# University School of Management Studies

Academic Year: 2021~22

## Project on:

## Stock Market Prediction using ARIMA Model

Post Graduate Diploma

In

Data Analytics

Under supervision of

Mr. Santanoo Pattnaik

Submitted By:-

Priyanshu Bhandari - 01016640621

Nishant - 01416640621

Date of Submission:-

15th Mar 2022

# Certificate

This is to certify that the dissertation entitled **"Stock Market prediction using ARIMA Model"** is the bona fide research work carried out by Priyanshu Bhandari and Nishant students of PGDDA, at University School of Management Studies, Guru Gobind Singh Indraprastha University during the year 2021-2022, in partial fulfilment of the requirements for the award of the degree of Post Graduate Diploma in Data Analytics and that the dissertation has not formed the basis for the award previously of any degree or diploma to the best of my knowledge and belief.

**Dr Santanoo Pattnaik**

**USMS**

**GGSIPU**

# Declaration

I hereby declare that this project work entitled "**Stock Market Prediction using ARIMA Model**" is a record of an original work done by us under the guidance of Dr. Santanoo Pattnaik and this project is submitted in partial fulfilment of the requirements for the award of the degree of Post Graduate Diploma in Data Analytics. The results embodied in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

Name: Priyanshu Bhandari

Enrollment: 01016640621

Name: Nishant

Enrollment: 01416640621

Course: PGDDA-2021~22

# Acknowledgement

Materials in this report have been drawn from a wide variety of sources including weekly journals, books, magazine and internet. Sources of these works are cited where they are discussed in the text, and I hope that I have made no omissions.

I am deeply indebted to our project in-charge **Mr. SANTANOO PATNAIK** without whose support and encouragement, this project couldn't have been accomplished.

I sincerely extend our deep felt regards for their able guidance throughout this training period, for providing us with eminent environment and essential resources to complete our project work.

# Table of Contents

# Abstract

Stock market is basically volatile and the prediction of its movement will be more useful to the stock traders to design their trading strategies. An intelligent forecasting will certainly abet to yield significant profits. Many important models have been proposed in the economics and finance literature for improving the prediction accuracy and this task has been carried out through the modelling based on time series analysis. The main aim of this paper is to check the stationarity in time series data and predicting the direction of change in stock market index using the stochastic time series ARIMA modelling. The **best fit ARIMA** model was chosen for forecasting the values of time series, viz., BSE_CLOSE and NSE_CLOSE by considering the **smallest values of AIC, BIC, RMSE, MAE, MAPE, Standard Error of Regression**, and **the relatively high Adjusted $R^2$ values**. Using this best fitted model, the predictions were made for the period ranging from 7th January, 2018 to 3rd June, 2018 (22 expected values) using the weekly data ranging from 6th January, 2014 to 31st December, 2017 (187 observed values). The results obtained from the study confirmed the prospective of ARIMA model to forecast the future time series in short-run and would assist the investing community in making the profitable investment decisions.

# Introduction

Stock market forecasting is an exercise to determine the future value of its performance index, viz., SENSEX, NIFTY. The successful prediction of any market's future index or a stock's future price will be more useful to the investing community to design optimal trading strategies and could yield significant profits. So, in the recent past, the concept of forecasting stock market and its return is gaining lot of attention by the researchers. It may be because of the fact that if the directions of change in the market movements are successfully predicted, the investors may be better guided. Sometimes, the forecasted trends of the market will help the policy makers and regulators of the stock market in making curative decisions. The profit making investments and day to day operations in capital market depends heavily on the forecasting ability.

Many practicing investors like Warren Buffett and other market researchers have proposed several models using various analytical methods, viz., fundamental analysis, technical analysis and analytical techniques, etc. to give more or less exact forecasting. In addition to the above methods of forecasting, some traditional time series models were also used for it. Mainly, there are two kinds of time series models for forecasting, viz., linear models and non-linear models. Some of the examples of linear models are moving average, exponential smoothing, time series regression, etc. One among the most common and popular linear models is the Autoregressive Integrated Moving Average (ARIMA) model proposed by Box and Jenkins (1976). In this paper, a modest attempt has been made to select the best fitted ARIMA model from different stochastic models that satisfies all the criteria of goodness of fit statistics for making the predictions and also to forecast the future values of stock market indices.

The remaining part of this paper is organized as follows: Section 2 contains the review of literature, statement of the problem, objectives of the study and research methodology. Section 3 deals with checking the stationarity in time series data using Augmented Dickey Fuller (ADF) Unit Root Test and by performing correlogram analysis. Section 4 of the paper highlights the selection of best suitable model from different stochastic models that satisfies all the criteria of goodness of fit statistics for making the predictions. The experimental results of ARIMA (p,d,q) forecasting model are presented in Section 5. Finally, the conclusion, limitations and scope for future research are explained in section 6.

# Literature review

It is pertinent to review the accessible literature connected to the time series modeling and forecasting using ARIMA model. Most of the literature is focused on the identification of suitable ARIMA time series model and forecasting the gold price, exchange rates, oil palm prices, inflation rates, electricity consumption, etc. Only few studies are available relating to the forecasting of stock prices and stock market indices. Hence, this paper has been mainly devoted to the studies related to the determination of best ARIMA time series model and forecasting of future stock prices and stock market indices.

Meyler and Kenny (1998) have developed ARIMA time series predicting model for predicting the inflation in Ireland. In their study, they have focused on maximizing the power of forecasting by minimizing forecast errors. Contreras, Rodrigo, Francisco and Antonio (2003) have examined the trends in daily prices of electricity in spot and forward contracts for mainland Spain and Californian Markets and provided the best suited ARIMA method to predict next day electricity prices.

Rangson and Tidia (2006) have conducted a study with an objective to find an appropriate ARIMA model for forecasting three types of oil palm price by considering the minimum Mean Absolute Percentage Error. The empirical analysis of the study show that ARIMA (2,1,0), (1,0,1) and (3,0,0) are the best models for forecasting the farm price of oil palm, wholesale price of oil palm and pure oil price of oil palm respectively.

In a study conducted by Jarrett and Kyper (2011) using the data developed by Pacific-Basin Capital Markets (PACAP) and the SINOFIN Information Services Inc, has demonstrated the usefulness of ARIMA-Intervention time series analysis as both an analytical and forecast tool. The study indicates the usefulness of the developed model in explaining the rapid decline in the values of the price index of Shanghai market during the world economic decline in China in 2008. The authors have concluded that the daily stock price index contains an autoregressive component; hence, it is better to forecast the stock returns using ARIMA model.

Banerjee (2014) has used the ARIMA Model for predicting stock market indices and also highlighted that they have an undue influence on the progress of Indian economy. The study has dealt with the identification of the best fit ARIMA model and after that predicted the SENSEX using the justified model.

Adebiyi and Adewumi (2014) have presented the procedure for developing ARIMA models for forecasting share prices during short-run. The results of the study have explained the power of ARIMA models in predicting the stock prices in short-run, which would help the investors in their decisions. A study was conducted by Jadhav, Kakade, Utpat and Deshpande (2015) for forecasting the Indian Share Market using ARIMA model and said that artificial neural networks (ANNs) are universal approximates that can be applied to a wide range of time series for forecasting futuristic values in share market and give bright scope for investment. But, in their study, they have proposed a novel hybrid model of ANN using ARIMA model instead of only artificial neural network for improving the predictive performance.

Guha and Bandyopadhyay (2016) have examined the application of ARIMA time series model to forecast the future gold price based on the past data from November, 2003 to January, 2014 to mitigate the risk in purchase of gold and, hence, to give guidelines for the investor when to buy or sell the yellow metal. The authors have opined that now-a-days gold has gained importance as one of the investment alternatives; it has become necessary to predict the price of gold with an appropriate method.

Savadatti (2017) has carried out a study to identify the best fitted ARIMA models for forecasting the area, production, and productivity of food grains for 5 years. Based on univariate time series analysis, the study has identified ARIMA (2,1,2), ARIMA (4,1,0), and ARIMA (3,1,3) models for forecasting the data on area, production, and productivity of food grains, respectively and these models were found to be adequate. The forecast values indicated that production and productivity have increased during the forecast period but that of area exhibited near stagnancy, calling for timely measures to enhance the supply of food grains to meet the increasing demand in the future years.

Wadhawan and Singh (2019) have examined the different volatility estimators for forecasting volatility with high accuracy by traders, option practitioners, and various players of the stock market. The study evaluated the efficiency and bias of various volatility estimators based on various error measuring parameters, viz., ME, RMSE, MAE, MPE, MAPE, MASE and ACFI. The study has identified Parkinson estimator as the most efficient volatility estimator. The study has also suggested that the forecasted values were accurate based on the values of MAE and RMSE.

It may be concluded that, many researchers have conducted the studies to give reason for the selection of ARIMA model for forecasting the time series data of a single variable with better accuracy. But, no researcher has focused on forecasting stock market indices in Indian context. The present work is an effort to forecast the indices of BSE and NSE based on the past 187 weeks using the best fitted ARIMA model.

# Statement of Problem

Because of dynamic and non-linear in nature, it is very tricky to predict the stock exchange movements precisely. But, it is necessary to forecast and uncover non-linearity of stock market; to enable the individual and institutional investors to design appropriate trading strategies and to achieve better results out of their investment endeavors. Hence, stock market forecasting has become a significant theme and motivated the researchers to build improved forecasting models. There are quite a few methods of statistical forecasting, viz., regression analysis, classical decomposition method, Box and Jenkins and smoothing techniques, with different degrees of accuracy. The accuracy of a forecasting model is based on the minimum errors of forecasting, viz., Root Mean Square Error, Mean Absolute Error, Standard Error of Regression, Adjusted R-square, Akaike Information Criterion, Bayesian Information Criterion, etc. Among several methods of time series forecasting, the Box and Jenkins method is quite accurate compared to other methods and may be applicable to all types of data movements. This paper is an attempt to test the stationarity in the given time series and selecting the best suitable ARIMA model (also known as Box-Jenkins Methodology) for short-term forecasting of BSE and NSE. The results obtained from the study could aid the investors in their investment decision-making process.

# Objective of the study

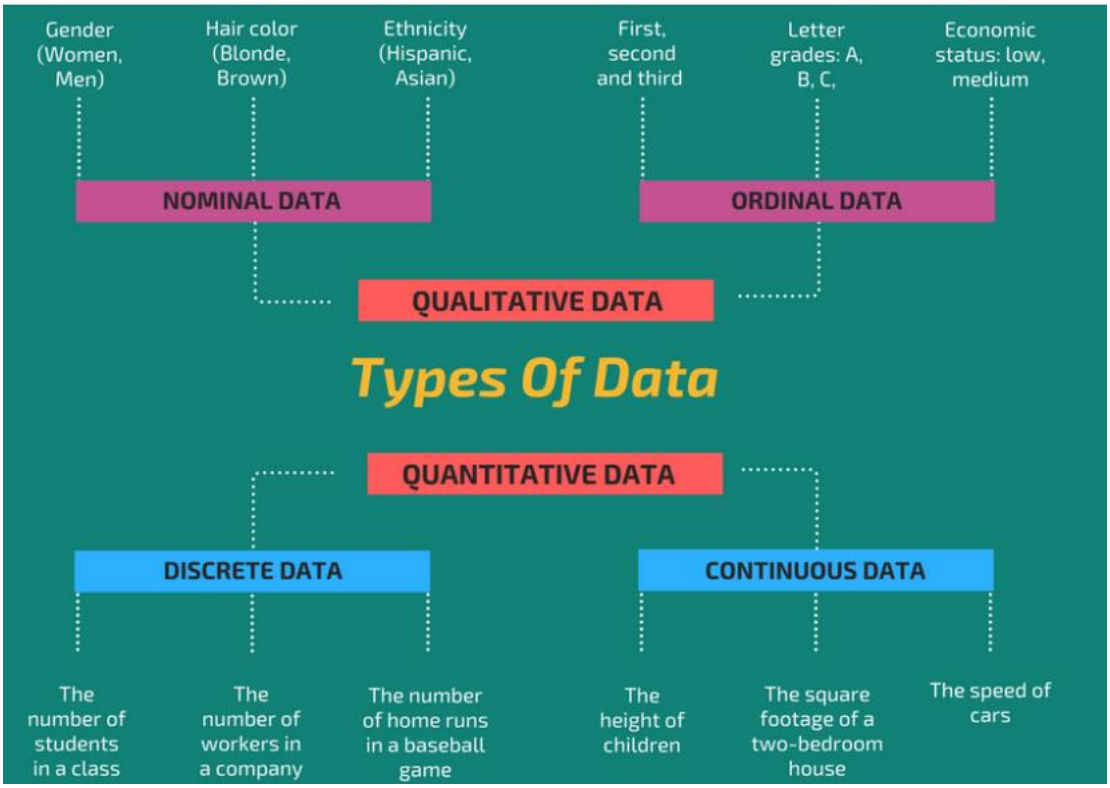
The objectives of the study are listed below:

1. To test the stationarity of the time series data compiled for the study, i.e., daily closing index values of BSE (BSE_CLOSE) and NSE (NSE_CLOSE).
2. To choose the optimum ARIMA model for estimating the series.
3. To forecast the indices of BSE and NSE using the selected time-series ARIMA model.

# Understanding the Terminology

**DATA:** Data is defined as a systematic record corresponding to a specific quantity. Basically, data can be summarized as a set of facts and figures which can be used to serve a specific usage or purpose. For instance, data can be used as a survey or an analysis. Data in a systematic and organized form is referred to as information. In addition to this, the source of data primary or secondary is also an essential factor.

# Types of Data




**Qualitative Data**
Qualitative data is used to represent some characteristics or attributes of the data. The facts and figures depicted by the qualitative data cannot be computed. These properties reflect observable attributes. These are non-numerical in nature. The qualitative data characteristics are exploratory on a larger end than being conclusive in nature. For instance, data on attributes such as honesty, loyalty, wisdom, and creativity for a set of persons defined can be considered as qualitative data.

**Examples:**

- Attitudes of people to a political system.
- Music and art
- Intelligence
- Beauty of a person

**Nominal Data**

Nominal data is a sub-category belonging to one of the types of qualitative information. Also known as the nominal scale, it is used to label the variables without providing the numerical value for them. Nominal data attributes can't either be ordered or measured. The nominal data can be both qualitative and quantitative in nature. For instance, some of the nominal data attributes are letters, symbols or gender, etc.

The examination of the nominal data is based on the usage of the grouping method. This method is based on the principle of the grouping of data into different categories. This is followed by the calculation of the frequency or the percentage of the data. The visualization of this data is done using the pie charts.

**Examples:**

- Gender (Women, Men)
- Eye color (Blue, Green, Brown)
- Hair color (Blonde, Brown, Brunette, Red, etc.)
- Marital status (Married, Single)
- Religion (Muslim, Hindu, Christian)

**Ordinal Data**

Ordinal data/variable is the specific type of data that follows a natural order. The difference between the data values is not determined in the case of nominal data. For instance, ordinal data variable is mostly found in surveys, economics, questionnaires, and finance operations.

The examination of the nominal data is based on the usage of visualization tools. The visualization of this data is done using the bar chart. The ordinal data can be expressed in the form of tables which have each row corresponding to the distinct category.

**Examples:**

- Feedback is recorded in the form of ratings from 1-10.
- Education level: elementary school, high school, college.
- Economic status: low, medium, and high.
- Letter grades: A, B, C, and etc.
- Customer level of satisfaction: very satisfied, satisfied, neutral, dissatisfied, very dissatisfied.

**Quantitative Data**
Quantitative data can be measured and is not just observable. The measurement of data is numerically recorded and represented. Calculations and interpretations can then be performed on the obtained results. Numerical data is indicated by quantitative data. For instance, data can be recorded about how many users found a product satisfactory in terms of the collected rating, and therefore, an overall product review can be generated.

**Examples:**
- Daily temperature
- Price
- Weights
- Income

**Discrete Data**
Discrete data refers to the data values which can only attain certain specific values. Discrete data can't attain a range of values. Discrete data can be represented using bar charts. For instance, ratings of a product made by the users can only be in discrete numbers.

**Examples:**
- The number of students in a class,
- The number of chips in a bag,
- The number of stars in the sky

**Continuous Data**
Continuous Data can contain values between a certain range that is within the highest and lowest values. The corresponding difference between the highest and lowest value of these intervals can be termed as the range of data. Continuous data can be tabulated in what is called a frequency distribution. The frequency distribution table can be computed for the range type of data. It can also be depicted using histograms. For example, the heights of the students in the class can be largely varying in nature, therefore, they can be divided into ranges to summarize the data.

**Examples:**
- Height and weight of a student,
- Daily temperature recordings of a place
- Wind speed measurement

# Data Collection

**Data collection** is essential for businesses, organizations, and even personal use. In the digital age data is one of the most valuable resources at your disposal.

Data collection is the process of gathering and categorizing relevant information that can then be used to make decisions about specific situations.

Depending on the source, data can also be classified as primary or secondary data.

1) **Primary data collection**
   Primary data, also known as raw data, is the data you collect yourself and are the first person to interpret. It's data that's gotten directly from the source. That could be in-person interviews, surveys sent out to your audience, or even courses. Put another way, you're the first person or group to interact with and draw conclusions from the data.
   Primary data is usually collected with a specific goal in mind but can be more challenging for the researcher to interpret. That's because the data is unstructured and needs to be arranged in a way that allows you to make meaningful decisions.

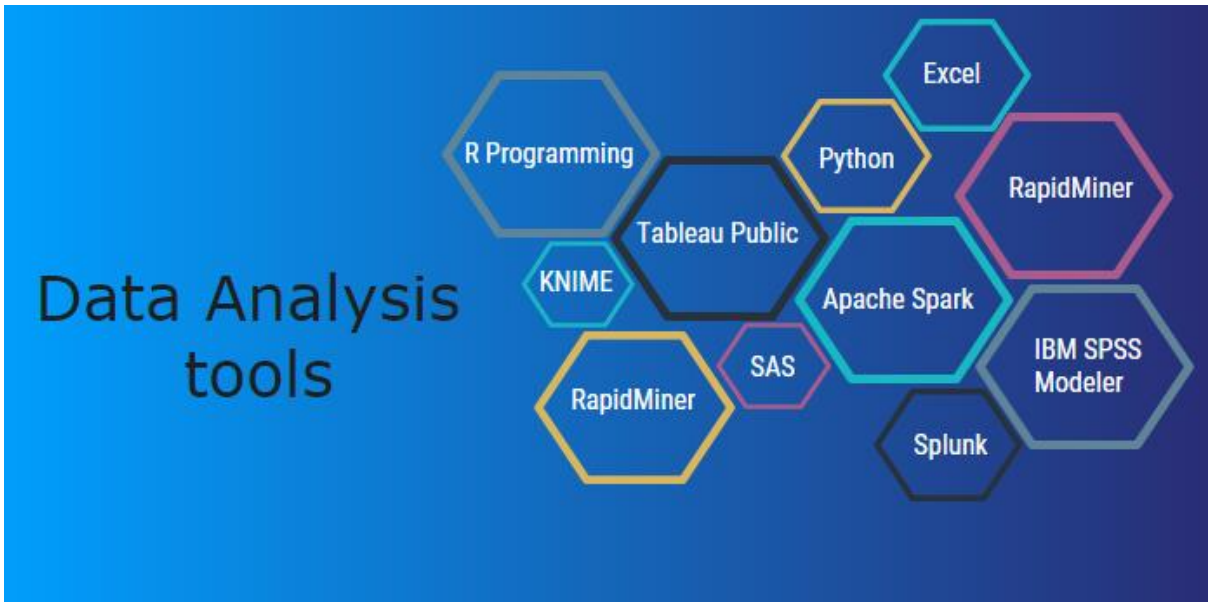
2) **Secondary data collection**

   Secondary data refers to information you use which has been collected, analysed, and structured by another person or group. Things like research papers, books, other websites, etc. can be considered primary data that, when used by you, are secondary data. An example of primary data is the Census of India.

# Data Analysis

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.

# Data Analysis tools




# Types of Analysis

Data Analysis can be separated and organized into 5 types:

1) **Descriptive analysis:**
   - The goal of this type of analysis is to Describe or Summarize a set of data.
   - This is also the very first analysis formed.
   - It Generates simple summaries about samples and measurements
   - common descriptive statistics (measures of central tendency, variability, frequency, position, etc)
   - The result is a very simple presentation of your data.

2) **Exploratory Analysis (EDA):**
   - The goal of exploratory data analysis is to examine or explore data and find relationships between variables which were previously unknown.
   - EDA helps you discover relationships between measures in your data, which are not evidence for the existence of the correlation.
   - Useful for discovering new connections — forming hypothesis and drives design planning and data collection

- Exploratory data analysis is cross-classified in two different ways where each method is either graphical or non-graphical. And then, each method is either univariate, bivariate or multivariate.

  i. **Univariate analysis**:
  This type of data consists of only one variable. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it. The example of a univariate data can be weight.

  ii. **Bivariate analysis**:
  This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables. Example of bivariate data can be temperature and ice cream sales in summer season.

  iii. **Multivariate Analysis**:
  Multivariate means involving multiple dependent variables resulting in one outcome. This explains that the majority of the problems in the real world are Multivariate. For example, we cannot predict the weather of any year based on the season. There are multiple factors like pollution, humidity, precipitation, etc.

### 3) Inferential Analysis:

- Inferential Analysis uses a small sample of data to infer about a larger population.
- The goal is all about using a small amount of information to extrapolate and generalize information to a larger group.
- Accuracy of inference depends heavily on sampling scheme; if the sample isn't representative of the population, the generalization will be inaccurate.

### 4) Predictive Analysis:

- It uses historical or current data to find patterns to make predictions about the future.
- Accuracy of the predictions depends on the input variables.

### 5) Prescriptive Analytics:

- Prescriptive Analytics is a form of advanced analytics which examines data or content to answer the question "What should be done?"
- It is characterized by techniques such as graph analysis, simulation, complex event processing, neural networks, recommendation engines, heuristics, and machine learning.

# Data Analysis vs Data Analytics

-

| S.No. | Data Analytics | Data Analysis |
|---|---|---|
| 1. | It is described as a traditional form or generic form of analytics. | It is described as a particularized form of analytics. |
| 2. | It includes several stages like the collection of data and then the inspection of business data is done. | To process data, firstly raw data is defined in a meaningful manner, then data cleaning and conversion are done to get meaningful information from raw data. |
| 3. | It supports decision making by analyzing enterprise data. | It analyzes the data by focusing on insights into business data. |
| 4. | It uses various tools to process data such as Tableau, Python, Excel, etc. | It uses different tools to analyze data such as Rapid Miner, Open Refine, Node XL, KNIME, etc. |
| 5. | Descriptive analysis cannot be performed on this. | A Descriptive analysis can be performed on this. |
| 6. | One can find anonymous relations with the help of this. | One cannot find anonymous relations with the help of this. |
| 7. | It does not deal with inferential analysis. | It supports inferential analysis. |

# Understanding the data

**Source of Data:**

- **Yahoo! Finance** is a media property that is part of the Yahoo! network.
- It provides financial news, data and commentary including stock quotes, press releases, financial reports, and original content.
- It also offers some online tools for personal finance management. In addition to posting partner content from other web sites, it posts original stories by its team of staff journalists.
- It is ranked 20th by SimilarWeb on the list of largest news and media websites.
- The daily closing indices of BSE and NSE are obtained from the website for the period from 1st Jan, 2018 to 25th Jan, 2022. From this range of data, the researcher will predict the values for 30 days up to 10th March, 2022.

**Tools Used:**

**i. RStudio**

- When you see powerful analytics, statistics, and visualizations used by data scientists and business leaders, chances are that the R language is behind them.
- Open-source R is the statistical programming language that data experts the world over use for everything from mapping broad social and marketing trends online to developing financial and climate models that help drive our economies and communities.
- R was first implemented in the early 1990's by Robert Gentleman and Ross Ihaka, both faculty members at the University of Auckland. Robert and Ross established R as an open-source project in 1995.
- RStudio provides open source and enterprise-ready professional software for data science.
- The reason for picking RStudio was that, because of popular belief, R Programming is best suited for a Time Series Analysis when compared to software like Python as the primary objective of R is Data analysis and Statistics.

**ii. MS EXCEL**

- MS Excel is a commonly used Microsoft Office application. It is a spreadsheet program which is used to save and analyse numerical data.
- It features calculation or computation capabilities, graphing tools, pivot tables.

**Time Series Data:**

- Time series data is a collection of observations obtained through repeated measurements over time.
- Plot the points on a graph, and one of your axes would always be time.
- Time series data can be useful for tracking daily, hourly, or weekly weather data. It can also be useful in tracking changes in application performance.
- In investing, a time series tracks the movement of data points, such as a security's price over a specified period of time with data points recorded at regular intervals. This can be tracked over the short term or the long term.

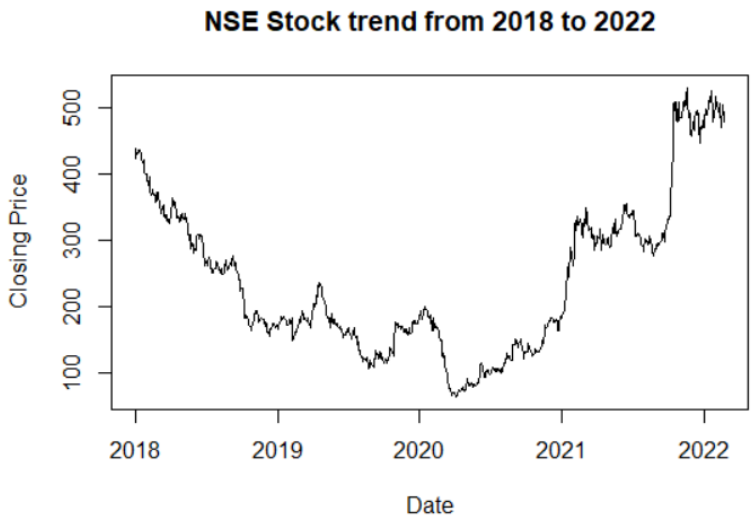
**Libraries used**:

```
library(xts)
library(zoo)
library(quantmod)
library(tseries)
library(timeSeries)
library(forecast)
library(readxl)
library(corrplot)
library(ggpubr)
```

**Getting a glimpse**:

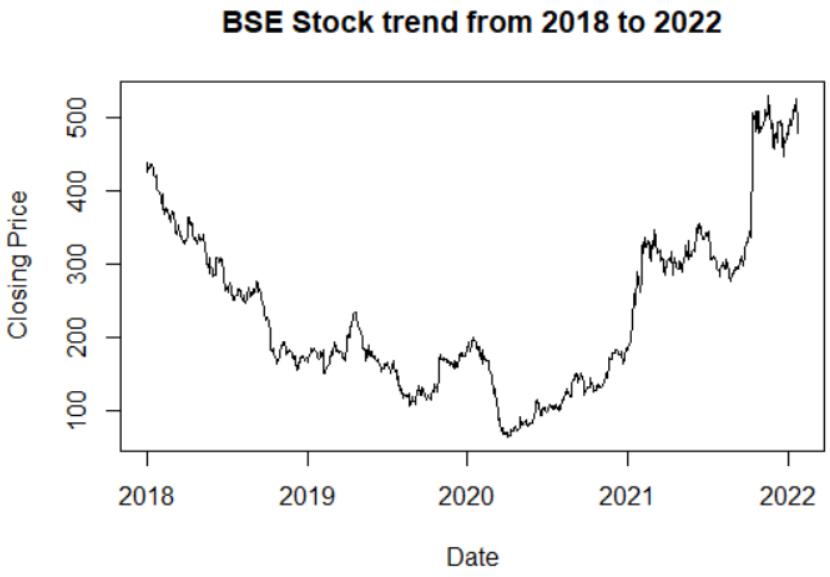
NSE stock data

```
> head(data,5)
        Date   Open   High    Low  Close Adj.Close   Volume
1 2018-01-01 430.95 436.40 422.25 424.45    424.45  6807536
2 2018-01-02 428.85 440.85 422.00 439.30    439.30 15331261
3 2018-01-03 440.40 441.40 431.95 433.90    433.90  9794953
4 2018-01-04 430.00 433.30 425.75 429.95    429.95  8395377
5 2018-01-05 431.25 436.35 429.80 431.60    431.60  7021611
```



BSE stock data

```
> head(data_BSE,5)
        Date  Open   High    Low  Close Adj.Close  Volume
1 2018-01-01 432.2 436.45 422.85 425.40    425.40  524326
2 2018-01-02 429.7 440.20 422.10 438.85    438.85 1316818
3 2018-01-03 439.5 441.45 432.25 433.20    433.20  526955
4 2018-01-04 430.0 433.00 426.05 429.70    429.70  430531
5 2018-01-05 432.0 436.30 430.10 430.80    430.80  358825
```



# Research Methodology

**Research Design**:

Keeping in view of the above listed objectives of the study, an exploratory research design and stochastic modeling has been adopted. Exploratory research is one which interprets the already available information and it lays particular emphasis on the analysis and interpretation of the available secondary data. Stochastic modeling is used for selecting the best ARIMA model and forecasting the time series using the selected model.

**Hypothesis**:

Hypothesis is usually considered as the principal instrument in research. Its main function is to suggest new experiments and observations. In fact, many experiments are carried out with the deliberate object of testing hypotheses. Decision-makers often face situations wherein they are interested in testing hypotheses on the basis of available information and then take decisions on the basis of such testing. In social science, where direct knowledge of population parameter(s) is rare, hypothesis testing is the often used strategy for deciding whether a sample data offer such support for a hypothesis that generalization can be made. Thus hypothesis testing enables us to make probability statements about population parameter(s). The hypothesis may not be proved absolutely, but in practice it is accepted if it has withstood a critical testing.

The null hypothesis in time series is generally defined as the presence of a unit root and the alternative hypothesis is stationarity (or trend-stationary).

H01: $\delta = 1$, there is unit root and the series (BSE_CLOSE and NSE_CLOSE) is non stationary.

Ha1: $\delta < 1$, there is no unit root and the series (BSE_CLOSE and NSE_CLOSE) is stationary.

**Methods for Analysis of Data:**

To select the best fitted ARIMA model, among several experiments conducted, many statistical tools are to be applied, viz., Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), etc.

The RMSE has been used as a standard metric to measure the model performance in stock market forecasting. While applying the RMSE, the underlying assumption is that the errors are unbiased and follow a normal distribution. It provides a complete picture of the error distribution and its value should be relatively low (Chai and Draxler, 2014). The RMSE can be calculated by using the following formula:

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x}_i)^2}{n}} \quad \text{---------- (1)}$$

Mean Absolute Error measures the average magnitude of the errors in a set of predictions, without considering their directions. It is the average over the test sample of the absolute differences between prediction and actual observations where all individual differences have equal weight. Hence, its value should be low. The MAE coefficient is given by the following equation (Chai and Draxler, 2014):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| x_i - \bar{x}_i \right| \quad \text{---------- (2)}$$

The mean absolute percentage error is a measure of prediction accuracy of a forecasting method. It usually expresses the forecasting accuracy of a model in percentage terms; hence, its value should be maximum. The MAPE formula as stated by Tofallis (2015):

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{x_i - \bar{x}_i}{x_i} \right| \quad \text{---------- (3)}$$

Bayesian Information Criterion also known as Schwarz Information Criterion (SIC) is a criterion for model selection among a finite set of models. It is based on the likelihood function, and it is closely related to Akaike Information Criterion (AIC). Mathematically, the BIC is an asymptotic result derived under the assumption that the data series is exponentially distributed. The BIC was developed by Gideon Schwarz (1978), who gave a Bayesian argument for adopting it.

$$\text{BIC} = \log\left(\frac{rss}{n}\right) + \frac{k}{n} \log n \quad \text{---------- (4)}$$

Where, “rss is residual sum of squares; k is the number of coefficients estimated, i.e., 1 + p + q + P + Q; and n is number of observations”.

# Augmented Dickey fuller (ADF) Test

The initial phase of structuring ARIMA model is to recognize the variable being predicted is stationary in time series or not. Most forecasting methods assume that a distribution has stationarity. A time series has stationarity if a shift in time does not cause a change in the shape of the distribution, i.e., the mean and auto-covariance of the series do not depend on time (Tsay, 2005). Unit roots are one cause for non-stationarity. An absence of stationary can cause unexpected behaviors in data series. Most real-life data sets just are non-stationary and we should make it stationary in order to get any useful predictions from it. Augmented Dickey Fuller (ADF) Unit root test tests whether, a time series variable is non-stationary and possesses unit root. A common example of a non-stationary series is the random walk. We may write the Random Walk Model (RWM) with stochastic process as (Rao and Mukherjee, 1971), (Garekos and Gramacy 2013):

$Y_t = \delta Y_{t-1} + u_t \quad (-1 \leq \delta \leq 1)$ ---------- (5)

Where t = time measured chronologically; and

$u_t$ = white noise error term.

For theoretical reasons, we manipulate equation - (5) by subtracting Yt-1 from both the sides to obtain -

$Y_t - Y_{t-1} = \delta Y_{t-1} - Y_{t-1} + u_t$

$Y_t - Y_{t-1} = (\delta - 1) Y_{t-1} + u_t$ ---------- (6), which can be written as

$\Delta Y_t = \beta Y_{t-1} + u_t$ ---------- (7)

Where $\beta = (\delta - 1)$, and $\Delta$ = first difference operator.

In practice, instead of estimating equation – (5), we estimate equation – (7) and test the hypothesis (null) that $\beta = 0$. If $\beta = 0$, then $\delta = 1$, i.e., we have a unit root, meaning that the time series under consideration is non stationary. Before we proceed to estimate equation – (7), it may be noted that if $\delta = 0$, equation – (7) will become –

$\Delta Y_t = (Y_t - Y_{t-1}) = u_t$ ---------- (8)

Since $u_t$ is the white noise error term, it is stationary, which means that the first differences of a random walk time series are stationary.

Before running the ADF test, one should inspect the data to figure out an appropriate regression model. We have three versions of the test.

| | | |
|---|---|---|
| Type 0 | No Constant, No Trend | $\Delta Y_t = \beta_1 Y_{t-1} + u_t$ |
| Type 1 | Constant, No Trend | $\Delta Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$ |
| Type 2 | Constant, Trend | $\Delta Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 t + u_t$ |

The Augmented Dickey Fuller adds lagged differences to the above models (Damodar N. Gujarati, 2004):

| | | |
|---|---|---|
| Type 0 | No Constant, No Trend | $\Delta Yt = \beta 1 Yt\text{-}1 + \sum_{t=1}^{m} \alpha\, i \Delta Y(t-1)$ |
| Type 1 | Constant, No Trend | $\Delta Yt = \beta 0 + \beta 1 Yt\text{-}1 + \sum_{t=1}^{m} \alpha\, i \Delta Y(t-1) + ut$ |
| Type 2 | Constant, Trend | $\Delta Yt = \beta 0 + \beta 1 Yt\text{-}1 + \beta 2t + \sum_{t=1}^{m} \alpha\, i \Delta Y(t-1) + ut$ |

Where

ut = Error term; and

ΔYt-i = Lagged differences.

Number of lagged differences would be added in the model is often decided numerically, so that the residuals are not serially correlated. Moreover, there are several options for choosing lags: Minimize Akaike‟s Information Criterion (AIC) or Bayesian Information Criterion (BIC), or drop lags until the last lag is statistically significant.

ADF test with intercept was applied on both series to test the data for stationarity. The null hypothesis is tested through „t-Statistics‟ which is given by the following formula:

$$t = \frac{\hat{\delta} - \delta_{H_0}}{SE\ of\ \hat{\delta}} \quad \text{---------- (9)}$$

If ‘t’ calculated is greater than the critical value, we do not reject the null hypothesis and the series under consideration would be non-stationary and has a unit root. On the other hand, if ‘t’ calculated is less than the critical value, we reject the null hypothesis and the series under consideration would be stationary and does not have the unit root. First, the series should be tested on level and if it does not become stationary, then we should test the series at the first and second difference sequentially. ‘P’ – value is also used to reject or accept the null hypothesis. If the ‘P’ – value is less than 0.05 (P<0.05), reject the null hypothesis and vice-versa.

**ADF Test using R**

R-programming language uses adf.test to run ADF test to check stationarity of data. And results the output as

ADF Test on NSE stock closing price.

```
> adf.test(closing_price_NSE)

        Augmented Dickey-Fuller Test

data:  closing_price_NSE
Dickey-Fuller = -1.7303, Lag order = 10, p-value = 0.6925
alternative hypothesis: stationary
```

ADF Test on BSE stock closing price

```
> adf.test((closing_price_BSE))

        Augmented Dickey-Fuller Test

data:  (closing_price_BSE)
Dickey-Fuller = -1.5819, Lag order = 10, p-value = 0.7553
alternative hypothesis: stationary
```

As clearly depicted from the results of ADF test for both BBSE_CLOSE and NSE_Close,

**P-values are greater than 0.05**. Which means the variable is failed to reject the null hypothesis. And there is presence of unit root in the variable. Hence both BSE_CLOSE and NSE_Close are not stationary and there is impact of time on both the data sets.

To convert the series into stationary form, we need to difference the datasets. We will do the same using R as

NSE_re1 = diff(data_NSE$Close) &

BSE_re1 = diff(data_BSE$Close)

Again running ADF test to check the stationarity of differenced series,

ADF test on differenced NSE_close

```
> adf.test(NSE_re1)

        Augmented Dickey-Fuller Test

data:  NSE_re1
Dickey-Fuller = -8.8235, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

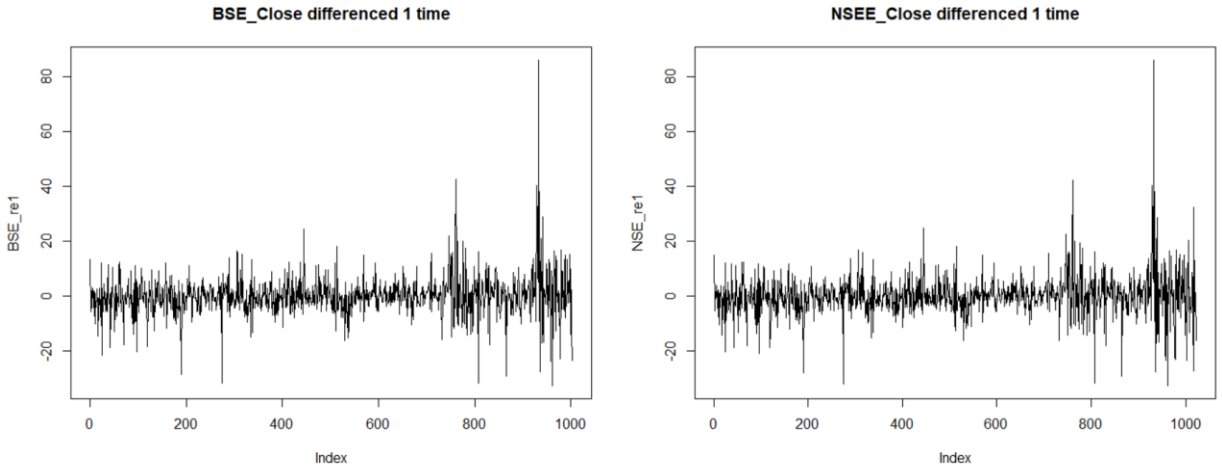
ADF Test on differenced BSE_CLOSE

```
> adf.test(BSE_re1)

        Augmented Dickey-Fuller Test

data:  BSE_re1
Dickey-Fuller = -8.758, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Now, both BSE_CLOSE and NSE_Close have a P-value less than 0.05. So, both the series are rejecting null hypothesis and becomes stationary after differenced one time.

Plotting differenced series for better understanding.

```
par(mfrow = c(1,2))
plot(BSE_re1, type = 'line', main = 'BSE_Close differenced 1 time')
plot(NSE_re1, type = 'line',main = 'NSEE_Close differenced 1 time')
```



# Correlogram Analysis

A Correlogram (also called Auto correlation function plot) is an image of correlation statistics and it gives a summary of correlation at different periods of time, i.e., serial correlation. Serial correlation is where an error at one point in time travels to a subsequent point of time. It is a commonly used tool for checking the randomness in a data set. A Correlogram contains Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). Autocorrelation refers to the way the observations in a time series are related to other and is measured by a simple correlation between current observation (Yt) and the observation „p" periods (lag p) from the current one (Yt-p) (Brooks, 2008) (Abdullah, 2012). The Autocorrelation coefficient at „lag p" is given by -

$$R_p = \frac{Cp}{Co} \text{ ---------- (10)}$$

where $C_p$ = the auto-covariance function; and

$C_0$ = the variance function.

$$c_p = \frac{1}{N}\sum_{t=1}^{N-p}(Y_t - \overline{Y})*(Y_{t+p} - \overline{Y}) \text{ -------- } (11) \text{ and}$$

$$c_0 = \frac{1}{N}\sum_{t=1}^{N}(Y_t - \overline{Y})^2 \text{ ---------- } (12).$$

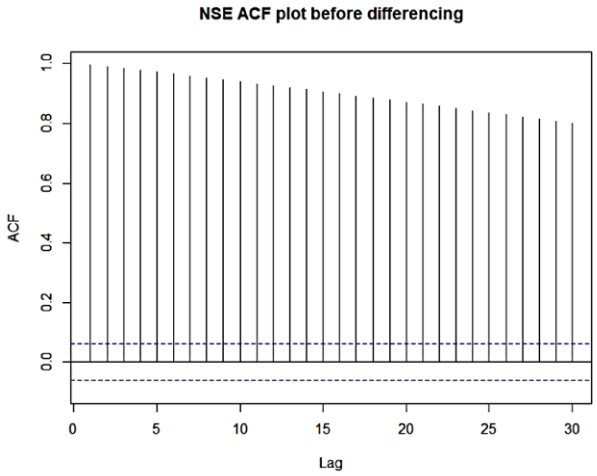
The resulting value of '$r_p$' will range between -1 and +1.

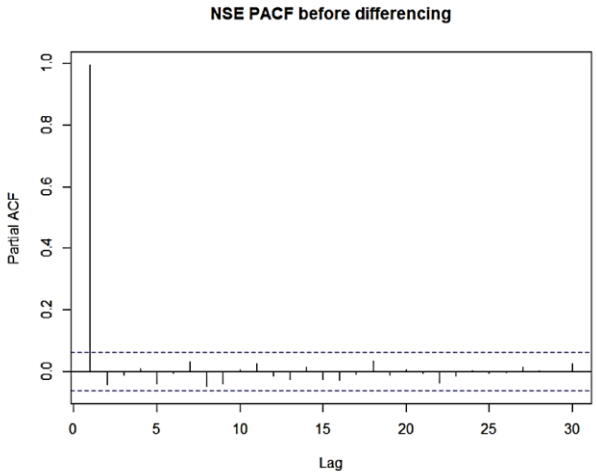
Partial Autocorrelations (PACF) are used to measure the degree of association between $Y_t$ and $Y_{t-p}$ when the effect of other time lags 1, 2, 3, ……, (p-1) are removed. The following figure No.2 represents the plot of Correlogram (ACF and PACF coefficients) of the time series BSE_CLOSE and NSE_CLOSE for lags 1 to 20 at the level (zero order difference). We may infer from the

Correlogram that, the ACF of BSE_CLOSE and NSE_CLOSE were dropped away very gradually; thus, the data in time series is non-stationary. Hence, there is a need to convert non-stationary series in to stationary by differencing.

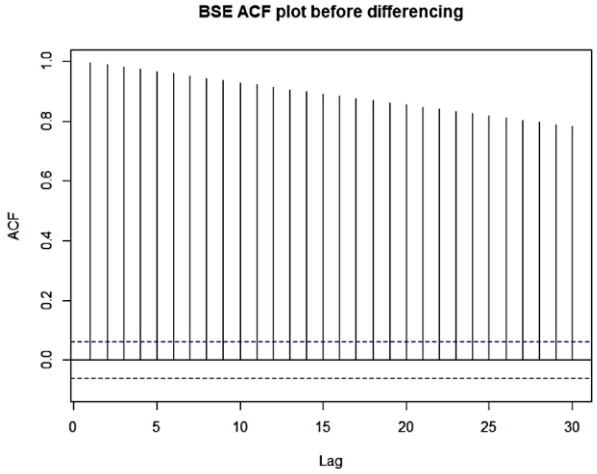
ACF and PACF before differencing the data for NSE_CLOSE and BSE_CLOSE

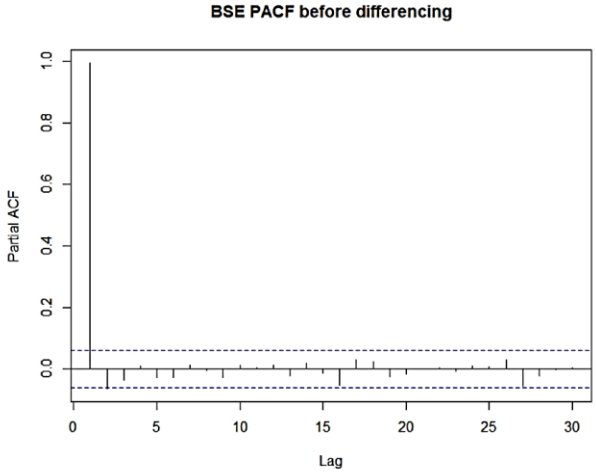
```
par(mfrow = c(1,2))
Acf(data_NSE$Close, main = 'NSE ACF Plot before differencing')
pacf(data_NSE$Close, main = 'NSE PACF plot before differencing')
```



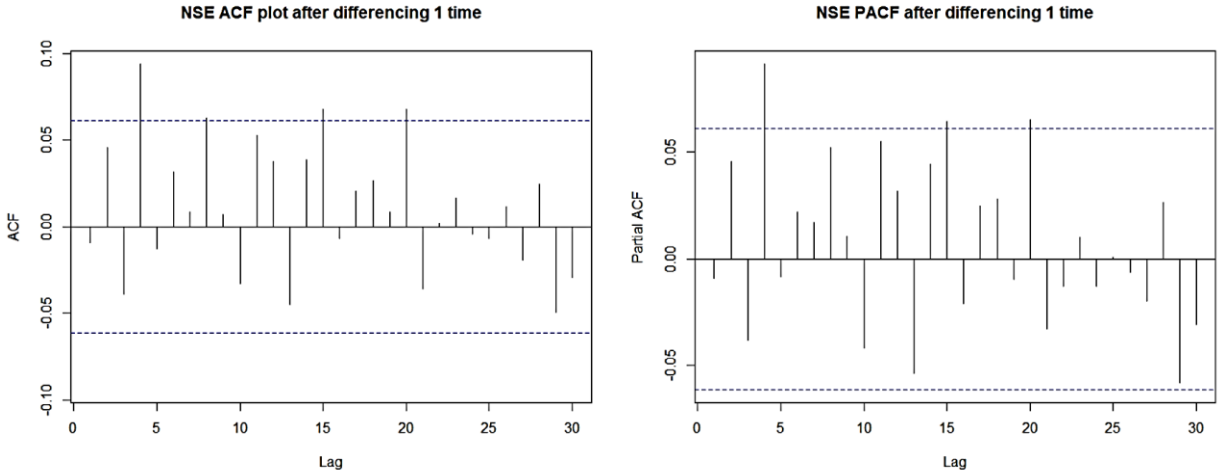

```
par(mfrow = c(1,2))
Acf(data_BSE$Close, main = 'BSE ACF plot before differencing')
pacf(data_BSE$Close, main = 'BSE PACF before differencing')
```



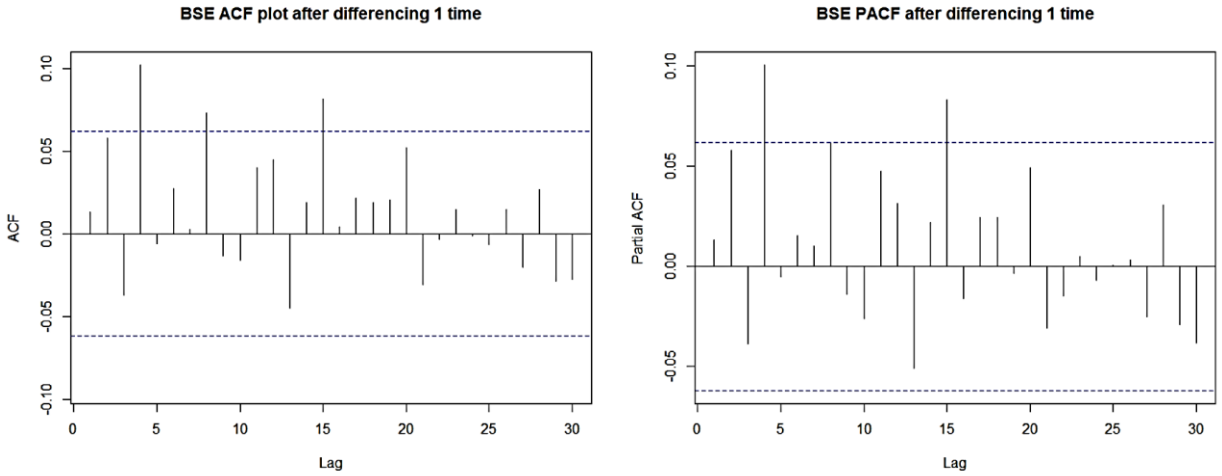



Below figure shows the spikes of Correlogram, auto correlation and partial auto correlation at the first order difference of the time series, viz., BSE_CLOSE and NSE_CLOSE.

```
par(mfrow = c(1,2))
Acf(NSE_re1, main = 'NSE ACF plot after differencing 1 time')
pacf(NSE_re1, main = 'NSE PACF after differencing 1 time')
```




```
par(mfrow = c(1,2))
Acf(BSE_re1, main = 'BSE ACF plot after differencing 1 time')
pacf(BSE_re1, main = 'BSE PACF after differencing 1 time')
```




The plots say that the first order difference of the data after transformation is random. If the model is fit, then the residuals of the model would contain the sequence of probable errors. Since spikes of ACFs and PACFs are insignificant, the residuals of the chosen ARIMA model are white noise, and, hence the time series data has become stationary.

The correlogram suggests that the transformed time series follow ARIMA (4,1,3) and ARIMA (3,1,2) model for NSE_CLOSE and BSE_CLOSE Respectively.

# ARIMA Model for forecasting

ARIMA model is the composition of series of steps for discovering the best model, supposing and identifying the different (ARIMA) models using available data in time series and forecasting the series using the best model. It is one of the well-known techniques for economic forecasting. ARIMA models are extremely capable to produce projections during short-term (Merh, Saxena and Pardasani, 2010). These are the best composite structural models, useful for short-term forecasts (Pai and Shenglin, 2005). In ARIMA model, the expected value of any variable is a "linear combination of past values and errors" (Hanke and Wichern, 2005), expressed as follows:

**Auto Regressive Model [AR(p)]**

An AR model is one in which '$Y_t$' depends only on its own past values, viz., $Y_{t-1}$, $Y_{t-2}$, $Y_{t-3}$, etc. Thus, $Y_t = f(Y_{t-1}, Y_{t-2}, Y_{t-3}, \ldots\ldots, \varepsilon_t)$. ---------- (13)

A common representation of an autoregressive model where it depends on „p" of its past values called AR(p) model is represented below:

$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-3} + \ldots\ldots + \beta_p Y_{t-p} + \varepsilon_t$---------- (14)

Where $Y_t$ = affecting (dependent) variable at time t.

$Y_{t-1}$, $Y_{t-2}$,……, $Y_{t-p}$ = Response variable at time lags t-1, t-2, ….., t-p, respectively.

$\beta_0$, $\beta_1$, $\beta_2$,……, $\beta_p$ = Coefficients to be estimated.

$\varepsilon_t$ = Error term at time t.

**Moving Average Model [MA(q)]**

A moving average model is one when Yt depends only on the random error terms whichfollow a white noise process, i.e., $Y_t = f(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \varepsilon_{t-3}, \ldots\ldots)$ ---------- (15)

A common representation of a moving average model where it depends on 'q' of its past values is called MA(q) model and is represented below:

$Y_t = \beta_0 + \varepsilon_t + \phi_1\varepsilon_{t-1} + \phi_2\varepsilon_{t-2} + \phi_3\varepsilon_{t-3} + \ldots\ldots + \phi_q\varepsilon_{t-q}$---------- (16)

The error terms εt is assumed to be white noise processes with mean zero and variance σ2.

Where Yt = Response variable (dependent) variable at time t.

$\beta_0$, = Constant mean of the process.

$\phi_1$, $\phi_2$, $\phi_3$,……, $\phi_q$ = coefficients to be estimated.

$\varepsilon_t$ = Error term at time t.

$\varepsilon_{t-1}$, $\varepsilon_{t-2}$, $\varepsilon_{t-3}$ ……. $\varepsilon_{t-q}$ = Errors in previous time periods that are incorporated in $Y_t$.

## Auto Regressive Moving Average (ARMA) Model

There are situations where the time series may be represented as a mix of both AR and MA models referred to as ARMA(p,q). The general form of such a time-series model, which depends on 'p' of its own past value and 'q' past values of white noise disturbances, taken the form:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-3} + \ldots\ldots + \beta_p Y_{t-p} + \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_2 \varepsilon_{t-2} + \phi_3 \varepsilon_{t-3} + \ldots\ldots + \phi_q \varepsilon_{t-q} \text{---- (17)}$$

## Selection of Appropriate ARIMA (p,d,q) Model

Model for non-seasonal series is called Autoregressive Integrated Moving Average Model, denoted by ARIMA (p,d,q). Here 'p' the order of autoregressive part, 'd' indicates the order of differencing, and 'q' indicates the order of moving average part. In general a series which is stationary after being differenced *'d times'* is said to be integrated of order 'd', denoted by *I(d)*. If the original series is stationary, *d=0* and the ARIMA models reduce to ARMA models. The time series data used for the present study, i.e., BSE_CLOSE and NSE_CLOSE has become stationary after the first order differencing. Since, there is no need for further differencing the series, it is necessary to adopt *d=1* (first difference) for ARIMA (p,d,q) model. To get the appropriate numbers for 'p' (in AR) and 'q' (in MA) in the model, we should check the Correlogram after first difference in time series

To choose one best ARIMA model amongst a numerous combinations present, the following criterions are used.

a. Comparatively low of Akaike/Bayesian/Schwarz Information Criterions (AIC/BIC).
b. Comparatively low S.E. of Regression.
c. Comparatively high adjusted R-square ($R^2$).
d. Root Mean Square Error (RMSE) should be relatively low.
e. Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) should be low.

**ARIMA models using R**

Running ARIMA models on NSE_CLOSE data.

```
Arima1 <- auto.arima(data_NSE$Close, seasonal = FALSE)
summary(Arima1)
```

```
> summary(Arima1)
Series: data_NSE$Close
ARIMA(1,2,0)

Coefficients:
          ar1
      -0.5270
s.e.   0.0267

sigma^2 = 84.93:  log likelihood = -3719.59
AIC=7443.17   AICc=7443.19   BIC=7453.03

Training set error measures:
                      ME     RMSE      MAE           MPE     MAPE     MASE        ACF1
Training set -0.03466565 9.202217 6.230204 -0.003191914 2.785304 1.27246 -0.1498527
```

```
Arima2 <- arima(data_NSE$Close, order = c(0,1,0))
summary(Arima2)
```

```
> summary(Arima2)

Call:
arima(x = data_NSE$Close, order = c(0, 1, 0))


sigma^2 estimated as 58.29:  log likelihood = -3531.06,  aic = 7064.13

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE        ACF1
Training set 0.05295355 7.631228 4.891821 -0.04048764 2.205913 0.9991081 -0.0090404
```

```
Arima3 <- arima(data_NSE$Close, order = c(1,1,0))
summary(Arima3)
```

```
> summary(Arima3)

Call:
arima(x = data_NSE$Close, order = c(1, 1, 0))

Coefficients:
          ar1
      -0.0091
s.e.   0.0314

sigma^2 estimated as 58.29:  log likelihood = -3531.02,  aic = 7066.04

Training set error measures:
                     ME     RMSE     MAE          MPE     MAPE      MASE          ACF1
Training set 0.05357833 7.630911 4.88897 -0.04084363 2.204878 0.9985259 0.0004833286
```

```
Arima4 <- arima(data_NSE$Close, order = c(0,1,1))
summary(Arima4)
```

```
> summary(Arima4)

Call:
arima(x = data_NSE$Close, order = c(0, 1, 1))

Coefficients:
          ma1
      -0.0084
s.e.   0.0301

sigma^2 estimated as 58.29:  log likelihood = -3531.02,  aic = 7066.05

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE      MASE          ACF1
Training set 0.05353122 7.630938 4.889189 -0.0408172 2.204959 0.9985706 -0.0003106329
```

```
Arima5 <- arima(data_NSE$Close, order = c(1,1,1))
summary(Arima5)
```

```
> summary(Arima5)

Call:
arima(x = data_NSE$Close, order = c(1, 1, 1))

Coefficients:
          ar1     ma1
      -0.8916  0.8555
s.e.   0.1038  0.1181

sigma^2 estimated as 57.93:  log likelihood = -3527.91,  aic = 7061.82

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE       ACF1
Training set 0.05488383 7.607665 4.871203 -0.04106446 2.201878 0.9948971 0.03343572
```

```
Arima6 <- arima(data_NSE$Close, order = c(2,1,0))
summary(Arima6)
```

```
> summary(Arima6)

Call:
arima(x = data_NSE$Close, order = c(2, 1, 0))

Coefficients:
          ar1     ar2
      -0.0088  0.0458
s.e.   0.0313  0.0313

sigma^2 estimated as 58.17:  log likelihood = -3529.95,  aic = 7065.9

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE       ACF1
Training set 0.05047602 7.622928 4.885563 -0.03879163 2.204339 0.99783 0.00187732
```

```
Arima7 <- arima(data_NSE$Close, order = c(2,1,1))
summary(Arima7)
```

```
> summary(Arima7)

Call:
arima(x = data_NSE$Close, order = c(2, 1, 1))

Coefficients:
          ar1     ar2     ma1
      -0.7542  0.0535  0.7515
s.e.   0.1335  0.0345  0.1306

sigma^2 estimated as 57.81:  log likelihood = -3526.83,  aic = 7061.66

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE      MASE         ACF1
Training set 0.05173542 7.599607 4.881471 -0.0390796 2.208603 0.9969942 0.001104118
```

```
Arima8 <- arima(data_NSE$Close, order = c(0,1,2))
summary(Arima8)
```

```
> summary(Arima8)

Call:
arima(x = data_NSE$Close, order = c(0, 1, 2))

Coefficients:
          ma1     ma2
      -0.0060  0.0384
s.e.   0.0315  0.0289

sigma^2 estimated as 58.19:  log likelihood = -3530.14,  aic = 7066.28

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE          ACF1
Training set 0.05085934 7.624337 4.886427 -0.03906471 2.204366 0.9980065 -0.001288154
```

```
Arima9 <- arima(data_NSE$Close, order = c(1,1,2))
summary(Arima9)
```

```
> summary(Arima9)

Call:
arima(x = data_NSE$Close, order = c(1, 1, 2))

Coefficients:
          ar1     ma1     ma2
      -0.8057  0.8055  0.0556
s.e.   0.1119  0.1144  0.0346

sigma^2 estimated as 57.8:  log likelihood = -3526.75,  aic = 7061.49

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE          ACF1
Training set 0.05169666 7.598982 4.882382 -0.03905702 2.209521 0.9971804 -0.001251682
```

```
Arima10 <- arima(data_NSE$Close, order = c(2,1,2))
summary(Arima10)
```

```
> summary(Arima10)

Call:
arima(x = data_NSE$Close, order = c(2, 1, 2))

Coefficients:
         ar1     ar2      ma1      ma2
      0.0633  0.8162  -0.0811  -0.7538
s.e.  0.1108  0.1073   0.1254   0.1212

sigma^2 estimated as 57.63:  log likelihood = -3525.21,  aic = 7060.42

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE        ACF1
Training set 0.0340419 7.587473 4.897603 -0.02045031 2.213363 1.000289 0.01083587
```

```
Arima11 <- arima(data_NSE$Close, order = c(3,1,0))
summary(Arima11)
```

```
> summary(Arima11)

Call:
arima(x = data_NSE$Close, order = c(3, 1, 0))

Coefficients:
          ar1     ar2      ar3
      -0.0071  0.0456  -0.0383
s.e.   0.0314  0.0313   0.0313

sigma^2 estimated as 58.08:  log likelihood = -3529.21,  aic = 7066.41

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE         ACF1
Training set 0.05320608 7.617364 4.883467 -0.04050059 2.206858 0.9974018 0.003581677
```

```
Arima12 <- arima(data_NSE$Close, order = c(0,1,3))
summary(Arima12)
```

```
> summary(Arima12)

Call:
arima(x = data_NSE$Close, order = c(0, 1, 3))

Coefficients:
          ma1     ma2      ma3
      -0.0008  0.0382  -0.0362
s.e.   0.0318  0.0288   0.0310

sigma^2 estimated as 58.11:  log likelihood = -3529.46,  aic = 7066.92

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE          ACF1
Training set 0.05306919 7.619251 4.885496 -0.04049951 2.207222 0.9978162 -0.003272269
```

```
Arima13 <- arima(data_NSE$Close, order = c(3,1,1))
summary(Arima13)
```

```
> summary(Arima13)

Call:
arima(x = data_NSE$Close, order = c(3, 1, 1))

Coefficients:
          ar1     ar2      ar3     ma1
      -0.6801  0.0394  -0.0299  0.6785
s.e.   0.1654  0.0379   0.0372  0.1627

sigma^2 estimated as 57.77:  log likelihood = -3526.51,  aic = 7063.02

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE          ACF1
Training set 0.05347348 7.597224 4.881248 -0.04016273 2.210656 0.9969487 0.0009050626
```

```
Arima14 <- arima(data_NSE$Close, order = c(1,1,3))
summary(Arima14)
```

```
> summary(Arima14)

Call:
arima(x = data_NSE$Close, order = c(1, 1, 3))

Coefficients:
          ar1     ma1     ma2      ma3
      -0.7372  0.7379  0.0399  -0.0320
s.e.   0.1422  0.1434  0.0371   0.0349

sigma^2 estimated as 57.76:  log likelihood = -3526.35,  aic = 7062.7

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE      MASE          ACF1
Training set 0.0534729 7.596035 4.883275 -0.0401623 2.212209 0.9973626 -0.001089464
```

```
Arima15 <- arima(data_NSE$Close, order = c(3,1,2))
summary(Arima15)
```

```
> summary(Arima15)

Call:
arima(x = data_NSE$Close, order = c(3, 1, 2))

Coefficients:
          ar1      ar2      ar3     ma1     ma2
      -1.0197  -0.3167  -0.0301  1.0185  0.3467
s.e.   1.1850   0.9645   0.0627  1.1845  0.9569

sigma^2 estimated as 57.72:  log likelihood = -3526.07,  aic = 7064.14

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE          ACF1
Training set 0.05378008 7.593938 4.885201 -0.04039689 2.213304 0.997756 0.0009272708
```

```
Arima16 <- arima(data_NSE$Close, order = c(2,1,3))
summary(Arima16)
```

```
> summary(Arima16)

Call:
arima(x = data_NSE$Close, order = c(2, 1, 3))

Coefficients:
          ar1      ar2     ma1     ma2      ma3
      -1.2166  -0.4326  1.2163  0.4669  -0.0151
s.e.   0.5002   0.4047  0.4996  0.4106   0.0504

sigma^2 estimated as 57.71:  log likelihood = -3525.93,  aic = 7063.85

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE           ACF1
Training set 0.05341439 7.592876 4.886116 -0.04004798 2.214682 0.9979429 -8.465131e-05
```

```
Arima17 <- arima(data_NSE$Close, order = c(3,1,3))
summary(Arima17)
```

```
> summary(Arima17)

Call:
arima(x = data_NSE$Close, order = c(3, 1, 3))

Coefficients:
          ar1     ar2     ar3     ma1      ma2      ma3
      -0.5136  0.7790  0.5729  0.5187  -0.7158  -0.5770
s.e.   0.3385  0.0922  0.2758  0.3304   0.1037   0.2481

sigma^2 estimated as 57.49:  log likelihood = -3524,  aic = 7062

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE     MASE          ACF1
Training set 0.03355404 7.578473 4.897084 -0.0191827 2.217593 1.000183 -0.009751374
```

Running ARIMA models on BSE_CLOSE data.

```
BSE_Arima1 <- auto.arima(data_BSE$Close, seasonal = FALSE)
summary(BSE_Arima1)
```

```
> summary(BSE_Arima1)
Series: data_BSE$Close
ARIMA(1,2,0)

Coefficients:
          ar1
      -0.5234
s.e.   0.0270

sigma^2 = 79.89:  log likelihood = -3616.17
AIC=7236.34   AICc=7236.35   BIC=7246.16

Training set error measures:
                     ME     RMSE      MAE           MPE     MAPE     MASE        ACF1
Training set -0.04844191 8.925027 6.053866 -0.004984484 2.767831 1.267342 -0.1430304
```

```
BSE_Arima2 <- arima(data_BSE$Close, order = c(0,1,0))
summary(BSE_Arima2)
```

```
> summary(BSE_Arima2)

Call:
arima(x = data_BSE$Close, order = c(0, 1, 0))


sigma^2 estimated as 55.97:  log likelihood = -3441.64,  aic = 6885.28

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE        ACF1
Training set 0.05336195 7.47759 4.772486 -0.04024536 2.194664 0.9990927 0.01331496
```

```
BSE_Arima3 <- arima(data_BSE$Close, order = c(1,1,0))
summary(BSE_Arima3)
```

```
> summary(BSE_Arima3)

Call:
arima(x = data_BSE$Close, order = c(1, 1, 0))

Coefficients:
         ar1
      0.0134
s.e.  0.0318

sigma^2 estimated as 55.96:  log likelihood = -3441.55,  aic = 6887.1

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE     MASE          ACF1
Training set 0.05233482 7.476921 4.776879 -0.0397316 2.196557 1.000012 -0.0007761983
```

```
BSE_Arima4 <- arima(data_BSE$Close, order = c(0,1,1))
summary(BSE_Arima4)
```

```
> summary(BSE_Arima4)

Call:
arima(x = data_BSE$Close, order = c(0, 1, 1))

Coefficients:
        ma1
      0.012
s.e.  0.030

sigma^2 estimated as 55.96:  log likelihood = -3441.56,  aic = 6887.12

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE          ACF1
Training set 0.05245659 7.476992 4.776386 -0.03979178 2.196332 0.9999092 0.0007101291
```

```
BSE_Arima5 <- arima(data_BSE$Close, order = c(1,1,1))
summary(BSE_Arima5)
```

```
> summary(BSE_Arima5)

Call:
arima(x = data_BSE$Close, order = c(1, 1, 1))

Coefficients:
          ar1     ma1
      -0.9074  0.8741
s.e.   0.0741  0.0851

sigma^2 estimated as 55.62:  log likelihood = -3438.52,  aic = 6883.04

Training set error measures:
                     ME    RMSE     MAE          MPE     MAPE      MASE       ACF1
Training set 0.05494972 7.45427 4.75863 -0.04079939 2.192952 0.9961921 0.05390515
```

```
BSE_Arima6 <- arima(data_BSE$Close, order = c(2,1,0))
summary(BSE_Arima6)
```

```
> summary(BSE_Arima6)

Call:
arima(x = data_BSE$Close, order = c(2, 1, 0))

Coefficients:
         ar1     ar2
      0.0129  0.0587
s.e.  0.0317  0.0318

sigma^2 estimated as 55.77:  log likelihood = -3439.84,  aic = 6885.69

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE        ACF1
Training set 0.04706951 7.464183 4.778665 -0.03737408 2.198996 1.000386 0.00234242
```

```
BSE_Arima7 <- arima(data_BSE$Close, order = c(2,1,1))
summary(BSE_Arima7)
```

```
> summary(BSE_Arima7)

Call:
arima(x = data_BSE$Close, order = c(2, 1, 1))

Coefficients:
          ar1     ar2     ma1
      -0.7407  0.0786  0.7614
s.e.   0.1158  0.0339  0.1126

sigma^2 estimated as 55.35:  log likelihood = -3436.07,  aic = 6880.14

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE         ACF1
Training set 0.04929327 7.436045 4.77755 -0.03786902 2.204964 1.000153 0.001058264
```

```
BSE_Arima8 <- arima(data_BSE$Close, order = c(0,1,2))
summary(BSE_Arima8)
```

```
> summary(BSE_Arima8)

Call:
arima(x = data_BSE$Close, order = c(0, 1, 2))

Coefficients:
         ma1     ma2
      0.0168  0.0502
s.e.  0.0320  0.0291

sigma^2 estimated as 55.8:  log likelihood = -3440.08,  aic = 6886.15

Training set error measures:
                     ME    RMSE      MAE          MPE     MAPE     MASE          ACF1
Training set 0.04791785 7.46592 4.778814 -0.03776388 2.198832 1.000418 -0.001903818
```

```
BSE_Arima9 <- arima(data_BSE$Close, order = c(1,1,2))
summary(BSE_Arima9)
```

```
> summary(BSE_Arima9)

Call:
arima(x = data_BSE$Close, order = c(1, 1, 2))

Coefficients:
          ar1     ma1     ma2
      -0.8184  0.8416  0.0799
s.e.   0.0979  0.1010  0.0339

sigma^2 estimated as 55.34:  log likelihood = -3435.96,  aic = 6879.91

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE         ACF1
Training set 0.04962593 7.435196 4.77845 -0.03797426 2.205982 1.000341 -0.00118121
```

```
BSE_Arima10 <- arima(data_BSE$Close, order = c(2,1,2))
summary(BSE_Arima10)
```

```
> summary(BSE_Arima10)

Call:
arima(x = data_BSE$Close, order = c(2, 1, 2))

Coefficients:
         ar1     ar2      ma1      ma2
      0.0299  0.8189  -0.0375  -0.7517
s.e.  0.0969  0.0942   0.1111   0.1078

sigma^2 estimated as 55.21:  log likelihood = -3434.86,  aic = 6879.72

Training set error measures:
                     ME     RMSE    MAE          MPE     MAPE     MASE        ACF1
Training set 0.03146921 7.426973 4.7913 -0.02079278 2.204641 1.003031 0.02311918
```

```
BSE_Arima11 <- arima(data_BSE$Close, order = c(3,1,0))
summary(BSE_Arima11)
```

```
> summary(BSE_Arima11)

Call:
arima(x = data_BSE$Close, order = c(3, 1, 0))

Coefficients:
         ar1     ar2      ar3
      0.0151  0.0590  -0.0388
s.e.  0.0317  0.0317   0.0318

sigma^2 estimated as 55.69:  log likelihood = -3439.1,  aic = 6886.2

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE        ACF1
Training set 0.05058443 7.458619 4.780052 -0.03895186 2.203633 1.000677 0.00393478
```

```
BSE_Arima12 <- arima(data_BSE$Close, order = c(0,1,3))
summary(BSE_Arima12)
```

```
> summary(BSE_Arima12)

Call:
arima(x = data_BSE$Close, order = c(0, 1, 3))

Coefficients:
         ma1     ma2      ma3
      0.0229  0.0495  -0.0401
s.e.  0.0322  0.0290   0.0318

sigma^2 estimated as 55.71:  log likelihood = -3439.28,  aic = 6886.56

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE          ACF1
Training set 0.0510254 7.459979 4.781888 -0.03914213 2.204446 1.001061 -0.003988487
```

```
BSE_Arima13 <- arima(data_BSE$Close, order = c(3,1,1))
summary(BSE_Arima13)
```

```
> summary(BSE_Arima13)

Call:
arima(x = data_BSE$Close, order = c(3, 1, 1))

Coefficients:
          ar1     ar2      ar3     ma1
      -0.6915  0.0680  -0.0212  0.7132
s.e.   0.1543  0.0387   0.0388  0.1513

sigma^2 estimated as 55.33:  log likelihood = -3435.93,  aic = 6881.85

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE         ACF1
Training set 0.05089571 7.434978 4.777895 -0.03852499 2.206304 1.000225 0.0006947777
```

```
BSE_Arima14 <- arima(data_BSE$Close, order = c(1,1,3))
summary(BSE_Arima14)
```

```
> summary(BSE_Arima14)

Call:
arima(x = data_BSE$Close, order = c(1, 1, 3))

Coefficients:
          ar1     ma1     ma2      ma3
      -0.7820  0.8049  0.0683  -0.0196
s.e.   0.1297  0.1319  0.0391   0.0356

sigma^2 estimated as 55.32:  log likelihood = -3435.81,  aic = 6881.62

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE         ACF1
Training set 0.05110131 7.434099 4.779168 -0.03856208 2.207305 1.000492 -0.00047913
```

```
BSE_Arima15 <- arima(data_BSE$Close, order = c(3,1,2))
summary(BSE_Arima15)
```

```
> summary(BSE_Arima15)

Call:
arima(x = data_BSE$Close, order = c(3, 1, 2))

Coefficients:
        ar1     ar2      ar3      ma1      ma2
      0.162  0.7862  -0.0466  -0.1461  -0.7128
s.e.  0.137  0.1000   0.0378   0.1340   0.1113

sigma^2 estimated as 55.15:  log likelihood = -3434.24,  aic = 6880.48

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE         ACF1
Training set 0.03125992 7.422388 4.791507 -0.01852379 2.207892 1.003075 0.001506464
```

```
BSE_Arima16 <- arima(data_BSE$Close, order = c(2,1,3))
summary(BSE_Arima16)
```

```
> summary(BSE_Arima16)

Call:
arima(x = data_BSE$Close, order = c(2, 1, 3))

Coefficients:
         ar1     ar2      ma1      ma2      ma3
      0.0965  0.7939  -0.0774  -0.7211  -0.0462
s.e.  0.1082  0.0948   0.1110   0.1066   0.0379

sigma^2 estimated as 55.13:  log likelihood = -3434.13,  aic = 6880.26

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE          ACF1
Training set 0.03181385 7.421563 4.79222 -0.01935381 2.208189 1.003224 -0.001609954
```

```
BSE_Arima17 <- arima(data_BSE$Close, order = c(3,1,3))
summary(BSE_Arima17)
```

```
> summary(BSE_Arima17)

Call:
arima(x = data_BSE$Close, order = c(3, 1, 3))

Coefficients:
          ar1     ar2     ar3     ma1      ma2      ma3
      -0.3509  0.7850  0.4001  0.3731  -0.7127  -0.4242
s.e.   0.4244  0.0922  0.3367  0.4157   0.1042   0.2997

sigma^2 estimated as 55.06:  log likelihood = -3433.49,  aic = 6880.98

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE     MASE          ACF1
Training set 0.03184591 7.416831 4.792662 -0.01887342 2.209486 1.003316 -0.004749775
```

For calculating the value of $R^2$ value for different ARIMA models we have used

$R^2$ = cor(fitted(Arima_model),closing_point)$^2$

And for calculating BIC for different ARIMA models, we have used

BIC = $(p+q+1)*\log_n(N) - 2*$logliklihood

Where p and q are from ARIMA (p,d,q) and N is the number of data points taken for running ARIMA model.

Below tables provide result for different AR(p) and MA(q) of ARIMA model. Using these values, the best fit model for predicting the time series BSE_CLOSE and NSE_CLOSE will be identified.

| Output for various ARIMA parameters for BSE_Close | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | AIC | BIC | RMSE | MAE | MAPE | Standard error of Regression | R^2 |
| (1,2,0) | 7236.34 | 7246.16 | 8.925027 | 6.053866 | 2.767831 | 1.267342 | 0.993685 |
| (0,1,0) | 6885.28 | 6896.212 | 7.47759 | 4.772486 | 2.194664 | 0.9990927 | 0.99553 |
| (1.1.0) | 6887.1 | 6896.965 | 7.476921 | 4.776879 | 2.196557 | 1.000012 | 0.995531 |
| (0.1.1) | 6887.12 | 6896.984 | 7.476992 | 4.776386 | 2.196332 | 0.9999092 | 0.995531 |
| (1,1,1) | 6883.04 | 6897.837 | 7.45427 | 4.75863 | 2.192952 | 0.9961921 | 0.995558 |
| (2,1,0) | 6885.69 | 6900.484 | 7.464183 | 4.778665 | 2.198996 | 1.000386 | 0.995547 |
| (2,1,1) | 6880.14 | 6899.87 | 7.436045 | 4.77755 | 2.204964 | 1.000153 | 0.99558 |
| (0,1,2) | 6886.15 | 6900.949 | 7.46592 | 4.778814 | 2.198832 | 1.000418 | 0.995545 |
| (1,1,2) | 6879.91 | 6899.642 | 7.435196 | 4.77845 | 2.205982 | 1.000341 | 0.995581 |
| (2,1,2) | 6879.72 | 6904.379 | 7.426973 | 4.7913 | 2.204641 | 1.003031 | 0.995595 |
| (3,1,0) | 6886.2 | 6905.925 | 7.458619 | 4.780052 | 2.203633 | 1.000677 | 0.995553 |
| (0,1,3) | 6886.56 | 6906.29 | 7.459979 | 4.881888 | 2.204446 | 1.001061 | 0.995552 |
| (3,1,1) | 6881.85 | 6906.516 | 7.434978 | 4.777895 | 2206304 | 1.000225 | 0.995581 |
| (1,1,3) | 6881.62 | 6906.281 | 7.434099 | 4.779168 | 2.207305 | 1.000492 | 0.995582 |
| (3,1,2) | 6880.48 | 6910.074 | 7.422388 | 4.791507 | 2.207892 | 1.003075 | 0.995601 |
| (2,1,3) | 6880.26 | 6909.852 | 7.421563 | 4.79222 | 2.208189 | 1.003224 | 0.995602 |
| (3,1,3) | 6880.98 | 6915.511 | 7.416831 | 4.792662 | 2.209486 | 1.003316 | 0.995608 |

Table No. - 1

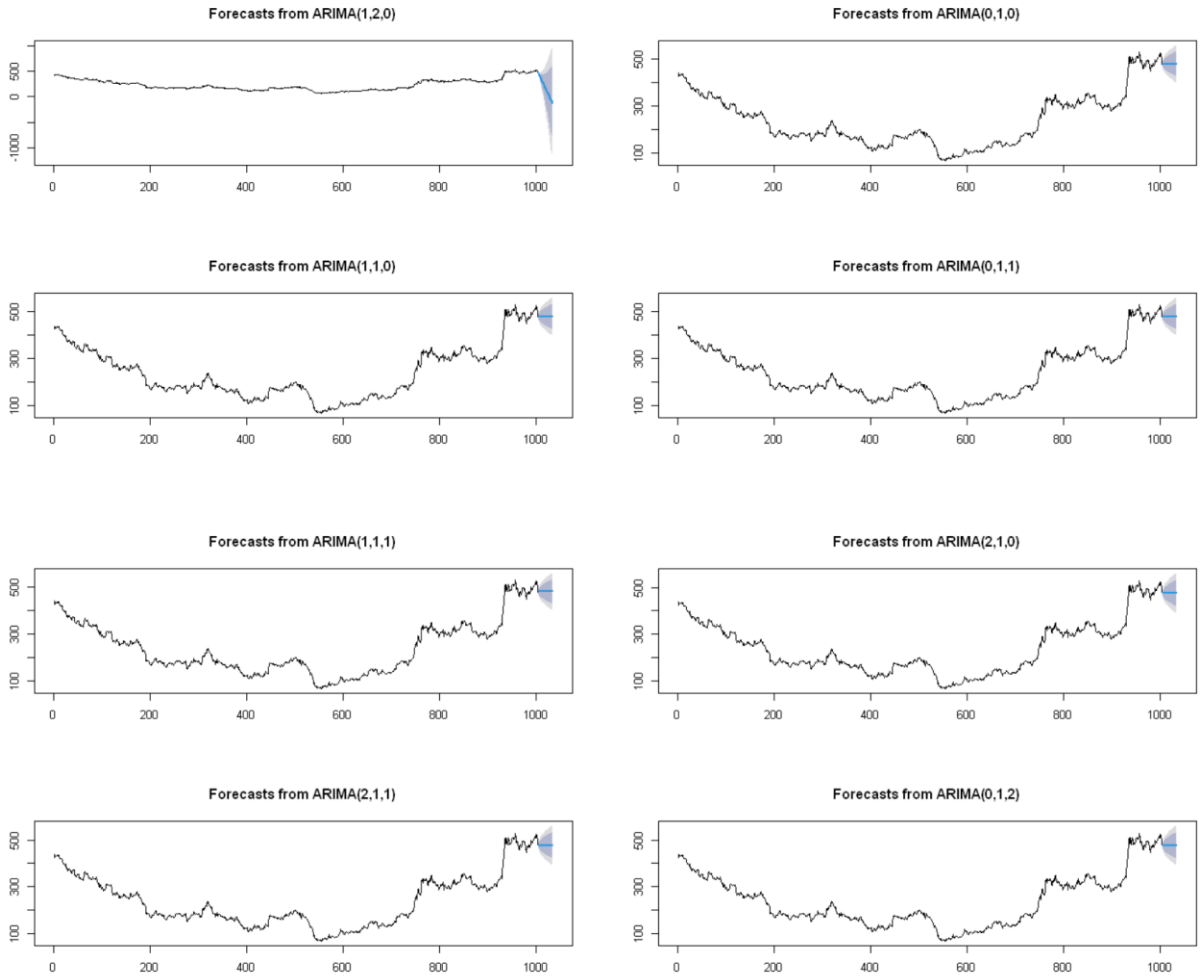
| Output for various ARIMA parameters for NSE_Close | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | AIC | BIC | RMSE | MAE | MAPE | Standard error of Regression | R^2 |
| (1,2,0) | 7443.17 | 7453.039 | 9.202217 | 6.230204 | 2.785304 | 1.27246 | 0.99381 |
| (0,1,0) | 7064.13 | 7069.058 | 7.631228 | 4.891821 | 2.205913 | 0.9991081 | 0.995713 |
| (1.1.0) | 7066.04 | 7075.906 | 7.630911 | 4.88897 | 2.204878 | 0.9985259 | 0.995713 |
| (0.1.1) | 7066.05 | 7075.913 | 7.630938 | 4.889189 | 2.204959 | 0.9985706 | 0.995713 |
| (1,1,1) | 7061.82 | 7076.619 | 7.607665 | 4.871203 | 2.201878 | 0.9948971 | 0.995739 |
| (2,1,0) | 7065.9 | 7080.701 | 7.622928 | 4.885563 | 2.204339 | 0.99783 | 0.995723 |
| (2,1,1) | 7061.66 | 7081.388 | 7.599607 | 4.881471 | 2.208603 | 0.9969942 | 0.995749 |
| (0,1,2) | 7066.28 | 7081.078 | 7.624337 | 4.886427 | 2.204366 | 0.9980065 | 0.995721 |
| (1,1,2) | 7061.49 | 7081.22 | 7.598982 | 4.882382 | 2.209521 | 0.9971804 | 0.995749 |
| (2,1,2) | 7060.42 | 7085.078 | 7.587473 | 4.897603 | 2.213363 | 1.000289 | 0.995766 |
| (3,1,0) | 7066.41 | 7086.144 | 7.617364 | 4.883467 | 2.206858 | 0.9974018 | 0.995729 |
| (0,1,3) | 7066.92 | 7086.649 | 7.619251 | 4.885496 | 2.207222 | 0.9978162 | 0.995726 |
| (3,1,1) | 7063.02 | 7087.681 | 7.597224 | 4.881248 | 2.210656 | 0.9969487 | 0.995751 |
| (1,1,3) | 7062.7 | 7087.363 | 7.596035 | 4.883275 | 2.212209 | 0.9973626 | 0.995755 |
| (3,1,2) | 7064.14 | 7093.73 | 7.593938 | 4.885201 | 2.213304 | 0.997756 | 0.995756 |
| (2,1,3) | 7063.85 | 7093.448 | 7.592876 | 4.886116 | 2.214682 | 0.9979429 | 0.995756 |
| (3,1,3) | 7062 | 7096.525 | 7.578473 | 4.897084 | 2.217593 | 1.000183 | 0.995776 |

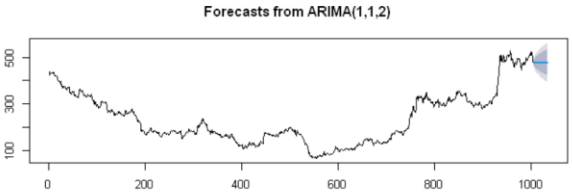
Table No. - 2

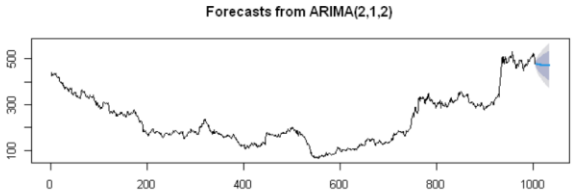
After checking the robustness of the statistics given in the above tables 1 and 2, it is found that **ARIMA model (1,1,1)** convinces most of the norms (lowest AIC, BIC, RMSE, MAE, MAPE, Standard Error of Regression, and the relatively high Adjusted $R_2$ values)……..(A)

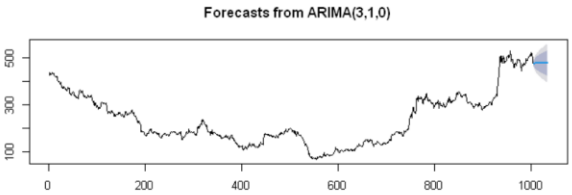
**Forecasting using different models to predict best model.**

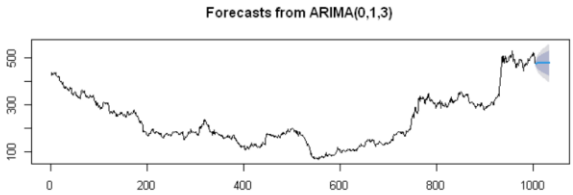
Let us plot different BSE_CLOSE ARIMA (p,d,q) models with actual share price to find best model for prediction.

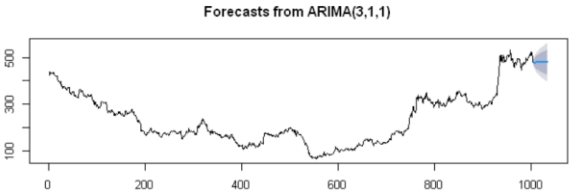


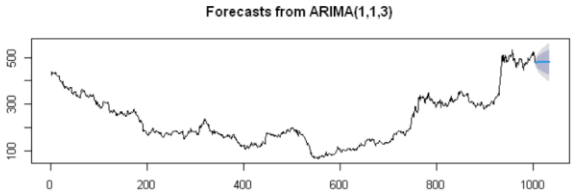

Forecasts from ARIMA(1,1,2)
100
300
500
0
200
400
600
800
1000

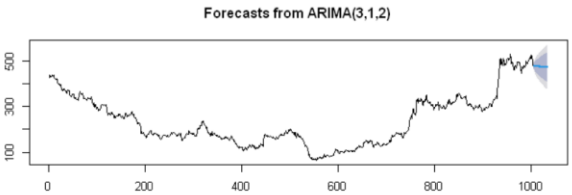

Forecasts from ARIMA(2,1,2)
100
300
500
0
200
400
600
800
1000

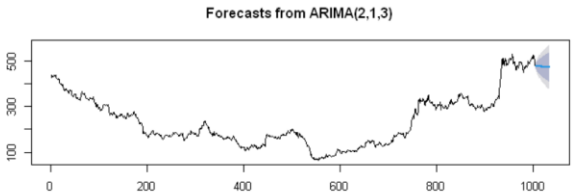

Forecasts from ARIMA(3,1,0)
100
300
500
0
200
400
600
800
1000

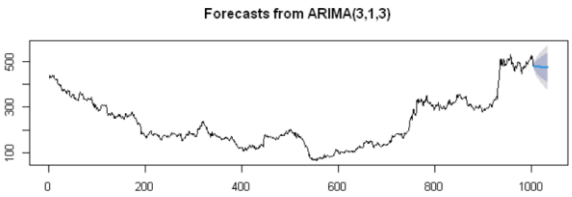

Forecasts from ARIMA(0,1,3)
100
300
500
0
200
400
600
800
1000


Forecasts from ARIMA(3,1,1)
100
300
500
0
200
400
600
800
1000

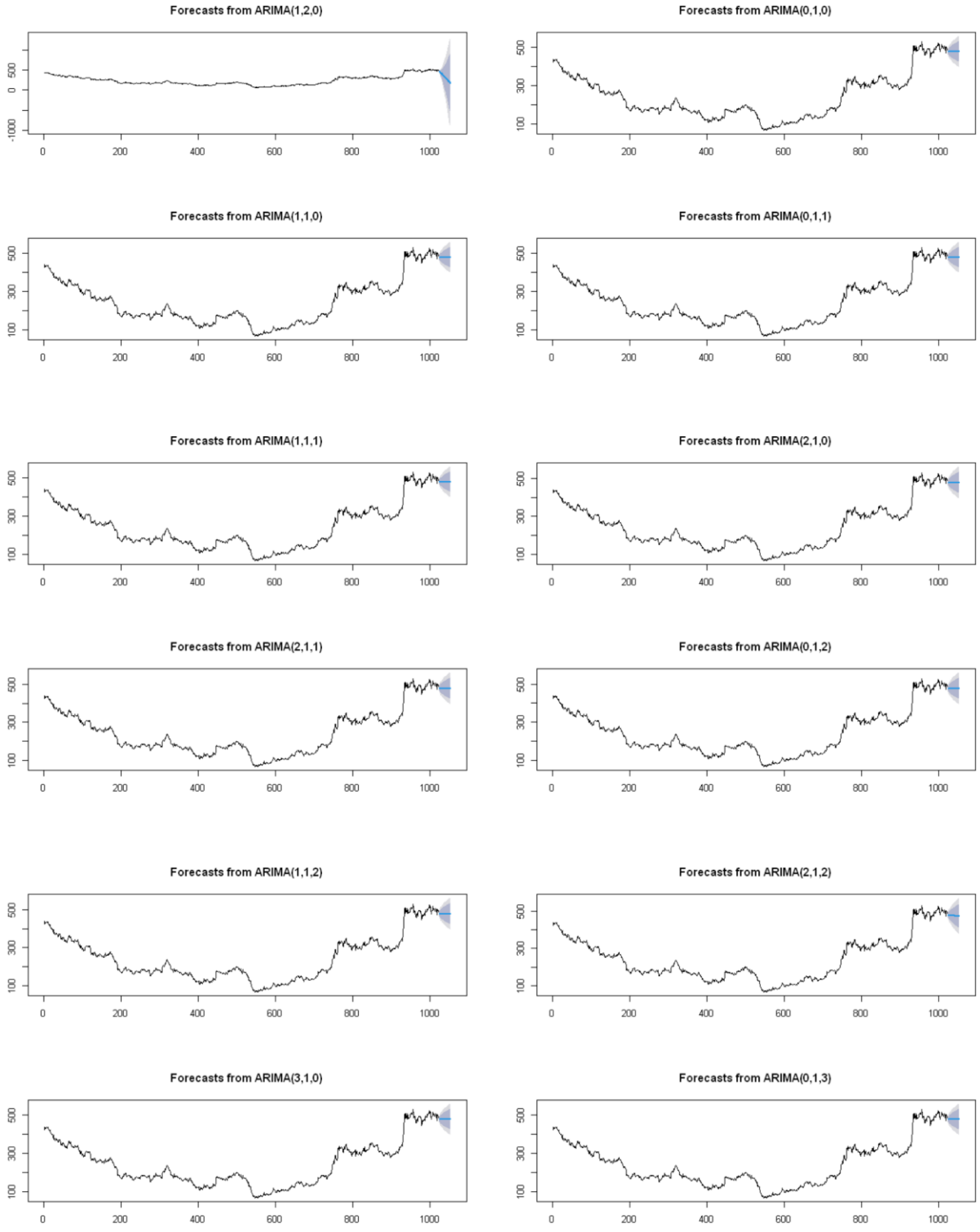

Forecasts from ARIMA(1,1,3)
100
300
500
0
200
400
600
800
1000

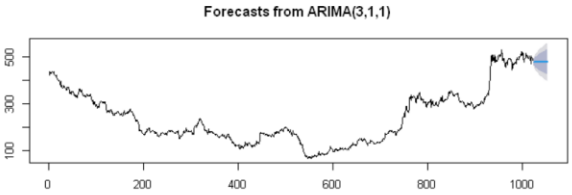

Forecasts from ARIMA(3,1,2)
100
300
500
0
200
400
600
800
1000

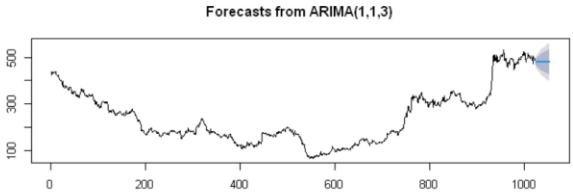

Forecasts from ARIMA(2,1,3)
100
300
500
0
200
400
600
800
1000

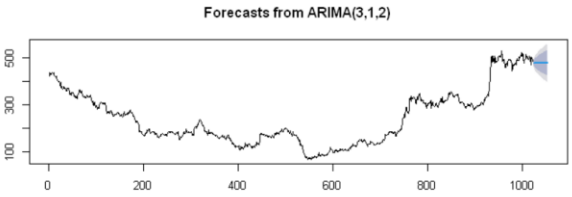

Forecasts from ARIMA(3,1,3)
100
300
500
0
200
400
600
800
1000

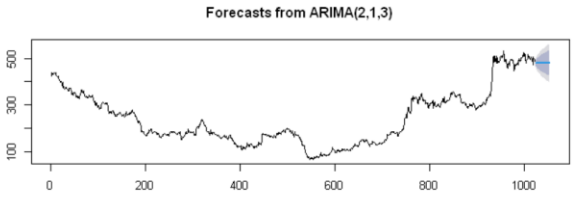
Let us plot different NSE_CLOSE ARIMA (p,d,q) models with actual share price to find best model for prediction.

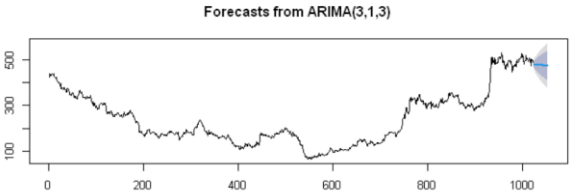



Forecasts from ARIMA(3,1,1)
100
300
500
0
200
400
600
800
1000

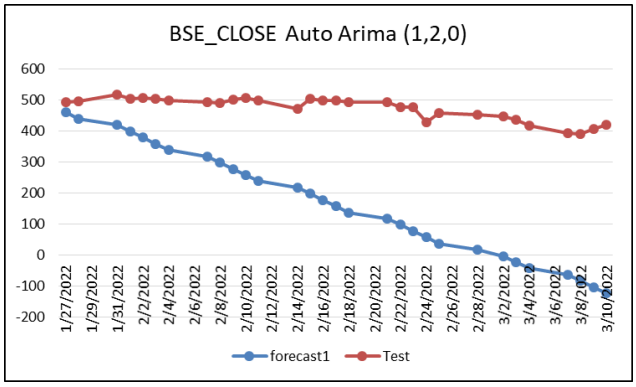

Forecasts from ARIMA(1,1,3)
100
300
500
0
200
400
600
800
1000

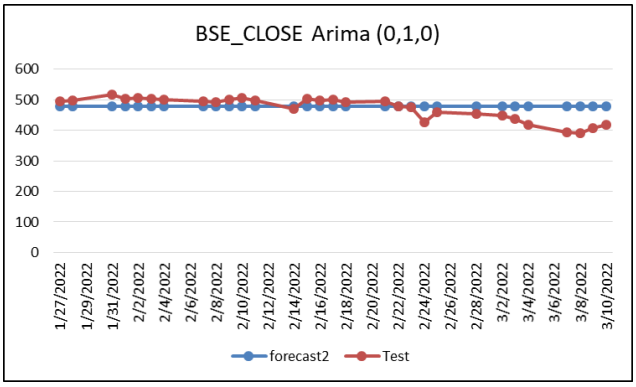

Forecasts from ARIMA(3,1,2)
100
300
500
0
200
400
600
800
1000

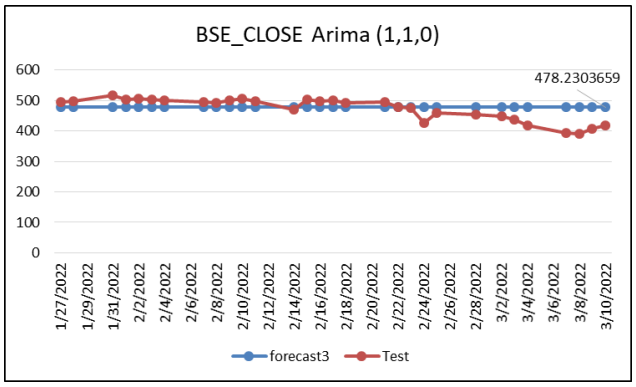

Forecasts from ARIMA(2,1,3)
100
300
500
0
200
400
600
800
1000

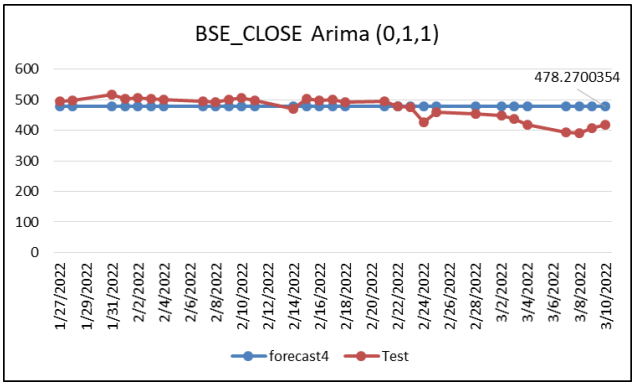

Forecasts from ARIMA(3,1,3)
100
300
500
0
200
400
600
800
1000

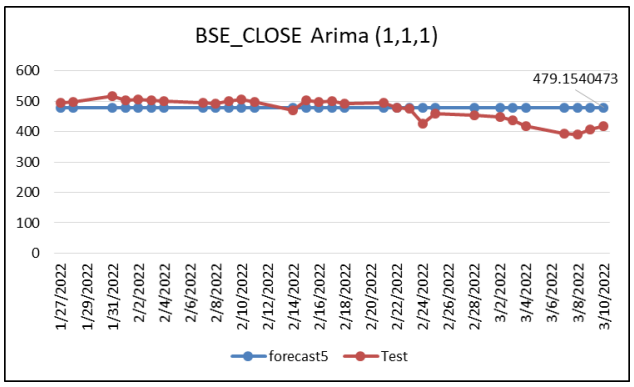
## Validating the data using test data (BSE_CLOSE from 26th Jan to 10th Mar'2022)

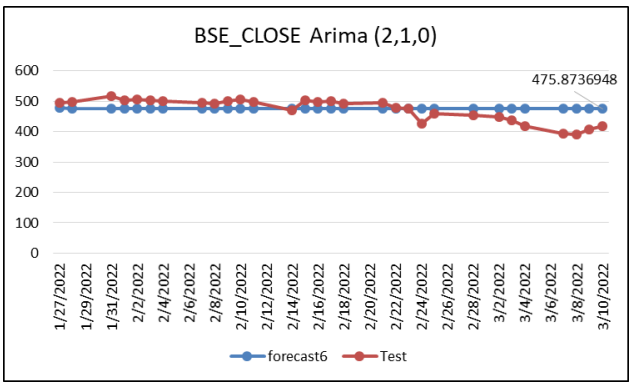


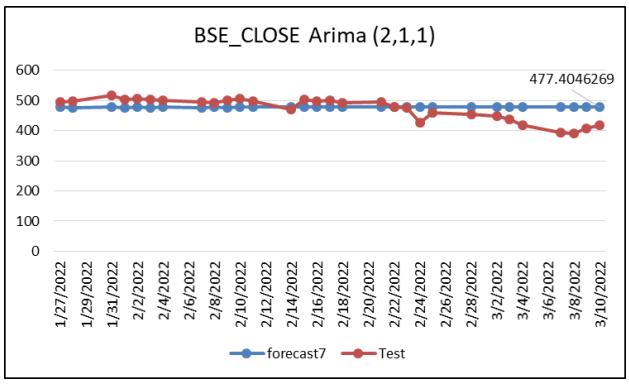



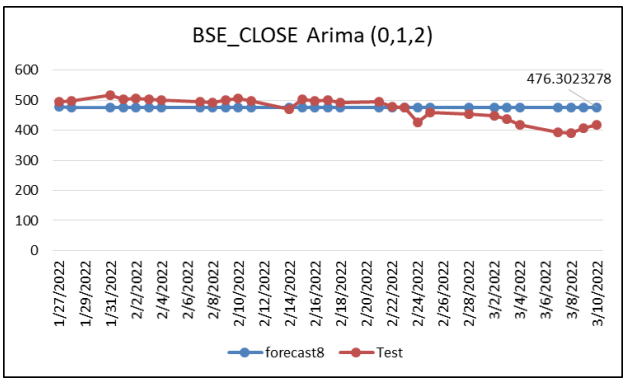



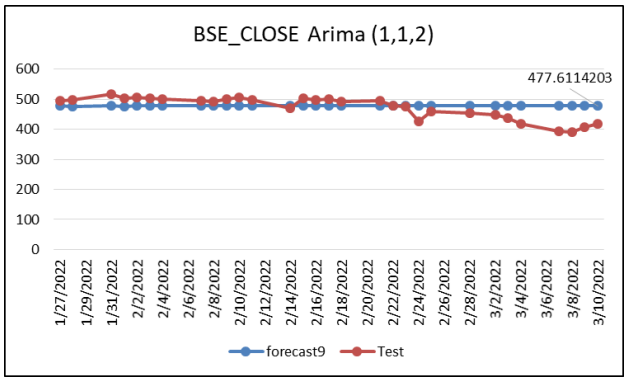



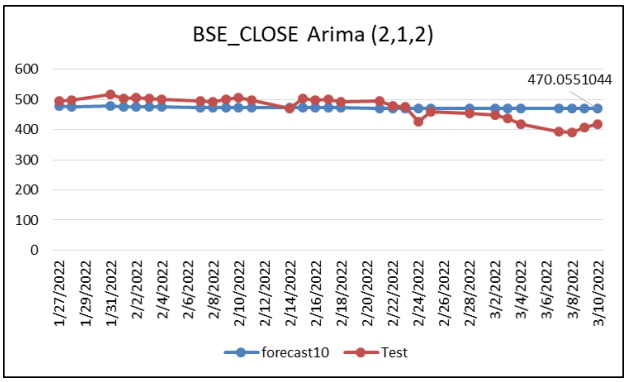



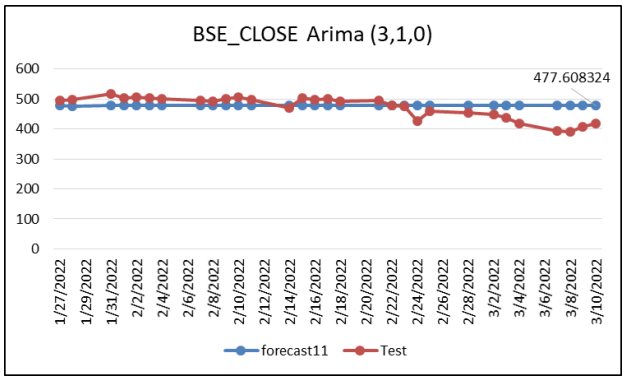



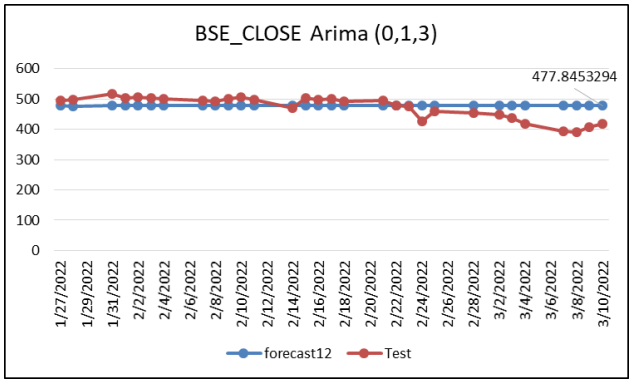



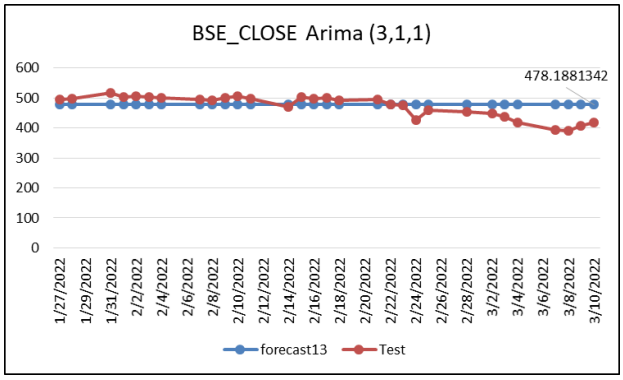



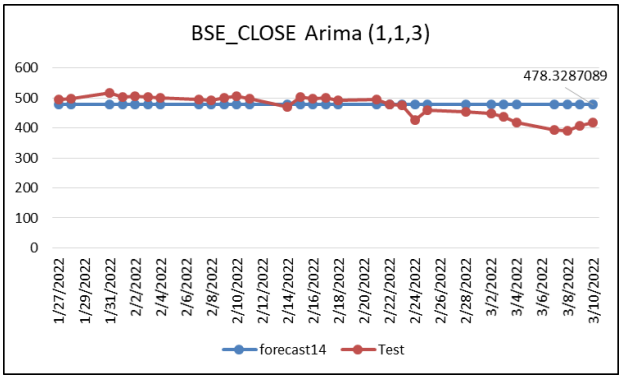

BSE_CLOSE Arima (1,1,2)
477.6114203
forecast9
Test

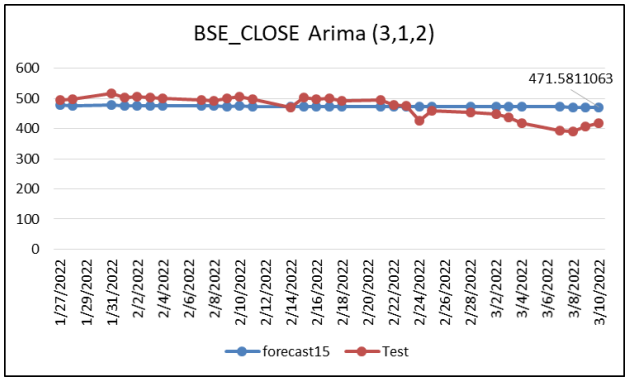

BSE_CLOSE Arima (2,1,2)
470.0551044
forecast10
Test

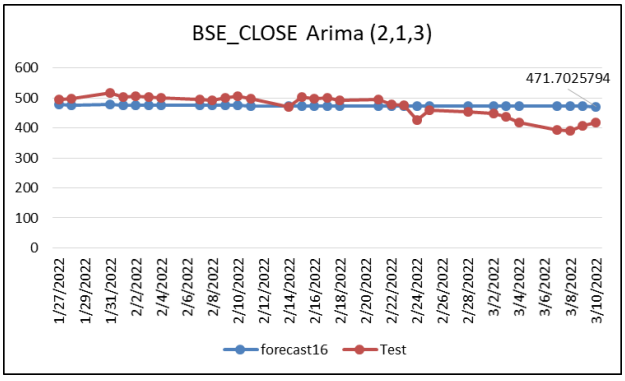

BSE_CLOSE Arima (3,1,0)
477.608324
forecast11
Test

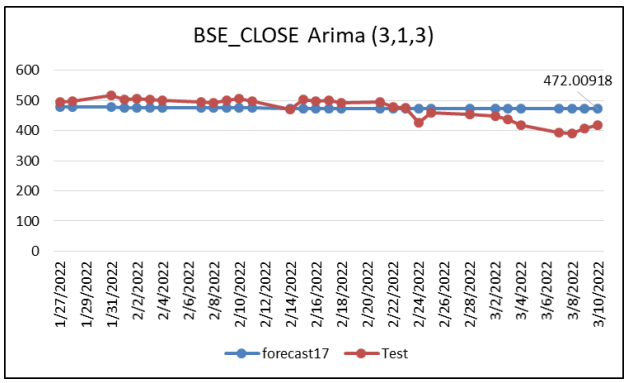

BSE_CLOSE Arima (0,1,3)
477.8453294
forecast12
Test


BSE_CLOSE Arima (3,1,1)
478.1881342
forecast13
Test

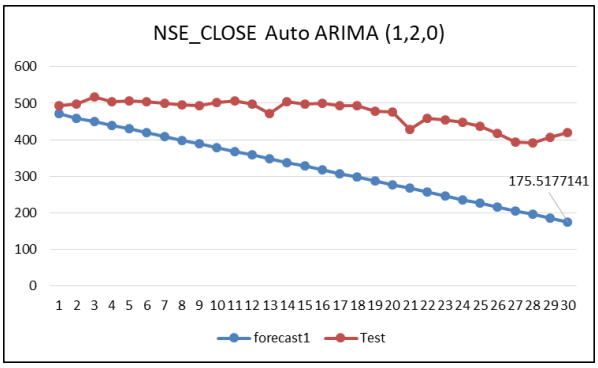

BSE_CLOSE Arima (1,1,3)
478.3287089
forecast14
Test

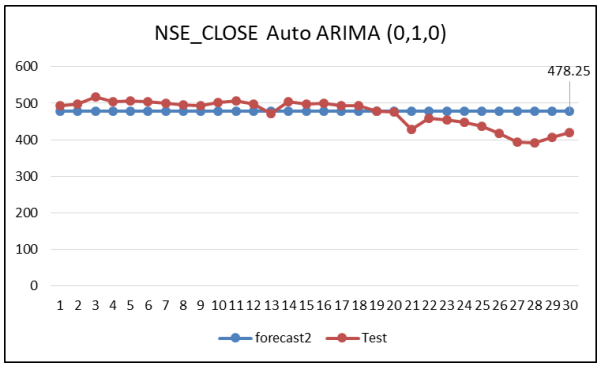

BSE_CLOSE Arima (3,1,2)
471.5811063
forecast15
Test

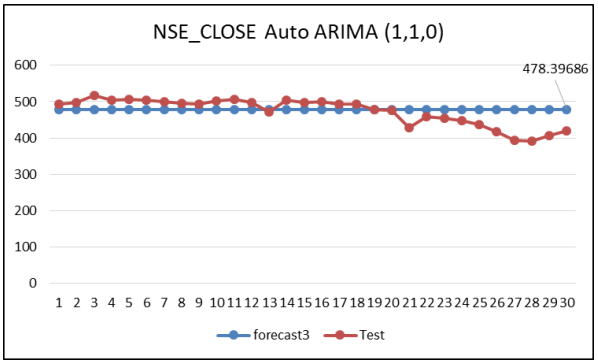

BSE_CLOSE Arima (2,1,3)
471.7025794
forecast16
Test

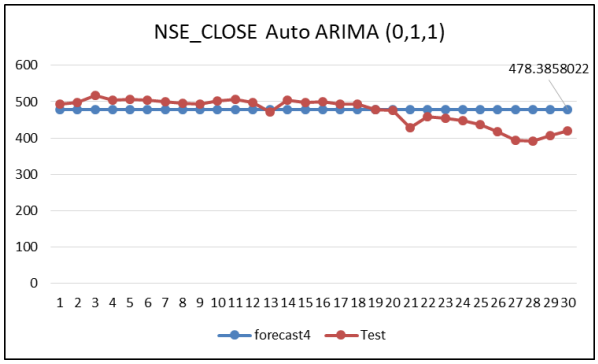


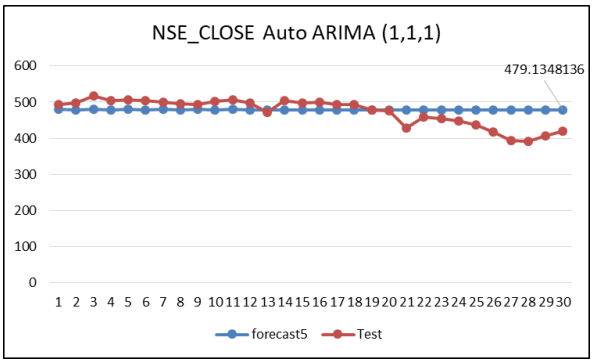

## Validating the data using test data (NSE_CLOSE from 26th Jan to 10th Mar'2022)

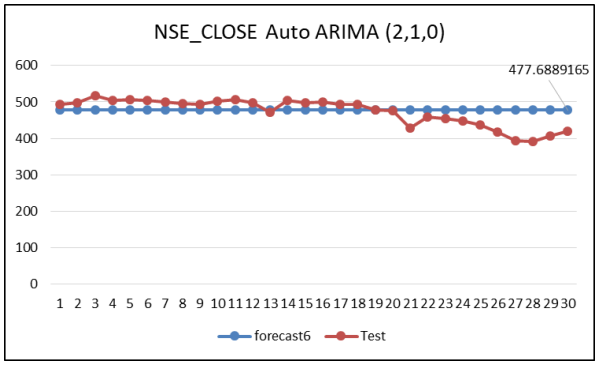


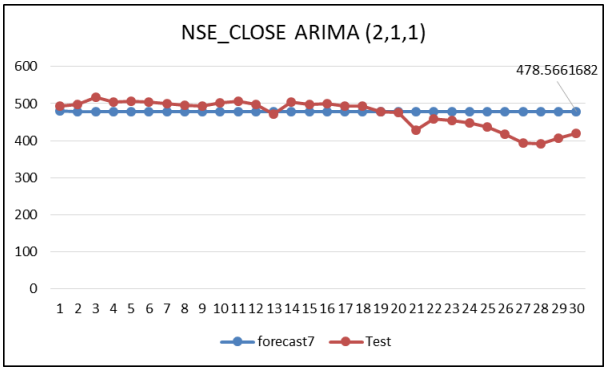



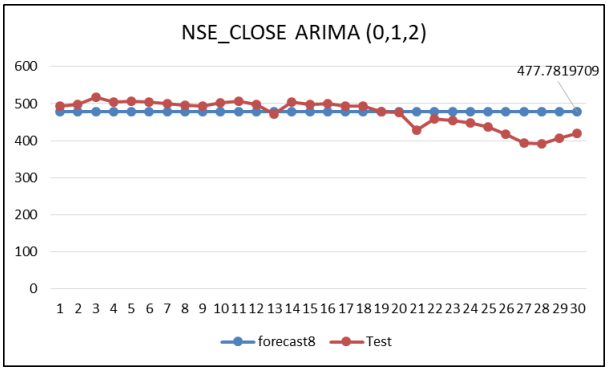



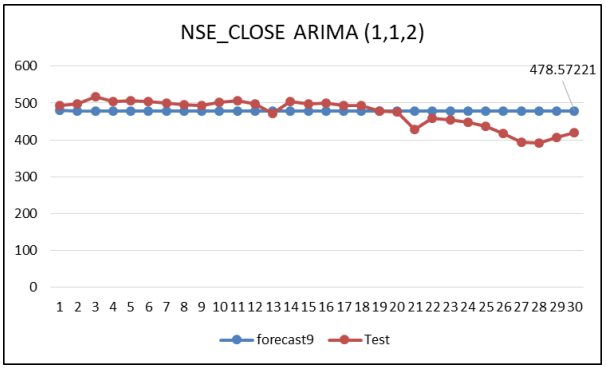



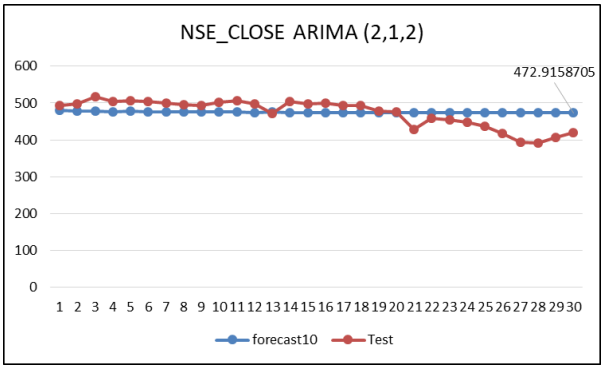



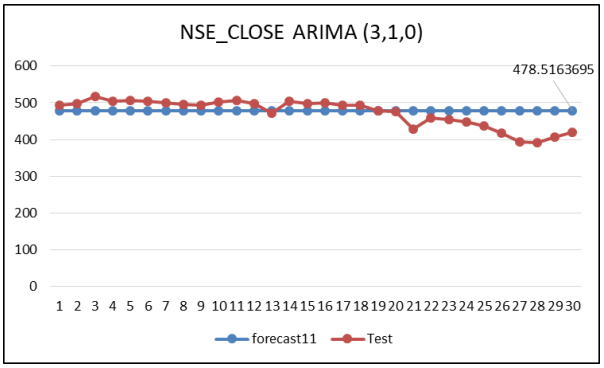



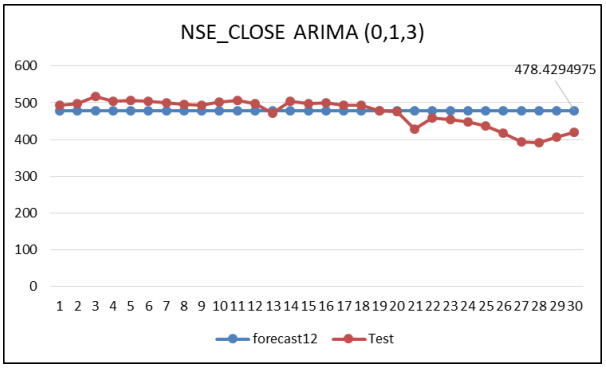

NSE_CLOSE ARIMA (2,1,1)
478.5661682
forecast7
Test

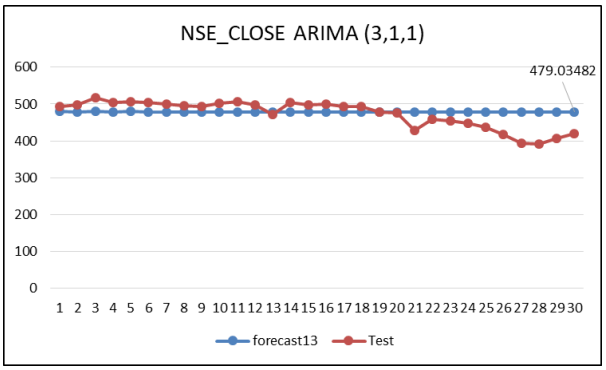

NSE_CLOSE ARIMA (0,1,2)
477.7819709
forecast8
Test

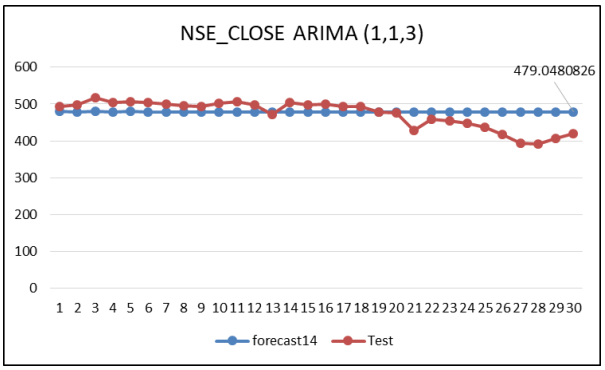

NSE_CLOSE ARIMA (1,1,2)
478.57221
forecast9
Test

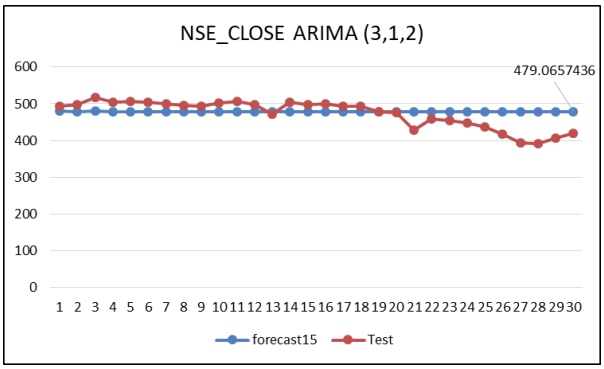

NSE_CLOSE ARIMA (2,1,2)
472.9158705
forecast10
Test

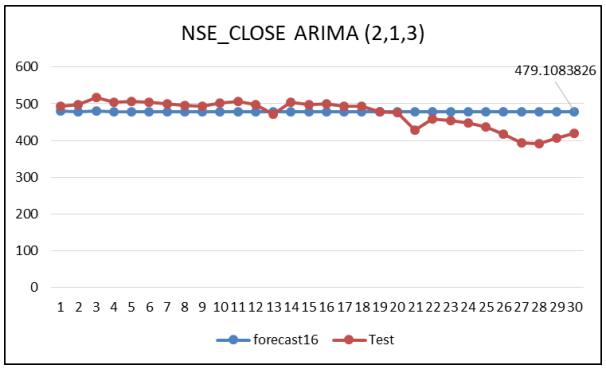

NSE_CLOSE ARIMA (3,1,0)
478.5163695
forecast11
Test

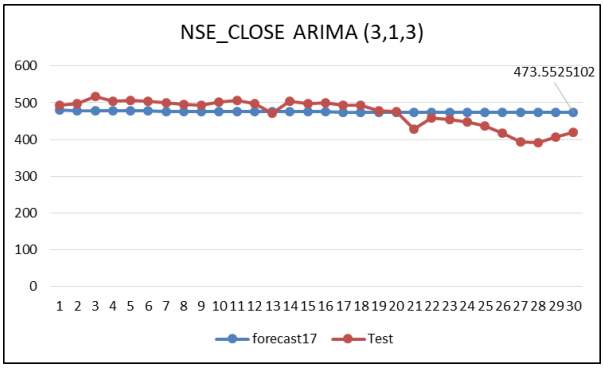

NSE_CLOSE ARIMA (0,1,3)
478.4294975
forecast12
Test


NSE_CLOSE ARIMA (3,1,1)
479.03482
forecast13
Test


NSE_CLOSE ARIMA (1,1,3)
479.0480826
forecast14
Test




From above validation, we can see **ARIMA (2,1,2) models** are most closely predicting the data for next 30 days……(B)

From point number (A) and (B)

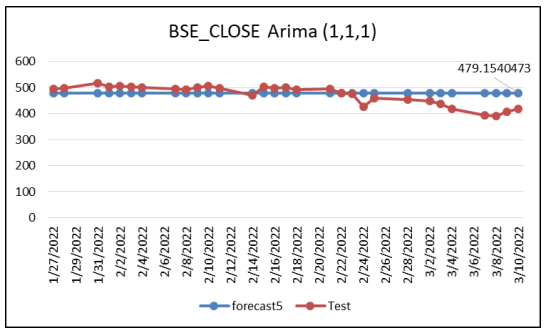
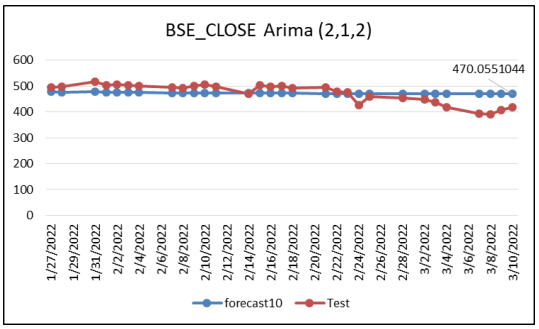
| BSE_CLOSE | | | | |
|---|---|---|---|---|
| MODEL | (1,1,1) | | (2,1,2) | |
| |  | |  | |
| Date | Forecast | Actual | Forecast | Actual |
| 1/27/2022 | 479.7682398 | 494.25 | 479.1668721 | 494.25 |
| 1/28/2022 | 478.6627783 | 497.399994 | 477.1630211 | 497.399994 |
| 1/31/2022 | 479.665892 | 517.5 | 477.608219 | 517.5 |
| 2/1/2022 | 478.7556503 | 504.299988 | 475.9806105 | 504.299988 |
| 2/2/2022 | 479.5816185 | 506.399994 | 476.2964749 | 506.399994 |
| 2/3/2022 | 478.8321215 | 504.600006 | 474.973097 | 504.600006 |
| 2/4/2022 | 479.5122272 | 500.549988 | 475.1921552 | 500.549988 |
| 2/7/2022 | 478.8950883 | 494.799988 | 474.1150115 | 494.799988 |
| 2/8/2022 | 479.4550902 | 492.5 | 474.2621643 | 492.5 |
| 2/9/2022 | 478.9469353 | 501.350006 | 473.3845072 | 501.350006 |
| 2/10/2022 | 479.4080433 | 507.299988 | 473.4787467 | 507.299988 |
| 2/11/2022 | 478.9896264 | 498.350006 | 472.7628636 | 498.350006 |
| 2/14/2022 | 479.3693047 | 471.700012 | 472.8186138 | 471.700012 |
| 2/15/2022 | 479.0247785 | 503.850006 | 472.2340537 | 503.850006 |
| 2/16/2022 | 479.3374072 | 498.200012 | 472.2622152 | 498.200012 |
| 2/17/2022 | 479.0537228 | 500 | 471.7843684 | 500 |
| 2/18/2022 | 479.3111426 | 493.100006 | 471.7931309 | 493.100006 |
| 2/21/2022 | 479.0775557 | 494.700012 | 471.40209 | 494.700012 |
| 2/22/2022 | 479.2895163 | 478.450012 | 471.3975643 | 478.450012 |
| 2/23/2022 | 479.0971798 | 477 | 471.0772101 | 477 |
| 2/24/2022 | 479.2717091 | 427.850006 | 471.0639182 | 427.850006 |
| 2/25/2022 | 479.1133384 | 459.600006 | 470.8011861 | 459.600006 |
| 2/28/2022 | 479.2570465 | 454.149994 | 470.7824397 | 454.149994 |
| 3/2/2022 | 479.1266434 | 447.600006 | 470.5667306 | 447.600006 |
| 3/3/2022 | 479.2449733 | 436.899994 | 470.5449248 | 436.899994 |
| 3/4/2022 | 479.1375988 | 417.5 | 470.3676306 | 417.5 |
| 3/7/2022 | 479.2350322 | 394.200012 | 470.3444688 | 394.200012 |
| 3/8/2022 | 479.1466196 | 391.799988 | 470.1985916 | 391.799988 |
| 3/9/2022 | 479.2268466 | 405.950012 | 470.1752597 | 405.950012 |
| 3/10/2022 | 479.1540473 | 419.549988 | 470.0551044 | 419.549988 |

Table No. - 3

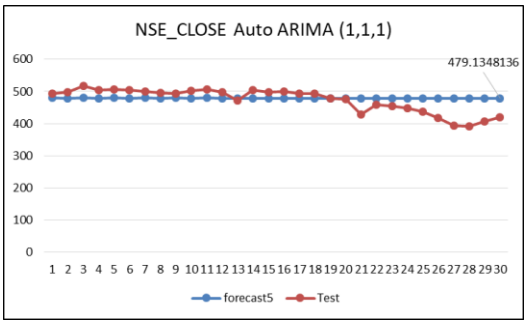
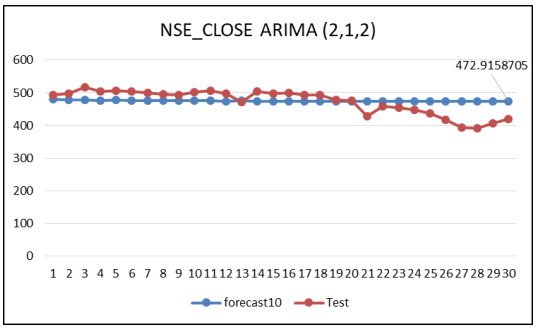
| NSE_CLOSE | | | | |
|---|---|---|---|---|
| MODEL | (1,1,1) | | (2,1,2) | |
| |  | |  | |
| Date | Forecast | Actual | Forecast | Actual |
| 1/27/2022 | 479.9790772 | 494.399994 | 479.6806601 | 494.399994 |
| 1/28/2022 | 478.4374072 | 497.299988 | 477.5796977 | 497.299988 |
| 1/31/2022 | 479.8119823 | 517.75 | 478.6144356 | 517.75 |
| 2/1/2022 | 478.5863914 | 504.299988 | 476.9651218 | 504.299988 |
| 2/2/2022 | 479.6791458 | 506.399994 | 477.7052924 | 506.399994 |
| 2/3/2022 | 478.7048303 | 504.5 | 476.4059705 | 504.5 |
| 2/4/2022 | 479.573544 | 500.600006 | 476.9278661 | 500.600006 |
| 2/7/2022 | 478.7989864 | 494.600006 | 475.9003926 | 494.600006 |
| 2/8/2022 | 479.489593 | 492.75 | 476.261337 | 492.75 |
| 2/9/2022 | 478.8738383 | 501.25 | 475.4455594 | 501.25 |
| 2/10/2022 | 479.422854 | 507.450012 | 475.6885335 | 507.450012 |
| 2/11/2022 | 478.9333437 | 498.850006 | 475.0380754 | 498.850006 |
| 2/14/2022 | 479.3697981 | 471.450012 | 475.1952249 | 471.450012 |
| 2/15/2022 | 478.9806491 | 504 | 474.6742685 | 504 |
| 2/16/2022 | 479.32762 | 498.450012 | 474.769564 | 498.450012 |
| 2/17/2022 | 479.0182558 | 499.950012 | 474.3503921 | 499.950012 |
| 2/18/2022 | 479.2940893 | 493.149994 | 474.4016441 | 493.149994 |
| 2/21/2022 | 479.0481522 | 494.450012 | 474.0627608 | 494.450012 |
| 2/22/2022 | 479.2674333 | 478.25 | 474.0831459 | 478.25 |
| 2/23/2022 | 479.0719191 | 477 | 473.8078406 | 477 |
| 2/24/2022 | 479.2462423 | 427.950012 | 473.8070557 | 427.950012 |
| 2/25/2022 | 479.0908132 | 459.75 | 473.5823028 | 459.75 |
| 2/28/2022 | 479.229396 | 454.049988 | 473.5674384 | 454.049988 |
| 3/2/2022 | 479.1058336 | 447.600006 | 473.3830551 | 447.600006 |
| 3/3/2022 | 479.2160036 | 437.100006 | 473.3592539 | 437.100006 |
| 3/4/2022 | 479.1177745 | 417.25 | 473.2072545 | 417.25 |
| 3/7/2022 | 479.205357 | 394 | 473.1782085 | 394 |
| 3/8/2022 | 479.1272672 | 391.75 | 473.0523089 | 391.75 |
| 3/9/2022 | 479.1968932 | 405.899994 | 473.020634 | 405.899994 |
| 3/10/2022 | 479.1348136 | 419.600006 | 472.9158705 | 419.600006 |

Table No. - 4

# Conclusion and Findings

The main objective of this paper is to study the stationarity of the indices of BSE and NSE and to forecast using the ARIMA model. For this purpose, the weekly closing indices of BSE and NSE are obtained from the website yahoofinance.com for the period from 6th June, 2014 to 3rd June, 2018. The ADF Test is administered to check for the presence of unit root to confirm the stationarity of index series. The results of the test confirmed the presence of Unit root and showed non-stationary. The ADF test has confirmed that the given time series are stationary at first difference.

For the present work, ARIMA (1,1,1) and ARIMA (2,1,2) model was chosen as the top model from seventeen different models because these gratify most of the norms of goodness of fit statistics, as other fifteen models have not satisfied such criterions. This best candidate model was selected for making predictions of BSE_CLOSE and NSE_CLOSE for the period ranging from 26$^{th}$ January, 2022 to 10$^{th}$ March, 2022 using the daily data ranging from 1$^{st}$ Jan 2018 to 25$^{th}$ Jan, 2022 which found to be same in both the cases. The study also made a comparison between predicted and actual performance of BSE_CLOSE and NSE_CLOSE during the sample period. The results of the best fitted model highlights the strength of ARIMA model to forecast the BSE_CLOSE and NSE_CLOSE satisfactory on short-term basis and would guide the individuals to select gainful investment options.

# Limitations

The ARIMA model has few constraints regarding the exactness of forecasting because of its wide usage for short-run predicting the values in the time series to notice the minor variations in the data due to hidden factors which cannot be studied (like current scenario of Russia and Ukraine war, or COVID-19 spread, or change in government policies or economic instability.. etc.). It turns out to be intricate to capture the accurate trend. Hence, this model turns out to be useless to predict long-run changes. Moreover, the forecasting using the ARIMA model would depend upon the hypothesis of linearity in historical data, however, there is no confirmation that BSE_CLOSE or NSE_CLOSE are linear in nature.

# Bibliography


https://www.influxdata.com/what-is-time-series-data

https://www.analyticsvidhya.com/blog/2021/04/exploratory-analysis-using-univariate-bivariate-and-multivariate-analysis-techniques

https://mran.microsoft.com/documents/what-is-r

https://www.geeksforgeeks.org/difference-between-data-analytics-and-data-analysis

https://finance.yahoo.com/quote/TATAMOTORS.NS?p=TATAMOTORS.NS&.tsrc=fin-srch

https://finance.yahoo.com/quote/TATAMOTORS.BO?p=TATAMOTORS.BO&.tsrc=fin-srch

https://en.wikipedia.org/wiki/Stock_market_prediction

https://www.youtube.com/watch?v=qaZNDKFnX_Y&t=1119s