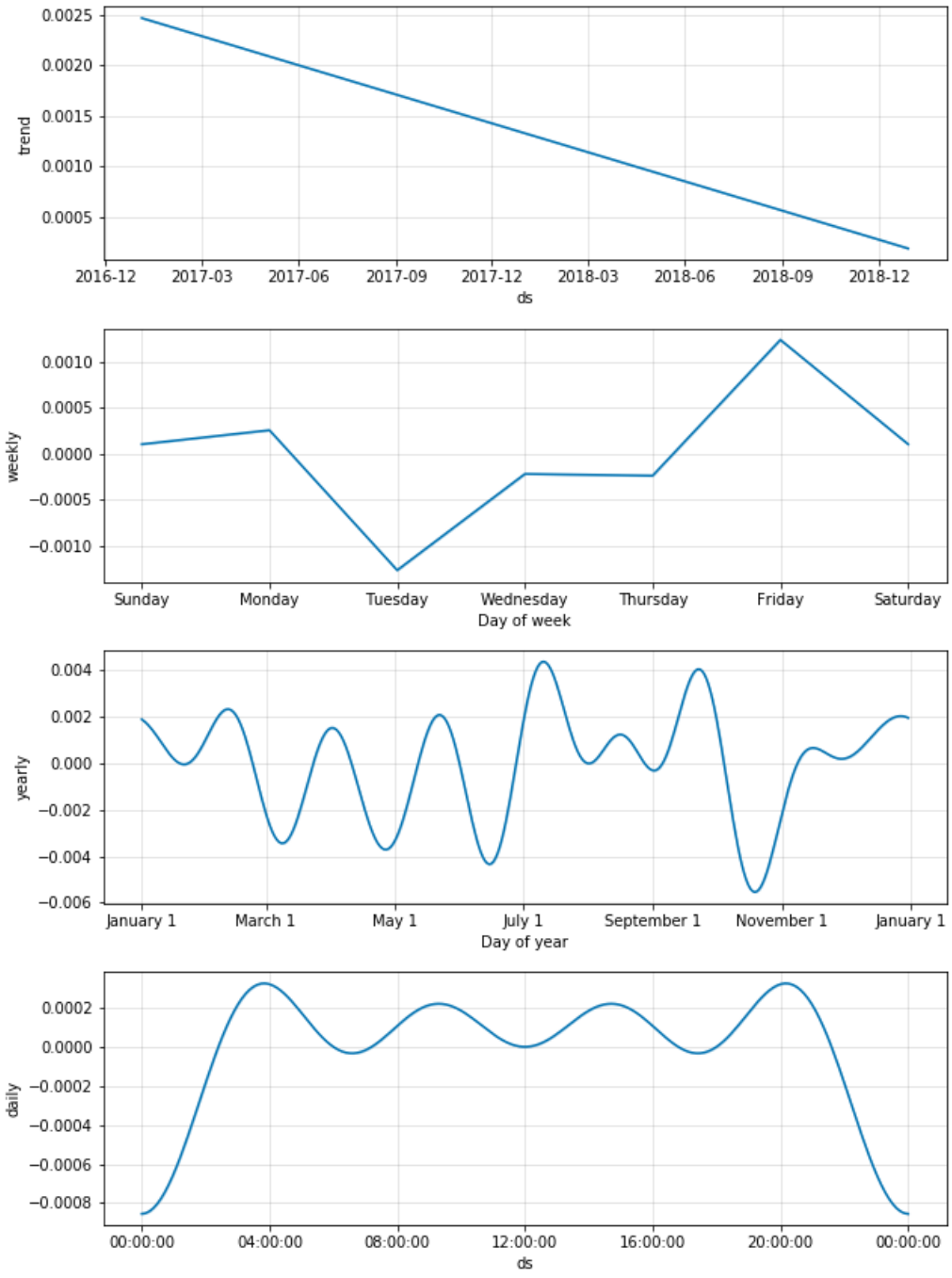

Figure 5.23: Components of LMT

Figure 5.24: Components for NOC

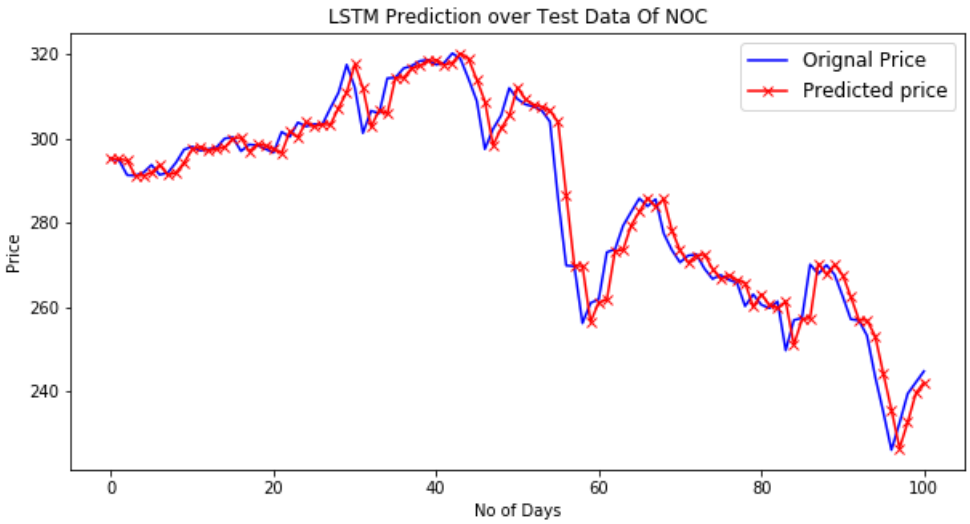

Figure 5.25: Results of NOC

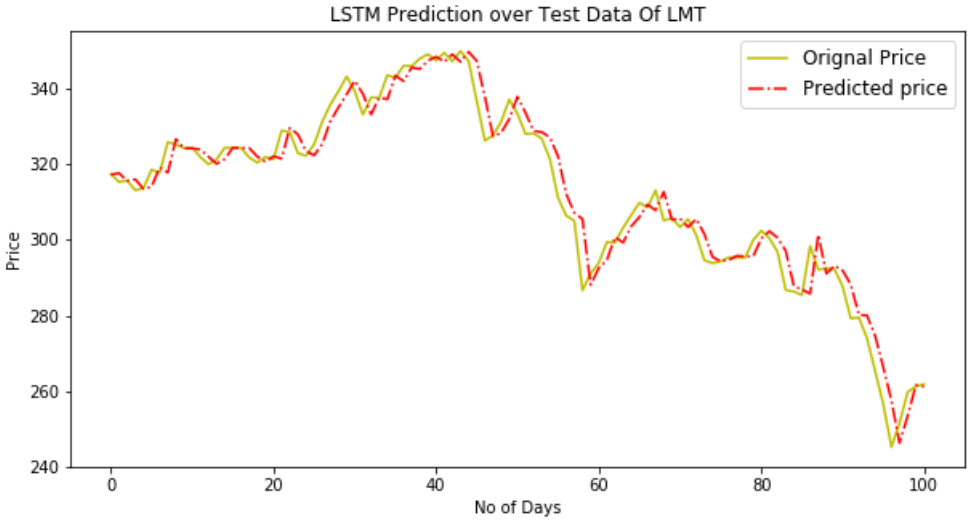

Figure 5.26: Result for LMT

Chapter 6

Result Analysis and Future Work

6.1 Comparison of the Algorithms

In this chapter I will show the comparative analysis of the algorithms based on their accuracy that is measured by RMSE value. Afterwards, I will discuss the future plan and overall experience of this research.

I used RMSE that is root mean square error to evaluate the accuracy of the used algorithms and those are listed below.

Algorithms	LMT [RMSE]	NOC [RMSE]
ARIMA	0.0%	5%
LSTM	5%	5%
FBP	0.01%	3.75%

Table 6.1: RMSE Value of different algorithms

From the above we can see FBP worked better on both the companies whereas ARIMA is the second best but LSTM also worked well. Though according to study RNN model should work better, since I used only 2 years of data that reasoned LSTM learn less and it could be the possible reason that LSTM became the least on both .

FBP is growing its popularity since the release of 2017. The only limitation to the model is, we do not know the whole algorithm as the documentation only says it is a additive linear model. Since, according to the documentation, data need not to be transformed, I assume that it has more than the linear model that can show such accuracy.
In this library, we can see other components like weekly, yearly and monthly cycle.It has methods that can show trend only by fitting the data in the prophet.

As we can see in fig. 6.1 and 6.2 both company has down trend of price with

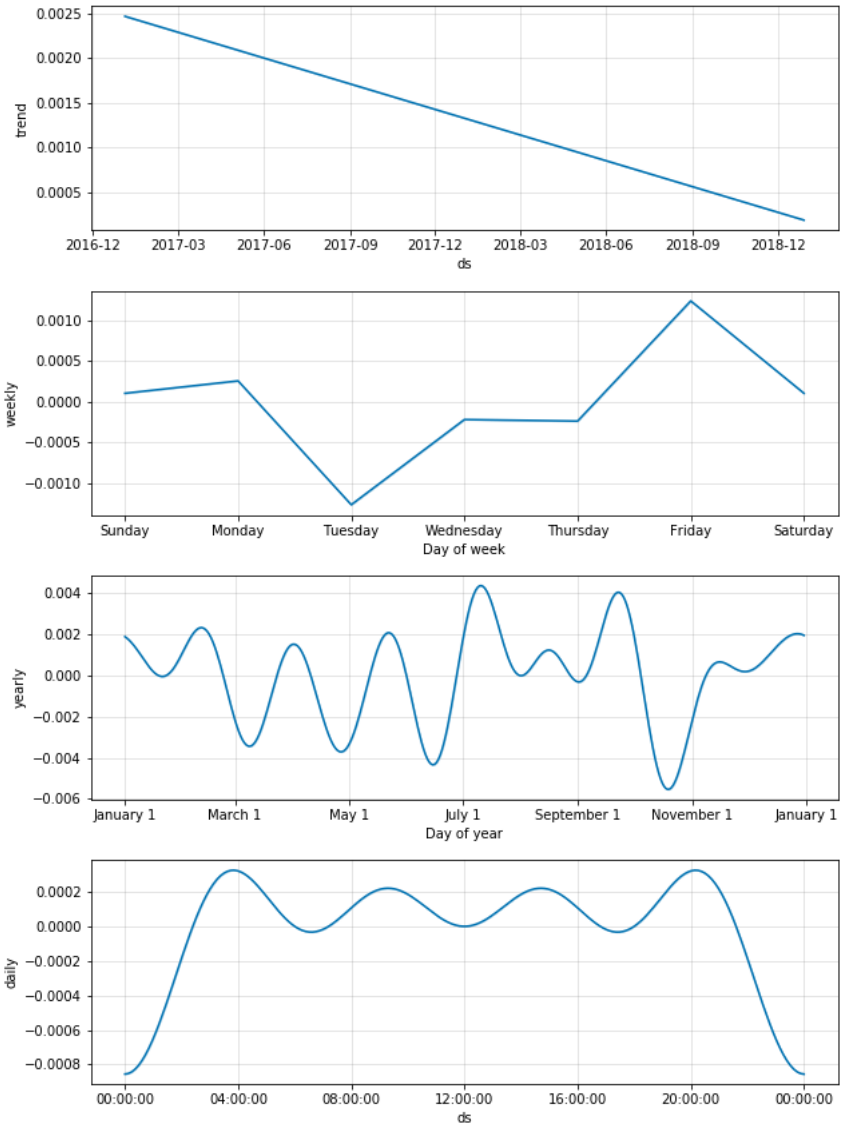

Figure 6.1: Components of LMT

months. On weekly cycle lmt price drops to negative on Tuesdays and starts rising from Thursdays and it reaches to highest price on Fridays.On the other hand NOC price stays low from Tuesdays to Thursdays and it reaches highest price on Fridays. From the graph we can say that for both the company Tuesdays are the right time to buy price and Fridays are the sell time. Yearly and daily cycles has fluctuations. If we think of a day, LMT price will start at a low price and may rise and NOC has the opposite behaviour.

Now, if I have to sort the algorithms according to simplicity/ease of use, I would say :

$$FBP > ARIMA > LSTM$$

I put LSTM at the end because the knowledge and research we have to go through before using RNN model because of their sophisticated theories are a lot harder

Figure 6.2: Components for NOC

than linear models. On the other hand, FBP has a very handy coding instructions in their documentations.

Furthermore, one other thing that was significant is, the two company has the same type of data set and the plots looked almost same. This could be the result of clustering.

In addition, though FBP documentation do not ask user to do data processing, if we do analysis with raw data then the accuracy of the forecasting technique lessens. For LSTM and ARIMA model, data processing is always done before using it for algorithm.

6.2 Proposed Model From Thesis Research

There are many financial services for price forecasting but they are very costly for local business. After learning these three techniques, I think FBP-LSTM hybrid model would be best for implementation. Though Facebook prophet needs less data processing other than LSTM but LSTM can be used as for text data like news paper headlines that is my future domain of research in stock price forecasting. I used FBP in a very basic way in my thesis but when I went through the full documentation, I came to know it can fill null values as and it has many more methods to play around with. So , after learning LSTM and FBP algorithms and their application in different domains I think it can be implemented in modern financial data analysis and forecasting systems.

6.3 Conclusion and Future Work

To sum up, I used three different algorithms to see their accuracy where FBP stood first in terms of ease and accuracy. Predicting stock price accurately is very hard but estimating the trend or price movement is less harder. Though RNN is a very powerful model, in case of less data going for regressive model would be a better way. Due to the vast availability of stock data in different stock price market , researchers are shifting towards deep learning models and it will be enriched in nearest future, hopefully.

For future research, I have thought of doing a combination of FBP and ARIMA as FBP is very handy to use and it has a very enriched library. Sentiment analysis is also a famous way of predicting future trends and NN models serve the best in this purpose. So, as a future project, I have thought of comparing FBP and ARIMA combination using numeric data to LSTM using newspaper headlines.

References

1. Box, G. E. and Tiao, G. C. (1975). Intervention analysis with applications to economic and environmental problems. Journal of the American Statistical Association, 70(349):70– 79.

2. Investopedia. (2019, June 24). Retrieved from https://www.investopedia.com/
3. P. Ming D. Phua and W. Lin, Neural network with genetically evolved algorithms for stocks prediction. Asia-Pacic Journal of Operational Research, 18:103–107, 2001.

4. E. H. Miller-Keith L. Sorensen and C. K. Ooi, The decision tree approach to stock selection. Journal of Portfolio Management, 27:42, 2000.

5. Wong, B. K., Bodnovich, T. A., and Selvi, Y. (1997). Neural network applications in business: A review and analysis of the literature (1988–1995). Decision Support Systems, 19(4):301–320.

6. Kryzanowski, L., Galler, M., and Wright, D. W. (1993). Using artificial neural networks to pick stocks. Financial Analysts Journal, 49(4):21–27

7. Yoon, Y., Guimaraes, T., and Swales, G. (1994). Integrating artificial neural networks with rule-based expert systems. Decision Support Systems, 11(5):497–507

8. A. C. H. Erol Egrioglu and U. Yolcu, Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks. Expert Systems with Applications,40:854–857,2013

9. J. Shah Shail Thakkar Priyank Patel and K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42:259–268, 2015.

10. K.-j.Kim, Financial time series forecasting using support vector machines. Neurocomputing, 55:307–319,2003

11. N. Yoshiteru Huang Wei and S.-Y. Wang, Forecasting stock market movement direction with support vector machine. Computers Operations Research, 32:2513–2522, 2005

12. Y. Du, "Application and analysis of forecasting stock price index based on combination of ARIMA model and BP neural network," 2018 Chinese Control And Decision Conference (CCDC), Shenyang, 2018, pp. 2854-2857.

13. J. Jagwani, M. Gupta, H. Sachdeva and A. Singhal, "Stock Price Forecasting Using Data from Yahoo Finance and Analyzing Seasonal and Nonseasonal Trend," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 462-467.

14. H. Chang Laiwan Yang and I. King, Support vector machine regression for volatile stock market prediction. IDEAL, 2017 2412:391–396

15.W. Li and J. Liao, "A comparative study on trend forecasting approach for stock price time series," 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, 2017, pp. 74-78.

16. Y. J. Chen, "Enhancement of stock market forecasting using a technical analysis based approach", 2014 IEEE 5th International Conference on Software Engineering and Service Science, Beijing, vol. 1, no. 4, pp. 702–705, 2014. [Online].

17. Fattah, Jamal Ezzine, Latifa Aman, Zineb El Moussami, Haj Lachhab, Abdeslam. (2018). Forecasting of demand using ARIMA model. International Journal of Engineering Business Management.

18. "ARIMA Model Python Example - Time Series Forecasting." Medium, Towards Data Science, 21 July 2019, towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7.

19. Forecasting: Principles and Practice. (2019, June 21) .

20. How to Check if Time Series Data is Stationary with Python. (2019, June 24). Retrieved from https://machinelearningmastery.com/time-series-data-stationary-python/

21. Forecasting: Principles and Practice." Chapter 6 Time Series Decomposition, otexts.com/fpp2/decomposition.html.2018

22. "A Demo of K-Means Clustering on the Handwritten Digits Data" Scikit,scikit-learn.org/stable/auto-examples/cluster/plotkmeansdigits.htmlsphx-glr-auto-examples-cluster-plot-kmeans-digits-py.(2019, June 24).

23. Hartigan, J. A., Wong, M. A. (1979). Algorithm AS 136 , A k-means clustering algorithm. Journal of the Royal Statistical Society.

24. A. A. Adebiyi, A. O. Adewumi and C. K. Ayo. Stock Price Prediction Using the ARIMA Model,in Proceedings of the 2014 UKSIM-AMSS 16th International Conference on Computer Modeling and Simulation, 2014.

25. M. A. Dempster, T. W. Payne, Y. Romahi, and G. W. Thompson, Computational learning techniques for intraday fx trading using popular technical indica-

tors,IEEE Transactions on neural networks, pp. 6-8, 2014

26. Prophet: Forecasting at scale. (2018, September 13).