STOCK PRICE PREDICTION

Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

ANUJ MEHTA-40415002716

To

Maharaja Surajmal Insitute of Technology
Affiliated to Guru Gobind Singh Indraprastha University
Janakpuri, New Delhi-58

June-July 2018

ACKNOWLEDGEMENT

We have taken efforts in this project. However it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Mr. Piyush Jain (Project Engineer and Course Coordinator) for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

We would like to express our gratitude towards our parents and member of CDAC for their kind cooperation and encouragement which help us in completion of this project.

We would like to express our special gratitude and thanks to institute person for giving us such attention and time.

Our thanks and appreciation also goes to our fellows in developing in the project and people who have willingly helped us out with their abilities.

ANUJ MEHTA

(40415002716)

CANDIDATE'S DECLARATION

It is hereby certified that the work which is being presented in the B. Tech summer internship Project Report entitled "Stock Price Prediction using ARMA model" in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology and submitted in the Department of COMPUTER SCIENCE & ENGINEERING department of MAHARAJA SURAJMAL INSTITUTE OF TECNOLOGY, New Delhi (Affiliated to Guru Gobind Singh Indraprastha University, Delhi) is an authentic record of my own work carried out during a period from June 2018 to July 2018 under the guidance of project mentor **Mr. Piyush Jain**.
The matter presented in the B. Tech summer internship Project Report has not been submitted by me for the award of any other degree of this or any other Institute.

ANUJ MEHTA
(40415002716)

PREFACE

The world is shrinking. New era is going to be an era of great technology and only those Engineers who will have the ability to move along with fast paced technology are going to survive. To move along with technology an Engineer must be equipped with the practical knowledge of work. Now the quality of knowledge is more important than the quantity. So a person without practical knowledge is nil no matter how many books he has studied.

The practical training is highly conducive for the development of

- Solid foundation of knowledge and personality
- Confidence
- Excellence and Self-discipline

It was our pleasure that we got an opportunity to undergo our six weeks industrial training in C-DAC, MOHALI.

In this report, we have tried to sum up the technical knowledge that we have gained during this precious training period.

ABSTRACT

Stock price forecasting is a popular and important topic in financial and academic studies. Share Market is an untidy place for predicting since there are no significant rules to estimate or predict the price of share in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis, etc. are all used to attempt to predict the price in the share market but none of these methods are proved as a consistently acceptable prediction tool.

In this project we attempt to implement an AutoRegressive Integrated Moving Average approach to predict stock market prices. AutoRegressive Integrated Moving Average (ARIMA) are very effectively implemented in forecasting stock prices, returns, and stock modeling, and the most frequent methodology is the Backpropagation algorithm. This project is for all the users as the prediction can be done on any listed companies in stock market. We outline the design of the ARIMA model with its salient features and customizable parameters. We select all the previous share price of a company in the past 25 years as our data. With the help of statistical analysis, the relation between the previous stock price and next stock price is formulated which can help in forecasting accurate results. Although, share market can never be predicted, due to its vague domain, this project aims at applying ARIMA model in forecasting the stock prices.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

Stock Market prediction and analysis is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. Stock market is the important part of economy of the country and plays a vital role in the growth of the industry and commerce of the country that eventually affects the economy of the country. Both investors and industry are involved in stock market and wants to know whether some stock will rise or fall over certain period of time. The stock market is the primary source for any company to raise funds for business expansions. It is based on the concept of demand and supply. If the demand for a company's stock is higher, then the company share price increases and if the demand for company's stock is low then the company share price decrease.

Another motivation for research in this field is that it possesses many theoretical and experimental challenges. The most important of these is the Efficient Market Hypothesis (EMH), the hypothesis says that in an efficient market, stock market prices fully reflect available information about the market and its constituents and thus any opportunity of earning excess profit ceases to exist. One of the example of big exchange is New York Stock Exchange.

The Nasdaq Stock, also known simply as NASDAQ is an American stock exchange. It is the second-largest exchange in the world by market capitalization, behind only the New York Stock Exchange located in the same city. The exchange platform is owned by NASDAQ, Inc. which also owns the Nasdaq Nordic (formerly known as OMX) and Nasdaq Baltic stock market network and several U.S. stock and options exchanges

Microsoft, Apple, Alphabet (Google), Facebook and Amazon are the some of the largest US Companies by market capitalization. The NASDAQ (National Association of Securities Dealers Automated Quotations) is an electronic stock exchange with more than 3,300 company listings. It currently has a greater trading volume than any other U.S. stock exchange, carrying out approximately 1.8 billion trades per day.

Due to involvement of many number of industries and companies, it contain very large sets of data from which it is difficult to extract information and analyze their trend of work manually. Stock market analysis and prediction will reveal the market

patterns and predict the time to purchase stock. The successful prediction of a stock's future price could yield significant profit. This is done using large historic market data to represent varying conditions and confirming that the time series patterns have statistically significant predictive power for high probability of profitable trades and high profitable returns for the competitive business investment.

1.1. Statement of the problem

Stock market is very vast and difficult to understand. It is considered too uncertain to be predictable due to huge fluctuation of the market. Stock market prediction task is interesting as well as divides researchers and academics into two groups, those who believe that we can devise mechanisms to predict the market and those who believe that the market is efficient and whenever new information comes up the market absorbs it by correcting itself, thus there is no space for prediction.

Investing in a good stock but at a bad time can have disastrous result, while investing in a stock at the right time can bear profits. Financial investors of today are facing this problem of trading as they do not properly understand as to which stocks to buy or which stocks to sell in order to get optimum result. So, the purposed project will reduce the problem with suitable accuracy faced in such real time scenario.

1.2. Objectives

The aims of this project are as follows:

1) To identify factors affecting share market

2) To generate the pattern from large set of data of stock market for prediction of stock price

3) To predict an approximate value of share price

4) To provide analysis for users through web application

The project will be useful for investors to invest in stock market based on the various factors. The project target is to create web application that analyses previous stock data of companies and implement these values in data mining algorithm to determine the value that particular stock will have in near future with suitable accuracy. These predicted and analyzed data can be observed by individual to know the financial

status of companies and their comparisons. Company and industry can use it to breakdown their limitation and enhance their stock value. It can be very useful to even researchers, stock brokers, market makers, government and general people.

The main feature of this project is to generate an approximate forecasting output and create a general idea of future values based on the previous data by generating a pattern. The scope of this project does not exceed more than a generalized suggestion tool.

1.3. System Overview

This system named "Stock Market Analysis and Prediction using Artificial Neural Networks" is a web application that aims to predict stock market value using Artificial Neural Network. This project is intended to solve the economic dilemma created in individuals that wants to invest in Stock Market.

1.4. System Features

1.4.1. Stock market prediction

Stock price movements are in somewhat repetitive in nature in the time series of stock values. The prediction feature of this system tries to predict the stock return in the time series value by training Neural Network which involves producing an output and correcting the error.

1.4.2. Market Analysis

A detailed analysis of Stock market is presented to the user. The analysis contains the performance of most of the listed companies for certain interval of days. The numbers and figures are represented in graphs and plots in the form of line charts.

2. LITERATURE REVIEW

A time series is a set of well-defined data items collected at successive points at uniform time intervals. Time series analysis is an important part in statistics, which analyzes data set to study the characteristics of the data and helps in predicting future values of the series based on the characteristics. Forecasting is important in fields like finance, industry, etc. Autoregressive and Moving Average (ARMA) model is an important method to study time series. The concept of autoregressive (AR) and moving average (MA) models was formulated by the works of Yule, Slutsky, Walker and Yaglom. Autoregressive Integrated Moving Average (ARIMA) is based on ARMA Model. The difference is that ARIMA Model converts a non-stationary data to a stationary data before working on it. ARIMA model is widely used to predict linear time series data. The ARIMA models are often referred to as Box-Jenkins models as ARIMA approach was first popularized by Box and Jenkins. Stock prices are not randomly generated values rather they can be treated as a discrete time series model and its trend can be analyzed accordingly, hence can also be forecasted. There are various motivations for stock forecasting, one of them is financial gain. A system that can identify which companies are doing well and which companies are not in the dynamic stock market will make it easy for investors or market or finance professionals make decisions. Having an excellent knowledge about share price movement in the future helps the investors and finances personals significantly. Since, it is necessary to identify a model to analyze trends of stock prices with relevant information for decision making, it recommends that transforming the time series using ARIMA is a better approach than forecasting directly, as it gives more accurate results

2.1. Time Series Modeling

2.1.1. Stationary Series

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. Most statistical forecasting methods are based on the assumption that the time series can be rendered approximately

stationary (i.e., "stationarized") through the use of mathematical transformations. A stationarized series is relatively easy to predict: you simply predict that its statistical properties will be the same in the future as they have been in the past! (Recall our famous forecasting quotes.) The predictions for the stationarized series can then be "untransformed," by reversing whatever mathematical transformations were previously used, to obtain predictions for the original series. (The details are normally taken care of by your software.) Thus, finding the sequence of transformations needed to stationarize a time series often provides important clues in the search for an appropriate forecasting model. Stationarizing a time series through differencing (where needed) is an important part of the process of fitting an ARIMA model, as discussed in the ARIMA pages of these notes.

Another reason for trying to stationarize a time series is to be able to obtain meaningful sample statistics such as means, variances, and correlations with other variables. Such statistics are useful as descriptors of future behavior only if the series is stationary. For example, if the series is consistently increasing over time, the sample mean and variance will grow with the size of the sample, and they will always underestimate the mean and variance in future periods. And if the mean and variance of a series are not well-defined, then neither are its correlations with other variables. For this reason you should be cautious about trying to extrapolate regression models fitted to nonstationary data.

Most business and economic time series are far from stationary when expressed in their original units of measurement, and even after deflation or seasonal adjustment they will typically still exhibit trends, cycles, random-walking, and other non-stationary behavior. If the series has a stable long-run trend and tends to revert to the trend line following a disturbance, it may be possible to stationarize it by de-trending (e.g., by fitting a trend line and subtracting it out prior to fitting a model, or else by including the time index as an independent variable in a regression or ARIMA model), perhaps in conjunction with logging or deflating. Such a series is said to be trend-stationary. However, sometimes even de-trending is not sufficient to make the series stationary, in which case it may be necessary to transform it into a series of period-to-period and/or season-to-season differences. If the mean, variance, and autocorrelations of the original series are not constant in time, even after detrending,

perhaps the statistics of the changes in the series between periods or between seasons will be constant. Such a series is said to be difference-stationary

2.1.2. Augmented Dickey–Fuller test

In statistics and econometrics, an augmented Dickey–Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend-stationarity. It is an augmented version of the Dickey–Fuller test for a larger and more complicated set of time series models.

The augmented Dickey–Fuller (ADF) statistic, used in the test, is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root at some level of confidence.

The testing procedure for the ADF test is the same as for the Dickey–Fuller test but it is applied to the model

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t,$$

Where α is a constant, β the coefficient on a time trend and p the lag order of the autoregressive process. Imposing the constraints α=0 and β =0corresponds to modelling a random walk with a drift. Consequently, there are three main versions of the test, analogous to the ones discussed on Dickey–Fuller test by including lags of the order p the ADF formulation allows for higher-order autoregressive processes. This means that the lag length p has to be determined when applying the test. One possible approach is to test down from high orders and examine the t-values on coefficients. An alternative approach is to examine information criteria such as the Akaike information criterion (AIC), Bayesian information criterion (BIC) or the Hannan–Quinn information criterion.

The unit root test is then carried out under the null hypothesis γ =0 against the alternative hypothesis of γ <0 Once a value for the test statistic

$$DF_\tau = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$$

is computed, it can be compared to the relevant critical value for the Dickey–Fuller Test. If the test statistic is less (this test is non symmetrical so we do not consider an

absolute value) than the (larger negative) critical value, then the null hypothesis of $\gamma=0$ is rejected and no unit root is present.

2.2. ARIMA MODEL

2.2.1. Autoregressive models

In a multiple regression model, we forecast the variable of interest using a linear combination of predictors. In an autoregression model, we forecast the variable of interest using a linear combination of past values of the variable. The term autoregression indicates that it is a regression of the variable against itself.

Thus, an autoregressive model of order p can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

Where εt is white noise. This is like a multiple regression but with lagged values of yt as predictors. We refer to this as an AR (p) model, an autoregressive model of order p

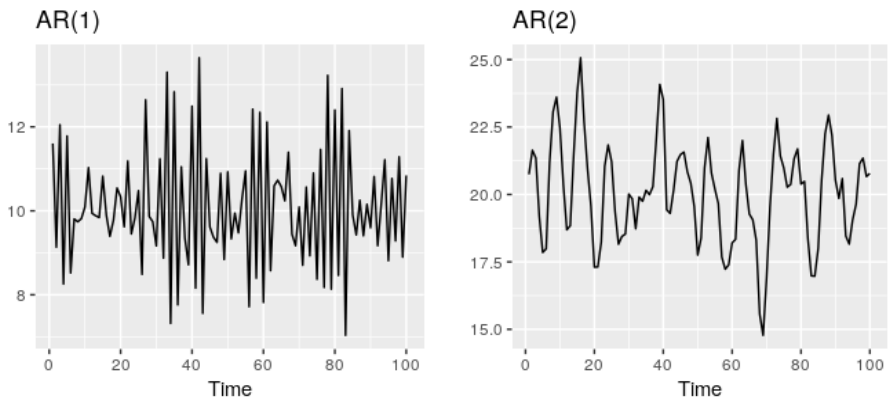

Figure 2.2.1: Two examples of data from autoregressive models with different parameters. Left: AR (1) with yt =18−0.8 yt −1+ εt. Right: AR (2) with yt =8+1.3 yt −1−0.7 yt −2+ εt. In both cases, εt is normally distributed white noise with mean zero and variance one.

2.2.2. Moving average models

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.

Where εt is white noise we refer to this as an MA (q) model, a moving average model of order q. we do not observe the values of εt, so it is not really a regression in the

usual sense. A moving average model is used for forecasting future values, while moving average smoothing is used for estimating the trend-cycle of past values.

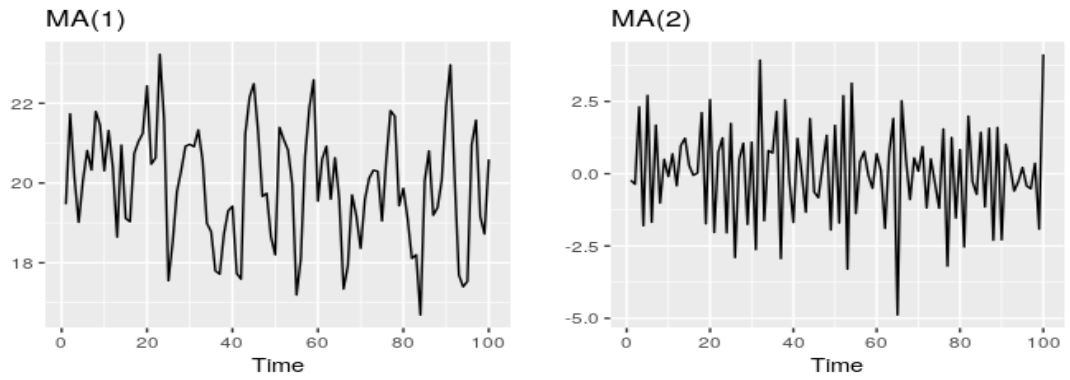

Figure 2.2.2: Two examples of data from moving average models with different parameters. Left: M (1) with yt =20+ εt +0.8 εt −1. Right: MA (2) with yt = εt − εt −1+0.8 εt −2. In both cases, ε is normally distributed white noise with mean zero and variance one.

2.2.3. ARIMA

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for AutoRegressive Integrated Moving Average (in this context, "integration" is the reverse of differencing). The full model can be written as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

Where y′t is the differenced series (it may have been differenced more than once). The "predictors" on the right hand side include both lagged values of yt and lagged errors. We call this an ARIMA (p, d, q) model, where

p = order of the autoregressive part;

d = degree of first differencing involved;

q = order of the moving average part.

The same stationarity and invertibility conditions that are used for autoregressive and moving average models also apply to an ARIMA model.

Many of the models we have already discussed are special cases of the ARIMA model, as shown in Table below

Special cases of ARIMA models.

White noise	ARIMA(0,0,0)
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregression	ARIMA(p,0,0)
Moving average	ARIMA(0,0,q)

3. REQUIREMENT ANALYSIS AND FEASIBILITY STUDY

3.1. Feasibility Study

Simply put, stock market cannot be accurately predicted. The future, like any complex problem, has far too many variables to be predicted. The stock market is a place where buyers and sellers converge. When there are more buyers than sellers, the price increases. When there are more sellers than buyers, the price decreases. So, there is a factor which causes people to buy and sell. It has more to do with emotion than logic. Because emotion is unpredictable, stock market movements will be unpredictable. It's futile to try to predict where markets are going. They are designed to be unpredictable.

The proposed system will not always produce accurate results since it does not account for the human behaviours. Factors like change in company's leadership, internal matters, strikes, protests, natural disasters and change in the authority cannot be taken into account for relating it to the change in Stock market by the machine.

The objective of the system is to give a approximate idea of where the stock market might be headed. It does not give a long term forecasting of a stock value. There are way too many reasons to acknowledge for the long term output of a current stock. Many things and parameters may affect it on the way due to which long term forecasting is just not feasible.

3.2. Requirement Analysis

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs. The requirement that the system needs is categorized into the functional and non-functional requirements. These requirements are listed below:

3.2.1 Functional Requirements

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

• The system should be able to generate an approximate share price.

• The system should collect accurate data in consistent manner.

3.2.2 Non-Functional Requirements

Non-functional requirement is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non- functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on these the non-functional requirements are as follows:

• The system should provide better accuracy.
• The system should have simple interface for users to use.
• To perform efficiently in short amount of time.

4. METHODOLOGY

The purposed method for developing the system consists of mainly three main steps. First, data is collected and sorted for relevancy from various sources. Second, data is converted to time series and check for stationarity by using ADF Test. Third, if non stationary, data is converted to stationary data by using differencing. At last an ARIMA MODEL is designed and a suitable algorithm yielding best accuracy is chosen to predict the stock value.

4.1. Data Sources

This project attempts to predict the stock value with respect to the stock's previous value and trends. It requires historic data of stock market as the project also emphasizes on data mining techniques. So, it is necessary to have a trusted source having relevant and necessary data required for the prediction. We will be using Alpha Vantage website (https://www.alphavantage.co/) as the primary source of data. Alpha Vantage Inc. is a leading provider of accessible APIs for financial market data including stocks, FX, and digital/crypto currencies. This website contains all the details such as: Opening value, Closing value, highest value, lowest value for each financial companies. The site is updated on daily basis and it is also a repository for years of stock market data for thousands of different companies.

There is an API provided by the website for providing data. We have used API to gathered all the required data from Alpha vantage website .Documentation is available on

4.2. Selection of Company

The stock market is a very fluctuating market. There are many companies of different sectors and the values as well as parameters can vary differently in time. In this case, same rules or logic for constructing a prediction model may not apply to the all the companies in NEPSE. So, this project performs analysis and prediction on MICROSOFT Stock Price Data.

4.3. ARIMA Design and Training

The main problem in predicting share market is that the share market is a chaos system. There are many variables that could affect the share market directly or indirectly. There are no significant relations between the variables and the price. We cannot draw any mathematical relation among the variables. There are no laws of predicting the share price using these variables.

The method used in this study to develop ARIMA model for stock price forecasting is explained in detail in subsections below. The tool used for implementation is R. Stock data used in this research work are historical daily stock prices obtained from two countries stock exchanged. The data composed of four elements, namely: open price, low price, high price and close price respectively. In this research the closing price is chosen to represent the price of the index to be predicted. Closing price is chosen because it reflects all the activities of the index in a trading day. To determine the best ARIMA model among several experiments performed, the following criteria are used in this study for each stock index.

• Relatively small of BIC (Bayesian or Schwarz Information Criterion)
• Relatively small standard error of regression (S.E. of regression)
• Relatively high of adjusted R2

4.3.1. Dataframe Creation

First of all, a data frame is created for training the Arima model. The collected data are then converted to Time Series Data, by using various libraries as Zoo, Xts, and Forecast packages available in R, which we use for training. Then we select only close_stock column. The data frame should be converted xts object. We use Microsoft Stock Price Data from 1995 to 2018

4.3.2 Visualize the Time Series

Plotting data is arguably the most critical step in the exploratory analysis phase. (We chose to emphasize the time-series object that has intervals from 1995 to 2014, a choice we will explain later!) Visualizing our time-series data enables us to make

inferences about important components, such as trend, seasonality, heteroskedasticity, and stationarity. Here is a quick summary of each:

1) Trend: We say that a dataset has a trend when it has either a long-term increase or decrease.

2) Seasonality: We say that a dataset has seasonality when it has patterns that repeat over known, fixed periods of time (e.g. monthly, quarterly, yearly).

3) Heteroskedasticity: We say that a data is heteroskedastic when its variability is not constant (i.e., its variance increases or decreases as a function of the explanatory variable).

4) Stationarity: A stochastic process is called stationary if the mean and variance are constant (i.e., their joint distribution does not change over time)..

We plot the graph by using PLOT function available in R

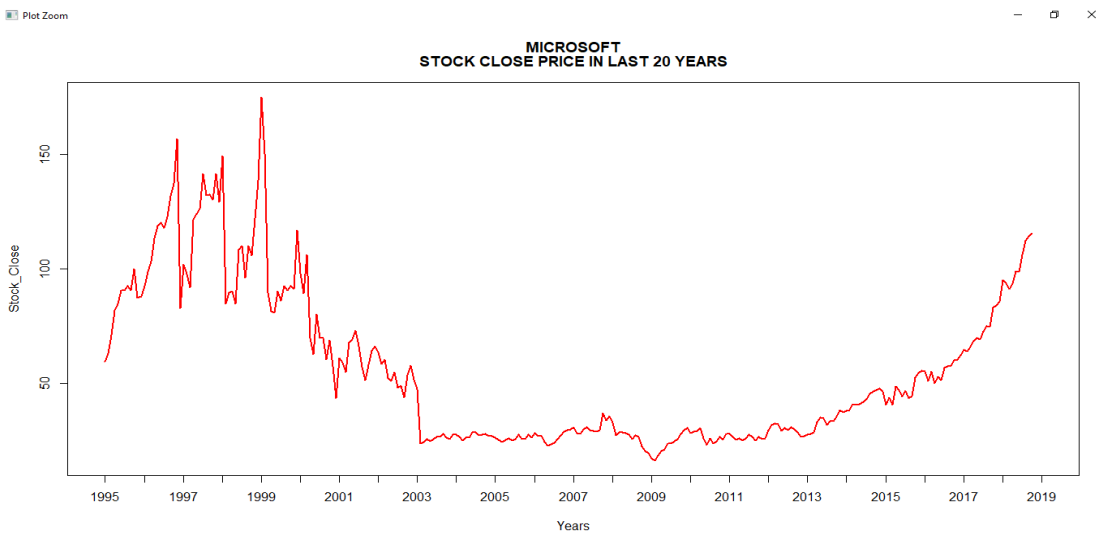

Figure 4.3.2: (a) Uncleaned Microsoft close_stock_price data from 1995-2018

We can infer from the graph itself that the data points follows an overall upward trend with some outliers in terms of sudden lower values. Now we need to do remove those outliers. We remove these outliers by using tsclean (Time Series clean) function available in r

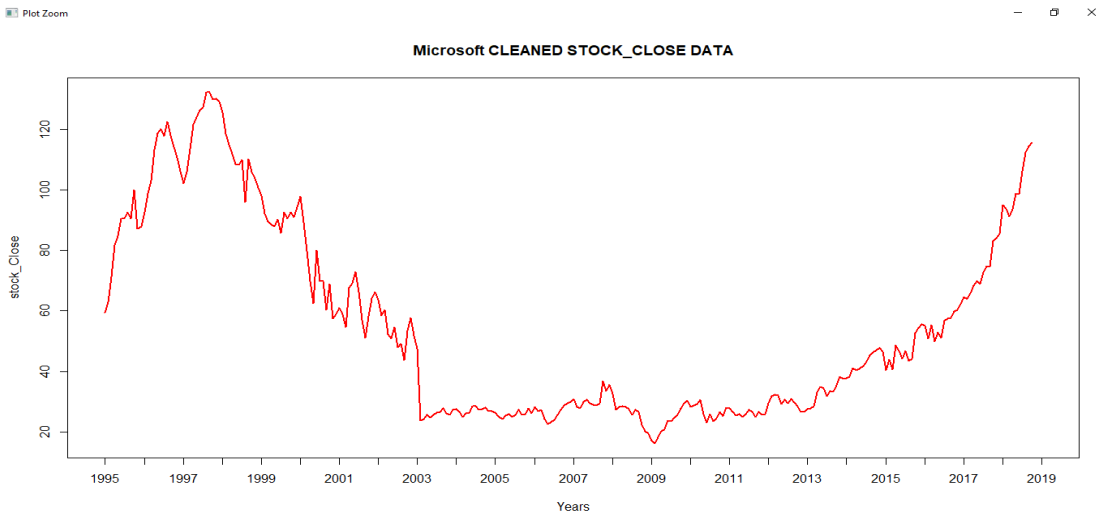

Figure 4.3.2 (b): Cleaned Microsoft close_stock_price data from 1995-2018

From the above graph we infer that stock prices first increases from

1) Stock prices first increases from 1995-1999
2) They gradually decreases from 1999-2003
3) Stock prices remains constant from 2003-2013 with a sudden downfall in 2009
4) Then They finally starts increasing from 2013-present day

We also notice that there is some seasonality present in our data.

4.3.3 Decompose a Time Series

Beyond understanding the trend of your time series, you want to further understand the anatomy of your data. For this reason, we will break down our time series into its seasonal component, trend, and residuals.

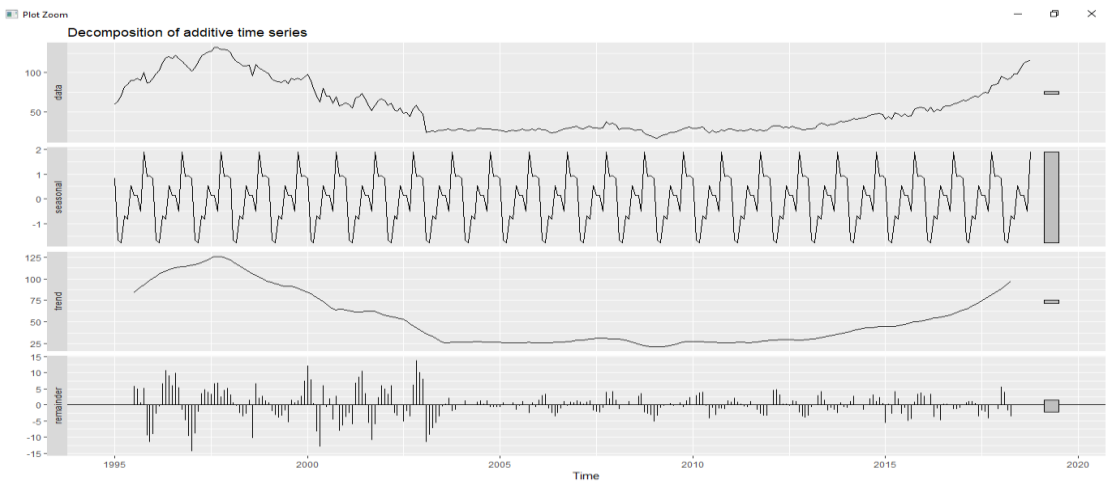

Figure 4.3.3: Decomposing Microsoft close_stock_price data

Important Inferences from seasonality trend

1) The variance and the mean value in May and June is much higher than rest of the months.

2) Even though the mean value of each month is quite different their variance is small. Hence, we have strong seasonal effect with a cycle of 12 months or less.

4.3.4 Stationarize the Series

Once we know the patterns, trends, cycles and seasonality, we can check if the series is stationary or not. Dickey – Fuller is one of the popular test to check the same. What if the series is found to be non-stationary?

There are three commonly used technique to make a time series stationary:

1. Detrending: Here, we simply remove the trend component from the time series. For instance, the equation of my time series is:

$x(t) = (\text{mean} + \text{trend} * t) + \text{error}$

We'll simply remove the part in the parentheses and build model for the rest.

2. Differencing: This is the commonly used technique to remove non-stationarity. Here we try to model the differences of the terms. For instance,

$x(t) – x(t-1) = \text{ARMA}(p, q)$

This differencing is called as the Integration part in AR(I)MA. Now, we have three parameters

p : AR, d : I , q : MA

3. Seasonality: Seasonality can easily be incorporated in the ARIMA model directly.

Let’s apply Augmented Dicker-Fuller test to our data and check for its stationarity

```
        Augmented Dickey-Fuller Test

data:  close_clean_data
Dickey-Fuller = 0.82807, Lag order = 6, p-value = 0.99
alternative hypothesis: stationary
```

Figure 4.3.4(a): ADF TEST ON STOCK DATA

You can see our p value for the ADF test is relatively high. Therefor null hypothesis (Non-stationary) cannot be rejected. Hence our time series is non-stationary. So We we need to address two issues before we test for stationary series again. One, we need to remove unequal variances. We do this using log of the series. Two, we need to address the trend component. We do this by taking difference of the series. Now, let’s test the resultant series.

```
        Augmented Dickey-Fuller Test

data:  differenced_data
Dickey-Fuller = -10.668, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

Figure 4.3.4(b): ADF TEST ON Differenced STOCK DATA

Since p value is less than 0.5, null hypothesis is rejected. Hence our series become stationary series

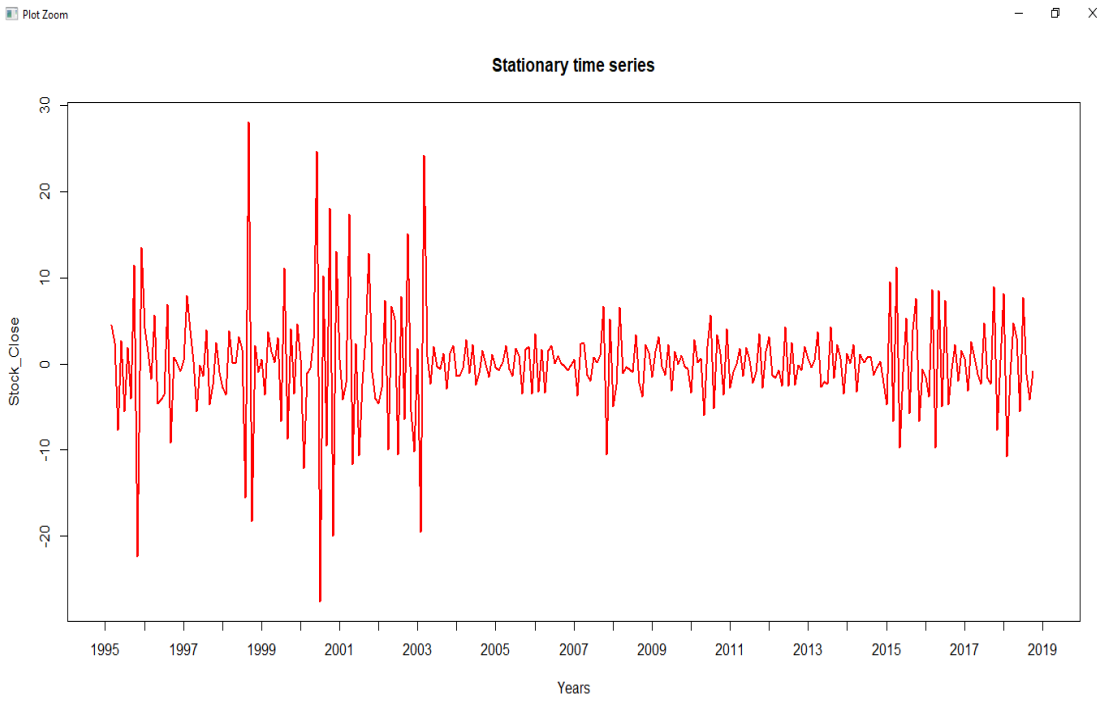

Figure 4.3.4 (c): Stationary close_stock_price data

4.4. Fitting an ARIMA model

The forecast package allows the user to automatically generate a set of optimal (p, d, q) using auto.arima(). This function searches through combinations of order parameters and picks the set that optimizes model fit criteria.

There exist a number of such criteria for comparing quality of fit across multiple models. Two of the most widely used are Akaike information criteria (AIC) and Bayesian information criteria (BIC). These criteria are closely related and can be interpreted as an estimate of how much information would be lost if a given model is chosen. When comparing models, one wants to minimize AIC and BIC.

While auto.arima() can be very useful, it is still important to complete steps 1-5 in order to understand the series and interpret model results. Note that auto.arima () also allows the user to specify maximum order for (p, d, q), which is set to 5 by default.

Hyndman–Khandakar algorithm for automatic ARIMA modelling

1. The number of differences $0 \leq d \leq 2$ is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
   a. Four initial models are fitted:
      - ARIMA(0, d, 0),
      - ARIMA(2, d, 2),
      - ARIMA(1, d, 0),
      - ARIMA(0, d, 1).

      A constant is included unless $d = 2$. If $d \leq 1$, an additional model is also fitted:
      - ARIMA(0, d, 0) without a constant.

   b. The best model (with the smallest AICc value) fitted in step (a) is set to be the "current model".

   c. Variations on the current model are considered:
      - vary p and/or q from the current model by ±1;
      - include/exclude c from the current model.

      The best model considered so far (either the current model or one of these variations) becomes the new current model.

   d. Repeat Step 2(c) until no lower AICc can be found.

Figure 4.4: Auto arima algorithm

We can specify non-seasonal ARIMA structure and fit the model to stationary data

```
> fit_seasonal
Series: differenced_data
ARIMA(3,0,2)(2,0,2)[12] with zero mean

Coefficients:
         ar1     ar2      ar3      ma1     ma2
      0.7608  0.1660  -0.1975  -1.8271  0.8449
s.e.  0.1310  0.0752   0.0673   0.1234  0.1174
        sar1     sar2    sma1    sma2
      -0.3226  -0.6233  0.4144  0.7782
s.e.   0.2554   0.1865  0.2180  0.1620

sigma^2 estimated as 17.05:  log likelihood=-803.8
AIC=1627.6   AICc=1628.4   BIC=1664.09
```

Note that value of d is 0, as our time series is stationary

4.5. Forecasting

Since we believe we've found the appropriate model, let's begin forecasting. Next we use the forecast function, ggplot2, and plotly to visualize the predictions for the years 2019-2023. Within the plots, the forecasted values are BLUE, the actual values are in RED, the 80% confidence intervals are encompassed in the LIGHT GREY bands, and 95% confidence intervals are encompassed in the DARK GREY bands.

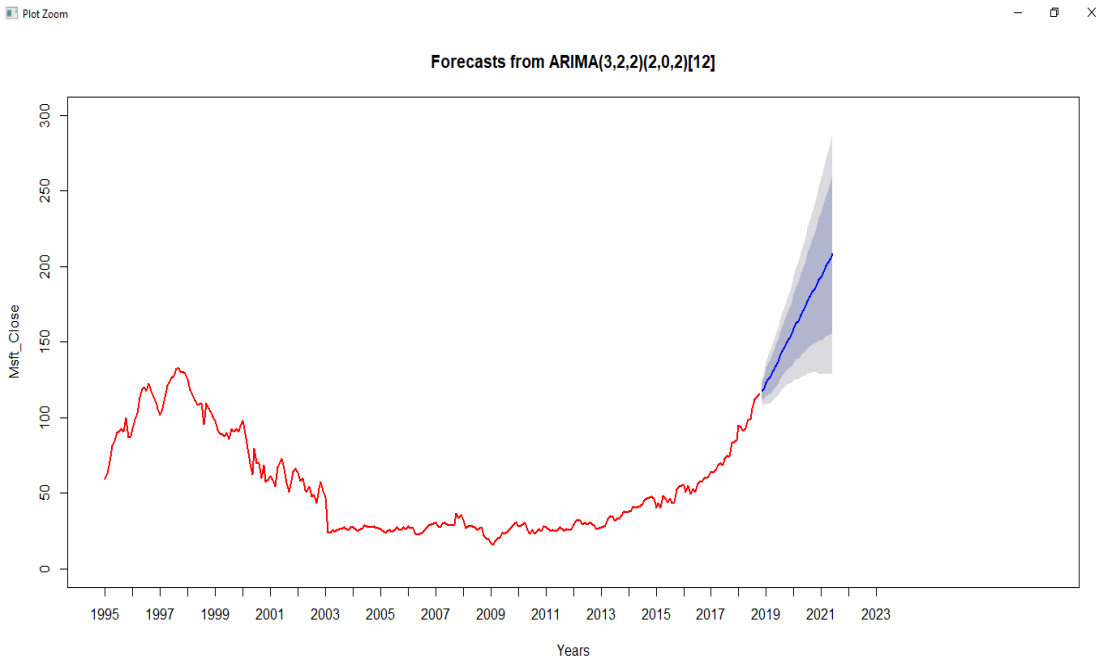

Figure 4.5: Forecasting plot

We can see that the model performs well and within the 80% and 95% confidence intervals. You can forecast values even further into the future by tuning the appropriate parameters. Please not that this forecast project is for educational purposes only and we do not recommend investing based on predictions made during this project. The stock market is very volatile.

5. ACCURACY AND RESULTS

5.1. Accuracy – "Mean Actual Percentage Error"

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation. It usually expresses accuracy as a percentage, and is defined by the formula:

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$

Where At is the actual value and Ft is the forecast value.

The difference between At and Ft is divided by the actual value At again. The absolute value in this calculation is summed for every forecasted point in time and divided by the number of fitted points n. multiplying by 100% makes it a percentage error.

We can calculate MAPE by using mape() function in "MATRICS" library in R

```
> mape(df$actual,df$predicted)
[1] 0.05866064
```

Hence our Mean Actual Percentage Error is 5.5866% which is good

5.2. RESULT

We first create a data frame df which contains two columns: Actual and Predicted and then plot them using plot() function

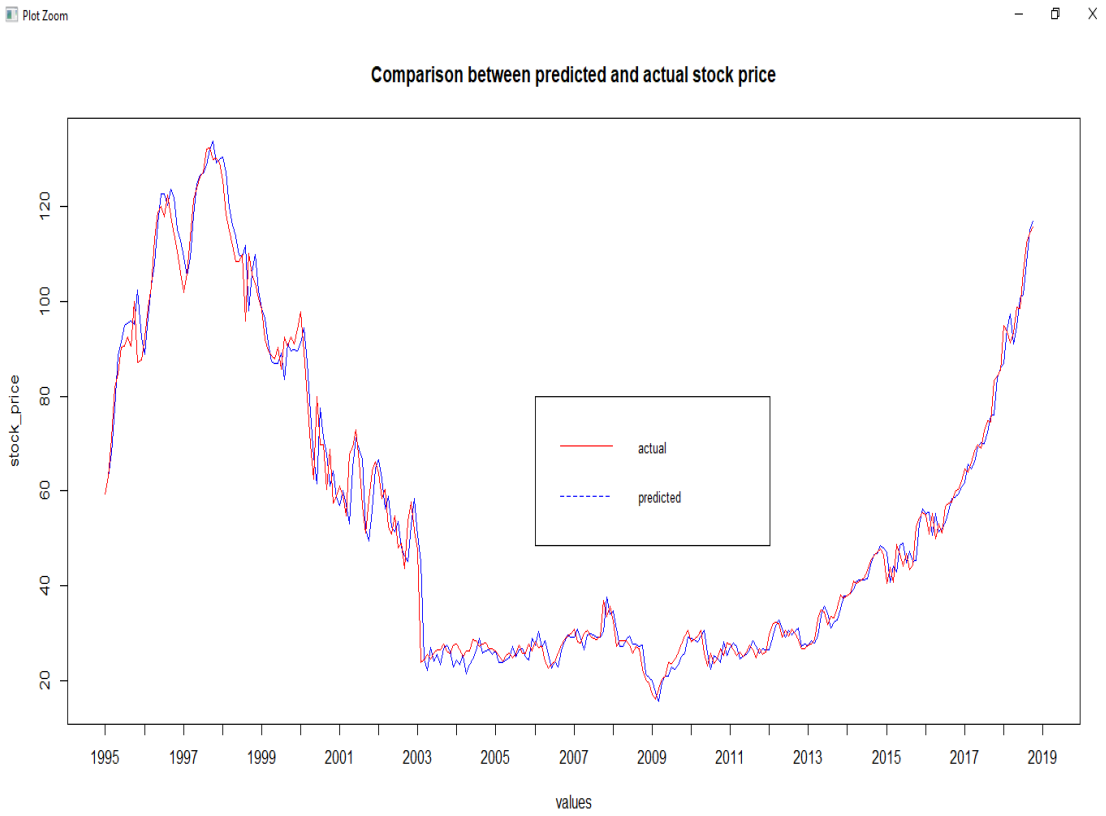

Figure 5.2: plot between actual and predicted value

‘

6. LIMITATION AND FUTURE ENHANCEMENT

6.1. Limitation

The main aim of this system is to provide a general idea of where the stock market is headed. It is only limited to a very basic prediction model. Thus, it cannot be used as a critical decision making tool. By incorporating only limited number of parameters, there is certain degree of accuracy. Since, there are many indeterminate parameters that directly affect stock market, each and every one of them cannot be taken into account. So, our model only depends on the relationship of our selected parameters with the share price.

At present, this system only performs analysis and prediction of only those companies whose data is available from 1995-2019. This system is limited to only certain users that have knowledge of stock market.

6.2. Future scope of improvement

- Potential improvement can be made to our data collection and analysis method.
- Future research can be done with possible improvement such as more refined data and more accurate algorithm.
- Implementation of discussion forums and economic news portal including other sector apart from hydropower and going in national level.

Bibliography

1. www.datascience.com

2. www.stackoverflow.com

3. www.datacamp.com

4. www.wikipedia.com

5. www.alphavantage.com