# “ARIMA modelling based Stock Prediction”

*A*

*Project Report*

*Submitted in partial fulfillment of the*

*Requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

***in***

# COMPUTER SCIENCE & ENGINEERING

**by**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| Anirban Mukherjee | R100214010 |
| Vibhor | R100214070 |
| Anshika Agarwal | R100214015 |

***Under the guidance of***

## Mr. Uppara Rajnikanth



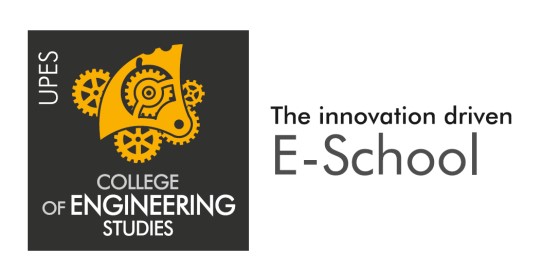
# Department of Computer Science & Engineering

**Centre for Information Technology**

**University of Petroleum & Energy Studies,**

**Dehradun, Uttarakhand**

**April 2018**



**CANDIDATE’S DECLARATION**

We hereby certify that the project work entitled “Machine Learning in Android Using Tensor Flow” in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Open Source and Open Standards and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of our work carried out during a period from January, 2018 to May, 2018 under the supervision of Mr Uppara Rajanikanth, Assistant Professor ,Center for Information Technology

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 01-04-2018

**Mr Uppara Rajanikanth**

Project Guide

# Dr. Ajay Prasad

Program Head, CSE- Open Source And Open Standards

School Of Computer Science & Engineering (SoCSE)

University of Petroleum & Energy Studies

# ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide Mr Uppara Rajanikanth, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, Mr Ajay Prasad, for his great support in doing our project in Python Programming Language at CIT.

We are also grateful to Dr. Manish Prateek, Associate Dean and Dr. Kamal Bansal, Dean COES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

**Name:** Anirban Mukherjee Vibhor Anshika Agarwal

**Roll No.:** R100214010 R100214070 R100214015

# ABSTRACT

‘Time’ is the most important factor which ensures success in a business. It’s difficult to keep up with the pace of time.  But, technology has developed some powerful methods using which we can ‘see things’ ahead of time. Don’t worry, I am not talking about Time Machine. Let’s be realistic here!

I’m talking about the methods of prediction & forecasting. One such method, which deals with time based data is **Time Series Modeling**. As the name suggests, it involves working on time (years, days, hours, minutes) based data, to derive hidden insights to make informed decision making.

Time series models are very useful models when you have serially correlated data. Most of business houses work on time series data to analyze sales number for the next year, website traffic, competition position and much more. However, it is also one of the areas, which many analysts do not understand.

**Keywords:**

Time Series Model

**TABLE OF CONTENTS**

**S.No. Contents Page No**

1. **Introduction 1**
   1. System Requirements 2
   2. Main Objective 3
   3. Sub Objectives 3
2. **System Analysis 4**
   1. Existing System 4
   2. Problem Statement 4
   3. Proposed System 4
3. **Implementation 5**
   1. Methodology 12
   2. Diagram 13
4. **Output screens 7**
5. **Limitations and Future Enhancements 11**
6. **Future Prospects 12**
7. **Conclusion 1**

**References 20**

# 1. INTRODUCTION

Forecasting involves predicting values for a variable using its historical data points or it can also involve predicting the change in one variable given the change in the value of another variable. Forecasting approaches are primarily categorized into qualitative forecasting and quantitative forecasting. Time series forecasting falls under the category of quantitative forecasting wherein statistical principals and concepts are applied to a given historical data of a variable to forecast the future values of the same variable. Some time series forecasting techniques used include:

* Autoregressive Models (AR)
* Moving Average Models (MA)
* Seasonal Regression Models
* Distributed Lags Models

[Shiny](http://shiny.rstudio.com/) is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.

1.1 Autoregressive Models:

In [statistics](https://en.wikipedia.org/wiki/Statistics) and [signal processing](https://en.wikipedia.org/wiki/Signal_processing), an autoregressive (AR) model is a representation of a type of [random process](https://en.wikipedia.org/wiki/Random_process); as such, it is used to describe certain time-varying processes in [nature](https://en.wikipedia.org/wiki/Natural_science), [economics](https://en.wikipedia.org/wiki/Economics), etc. The autoregressive model specifies that the output variable depends [linearly](https://en.wikipedia.org/wiki/Linear_prediction) on its own previous values and on a [stochastic](https://en.wikipedia.org/wiki/Stochastic_variable) term (an imperfectly predictable term); thus the model is in the form of a stochastic [difference equation](https://en.wikipedia.org/wiki/Difference_equation).

Together with the [moving-average (MA) model](https://en.wikipedia.org/wiki/Moving-average_model), it is a special case and key component of the more general [ARMA](https://en.wikipedia.org/wiki/Autoregressive%E2%80%93moving-average_model) and [ARIMA](https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average) models of [time series](https://en.wikipedia.org/wiki/Time_series), which have a more complicated stochastic structure; it is also a special case of the [vector autoregressive model](https://en.wikipedia.org/wiki/Vector_autoregressive_model) (VAR), which consists of a system of more than one interlocking stochastic difference equation in more than one evolving random variable.

1.2 Moving Average Model:

In [time series analysis](https://en.wikipedia.org/wiki/Time_series_analysis), the moving-average (MA) model is a common approach for modeling [univariate](https://en.wikipedia.org/wiki/Univariate) time series. The moving-average model specifies that the output variable depends [linearly](https://en.wikipedia.org/wiki/Linear_prediction) on the current and various past values of a [stochastic](https://en.wikipedia.org/wiki/Stochastic) (imperfectly predictable) term.

Together with the [autoregressive (AR) model](https://en.wikipedia.org/wiki/Autoregressive_model), the moving-average model is a special case and key component of the more general [ARMA](https://en.wikipedia.org/wiki/Autoregressive%E2%80%93moving-average_model) and [ARIMA](https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average) models of [time series](https://en.wikipedia.org/wiki/Time_series), which have a more complicated stochastic structure.

The moving-average model should not be confused with the [moving average](https://en.wikipedia.org/wiki/Moving_average), a distinct concept despite some similarities.

Contrary to the AR model, the finite MA model is always [stationary](https://en.wikipedia.org/wiki/Stationary_process).

1.3 Seasonal Regression Model:

Seasonal fluctuations in a time series can be contrasted with cyclical patterns. The latter occur when the data exhibits rises and falls that are not of a fixed period. These fluctuations are usually due to economic conditions and are often related to the "business cycle." The period of time usually extends beyond a single year and the fluctuations are usually of at least two years

* A [run sequence plot](https://en.wikipedia.org/wiki/Run_sequence_plot) will often show seasonality
* A seasonal plot will show the data from each season overlapped[[4]](https://en.wikipedia.org/wiki/Seasonality#cite_note-4)
* A [seasonal subseries plot](https://en.wikipedia.org/wiki/Seasonal_subseries_plot) is a specialized technique for showing seasonality
* Multiple [box plots](https://en.wikipedia.org/wiki/Box_plot) can be used as an alternative to the seasonal subseries plot to detect seasonality
* An [autocorrelation plot](https://en.wikipedia.org/wiki/Autocorrelation_plot) (ACF) and a spectral plot can help identify seasonality.
* Seasonal Index measures how much the average for a particular period tends to be above (or below) the expected value

1.4 Distributed Lag Models:

In [statistics](https://en.wikipedia.org/wiki/Statistics) and [econometrics](https://en.wikipedia.org/wiki/Econometrics), a distributed lag model is a model for [time series](https://en.wikipedia.org/wiki/Time_series) data in which a [regression](https://en.wikipedia.org/wiki/Linear_regression) equation is used to predict current values of a [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) based on both the current values of an [explanatory variable](https://en.wikipedia.org/wiki/Explanatory_variable) and the lagged (past period) values of this explanatory variable.

The starting point for a distributed lag model is an assumed structure of the form {\displaystyle y\_{t}=a+w\_{0}x\_{t}+w\_{1}x\_{t-1}+w\_{2}x\_{t-2}+...+{\text{error term}}}or the form {\displaystyle y\_{t}=a+w\_{0}x\_{t}+w\_{1}x\_{t-1}+w\_{2}x\_{t-2}+...+w\_{n}x\_{t-n}+{\text{error term}},}where yt is the value at time period t of the dependent variable y, a is the intercept term to be estimated, and wi is called the lag weight (also to be estimated) placed on the value periods previously of the explanatory variable x. In the first equation, the dependent variable is assumed to be affected by values of the independent variable arbitrarily far in the past, so the number of lag weights is infinite and the model is called an infinite distributed lag model. In the alternative, second, equation, there are only a finite number of lag weights, indicating an assumption that there is a maximum lag beyond which values of the independent variable do not affect the dependent variable; a model based on this assumption is called a finite distributed lag model.

# Requirement Analysis (System Requirements)

**Software**

* RStudio

**Hardware**

* Ram 2GB and above

**Libraries**

* Tidyverse
* Timeseries
* Forecast
* Shiny
* Quantmod
* Quandl

## 2. Objective

The objective of this project is to predict the stock prices using time series analysis that is the ARIMA modelling. ARIMA modelling is used to generate a white noise resemblance type time series out of the given data (stock data). The ARIMA stands for Auto Regressive Integrated Moving Average.

The terms Auto Regression and Moving Average are the two functions that is used to determine the type of series we want to predict. Usually in R the two terms are p and q separated by d that is the lags or the differencing. The main objective is to come up with the least p value and keep the AIC (Akakie) and BIC (Bayesian) factors low as possible so that the resulting time series resembles a white noise and the errors are minimizes when calculated by least square estimates.

3. Problem Statement:

To find the correct model for stock prediction using the ARIMA analysis. This is achieved by analyzing the ACF (the autocorrelation) and PACF (Partial autocorrelation) Factors to determine the pa nd q values respectively. The value of “d” is usually chosen by hit and trail method by calling the summary function in the fitted model with a value of “d”.

The 80% and 95% accuracy band is predicted using the forecast function is tested upon the tested part of the dataset. The dataset is an S&P index stock prices obtained from yahoo finance. The idea is to match to a precision value for the test data so that we get the least squared sum for their forecasted and actual data.

4. Implementation

**4.1. Testing and Ensuring Stationarity**

To model a time series with the Box-Jenkins approach, the series has to be stationary. A stationary time series means a time series without trend, one having a constant mean and variance over time, which makes it easy for predicting values.

**4.2. Testing for stationarity**

We test for stationarity using the Augmented Dickey-Fuller unit root test. The p-value resulting from the ADF test has to be less than 0.05 or 5% for a time series to be stationary. If the p-value is greater than 0.05 or 5%, you conclude that the time series has a unit root which means that it is a non-stationary process.

**4.3 Differencing**

To convert a non-stationary process to a stationary process, we apply the differencing method. Differencing a time series means finding the differences between consecutive values of a time series data. The differenced values form a new time series dataset which can be tested to uncover new correlations or other interesting statistical properties.

We can apply the differencing method consecutively more than once, giving rise to the “first differences”, “second order differences”, etc.

We apply the appropriate differencing order (d) to make a time series stationary before we can proceed to the next step.

**4.4. Identification of p and q**

In this step, we identify the appropriate order of Autoregressive (AR) and Moving average (MA) processes by using the Autocorrelation function (ACF) and Partial Autocorrelation function (PACF).  Please refer to our blog, [“Starting out with Time Series”](https://www.quantinsti.com/blog/starting-time-series/) for an explanation of ACF and PACF functions.

**4.5 Identifying the p order of AR model**

For AR models, the ACF will dampen exponentially and the PACF will be used to identify the order (p) of the AR model. If we have one significant spike at lag 1 on the PACF, then we have an AR model of the order 1, i.e. AR(1). If we have significant spikes at lag 1, 2, and 3 on the PACF, then we have an AR model of the order 3, i.e. AR(3).

**4.6 Identifying the q order of MA model**

For MA models, the PACF will dampen exponentially and the ACF plot will be used to identify the order of the MA process. If we have one significant spike at lag 1 on the ACF, then we have an MA model of the order 1, i.e. MA(1). If we have significant spikes at lag 1, 2, and 3 on the ACF, then we have an MA model of the order 3, i.e. MA(3).

**4.7. Estimation and Forecasting**

Once we have determined the parameters (p,d,q) we estimate the accuracy of the ARIMA model on a training data set and then use the fitted model to forecast the values of the test data set using a forecasting function. In the end, we cross check whether our forecasted values are in line with the actual values.

5. Procedure

5.1. First we pull the dataset and clean it. Then we apply the plotting method to analysie the type of data we are dealing with.

library(quantmod);library(tseries);

library(timeSeries);library(forecast);library(xts);

# Pull data from Yahoo finance

getSymbols('TECHM.NS', from='2012-01-01', to='2015-01-01')

# Select the relevant close price series

stock\_prices = TECHM.NS[,4]

In the next step, we compute the logarithmic returns of the stock as we want the ARIMA model to forecast the log returns and not the stock price. We also plot the log return series using the plot function.

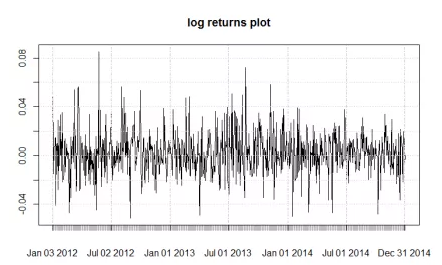
# Compute the log returns for the stock

stock = diff(log(stock\_prices),lag=1)

stock = stock[!is.na(stock)]

# Plot log returns

plot(stock,type='l', main='log returns plot')



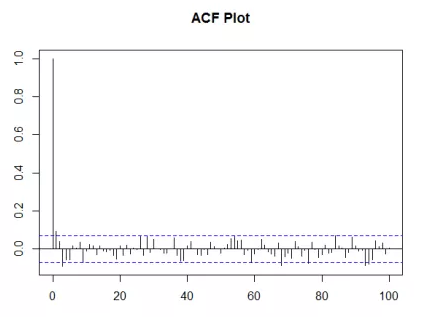
5.2. Finding the ACF and PACF values.

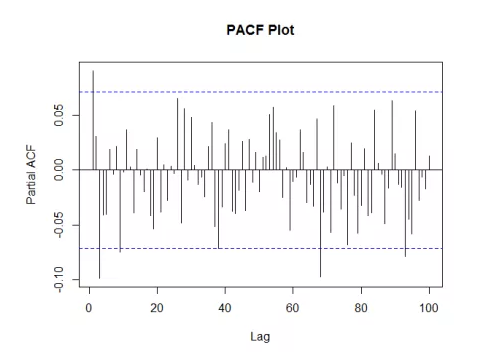
# Apply the ACF and PACF functions

par(mfrow = c(1,1))

acf.stock = acf(stock[c(1:breakpoint),], main='ACF Plot', lag.max=100)

pacf.stock = pacf(stock[c(1:breakpoint),], main='PACF Plot', lag.max=100)





We know that for AR models, the ACF will dampen exponentially and the PACF plot will be used to identify the order (p) of the AR model. For MA models, the PACF will dampen exponentially and the ACF plot will be used to identify the order (q) of the MA model. From these plots let us select AR order = 2 and MA order = 2. Thus, our ARIMA parameters will be (2,0,2).

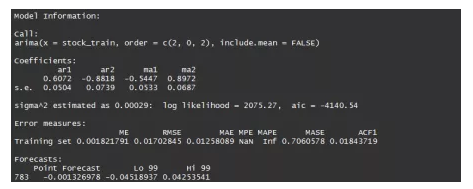
Our objective is to forecast the entire returns series from breakpoint onwards. We will make use of the For Loop statement in R and within this loop we will forecast returns for each data point from the test dataset.

In the code given below, we first initialize a series which will store the actual returns and another series to store the forecasted returns.  In the For Loop, we first form the training dataset and the test dataset based on the dynamic breakpoint.

We call the arima function on the training dataset for which the order specified is (2, 0, 2). We use this fitted model to forecast the next data point by using the forecast.Arima function. The function is set at 99% confidence level. One can use the confidence level argument to enhance the model. We will be using the forecasted point estimate from the model. The “h” argument in the forecast function indicates the number of values that we want to forecast, in this case, the next day returns.

We can use the summary function to confirm the results of the ARIMA model are within acceptable limits. In the last part, we append every forecasted return and the actual return to the forecasted returns series and the actual returns series respectively.

5.3. finding the coefficients for the liner model by checkresiudals function on the model.

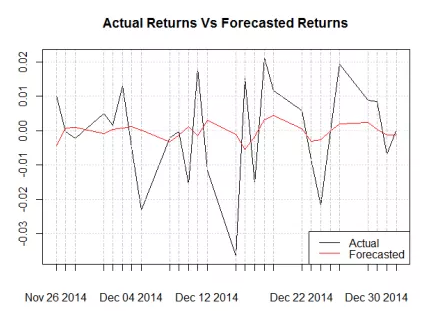


From the coefficients obtained, the return equation can be written as:

**Yt = 0.6072\*Y(t-1)  – 0.8818\*Y(t-2) – 0.5447ε(t-1)+ 0.8972ε(t-2)**

The standard error is given for the coefficients, and this needs to be within the acceptable limits. The Akaike information criterion (AIC) score is a good indicator of the ARIMA model accuracy. Lower the AIC score better the model. We can also view the ACF plot of the residuals; a good ARIMA model will have its autocorrelations below the threshold limit. The forecasted point return is -0.001326978, which is given in the last row of the output.

5.4. Plotting the graph of forecasted.

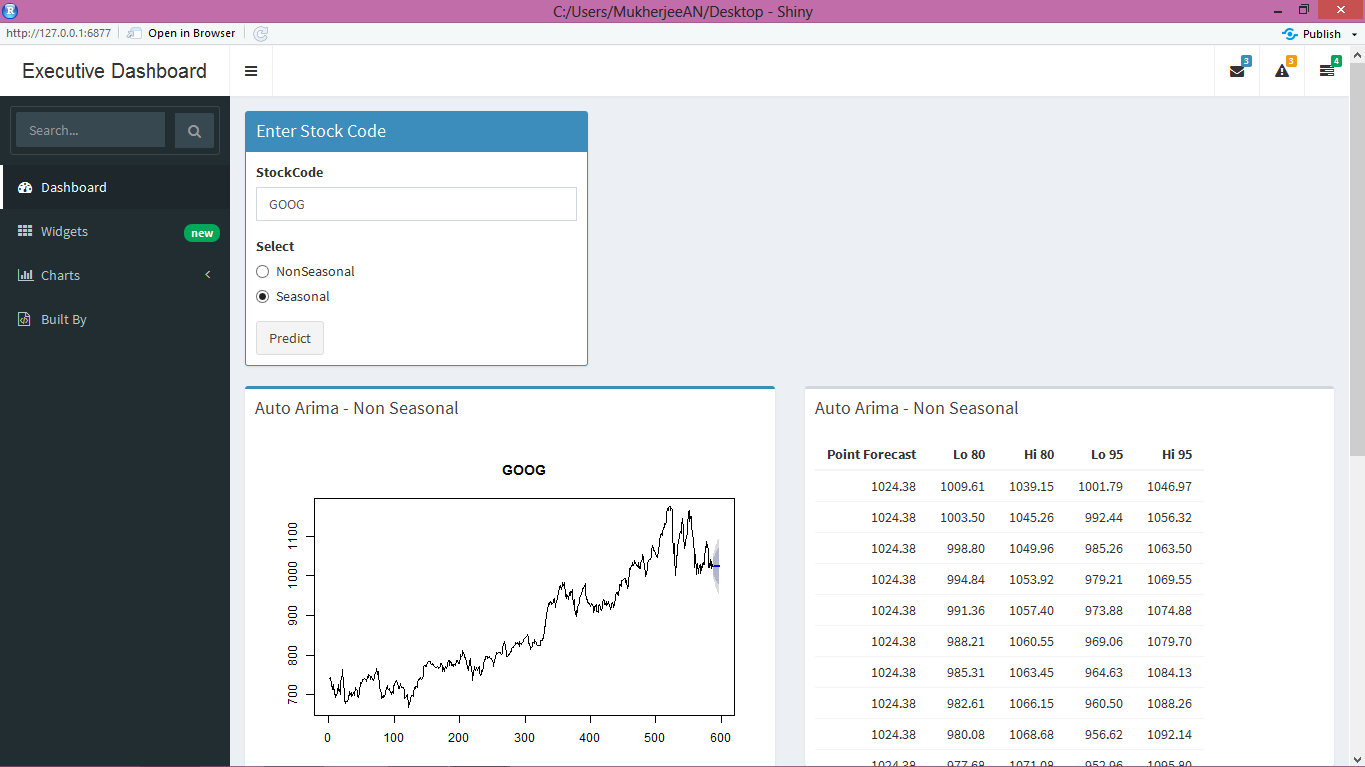


4. Output Screens

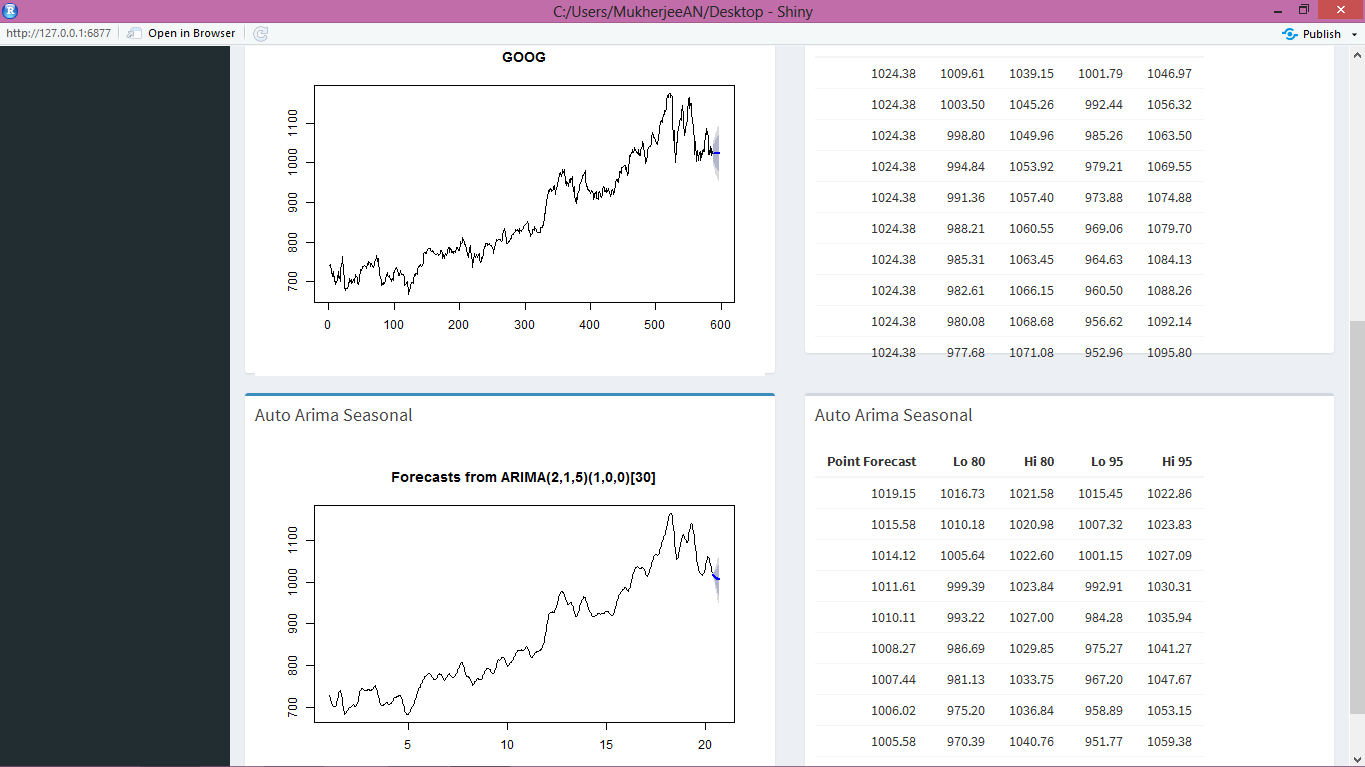
The google stock data was used for forecasting function.

The two different types of ARIMA forecast that is seasonal and non-seasonal are depicted in the screenshots below

1. Layout of the shiny web app and input is of google stock and ticker symbol is GOOG (NASDAQ)



1. The Seasonal and the non-seasonal types of forecast along with the parameters and the OHLCV data (open, high, low, close, volume) is projected by its side.



Project link : <https://github.com/Anirban87/Rprograms/blob/master/SHiny_ARIMA_forecast.R>

5. Limitations

The limitation is based on the fact that this is a very naïve method to predict as there are a lot of independent variables upon which the data should be predicted as the more data we have the greater accuracy we achieve in forecasting as decision variables are linearly dependent on the independent variable or the forecasting variable.

Appling liner regression model which is widely used by the insurance models to predict the premiums that is again a part of forecasting. That is a kind of future scope for this project.

6. Future prospects

The future of the project finds its place in the fact that it can be used as a part of entire portfolio management in quantitative analysis. The goal of the portfolio is to make supervised decision that the book should contain diverse stocks and assets so that in case of a loss of a particular asset that other can recover it.

The ARIMA based prediction can be used as a plugin as of an entire quantitative asset management and could prove to be a useful tool in real time stock prediction.

7. Conclusion

To conclude, in this post we covered the ARIMA model and applied it for forecasting stock price returns using R programming language. We also crossed checked our forecasted results with the actual returns. In our upcoming posts, we will cover other time series forecasting techniques and try them in Python/R programming languages.

8. References

<https://www.analyticsvidhya.com/blog/2017/09/comparative-stock-analysis/>

<https://www.r-bloggers.com/forecasting-stock-returns-using-arima-model/>

[www.wikipedia.org](http://www.wikipedia.org)

<https://www.otexts.org/fpp/8/1>

<https://onlinecourses.science.psu.edu/stat462/node/78>