A Project Report on

Stock Price Prediction using Time Series analysis

Submitted in partial fulfilment of the requirement for the award of Bachelor of Engineering in Computer Science and Engineering

Submitted by

Kumar Saurav (Roll No.- 2015-1055)
Pragati Kumar (Roll No.- 2015-1042)
Mohit Kumar Singh (Roll No.- 2015-1051)

Under the supervision of
Mr.Biswanath Pal
Assistant Professor
Department of Computer Science and Engineering
University Institute of Technology
The University of Burdwan
Golapbag (North), Burdwan- 713104, W.B.

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our faculty **Mr.Biswanath Pal** (Assistant professor, Dept. of CSE, UIT BU) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We also want to express our gratitude towards **Dr.Souvik Bhattacharya**, In-charge, Department of Computer Science and Engineering, University Institute of Technology, Burdwan University and all those who offered helping hands to complete this project directly or indirectly.

We are highly obliged to our project team members for the valuable information and participation. We are grateful for their cooperation during the period of our assignment.

Kumar Saurav
------------------------------------- Roll No. 2015-1055
Reg. No. A3146 of 2015-16

Pragati Kumar
------------------------------------- Roll No. 2015-1042
Reg. No. A2802 of 2015-16

Mohit Kumar Singh
------------------------------------- Roll No. 2015-1051
Reg. No. A3183 of 2015-16

UNIVERSITY INSTITUTE OF TECHNOLOGY
THE UNIVERSITY OF BURDWAN

CERTIFICATE

This is to be certify that a project work titled – "*Stock Price Prediction using Time Series analysis*" which is submitted by **Kumar Saurav** (B.E. 4th year, CSE Dept. , Roll No- 2015-1055, Reg. No.- A-3146 of 2015-16) , **Mohit Kumar Singh** (B.E. 4th year, CSE Dept. , Roll No- 2015-1051, Reg. No.- A-3183 of 2015-16) and **Pragati Kumar** (B.E. 4th year, CSE Dept. , Roll No. 2015-1042, Reg. No.- A2802 of 2015-16) towards the partial fulfilment of the requirements for the award of **Bachelor of Engineering degree in Computer Science & Engineering** at **University Institute of Technology, The University of Burdwan.** This project has been prepared as per the regulation of the **University of Burdwan**.

Mr. Biswanath Pal
Project Supervisor,
Associate Professor
Department of Computer
Science and Engineering

University Institute of
Technology, The University of
Burdwan

Dr. Souvik Bhattacharya
Assistant Professor,
In-charge Department of
Computer Science and
Engineering

University Institute of
Technology, The University of
Burdwan

[It may be noted that any act of Plagiarism if found in the project dissertation /document/report is the sole responsibility of the project scholar and the signatories of this certificate has no role whatsoever]

TABLE OF CONTENTS

1. ABSTRACT

Prediction of stock market has been an attractive topic to the stock brokers and the researchers from various fields. There have been numerous studies to predict the price of the stocks of a particular company using various machine learning techniques. In this paper we would be using k-nearest neighbor algorithm to predict the stock price of the company. Also we would be comparing their accuracy so that we can understand which regression technique is better.

Stock prices prediction is interesting and challenging research topic. Developed countries' economies are measured according to their power economy. Currently, stock markets are considered to be an illustrious trading field because in many cases it gives easy profits with low risk rate of return. Stock market with its huge and dynamic information sources is considered as a suitable environment for data mining and business researchers.

In this paper, we applied ARIMA algorithm in order to predict stock prices for sample of top 500 SMP500 company's stock price record listed on their official website to assist investors, management, decision makers, and users in making correct and informed investments decisions.

According to the results, this algorithm is robust with small error ratio; consequently the results were rational and also reasonable. In addition, depending on the actual stock prices data; the prediction results were close and almost parallel to actual stock prices.

2. INTRODUCTION

2.1 Motivation

In recent years, Artificial Intelligence (AI) has been a significant technology that is being applied in making of driverless cars, intelligent robots, image and speech recognition, automatic translations, and medical assistants [9]. Hence, forecasting stock market in the light of AI and by utilizing different machine learning methods has been an important issue in financial and economic fields. Consequently, it has made the researchers think to come up with reliable predictive models over the decade [2]. The urge to predict stock prices more accurately is making the researchers to work for the betterment of the current predictive machine learning models. The reason is that shareholders and investors have the freedom to make plans and strategical approaches towards taking decision about investments and future activities. This leads the organizations and individuals to get any predictive method that ensures more income from the stock market easily along with minimum investment risk.

In finance, forecasting stock market is considered to be one of most difficult tasks to do till now because of the stochastic behaviors and complex dependencies of stock market [11]. Because of the unpredictive nature of stock market there exists no certain models of machine learning that can precisely forecast about stock market and there is more work to perform in this sector which is the inspiring factor for us to research and build a better predictive system. Various methods of machine learning had been applied for forecasting stock market throughout the recent years. Among them models such as Support Vector Regression (SVR),
Artificial Neural Networks (ANNs), Bayesian Neural Network (BNN) and so on had been exploited to improve time series forecasting [11]. Moreover, different hybrid methods had also been generated to improve the efficiency of prediction. Yet, there is little evidence about their relative performance as standard forecasting models. ARIMA has been extensively

used for its efficiency in financial time series forecasting especially for short-term prediction than the most used neural network techniques.

2.2 Objective

Prediction is a procedure to make assumption of future in the light of existing information. The more exact the prediction, the simpler it could be to settle on a choice for future. As we discussed before, predicting stock market accurately is a difficult task to do because of the dynamic nature of the stock market. If stock market rises, then a country's financial development would be high and vice-versa. In recent years, we can use huge amount of data and analyze those due to the development of computer technology. There are two approaches to stock market prediction. One approach focuses on the historical data, another approach focuses on data aside from the historical data. Regarding the first approach there are two types of methods, fundamental method and technical method. Technical method is a type of method that uses historical data. We can use various types of technical methods to predict a stock market such as Neural Network, Evolutionary Algorithms, Support Vector Machine, Neuro-Fuzzy, Hidden Markov model and decision tree. In this paper we use Time Series (TS) techniques to predict a stock market that how it will behave in the upcoming thirty days by using some well-known companies last twenty years historical data. Stock prices can be treated as a discrete time series model which is in the light of an arrangement of well-defined numerical data items collected at successive points at regular intervals of time. Since, it is fundamental to distinguish a model with the end goal to analyze trends of stock prices with adequate information for decision making, it recommends that transforming the TS using Autoregressive Integrated Moving Average (ARIMA) is a better algorithmic approach than forecasting directly. ARIMA model converts a non-stationary data to a stationary data before working on it. It is one of the most popular models to predict linear time series data.

2.3 Background Study

Time series analysis is a statistical method that analyses and manipulates time series data. Time series is made of data points collected at constant time intervals. Time series analysis unlike regression analysis is very useful in order to get the important characteristics and statistics of a time series data. Time series analysis has the features to help us understand the underlying factors that lead to a specific trend in time series data points and thus help us predict data points. The first step towards time series analysis is to check if the time series data is stationary. The two main reasons behind non-stationarity are:
Trend: The mean of a time series is variable over time. For example, the average number of car users is growing over time.

- Seasonality: Variations at a particular time-interval such as, people might buy cars in a particular month because of salary increment or festival.

To check if a time series is stationary, the following tests are performed:

- Plotting Rolling Statistics: Plots the moving average or variance and check if it varies with time. This is more like a visual representation.
- Dickey-Fuller Test: Unlike the first one this a statistical test to check stationarity. In this test, null hypothesis considers time series as non-stationary. Results are the test-statistic and some critical-values for different confidence levels. The series is said to be stationary if the null hypothesis gets rejected when the test-statistic becomes less than the critical-value.

The main purpose is to reduce these features from the time series by estimating the trend and seasonality in the series. The following techniques can be useful to model or estimate trend and seasonality:

- Aggregation: Considers averages for a time period like month/week.
- Smoothing: Considers rolling averages.
- Polynomial Fitting: Fits a regression model.

According to different problem solving, any of the techniques can be used. After the estimation, trend and seasonality can be reduced by using the following methods:

• Differencing: It is the most common way to reduce non-stationary features by taking the difference of observation between a particular instant with a previous instant. Trend and seasonality reduction can be improved by changing the order of differencing.
• Decomposing: It separately models the trend and seasonality of the time series and the rest of it is returned so that the residuals can be modeled.
After these steps, forecasting techniques can be applied on the non-stationary series. In final step, trend and seasonality constraints are applied back to convert the predicted values into the original scale.

2.4 Related Works

The Stock market prediction has been an important exertion in business and finance for many years. Correct prediction of stock market is very important for the investors to determine that if it would be better to buy any specific stock or not. There have been a significant number of studies and analysis done by many enthusiasts who applied previously established prediction models to acquire more accuracy.

Artificial Neural Network based method is the first technique to be used for the stock market trend prediction [14]. Neural Networks (NNs) have been proved to be predicting the future value of a stock market with a good accuracy. NNs can deal with uncertain, fuzzy or insufficient data that is very volatile and for this reason, NNs have become very important method for stock market prediction [10]. According toWong, Bodnovich and Selvi [12], the most frequent areas of NNs applications in past 10 years are production/operations (53.5%) and finance (25.4%). NNs in finance have their most frequent applications in stock performance and stock selection predictions. Benefit of NNs applications is in their ability to deal with uncertain and robust data. Therefore, NNs can be efficiently used in stock markets, to predict either stock prices or stock returns. However, NNs require very large number of previous cases [7][13] and the best network architecture is still unknown [10]. In some cases, for complicated networks, reliability of results may decrease [13].

Labiad, B., Berrado, A., Benabbou, L. (2016) did an analysis on Moroccan Stock Exchange for Short Term Stock Movements Classification and found 89% accuracy in shortest CPU time [8]. They have used Random Forests, Gradient Boosted trees and Support Vector Machine (SVM) techniques. They used a technical indicator as input variable. Then they performed a feature selection and sample selection steps to improve prediction accuracy.

There prediction was for a very short term (10 minutes ahead). They took eight years of previous intraday prices of Maroc Telecom (IAM) stocks to evaluate the performance of their selected models. Their experiment shows higher accuracy in RF and GBT techniques than SVM. They came to a conclusion that less complex data and reduced training time of RF and GBT are suitable for short term forecasting.

Another stock market prediction technique is used by Aditya Gupta and Bhuwan Dhingra in 2012 [5]. Hidden Markov Models (HMM's) have been applied to forecast and predict the stock market in their work. They have used historical data of different stocks to forecast the next day's stock values through Maximum a Posteriori HMM approach.

In their approach, they considered the fractional change in Stock value and the intra-day high and low values of the stock to train the continuous HMM. This HMM is then used to make a Maximum a Posteriori decision over all the possible stock values for the next day. HMM's have been successful in analyzing and predicting time depending phenomena, or time series [6]. Hidden Markov Models are based on a set of unobserved underlying states amongst which transitions can occur and each state is associated with a set of possible observations. The stock market can also be seen in a similar manner. The underlying states, which determine the behavior of the stock value, are usually invisible to the investor. The transitions between these underlying states are based on company policy, decisions and economic conditions etc. The visible effect which reflects these is the value of the stock. In their model, they have used the daily fractional change in the stock value, and the fractional deviation of intra-day high and low. They used four different stocks and separate HMM is trained for each stock. After testing their approach in different stocks, they compared the performance to some of the existing methods using HMMs and Artificial Neural Networks using Mean Absolute Percentage Error (MAPE).

3. Methodology

Time Series Analysis

3.1 Definition

TS is a series of data points indexed in time order. Most commonly, a TS is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discretetime data. It is mathematically defined as a set of vectors x(T),T = 0,1,2,... where T represents the time elapsed we can denote the observations by Y1,Y2, . . .YT . The variable x(T) is treated as a random variable. A TS is uni-variate when it contains records of a single variable. If records of more than one variable are considered, then it is called multivariate. Again TS can be continuous or discrete. If observations are measured at every instances of time then it is called a continuous TS, whereas a discrete time series contains observations measured at discrete points of time. For example flow of river, concentration of a chemical process, temperature reading etc. can be recorded as a continuous TS. On the other hand population of a country, production of a company, exchange rates between two different currencies may present discrete TS. The consecutive calculations are usually recorded in a discrete TS at equally spaced time intervals such as hourly, daily, weekly, monthly or yearly time separations. The variable observed in a discrete time series is assumed to be measured by the real number scale as a continuous variable. By merging data together over a specified time interval we can easily transform continuous TS to a discrete one. There are a number of important interests in a TS such as Smoothing, Modeling, Forecasting, Control.

- Smoothing: The observed Yt are assumed to be the result of "noise" values t additively contaminating a smooth signal t.

$$Y_t = \eta_t + \varepsilon_t \qquad (3.1)$$

We may wish to recover the values of the underlying η_t .

- **Modelling:** We may wish to develop a simple mathematical model which explains the observed pattern of $Y_1, Y_2, \ldots Y_T$. This model may depend on unknown parameters and these will need to be estimated.

- **Forecasting:** On the basis of observations $Y_1, Y_2, \ldots Y_T$, we may wish to predict what the value of Y_{T+L} will be (L>1), and possibly to give an indication of what the uncertainty is in the prediction.

- **Control:** We may wish to intervene with the process which is producing the Y_t values in such a way that the future values are altered to produce a favorable outcome.

3.2 Components of a Time Series

A time series in general is supposed to be affected by four main components, which can be separated from the observed data. These components are: Trend, Cyclical, Seasonal and Irregular components. A brief description of these four components is given here. The general tendency of a time series to increase, decrease or stagnate over a long period of time is termed as Secular Trend or simply Trend. Thus, it can be said that trend is a long term movement in a time series. For example, series relating to population growth, number of houses in a city etc. show upward trend, whereas downward trend can be observed in series relating to mortality rates, epidemics, etc. Seasonal variations in a time series are fluctuations within a year during the season. The important factors causing seasonal variations are: climate and weather conditions, customs, traditional habits, etc. For example sales of ice-cream increase in summer, sales of woollen cloths increase in winter. Seasonal variation is an important factor for businessmen, shopkeeper and producers for making proper future plans. The cyclical variation in a time series describes the medium-term changes in the series, caused by circumstances, which repeat in cycles. The duration of a cycle extends over longer period of time, usually two or more years. Most of the economic and financial time series show some kind of cyclical variation. For example a business cycle consists of four phases such as I) Prosperity II) Decline III) Depression IV) Recovery. A typical business cycle is shown in Fig. 3.1.

Irregular or random variations in a time series are caused by unpredictable influences which are not regular and also do not repeat in a particular pattern. These variations are caused by incidents such as war, strike, earthquake, flood, revolution, etc. There is no defined statistical techniques for measuring random fluctuations in a time series.

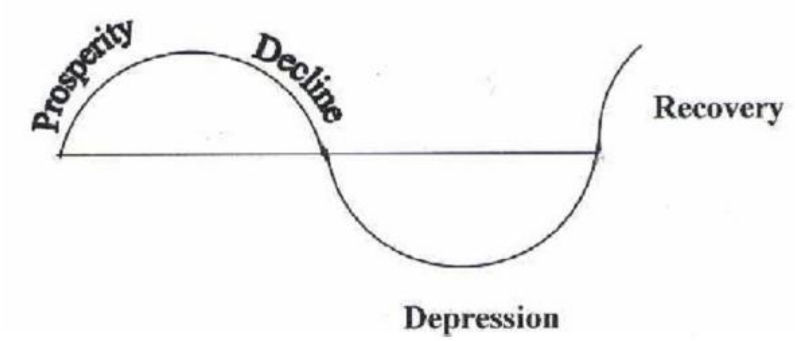


Fig. 3.1 Four phases business cycle

Considering the effects of these four components, two different types of models are generally used for a time series known as Multiplicative and Additive models.
Multiplicative Model: $Y(t) = T(t)\times S(t)\times C(t)\times I(t)$.
Additive Model: $Y(t) = T(t) + S(t) + C(t) + I(t)$.

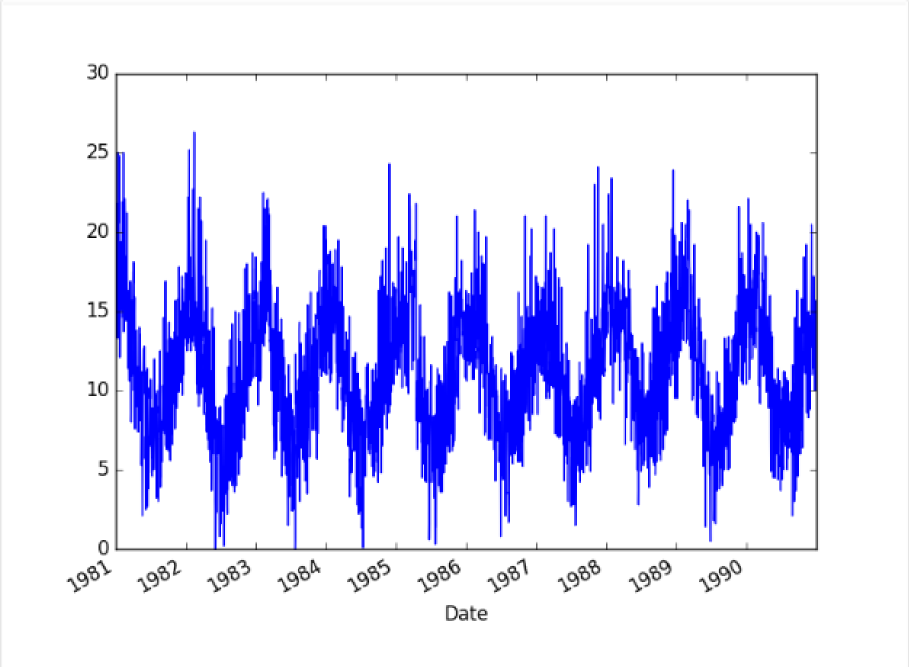
Here $Y(t)$ is the observation and $T(t)$, $S(t)$,$C(t)$ and $I(t)$ are respectively the trend, seasonal, cyclical and irregular variation at time t. Multiplicative model is based on the assumption that the four components of a time series are not necessarily independent and they can affect one another;

whereas in the additive model it is assumed that the four components are independent of each other.

3.3 Time Series Visualization

Plots of the raw sample data can provide valuable diagnostics to identify temporal structures like trends, cycles, and seasonality that can influence the choice of model. There are 6 different types of visualizations that we can use on our time series data. They are:

1. **Line Plot**: The first, and perhaps most popular, visualization for time series is the line plot. In Fig. 3.4, time series is shown on the X-axis with observation values along the Y-axis. This is an example of visualizing the Minimum Daily Temperatures data-set directly as a line plot.



2. **Histogram and Density Plot**: This means a plot of the values without the temporal ordering. Some linear time series forecasting methods assume a well-behaved distribution of observations. This can be explicitly checked using tools like statistical hypothesis tests. But plots can provide a useful first check of the distribution of observations both on raw observations and after any type of data transform has been performed. Fig. 3.5 shows a histogram plot of the observations in the Minimum Daily Temperatures

data-set. A histogram groups values into bins, and the frequency or count of observations in each bin can provide insight into the underlying distribution of the observations.

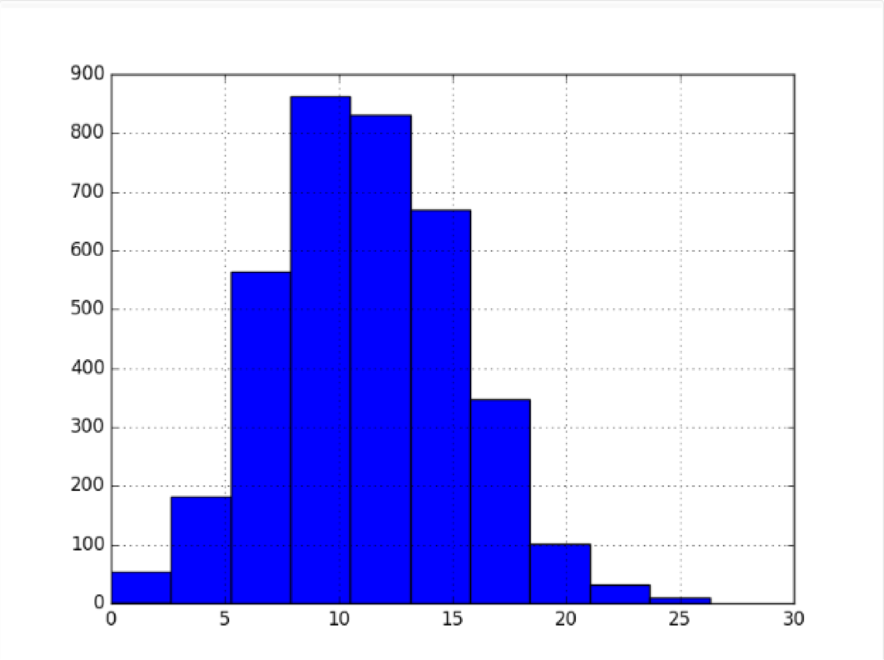


Fig. 3.3 Histogram and Density Plot

3. **Box and Whisker Plot**: Another type of plot that is useful to summarize the distribution of observations is the box and whisker plot. This plot draws a box around the 25th and 75th percentiles of the data that captures the middle 50% of observations. A line is drawn at the 50th percentile (the median) and whiskers are drawn above and below the box to summarize the general extents of the observations. Dots are drawn for outlines outside the whiskers or extents of the data. Fig. 3.6 is an example of grouping the Minimum Daily Temperatures data-set by years. A box and whisker plot is then created for each year and lined up side-by-side for direct comparison. Comparing box and whisker plots by consistent intervals is a useful tool. Within an interval, it can help to spot outlines (dots above or below the whiskers). Across intervals, in this case years, we can look for multiple year trends, seasonality, and other structural information that could be modeled.

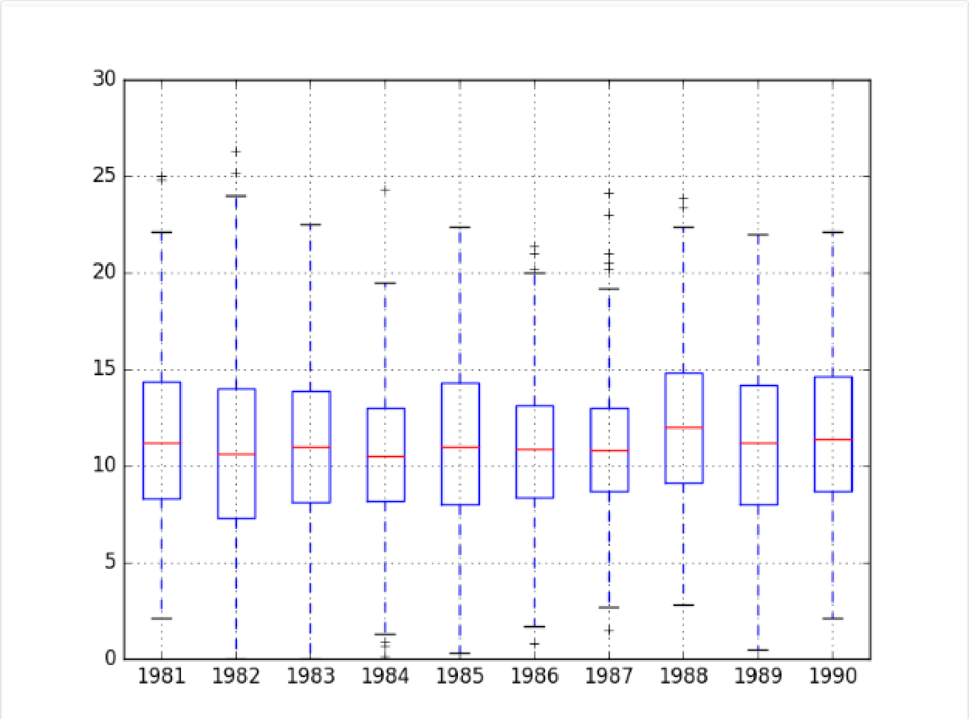


Fig. 3.4 Time Series Box and Whisker Plot

4. **Heat Map**: A matrix of numbers can be plotted as a surface, where the values in each cell of the matrix are assigned a unique color. This is called a heat-map, as larger values can be drawn with warmer colors (yellows and reds) and smaller values can be drawn with cooler colors (blues and greens). Fig. 3.7 is an example of creating a heat-map of the Minimum Daily Temperatures data. For convenience, the matrix is rotation (transposed) so that each row represents one year and each column one day.

This provides a more intuitive, left-to-right layout of the data. The plot shows the cooler minimum temperatures in the middle days of the years and the warmer minimum temperatures in the start and ends of the years, and all the fading and complexity in between.

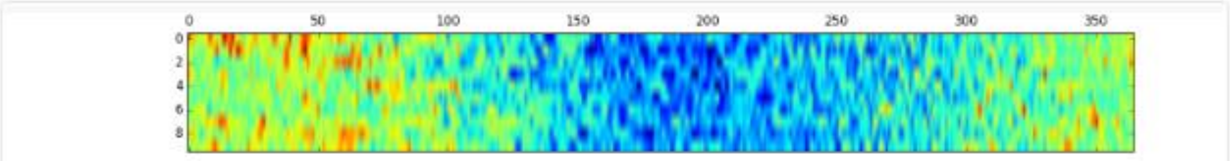


Fig. 3.5 Heat Map Plot

5. **Lag Plot or Scatter Plot**: Time series modeling assumes a relationship between an observation and the previous observation. Previous observations in a time series are called lags, with the observation at the previous time step called lag1, the observation at two time steps ago lag2, and so on. A useful type of plot to explore the relationship between each observation and a lag of that observation is called the scatter plot. Fig 3.8 shows f a lag plot for the Minimum Daily Temperatures dataset.

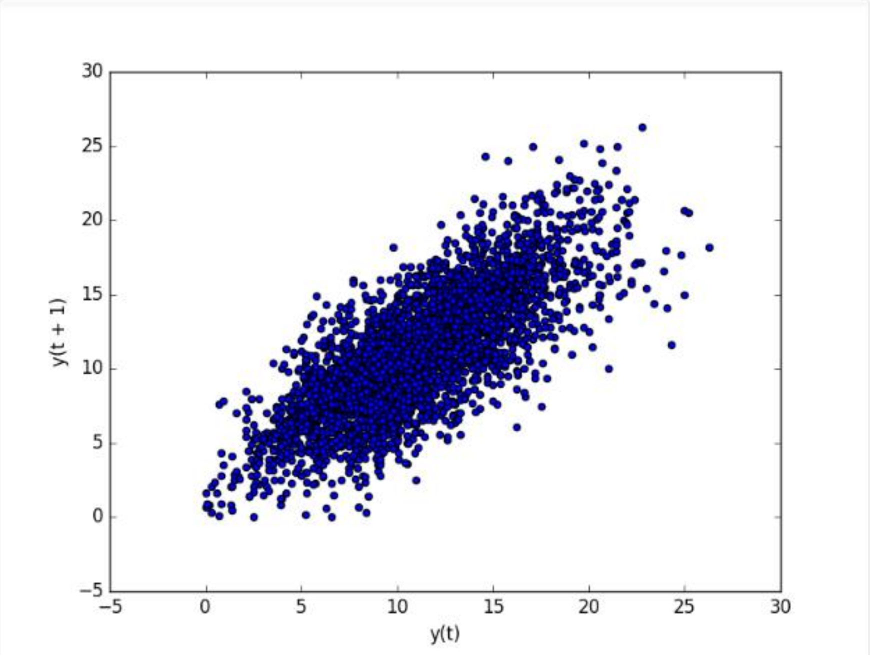


Fig. 3.6 Time Series Lag Scatter Plot

6. **Auto-correlation Plot**: We can quantify the strength and type of relationship between observations and their lags. In statistics, this is called correlation, and when calculated against lag values in time series, it is called auto-correlation (self-correlation). A correlation value calculated between two groups of numbers, such as observations and their lag1 values, results in a number between -1 and 1. The sign of this number indicates a negative or positive correlation respectively. A value close to zero suggests a weak correlation, whereas a value closer to - 1 or 1 indicates a strong correlation. Correlation values, called correlation coefficients, can be calculated for each observation and different lag values. Once calculated, a plot can be created to help better understand how this relationship changes over the lag. This

type of plot is called an auto-correlation plot and Pandas provides this capability built in, called the auto-correlation plot

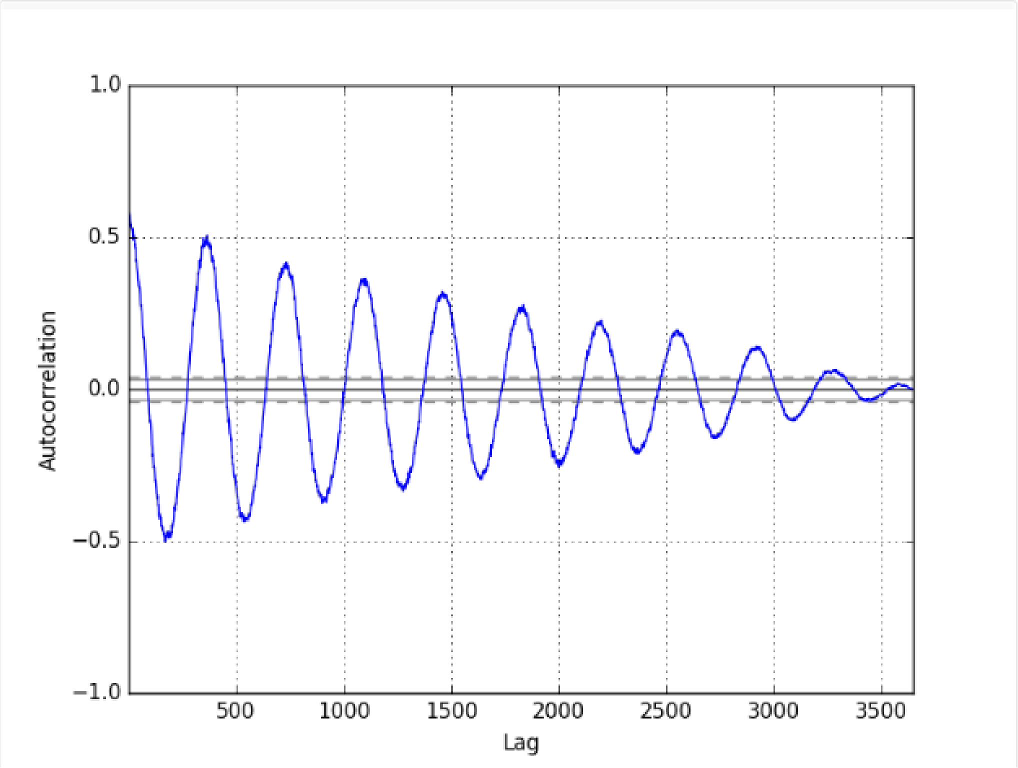


Fig. 3.7 Time Series Auto-Correlation Plot

3.4 Checking Stationarity

After plotting the TS data the next step is to determine whether the given series is stationary or not.

1. **Visual Test**: Consider the second example we gave in the previous section. We were able to identify the series in which mean and variance were changing with time, simply by looking at each plot. Similarly, we can plot the data and determine if the properties of the series are changing with time or not. Although its very clear that we have a trend (varying mean) in the above series, this visual approach might not always give accurate results. It is better to confirm the observations using some statistical tests.

2. **Statistical Test**: Instead of going for the visual test, we can use statistical tests like the unit root stationary tests. Unit root indicates that the statistical properties of a given series are not constant with time, which is the condition for stationary time series. Here is the mathematics explanation of the same: Suppose we have a time series:

$$y_t = a * y_{t-1} + \varepsilon_t \qquad (3.2)$$

Where y_t is the value at the time instant t and ε_t is the error term. In order to calculate y_t we need the value of y_{t-1}, which is:

$$y_{t-1} = a * y_{t-2} + \varepsilon_{t-1} \qquad (3.3)$$

If we do that for all observations, the value of y_t will come out to be:

$$y_t = a^n * y_{t-n} + \sigma\varepsilon_{t-i} * a^i \qquad (3.4)$$

If the value of a is 1 (unit) in the above equation, then the predictions will be equal to the y_{t-n} and sum of all errors from t-n to t, which means that the variance will increase with time. This is knows as unit root in a time series. We know that for a stationary time series, the variance must not be a function of time. The unit root tests check the presence of unit root in the series by checking if value of a=1.

The followings can be useful as well to check the stationarity of a time series:

1. **Augmented Dickey-Fuller Test (ADF):** The Dickey Fuller test is one of the most popular statistical tests. It can be used to determine the presence of unit root in the series, and hence help us understand if the series is stationary or not. The null and alternate hypothesis of this test are:

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

If we fail to reject the null hypothesis, we can say that the series is non-stationary. This means that the series can be linear or difference stationary.

Test for stationarity: If the test statistic is less than the critical value, we can reject the nullhypothesis (aka the series is stationary). When the test statistic is greater than the critical value,we fail to reject the null hypothesis (which means the series is not stationary).

2. **Kwiatkowski-Phillips-Schmidt-Shin (KPSS):** KPSS is another test for checking the stationarity of a time series (slightly less popular than the Dickey Fuller test). The null and alternate hypothesis for the KPSS test are opposite that of the ADF test, which often creates confusion.

3.5 Making Time Series Stationary

We have already come across that in order to forecast using time series, first, we have to make the time series stationary. Stationary time series refers to a data model in which statistical properties such as mean, variance and auto-correlation I are constant over time.

Stationarity is important, without stationarity, the data model will lack accuracy at some points of time. In this thesis, our principal motive is to predict the stock market using time series and we will use the ARIMA model to reach there. A data model can be approximated close to perfection using the ARIMA model only if the data set is stationary. Another reason for the need of a stationary time series is to acquire significant mean, variance and correlation with various variables samples. For example, if a series is consistently fluctuating with time, the mean and variance values will fluctuate along, which will underestimate mean and variance. If the properties like mean and variance of the series is undefined or not properly defined, then the correlations between different properties in the series will also remain indefinite. A data set can either be stationary or non-stationary. A time series data model can be rendered to approximate stationarity through some mathematical conversion. We will use that stationary series for the prediction and assume that the statistical properties will remain constant in the future. When the simulations are done using the transformed stationary data model and the anticipated results are achieved, then we will have to add the properties that were reduced from the original series to make it stationary. A unit root test is required to determine if the data model is stationary, if not, then differencing is applied to transform the data set into a stationary one. In our project, we are going to make the time series stationary applying three different methods; Differencing, Log Transformation and decomposing.

3.5.1 Differencing

In data science, differencing technically refers to the consecutive difference between two different sets of data. Differencing removes the changes in the levels of a time series and stabilizes the mean value for the dataset, which means, if there exist any trend or seasonality, is also reduced. A more general perception of differencing is taking derivative. For example, when trying to model a coordinate (x) that changes non-linearly with time and also correlation with other properties is random, differencing transforms it into a nearly linear state and its correlations with different properties also come close to a linear state. ACF plot also helps identify a non-stationary data set, for example, the ACF will go down to zero quickly for a stationary data set, on the other hand, the ACF of of a non stationary data set drops to zero relatively slowly.

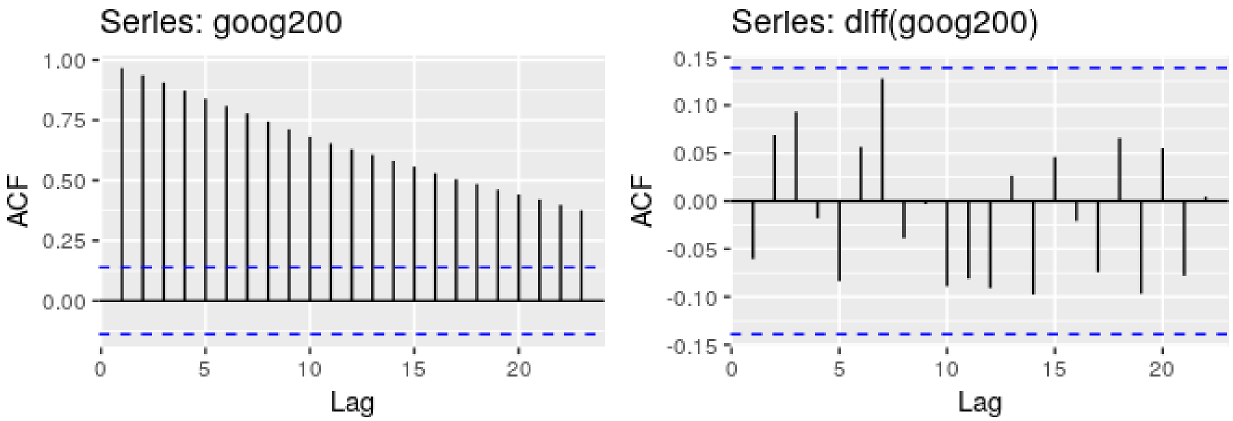


Fig. 3.8 ACF of Google Stock Price

In the Fig. 3.10, we see two different images of google stock price, the one in the right is differenced daily stock price of google which is similar to a white noise series, the one in the left is the ACF of google stock price.

```
Box.test(diff(goog200), lag=10, type="Ljung-Box")
> Box-Ljung test
> data: diff(goog200)
> X-squared = 11, df = 10, p-value = 0.4
```

Random Walk Model

A differenced time series is the switching of states in between successive observations in the original series, it is usually written as

$$y_t' = y_t - y_{t-1} \tag{3.5}$$

The differenced time series will only consist of t-1 values, as it is not possible to difference y1 for the first observation. When the series is differenced for first observation and series contains white noise, the model for the initial series can be expressed as

$$y_t - y_{t-1} = \varepsilon_t \tag{3.6}$$

ε_t denotes white noise , and we get the 'random walk model' rearranging the equation

$$y_t = y_{t-1} + \varepsilon_t \tag{3.7}$$

$$y_t = y_{t\ 1+\varepsilon_t} \tag{3.8}$$

Most financial and economic time series data are far from stationarity as they are expressed in their own set of units, random walk models are extensively used for that type of non-stationary data set. Random walk typically have long periods of trends up-down, it also shows sudden and unpredictable changes in the direction. However, random walk model still exhibits cyclic, trends and other non-stationary behavior, and the future movements are unpredictable, thus the forecast it provides is more likely to be a naive one.

Second Order Differencing

Mostly the differenced time series model is not stationary after the first iteration and it may be necessary to difference the data again to acquire a stationary series. Then the series looks like

$$y''t \ = \ y't \ - y't - 1$$
$$= \ (yt \ - yt - 1) - (yt - 1 - yt - 2)$$
$$= \ yt \ - 2yt - 1 + yt - 2$$

After this second iteration, y"t will have have T-2 values, it is typically not necessary to go beyond a second-order differencing.

Seasonal Differencing

Seasonal difference in time series is technically the difference between two consecutive seasonal observations. For monthly data, seasonal difference is denoted by:

$$y't \ = \ yt \ - ytm \ (3.9)$$

Here, m is the number of seasons. This is also known as "lag-m difference", hence, we deduct the observation after lag-m seasons. If seasonally differenced data is white noise, then the appropriate model would be:

$$yt \ = \ yt - m + \varepsilon t \ (3.10)$$

Forecasts from this model are seasonal naive forecast and are equal to the last observation from the relevant season. The transformation and differencing have made the series look relatively stationary.

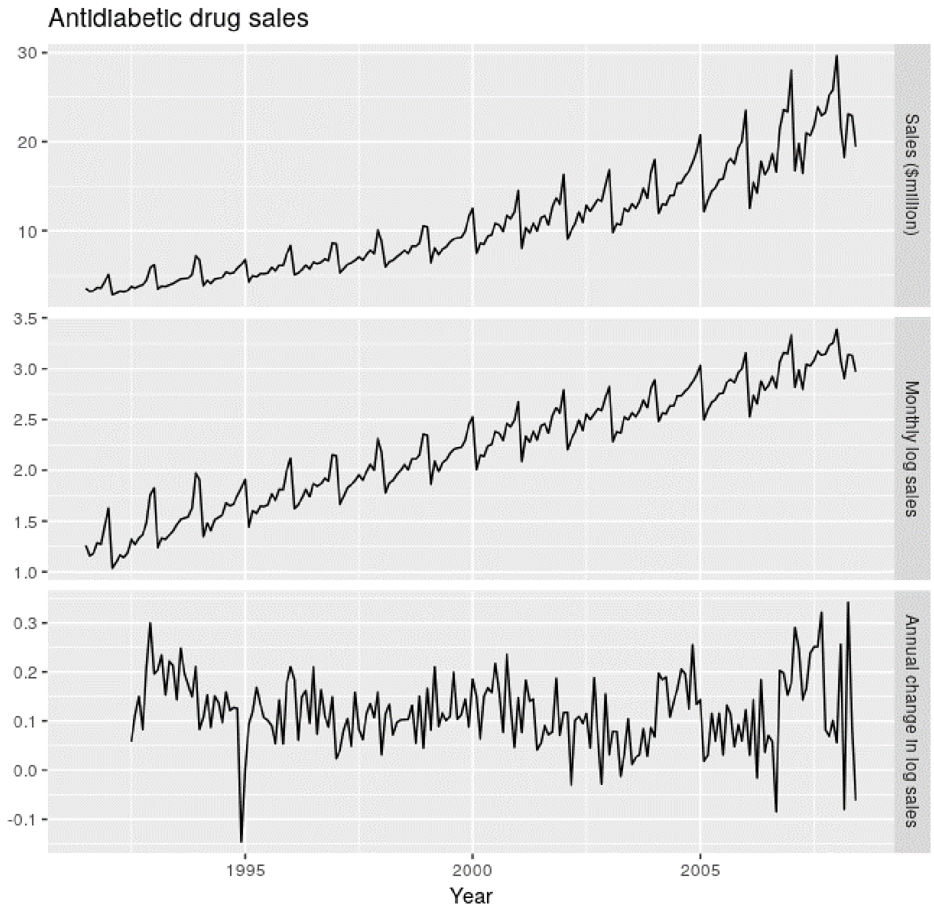


Fig. 3.9 Antibiotics Drug Sale in Australia

Fig. 3.11 shows the seasonal differences of the logarithm of the monthly scripts for antidiabetic drugs sold in Australia.

cbind("Sales ($million)" = a10, "Monthly log sales" = log(a10), "Annual change in log sales" = diff(log(a10),12)) %>% autoplot(facets=TRUE) + xlab("Year") + ylab("") + ggtitle("Antidiabetic drug sales")

To distinguish seasonal differencing from ordinary differencing, ordinary differencing is explained as lag-1 differencing, both lag-1 difference and seasonal difference is needed to obtain stationary data. If seasonal differencing is applied at the first place and then lag-1 differencing, the results would be no different. But, if the data shows a strong seasonal pattern, it is recommended to do seasonal differencing first. If a seasonally differenced series is denoted by,

$$y't \ = \ yt \ - yt - m$$

then the twice differenced series would be,

$$y''t \ = \ y't \ - y't - m$$

It is important that both the lag-1 differencing and seasonal differencing be used, because, lag-1 difference represents the change between two consecutive observations and seasonal represents the change between one year to the next year.

3.5.2 Logarithmic Transformation

In economic analysis and forecasting, there are so many properties and some of them are used in logarithms. Time series analysis also uses log transformation to stabilize the variance of a series, it makes highly skewed distributions less skewed. It is typically used when properties are multiplicatively related and data distribution is positive and highly skewed, for example, log transformation is used in series that are greater than zero and grows exponentially. Fig. 3.12 shows a plot of an airline passenger-miles series that has exponential growth and the variability of the series increases with time.

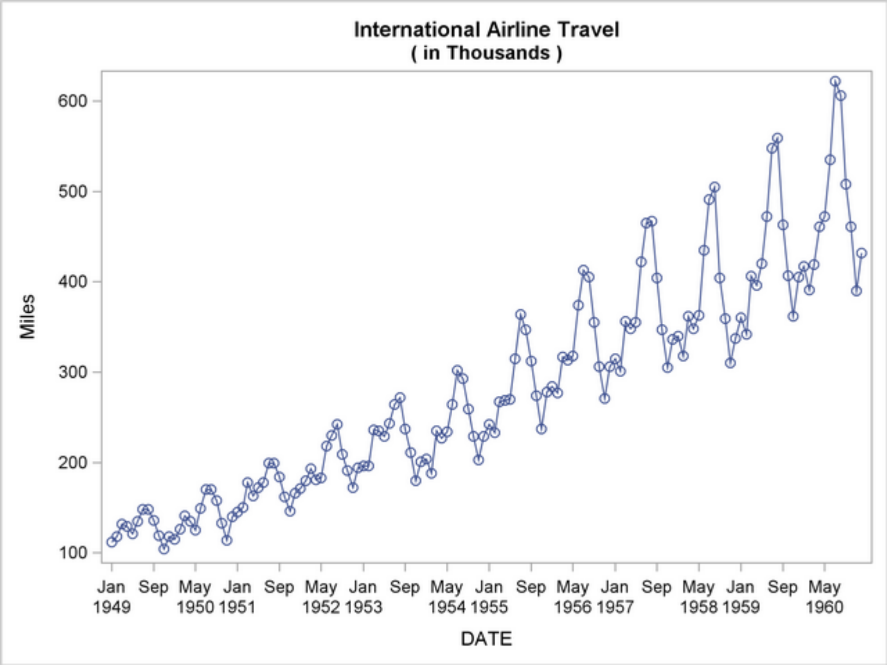


Fig. 3.10 Plot of Airline Passenger-Miles Series

The following pseudo code computes the logarithms of the airline series:

```
data lair;
set sashelp.air;
logair = log( air );
run;
```

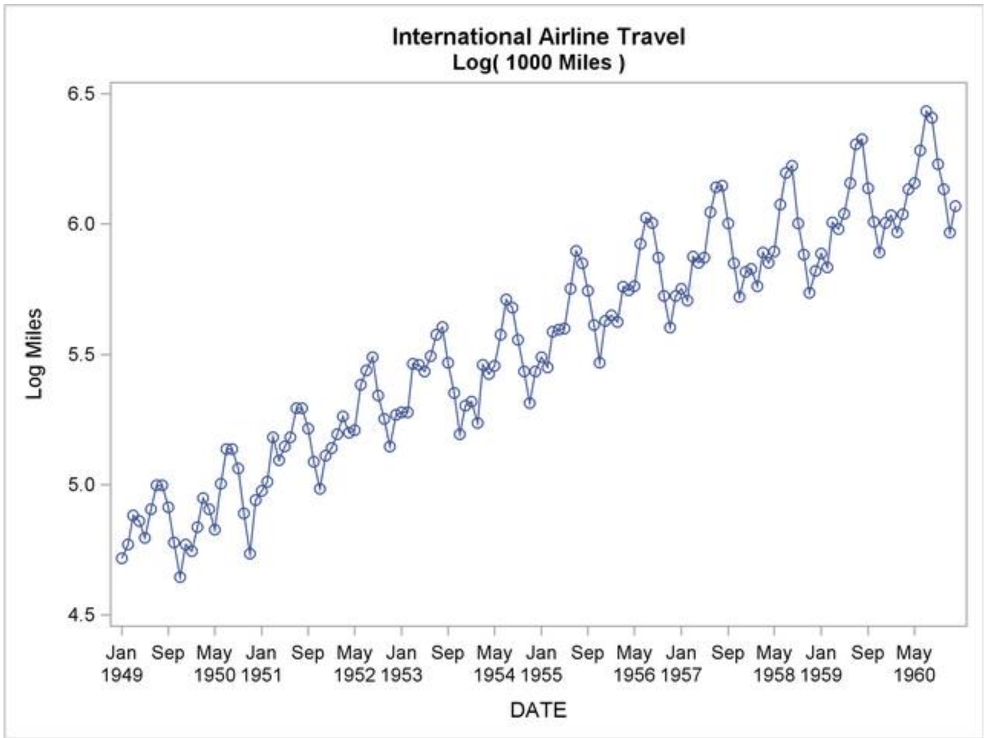


Fig. 3.11 Logged Airline Passenger-Miles Series

Fig. 3.13 shows a plot of the log-transformed airline series. Here, we can see that the logged series showing a linear trend and the variance is constant. Log transformation linearizes growth in a graph but it does not remove any upward trend, if logged data shows any upward trend then we should use some other model that includes a trend factor, for example, we can use Random Walk model.

3.5.3 Decomposition

Time series data shows various patterns and it is helpful to divide the model into different components. Decomposing a time series involves thinking the series as combination of trend, seasonality, level and noise. They are divided into two groups; systematic components and non-systematic components, Systematic components include level, trend, seasonality and non-systematic components include noise. There are two commonly used methods for decomposing these components, Additive decomposition and multiplicative decomposition.

Additive Decomposition

There are few steps to compute additive decomposition

step 1 Suppose, m be an even number, compute the trend-cycle component Tt by 2×m-MA.

If m is an odd number, compute the trend-cycle component Tt using an m-MA

step 2 Calculate the detrended series using: yt −Tt .

step 3 Estimate the seasonal components for all seasons averaging the detrended values for that season. The seasonal values are adjusted to ensure that they result to zero. The seasonal component is acquired by composing together the monthly values, and then replicating that value for each year of data and is denoted St.

step 4 The remainder component is then calculated by deducting the estimated seasonal and trend-cycle components:

$$Rt \ = \ yt \ - Tt \ - St \ (3.11)$$

An additive model is linear, changes over time are consistent and always made by same amount.

Multiplicative decomposition

Multiplicative decomposition steps are similar, except that the subtractions are replaced by divisions.

Step 1 Suppose m is an even number, compute the trend-cycle component Tt using 2×m-MA.

If m is an odd, compute the trend-cycle component Tt using an m-MA.

Step 2 Calculate the detrended series: yt/Tt

Step 3 Estimate the seasonal component for each season by averaging the detrended values for that season. The seasonal component St is gained by stringing together all monthly entries, and then replicating the sequence for each year.

Step 4 The remainder component is then calculated by dividing the estimated seasonal and trend-cycle components:

$$Rt = \frac{yt}{TtSt} \quad (3.12)$$

elecequip autoplot() + xlab("Year") + ggtitle("Classical multiplicative decomposition of electrical equipment index")

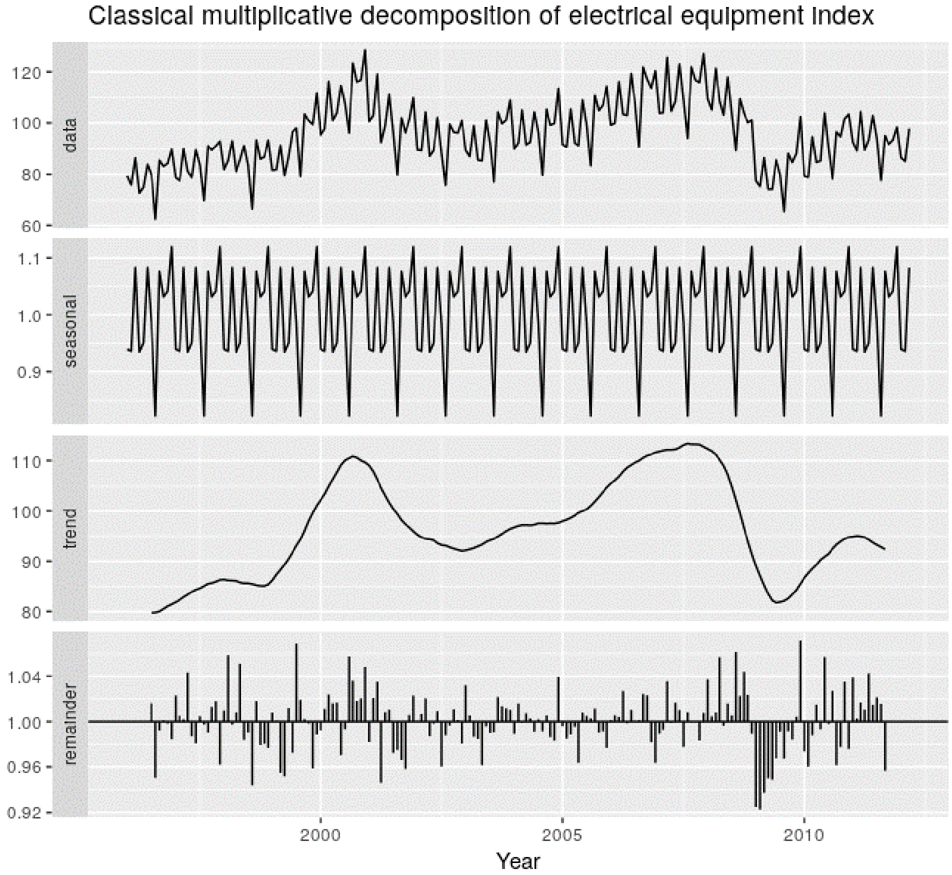


Fig. 3.12 Classical Multiplicative Decomposition for Electrical Equipment

3.6 Plotting ACF & PACF

After removing stationarity from a time series by applying differencing, the next step is to find Auto Regressive (AR) or Moving Average (MA) terms which are needed to correct any auto-correlation in the differenced series in order to fit an ARIMA model. This is usually done by plotting auto-correlation Function (ACF) and partial auto-correlation Function (PACF) graph.

Auto-correlation Function: Auto-correlation function correlates a time series with its own past and future values. Simply, ACF measures and explains the internal association between observations within the time series. If correlation exists within a time series, current and past values can be exploited in order to predict the future values. Thus, auto-correlation helps predict as well as time series modeling. For example, for a stock price time series one might take any spot in time and tries to understand how the time series look like in four weeks on the average, compared to today. It, in other words, determines the strength of the internal association within the time series at a period of four weeks. There are three types of such relations: strong and positive (i.e. today is very similar to the series in four weeks), strong and negative (i.e. today is very dissimilar to the series in four weeks), and weak or no relationship (i.e. no similarity between them). Moreover, time periods could be other than four weeks such as: periods of one week or periods of one year.

ACF for a stationary series is $\rho k : k \geq 0$, where k is the correlation coefficient of Xt with $Xt-k$ where Xt with $Xt-k$ are observations in time series X, at time t and tk respectively. Occasionally, ρ is used for $k<0$ and ρ is an even function meaning $\rho-k = \rho k$. ACF is undefined for a non-stationary time series. For AR (1) with parameter α the ACF is $\rho k = \alpha|k|$ and For MA (1) with parameter β, $\rho k = 0$ except for $\rho 0 = 1$ and $\rho 1 = \beta / (1 + \beta 2)$. ACF plot is nothing but a bar chart of the coefficients of correlation between the time series and lags of itself. At various lags ACF summarizes the correlation of a time series. For instance, the ACF for a time series Xt is given by:

$$Corr(Xt\ ,Xt-k)\ (3.13)$$

where k is the time gap being considered which is called the lag. A lag 1 auto-correlation is the correlation between values that are one time period apart meaning a lag k autocorrelation is the correlation between values that are k time periods apart.

Partial Auto-correlation Function: Partial auto-correlation function determines the partial correlation of a time series with its own lagged values. In contrast with the autocorrelation function, PACF controls for the values at all shorter lags of the series. In time series analysis, PACF plays a vital role by identifying the extent of the lag in an autoregressive model.

Box–Jenkins approach to time series modelling introduced the use of this function whereby plotting the partial auto-correlation function helps determine the appropriate lags p in an AR (p) model, in a mixed ARMA (p, q) model or in an extended

ARIMA (p, d, q) model [3]. The PACF is denoted by φk and said to be the conditional correlation of Xt and Xt−k given all the values from t-k+1 to t-1. Moreover, theoretical relation between the partial auto-correlation function and the auto-correlation function can be exploited to estimate partial auto-correlation. The most commonly used tool to determine the order of an auto-regressive model is partial auto-correlation plots. For an AR(p) model, the partial auto-correlation is 0 at lag p+1 and greater. If the auto-correlation plot determines that an AR model may be appropriate to apply, then the partial auto-correlation plot can be examined for identifying the order. The partial auto-correlations for all higher lags are essentially 0. An indication of the sampling uncertainty of the PACF can be placed on the plot which helps for this purpose and it is constructed on the basis that the true value of the PACF is 0 at any given positive lag. The following table summarizes the ACF and PACF behavior for the time series models.

Table 3.1 Significance of ACF & PACF in time series.

Model	ACF	PACF
AR(p)	Tails off gradually	Cuts off after p lags
MA(q)	Cuts off after q lags	Tails off gradually
ARMA(p,q)	Tails off gradually	Tails off gradually

3.7 Forecasting Models

Auto Regressive (AR): Unlike random time series model, in auto-regressive time series model the current observation Xt depends not only on the current errors but also on the previous time instances. This can be expressed as:

$$Xt = \alpha 1Xt - 1 + \alpha 2Xt - 2 + \ldots\ldots + \alpha pXt - p + \varepsilon t + \lambda \, (3.14)$$

Where α1,α2, . . . ,αp are the coefficients which are tweaked to generate different set of time series data which determines how strongly what happens today depends on previous time instances, is a constant, εt is known as white noise process, and Xt−1 ,Xt−2 ,. . . , Xt−p are previous time instances where p denotes the number of lags of a series also determines the order of an AR model. This lag order parameter, p takes on any positive integer value, and theoretically it can approach infinity. The value εt is considered as the error in forecasting the current value, Xt based totally on a linear combination of its past observations. Since only the random errors at time t-1 determines Xt−1 and earlier, and εt is serially uncorrelated, εt and Xt−p must be uncorrelated for all t and whenever p exceeds 0. Now, selecting the order of AR model is very important before forecasting a time series. Usually, the model parameters can be found by solving a set of linear equation obtained by minimizing the mean squared error. The characteristic of this error is that it decreases as the order of the AR model increases. One of the most common techniques is choosing the numbers of terms in AR model which is known as the model order, p. When the value of this order, p is too low, a highly smoothed spectrum of a time series can be achieved. On the contrary, if an AR model has too high order, there is a risk of getting spurious low-level peaks in the spectrum.

Partial auto-correlation function plays an important role determining the correct order of an auto-regressive model. The following figures illustrates the process of order selection of AR

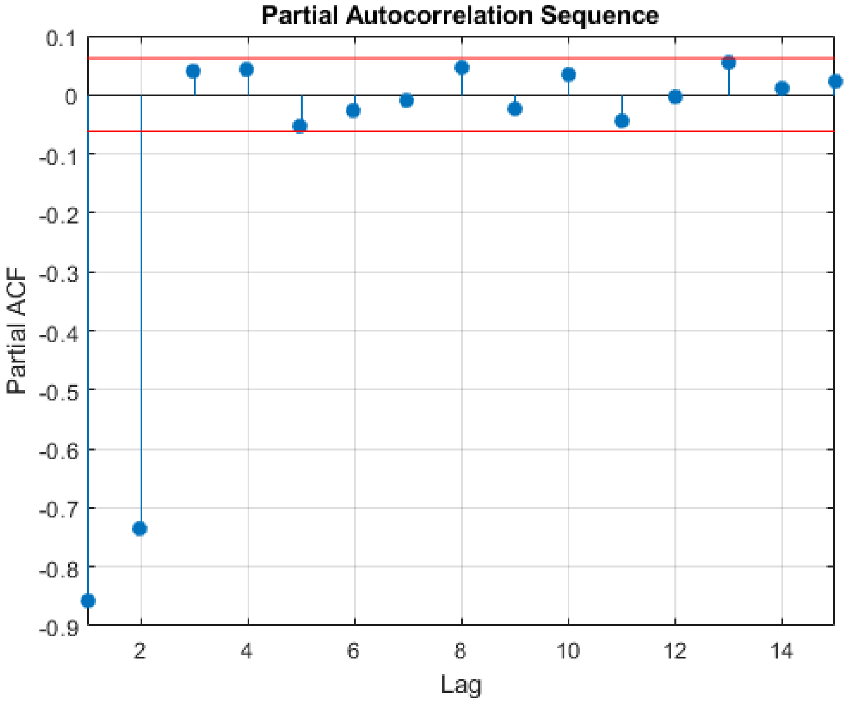


Fig. 3.13 PACF graph of a time series

Fig. 3.15 is a PACF graph of a time series plotted with 95% confidence intervals. It can clearly be seen that only at lags 1 and 2 the values of the partial auto-correlation sequence exceed the 95% confidence level. This consequently says that the appropriate order for the AR model is 2. Practically, a time series is observed without any prior information about model order. Nevertheless, the partial auto-correlation function is a crucial technique for appropriate order selection in stationary auto-regressive time series.

Moving Average (MA): Unlike auto-regressive method, moving average(MA) model considers current errors and error in the previous time instances to estimate current observation.

If the current observation is Xt , the MA model can be expressed as :

$$Xt \ = \ \beta 1\varepsilon t - 1 + \beta 2\varepsilon t - 2 + \ldots\ldots + \beta q\varepsilon t - q + \varepsilon t \ + \lambda \ (3.15)$$

Where β1,β2, . . . ,βq are the coefficients which are used to generate different set of time series data which determines how strongly what happens today depends on error in the previous time instances, is a constant, εt−i is known as white noise processes which denotes current and previous random instances, q denotes the number of lags of a series also determines the order of an MA model. Selecting the order of the MA model is also the trickiest part in order to forecast a time series. The auto-correlation function (ACF) is an important tool to calculate the correct order for MA model just like the PACF is useful to calculate the order for AR process that is the ACF determines how many MA terms are needed in order to remove the remaining auto-correlation from the differenced time series.

For example, if the auto-correlation is significant at lag q but not at any higher lags meaning that the ACF "cuts off" at lag q then exactly q MA terms should be used in forecasting the time series.

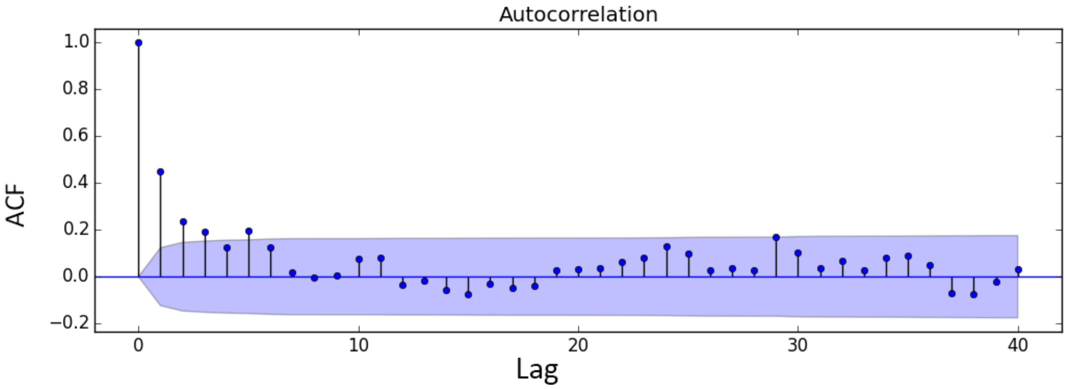


Fig. 3.14 ACF graph of a Time Series

To estimate the order of MA model the following observations can be made from an ACF graph:

- Lag where ACF values die out sufficiently
- Whether the ACF valurs over differencing
- ACF graph shows any significant and easily interpretable peaks at certain lags By considering these things to observe, from Fig. 3.16 it can be easily

seen that ACF values die out at lag 4. Hence, the order of the MA model would be 4.

Auto-regressive Moving Average (ARMA): Usually, in real world, a time series shows some auto-regressive behavior as well as some moving average behavior. To deal with that ARMA model is introduced that is an ARMA model is consisted of AR and MA terms. For example, an ARMA (p, q) has following terms:

- AR(p): The model has p AR terms
- MA(q): The model has q MA terms

So, in an ARMA model current observation, Xt for a time series can be expressed as:

$$\begin{aligned} Xt \; &= \; Auto - regressiveBehavior + MovingAverageBehavior \\ &\qquad + RandomBehavior \\ &= \; (\alpha 1Xt - 1 + \alpha 2Xt - 2 + \ldots\ldots + \alpha pXt - p) + (\beta 1\varepsilon t - 1 + \beta 2\varepsilon t \\ &\qquad - 2 + \ldots\ldots + \beta q\varepsilon t - q) + \varepsilon t \; + \lambda \end{aligned}$$

So, to apply ARMA model on a time series, it must be a stationary series and correct values of p, q should be interpreted from the ACF and PACF.

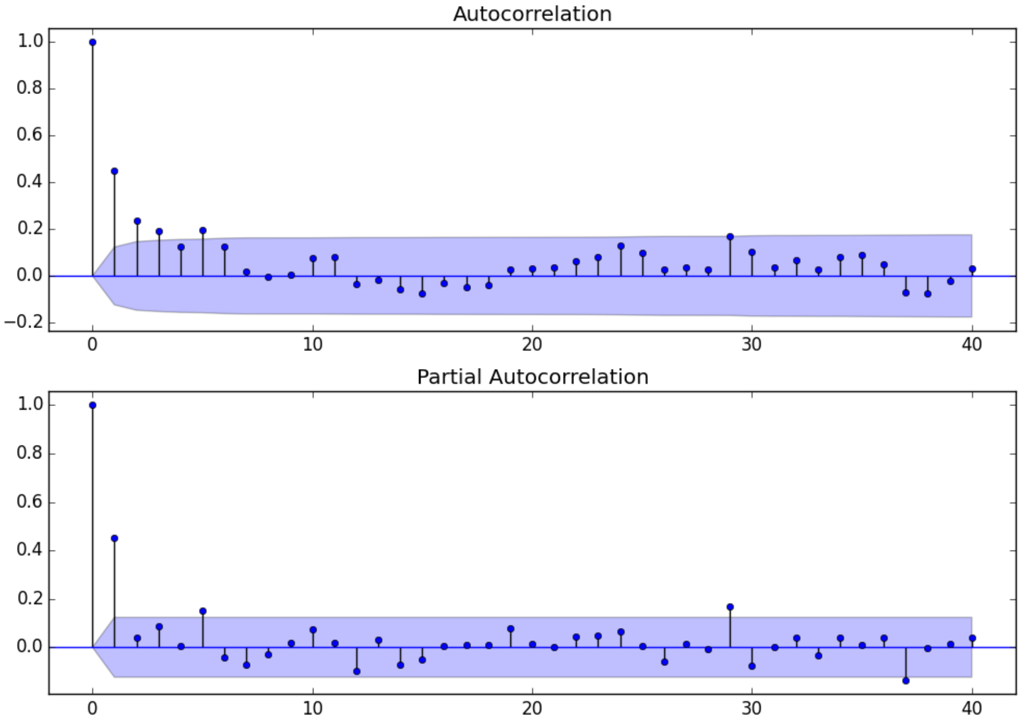


Fig. 3.15 ACF & PACF plot of a time series

For correct AR MA terms selection the steps of AR order selection and MA order selection can be combined. Having that in mind, from Fig. 3.17 it can be decided that the most appropriate order of the ARMA model can be 4, 2 as ACF values die out at lag 4 and

PACF shows spikes at 1 and 2 where 4 2 denotes the value of p,q respectively. However, there could be another way select the correct values of p q that is there are two significant spikes in the PACF plot and one significant spike in the ACF plot where the values die out.

Thus, different observations can be done from ACF and PACF so that the orders p and q of the ARMA model can be chosen correctly.

Auto-regressive Integrated Moving Average (ARIMA): auto-regressive Integrated Moving Average (ARIMA) model is made on the basis of ARMA Model. The main thing that differs from ARMA model is that ARIMA Model make a non-stationary time series to a stationary series before working on it. ARIMA model has widely been applied for forecasting linear time series data [4]. The objective of ARIMA model, also called the Box-Jenkins model, is to identify and estimate a statistical model that can be considered as having generated the sample data which makes stationarity an important pre-requisite of the model. In real world, very few time series are stationary but integrated. In that case, the technique of differencing is applied to convert a non-stationary time series to a stationary time series. More generally, when a time series becomes stationary after differencing d times, the series is referred as I(d). Therefore, if ARMA (p, q) model is applied to a time series which is I(d), it is said that original time series is ARIMA (p, d, q) where p, q denotes AR terms and

MA terms of the series respectively. According to the Box Jenkins methodology, the values of p and q for AR and MA respectively can be calculated by using the correlogram. The ACF graph helps find the correct value of q while the PACF graph is helpful for finding the value of p. While PACF values die out or cut off after lag p for AR (p) model, auto-regressive of order p, for an MA (q) model, moving average of order q, ACF values die out or cut off after lag q. This process of choosing the best order of AR and

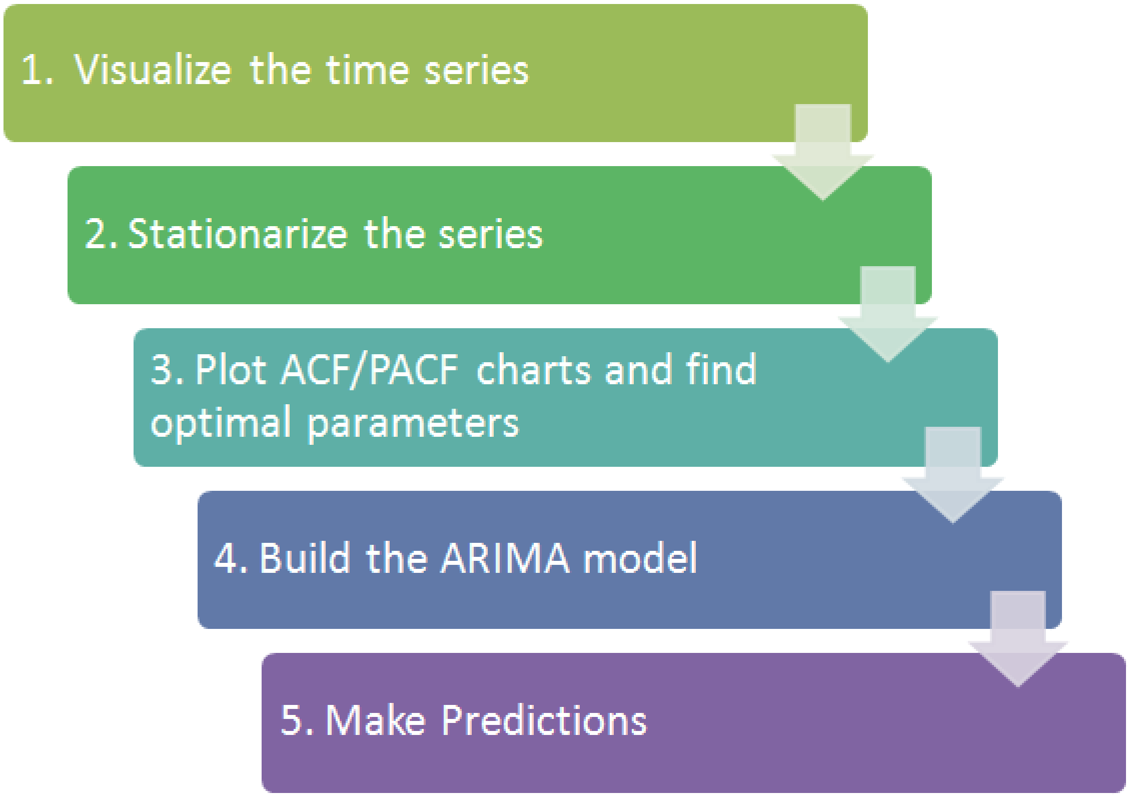
MA model can be confirmed by least values of Akaike's Information Criterion (AIC) where the minimum value of AIC is considered as most suitable[1]. Moreover, model performance estimation can be carried out by the Root mean Square Error (RMSE) values and Mean Absolute Percent Error (MAPE) values. Apart from these techniques for model diagnosis the prediction accuracy is also measured by an accuracy measure which is defined as Accuracy Percent and can be expressed as:

$$AccuracyPercent\ =\ (1 - residual/actualseriesvalue)\ *\ 100 \quad (3.16)$$

where residual is the absolute difference between actual and estimated values. In this paper, we will consider the AIC score to estimate the performance of the time series forecasting models.

Overview of the Framework

This framework(shown below) specifies the step by step approach on '**How to do a Time Series Analysis**':


1. Visualize the time series
2. Stationarize the series
3. Plot ACF/PACF charts and find optimal parameters
4. Build the ARIMA model
5. Make Predictions

4. DATA SET DESCRIPTION

The data set was extracted from top 500 companies of SMP500 under NYSE (New York Stock Exchange) for time period of 5 Yrs.

Each company has approximately 1258 records with six attributes including closing price, low price, high price, opening price, adjacent closing price and volume.

We have chosen ‘Opening Price’ of the stock for prediction.

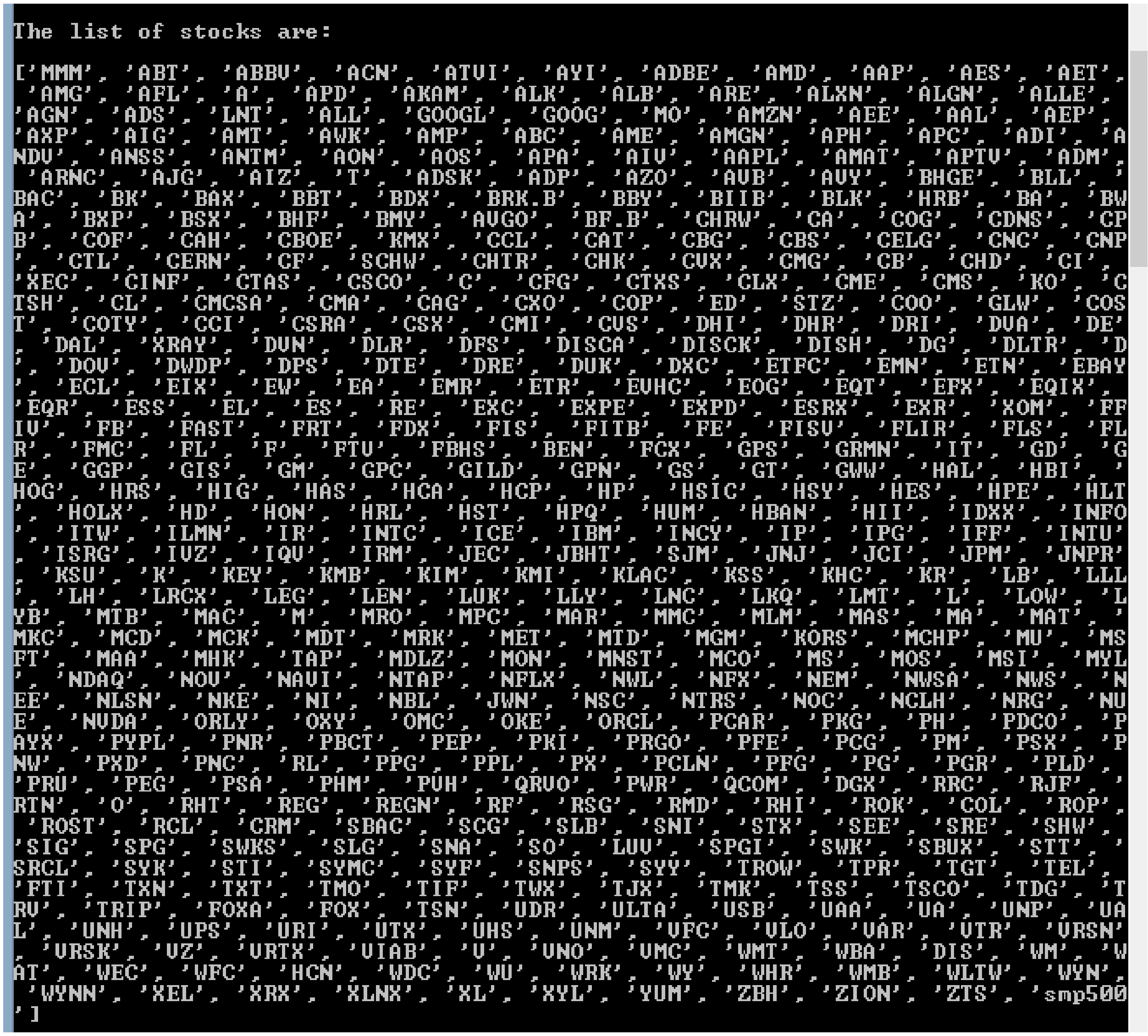
Data set is split into three parts: Training data set, Validation data set and Testing data set.

Our Training dataset consist of 60% of the data, Validation consist of 20% of the data while Testing consist of 20% of the data.

5. Implementation and Results

5.1 Implementation

In this paper, we have taken data of top 500 companies of SMP500 under NYSE. We can select any stock to train or view trained stocks by giving proper command.



```
The list of stocks are:

['MMM', 'ABT', 'ABBV', 'ACN', 'ATVI', 'AYI', 'ADBE', 'AMD', 'AAP', 'AES', 'AET',
 'AMG', 'AFL', 'A', 'APD', 'AKAM', 'ALK', 'ALB', 'ARE', 'ALXN', 'ALGN', 'ALLE',
'AGN', 'ADS', 'LNT', 'ALL', 'GOOGL', 'GOOG', 'MO', 'AMZN', 'AEE', 'AAL', 'AEP',
'AXP', 'AIG', 'AMT', 'AWK', 'AMP', 'ABC', 'AME', 'AMGN', 'APH', 'APC', 'ADI', 'A
NDV', 'ANSS', 'ANTM', 'AON', 'AOS', 'APA', 'AIV', 'AAPL', 'AMAT', 'APTV', 'ADM',
 'ARNC', 'AJG', 'AIZ', 'T', 'ADSK', 'ADP', 'AZO', 'AVB', 'AVY', 'BHGE', 'BLL', '
BAC', 'BK', 'BAX', 'BBT', 'BDX', 'BRK.B', 'BBY', 'BIIB', 'BLK', 'HRB', 'BA', 'BW
A', 'BXP', 'BSX', 'BHF', 'BMY', 'AVGO', 'BF.B', 'CHRW', 'CA', 'COG', 'CDNS', 'CP
B', 'COF', 'CAH', 'CBOE', 'KMX', 'CCL', 'CAT', 'CBG', 'CBS', 'CELG', 'CNC', 'CNP
', 'CTL', 'CERN', 'CF', 'SCHW', 'CHTR', 'CHK', 'CVX', 'CMG', 'CB', 'CHD', 'CI',
'XEC', 'CINF', 'CTAS', 'CSCO', 'C', 'CFG', 'CTXS', 'CLX', 'CME', 'CMS', 'KO', 'C
TSH', 'CL', 'CMCSA', 'CMA', 'CAG', 'CXO', 'COP', 'ED', 'STZ', 'COO', 'GLW', 'COS
T', 'COTY', 'CCI', 'CSRA', 'CSX', 'CMI', 'CVS', 'DHI', 'DHR', 'DRI', 'DVA', 'DE'
, 'DAL', 'XRAY', 'DVN', 'DLR', 'DFS', 'DISCA', 'DISCK', 'DISH', 'DG', 'DLTR', 'D
', 'DOV', 'DWDP', 'DPS', 'DTE', 'DRE', 'DUK', 'DXC', 'ETFC', 'EMN', 'ETN', 'EBAY
', 'ECL', 'EIX', 'EW', 'EA', 'EMR', 'ETR', 'EVHC', 'EOG', 'EQT', 'EFX', 'EQIX',
'EQR', 'ESS', 'EL', 'ES', 'RE', 'EXC', 'EXPE', 'EXPD', 'ESRX', 'EXR', 'XOM', 'FF
IV', 'FB', 'FAST', 'FRT', 'FDX', 'FIS', 'FITB', 'FE', 'FISV', 'FLIR', 'FLS', 'FL
R', 'FMC', 'FL', 'F', 'FTV', 'FBHS', 'BEN', 'FCX', 'GPS', 'GRMN', 'IT', 'GD', 'G
E', 'GGP', 'GIS', 'GM', 'GPC', 'GILD', 'GPN', 'GS', 'GT', 'GWW', 'HAL', 'HBI', '
HOG', 'HRS', 'HIG', 'HAS', 'HCA', 'HCP', 'HP', 'HSIC', 'HSY', 'HES', 'HPE', 'HLT
', 'HOLX', 'HD', 'HON', 'HRL', 'HST', 'HPQ', 'HUM', 'HBAN', 'HII', 'IDXX', 'INFO
', 'ITW', 'ILMN', 'IR', 'INTC', 'ICE', 'IBM', 'INCY', 'IP', 'IPG', 'IFF', 'INTU'
, 'ISRG', 'IVZ', 'IQV', 'IRM', 'JEC', 'JBHT', 'SJM', 'JNJ', 'JCI', 'JPM', 'JNPR'
, 'KSU', 'K', 'KEY', 'KMB', 'KIM', 'KMI', 'KLAC', 'KSS', 'KHC', 'KR', 'LB', 'LLL
', 'LH', 'LRCX', 'LEG', 'LEN', 'LUK', 'LLY', 'LNC', 'LKQ', 'LMT', 'L', 'LOW', 'L
YB', 'MTB', 'MAC', 'M', 'MRO', 'MPC', 'MAR', 'MMC', 'MLM', 'MAS', 'MA', 'MAT', '
MKC', 'MCD', 'MCK', 'MDT', 'MRK', 'MET', 'MTD', 'MGM', 'KORS', 'MCHP', 'MU', 'MS
FT', 'MAA', 'MHK', 'TAP', 'MDLZ', 'MON', 'MNST', 'MCO', 'MS', 'MOS', 'MSI', 'MYL
', 'NDAQ', 'NOV', 'NAVI', 'NTAP', 'NFLX', 'NWL', 'NFX', 'NEM', 'NWSA', 'NWS', 'N
EE', 'NLSN', 'NKE', 'NI', 'NBL', 'JWN', 'NSC', 'NTRS', 'NOC', 'NCLH', 'NRG', 'NU
E', 'NVDA', 'ORLY', 'OXY', 'OMC', 'OKE', 'ORCL', 'PCAR', 'PKG', 'PH', 'PDCO', 'P
AYX', 'PYPL', 'PNR', 'PBCT', 'PEP', 'PKI', 'PRGO', 'PFE', 'PCG', 'PM', 'PSX', 'P
NW', 'PXD', 'PNC', 'RL', 'PPG', 'PPL', 'PX', 'PCLN', 'PFG', 'PG', 'PGR', 'PLD',
'PRU', 'PEG', 'PSA', 'PHM', 'PVH', 'QRVO', 'PWR', 'QCOM', 'DGX', 'RRC', 'RJF', '
RTN', 'O', 'RHT', 'REG', 'REGN', 'RF', 'RSG', 'RMD', 'RHI', 'ROK', 'COL', 'ROP',
 'ROST', 'RCL', 'CRM', 'SBAC', 'SCG', 'SLB', 'SNI', 'STX', 'SEE', 'SRE', 'SHW',
'SIG', 'SPG', 'SWKS', 'SLG', 'SNA', 'SO', 'LUV', 'SPGI', 'SWK', 'SBUX', 'STT', '
SRCL', 'SYK', 'STI', 'SYMC', 'SYF', 'SNPS', 'SYY', 'TROW', 'TPR', 'TGT', 'TEL',
'FTI', 'TXN', 'TXT', 'TMO', 'TIF', 'TWX', 'TJX', 'TMK', 'TSS', 'TSCO', 'TDG', 'T
RV', 'TRIP', 'FOXA', 'FOX', 'TSN', 'UDR', 'ULTA', 'USB', 'UAA', 'UA', 'UNP', 'UA
L', 'UNH', 'UPS', 'URI', 'UTX', 'UHS', 'UNM', 'VFC', 'VLO', 'VAR', 'VTR', 'VRSN'
, 'VRSK', 'VZ', 'VRTX', 'VIAB', 'V', 'VNO', 'VMC', 'WMT', 'WBA', 'DIS', 'WM', 'W
AT', 'WEC', 'WFC', 'HCN', 'WDC', 'WU', 'WRK', 'WY', 'WHR', 'WMB', 'WLTW', 'WYN',
 'WYNN', 'XEL', 'XRX', 'XLNX', 'XL', 'XYL', 'YUM', 'ZBH', 'ZION', 'ZTS', 'smp500
']
```

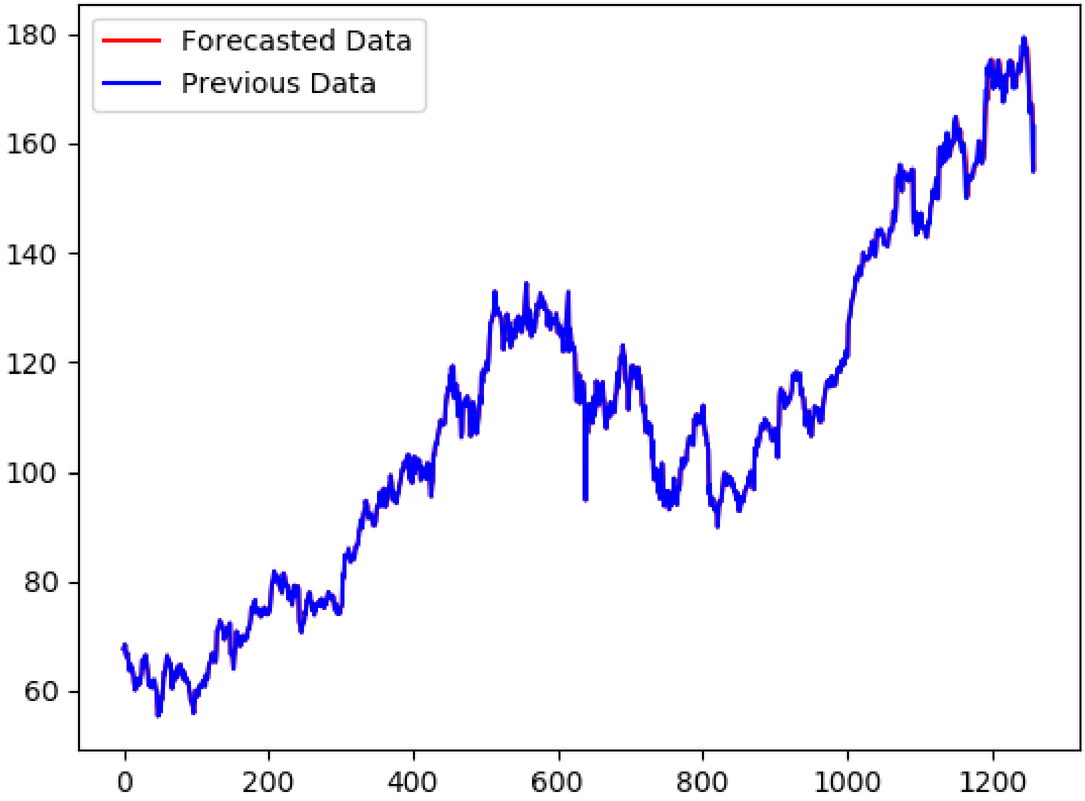
After choosing the stock for training, first we stationarize the dataset after that we iterated over different values of (p,d,q) and chosen that (p,d,q) for which aic value is minimum.

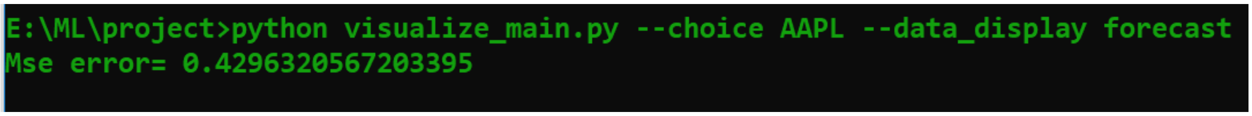
We have chosen 'Opening price' of the stock for prediction. After getting optimal values of (p,d,q) we predict the stock price of next day. We check the error between predicted and actual data then we again add the actual data to previous data and again predict the price for next day.

5.2 Result analysis

The graphs below represents the actual prices and predicted stock prices of the next day for stocks of different companies.

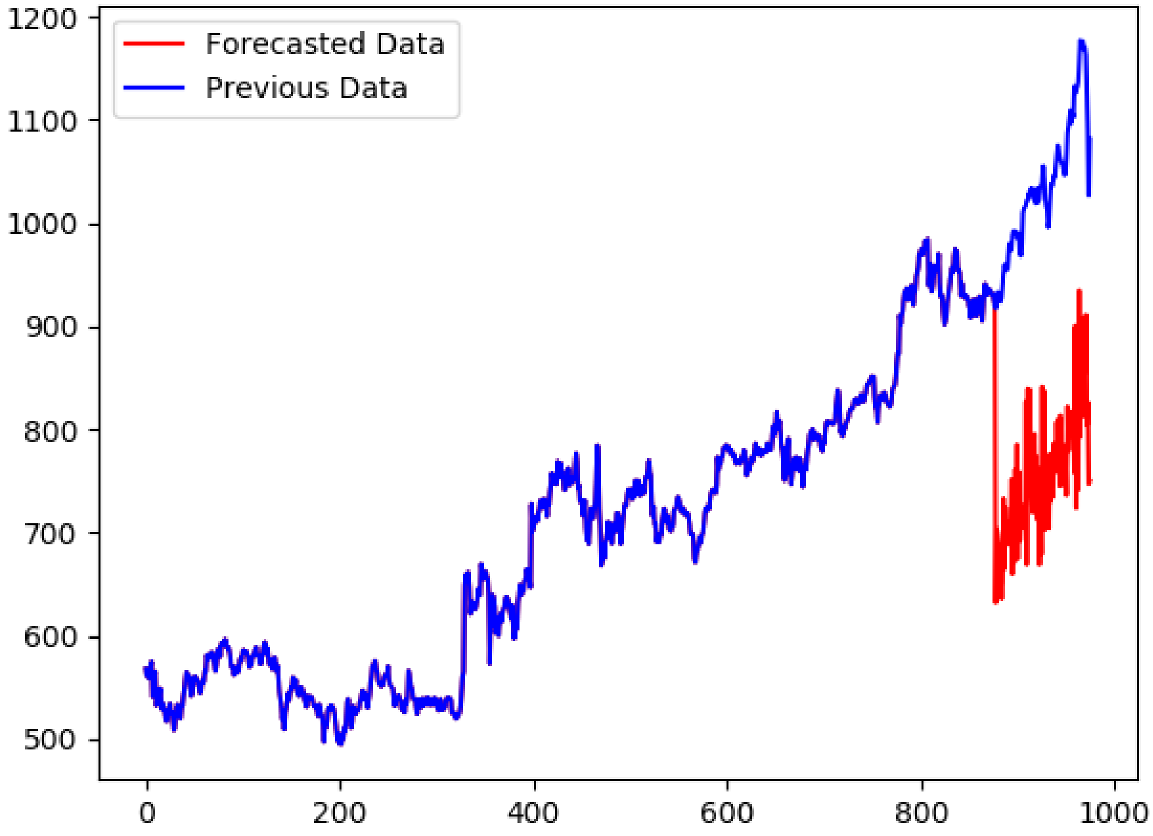
Mean Squared Error values for most of the stocks significantly small with respect to actual data.

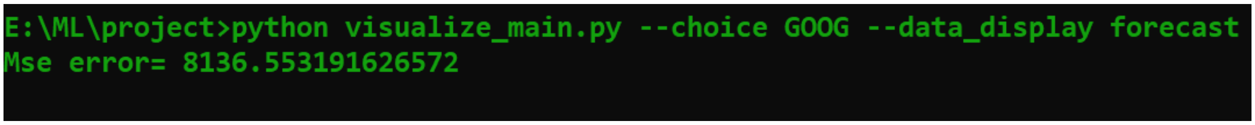




```
E:\ML\project>python visualize_main.py --choice AAPL --data_display forecast
Mse error= 0.4296320567203395
```

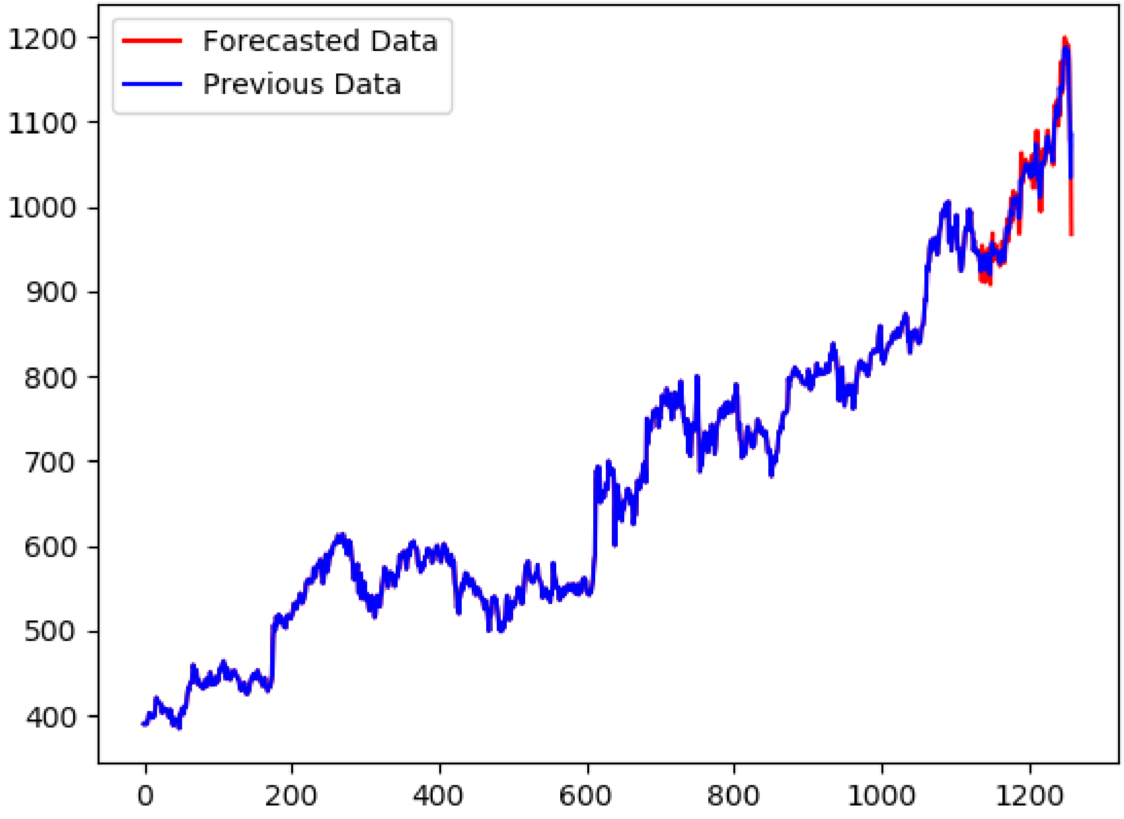
Fig: Stock price prediction of company code-'AAPL'

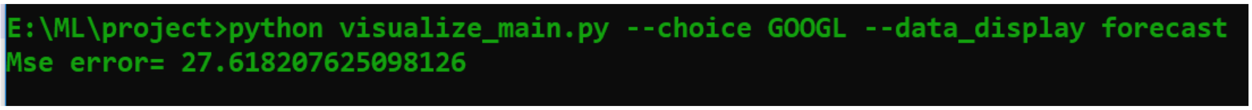




```
E:\ML\project>python visualize_main.py --choice GOOG --data_display forecast
Mse error= 8136.553191626572
```

Fig: Stock price prediction of company code-'GOOG'





```
E:\ML\project>python visualize_main.py --choice GOOGL --data_display forecast
Mse error= 27.618207625098126
```

Fig: Stock price prediction of company code-'GOOGL'

6. CONCLUSION

- Our model has assumed that there is no seasonality in data to prevent manual choice of parameters.
- Our model predicted the future prices for stocks in a very general sense, strongly. We believe this is not because of the model itself, but because

of the market performance during and after the provided data. Our model was mostly unsuccessful in ignoring the noise, even with our best efforts to station the data.

- We predicted the stock price of next day given the past stock prices and mean squared error are significantly low for most of the stocks.
- It can be efficiently used for short term prediction.

7. FUTURE SCOPE

This model is very beneficial to predict stock prices for short term. But stock price not only depends upon previous stock prices but also on status of the company in the market. We can use sentiment analysis for that.

Apart from that, to prevent manual intervention, we didn't accounted for seasonality in the data. For Better performance of the algorithm, we need to choose parameters manually for seasonality and use SARIMA instead of ARIMA.

REFERENCES

[1] Anderson, D. R. (2007). Model based inference in the life sciences: a primer on evidence. Springer Science & Business Media.

[2] Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. (2014). Stock price prediction using the arima model. In Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on, pages 106–112. IEEE.

[3] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

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

[5] Gupta, A. and Dhingra, B. (2012). Stock market prediction using hidden markov models. In Engineering and Systems (SCES), 2012 Students Conference on, pages 1–4. IEEE.

[6] Hassan, M. R. (2009). A combination of hidden markov model and fuzzy model for stock market forecasting. Neurocomputing, 72(16-18):3439–3446. [7] Kryzanowski, L., Galler, M., and Wright, D. W. (1993). Using artificial neural networks to pick stocks. Financial Analysts Journal, 49(4):21–27.

[8] Labiad, B., Berrado, A., and Benabbou, L. (2016). Machine learning techniques for short term stock movements classification for moroccan stock exchange. In Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on, pages 1–6. IEEE.

[9] Mostafa, M. M. (2010). Forecasting stock exchange movements using neural networks: Empirical evidence from kuwait. Expert Systems with Applications, 37(9):6302–6309. [10] Schoneburg, E. (1990). Stock price

prediction using neural networks: A project report. Neurocomputing, 2(1):17–27.

[11] Wei, L.-Y. (2013). A hybrid model based on anfis and adaptive expectation genetic algorithm to forecast taiex. Economic Modelling, 33:893–899.

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

[13] https://machinelearningmastery.com/sarima-for-time-series
/forecasting-in-python/

[14] https://people.duke.edu/~rnau/411arim.htm

[15] https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/

[16] https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/

[17] https://www.youtube.com/watch?v=d4Sn6ny_5LI

[18] P. Pai, C. Lin, "A hybrid ARIMA and support vector machines model in stock price prediction", *Omega*, vol. 33, pp. 497-505, 2005.

[19] JJ. Wang, JZ. Wang, Z.G. Zhang, S.P. Guo, "Stock index forecasting based on a hybrid model", *Omega*, vol. 40, pp. 758-766, 2012.

[20] http://ijssst.info/Vol-15/No-4/data/4923a105.pdf