DMA LAB PROJECT REPORT

On

Stock Forecasting

# A Project Report submitted in partial fulfilment of

the requirement for the award of

## Bachelor of Engineering

**IN**

**OF COMPUTER SCIENCEANDENGINEERING**

**Submitted By:**

**Mohsin Majeed**

**(Roll no. CO17363)**

**Jaswinder kaur**

**(Roll no. CO17332)**

**Parina**

**(Roll no. CO17343)**

****

## Chandigarh college of Engineering And Technology

### (DEGREE WING)

Goverment Institute under Chandigarh (UT)Administration,Affilated to Punjab University

Chandigarh

SECTOR -26,Chandigarh,PIN-160019

Contents

Student Declaration………….………………………………………………..……….…i

Certificate by the Guide……………………………………………………………........ii

Acknowledgement…………………………………………………………………….......iii

Abstract…………………..…………………………………………………………….....iv

Table of Content…………………..……………………………………………………....v

Chapter 1 Data Science…………………………………………………………...……

1.1 What is Data Science ………………………………………..…………..

1.2 The Data science process ………………………………………………

1.3 Data Science Foundations ………………………………….…….…….

Chapter 2.R ………………………………………………………………………

2.1 R ……………………………………………………………………….

2.2 What is R used for ………………………………………………………

2.3 Why Use R ……………………………………………………………….

2.4 Should You Choose R …………………………………………………..

2.5 Shiny Dashboard … ……………………………………………………..

Chapter3. Introduction to Project ……………………………………………………..

3.1 Problem Statement …………………………………………………….

3.2 Objective ……………………………………………………….………

3.3 Source Code …………………………………………………………….

Chapter4 Output …………………………………………………………………………

**STUDENT DECLARTION**

This is declare that the work presented in this report entitled

“**STOCK FORECASTING**” ,in the fulfilment of the requirement for the award of the Bachelor of Engineering in Computer Sciences & Engineering, submitted in CSE Department , Chandigarh Collage of Engineering & Technology (Degree Wing) affiliated to Punjab University Chandigarh is an authentic record of my/our work carried out during my degree under the guidance of Dr.Varun Gupta .The work reported in this has not submitted by me for award of any other degree or diploma.

Mohsin Majeed (CO17363)

Jaswinder Kaur(CO17332)

Parina(CO17343)

Place: CCET

**ACKNOWLEDGEMENT**

I would like to express My whole hearted gratitude towards my esteemed guide **Dr. Varun Gupta(Assistant Professor, C.S.E, CCET)** for his able guidance, monitoring and encouraging attitude and searching supervision throughout my study which enable me to successfully complete my project. Without his help and valuable suggestion, this report would not have been the light of the day. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I would also like to express a deep sense of gratitude to **Dr. Sunil. K Singh (Head Of Department C.S.E, CCET)** for their cordial support, valuable information and guidance which helped me in completing this task through various stages.

Lastly, I would like to thank almighty and my parents for their moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.

MOHSIN MAJEED

(CO17363)

**ABSTRACT**

Sales forecasting is the process of estimating future sales and revenue in order to enable companies to make informed business decisions and predict short-term and long-term performance. Companies can base their forecasts on past sales data, industry-wide comparisons, and economic trends. The problem of sales forecasting can be classified as a time-series forecasting, because the time is the domain in which the data (sales or revenue) got changed.

Sales forecasting is the process of estimating future sales. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance. Companies can base their forecasts on past sales data, industry-wide comparisons, and economic trends.

It is easier for established companies to predict future sales based on years of past business data. Newly founded companies have to base their forecasts on less-verified information, such as market research and [competitive intelligence](https://trackmaven.com/marketing-dictionary/competitive-intelligence/) to forecast their future business.

Sales forecasting gives insight into how a company should manage its workforce, cash flow, and resources. In addition to helping a company allocate its internal resources effectively, predictive sales data is important for businesses when looking to acquire investment capital.

Sales forecasting allows companies to:

* Predict achievable sales revenue;
* Efficiently allocate resources

# TABLE OF CONTENTS

* Data Science
* R langauge
* Introduction to Project
* Project Code
* Project Output

**Chapter : 1**

**Data Science**

# 1.1What is Data Science?

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining. ~Wikipedia

Data Science is a field where we apply ‘science’ to available ‘data’ in order to get the ‘patterns’ or ‘insights’ which can help a business to optimize operations or improvise decisions.

# Why Data Science is important?

Every business has data but its business value depends on how much they know about the data they have.

Data Science has gained importance in recent times because it can help businesses to increase business value of its available data which in turn can help them to take competitive advantage against their competitors.

It can help us to know our customers better, it can help us to optimize our processes, it can help us to take better decisions. Because of data science, data has become strategic asset.

# How to do Data Science?

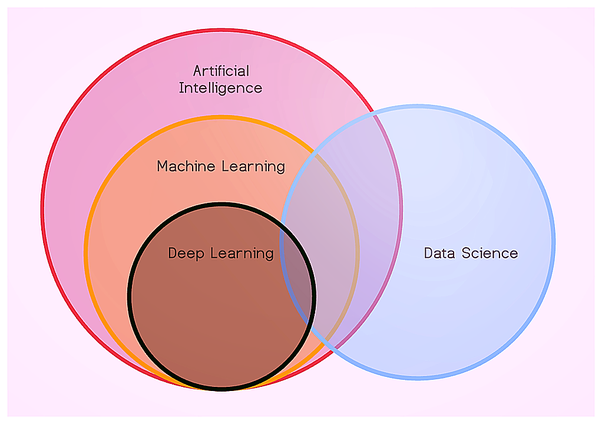
A typical data science process looks like this, which can be modified for specific use case:

* Understand the business
* Collect & explore the data
* Prepare & process the data
* Build & validate the models
* Deploy & monitor the performance

Data Science is not a certain or a single one realm, it’s like a combination of various disciplines that are focusing on analyzing data and finding the best solutions based on them. Initially, those tasks were held by math or statistics specialists, but then data-experts began to use machine learning and artificial intelligence, which added optimization and computer science as a method for analyzing data. This new approach turned out to be much faster and effective, and so extremely popular.

So all-in-all, the popularity of Data Science lies in the fact it encompasses the collection of large arrays of structured and unstructured data and their conversion into human-readable format, including visualization, work with statistics and analytical methods — machine and deep learning, probability analysis and predictive models, neural networks and their application for solving actual problems.

Artificial Intelligence, Machine Learning, Deep Learning, and Data Science — undoubtedly, these major terms are the most popular today. And although they are somehow related, they are not the same. So, before jumping into any of those realms, it is mandatory to feel the difference.



* Artificial Intelligence is the realm focusing on the creation of intelligent machines that work and react like humans. AI as a study dates back to 1936 when Alan Turing build first AI-powered machines. Despite quite a long history, today AI in most areas is not yet able to completely replace a human. And the competition of AI with humans in chess, and data encryption are two sides of the same coin.
* Machine learning is a creating tool for extracting knowledge from data. In ML models can be trained on data independently or in stages: training with a teacher, that is, having human-prepared data or training without a teacher, working with spontaneous, noisy data.
* Deep learning is the creation of multi-layer neural networks in areas where more advanced or fast analysis is needed and traditional machine learning cannot cope. “Depth” provides more than one hidden layer of neurons in the network that conducts mathematical calculations.
* Big Data — work with huge amounts of often unstructured data. The specifics of the sphere are tools and systems capable of withstanding high loads.
* Data Science is the addition of meaning to arrays of data, visualization, collection of insights, and making decisions based on these data. The field specialists use some methods of machine learning and Big Data — cloud computing, tools for creating a virtual development environment and much more. Data Science’s tasks summed up well by this [Venn diagram](http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram) created by Drew Conway:

**1.2 The Data science process:**

When a non-technical supervisor asks you to solve a data problem, the description of your task can be quite ambiguous at first. It is up to you, as the data scientist, to translate the task into a concrete problem, figure out how to solve it and present the solution back to all of your stakeholders. We call the steps involved in this workflow the “Data Science Process.” This process involves several important steps:

* Frame the problem: Who is your client? What exactly is the client asking you to solve? How can you translate their ambiguous request into a concrete, well-defined problem?
* Collect the raw data needed to solve the problem: Is this data already available? If so, what parts of the data are useful? If not, what more data do you need? What kind of resources (time, money, infrastructure) would it take to collect this data in a usable form?
* Process the data (data wrangling): Real, raw data is rarely usable out of the box. There are errors in data collection, corrupt records, missing values and many other challenges you will have to manage. You will first need to clean the data to convert it to a form that you can further analyze.
* Explore the data: Once you have cleaned the data, you have to understand the information contained within at a high level. What kinds of obvious trends or correlations do you see in the data? What are the high-level characteristics and are any of them more significant than others?
* Perform in-depth analysis (machine learning, statistical models, algorithms): This step is usually the meat of your project,where you apply all the cutting-edge machinery of data analysis to unearth high-value insights and predictions.
* Communicate results of the analysis: All the analysis and technical results that you come up with are of little value unless you can explain to your stakeholders what they mean, in a way that’s comprehensible and compelling. Data storytelling is a critical and underrated skill that you will build and use here.

## 1.3 Data Science Foundations

* **Programming :** Your first task will be to choose either you’ll use [Python](https://www.python.org/) or [R](https://www.r-project.org/) (I’ll leave you some help [here](https://www.dataquest.io/blog/python-vs-r/), [here](https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis) and [here](https://thenextweb.com/dd/2016/04/08/start-using-python-andor-r-data-science-one-best/)) and then immerse yourself into coding.
* **Linear Algebra :** As you’ll be working with data you’ll want to know how to represent data sets as matrices, and understand concepts like vectorization and orthogonality.
* **Calculus :** Many of the models you’ll write and use will use tools like derivatives, integrals and optimization to compute and find a solution to your problem more rapidly.
* **Probability :** While you use data science, many times you’ll be working to predict something in the future so you’ll want to know how likely something is to happen or why two events are related.
* **Statistics :** In order to describe the information you’ll be analyzing, things like the mean or percentiles will come in handy, also tests to check your hypothesis will appear along the way.
* **Machine Learning :**Maybe the core of data science, at some point during your project you’ll want to predict something and that’s when machine learning kicks in.

**Chapter -: 2**

**R LANGUAGE**

**2.1R**

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation’s GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

## The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

* an effective data handling and storage facility,
* a suite of operators for calculations on arrays, in particular matrices,
* a large, coherent, integrated collection of intermediate tools for data analysis,
* graphical facilities for data analysis and display either on-screen or on hardcopy, and
* a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

## 2.2 What is R used for?

* Statistical inference
* Data analysis
* Machine learning algorithm

## R by Industry

If we break down the use of R by industry, we see that academics come first. R is a language to do statistic. R is the first choice in the healthcare industry, followed by government and consulting.

## R package

The primary uses of R is and will always be, statistic, visualization, and machine learning. The picture below shows which R package got the most questions in Stack Overflow. In the top 10, most of them are related to the workflow of a data scientist: data preparation and communicate the results.

All the libraries of R, almost 12k, are stored in CRAN. CRAN is a free and open source. You can download and use the numerous libraries to perform Machine Learning or time series analysis.

## Communicate with R

R has multiple ways to present and share work, either through a markdown document or a shiny app. Everything can be hosted in Rpub, GitHub or the business's website.

Below is an example of a presentation hosted on Rpub

Rstudio accepts markdown to write a document. You can export the documents in different formats:

* Document :
  + HTML
  + PDF/Latex
  + Word
* Presentation
  + HTML
  + PDF beamer

## 2.3 Why use R?

Data science is shaping the way companies run their businesses. Without a doubt, staying away from Artificial Intelligence and Machine will lead the company to fail. The big question is which tool/language should you use?

They are plenty of tools available in the market to perform data analysis. Learning a new language requires some time investment. The picture below depicts the learning curve compared to the business capability a language offers. The negative relationship implies that there is no free lunch. If you want to give the best insight from the data, then you need to spend some time learning the appropriate tool, which is R.

On the top left of the graph, you can see Excel and PowerBI. These two tools are simple to learn but don't offer outstanding business capability, especially in term of modeling. In the middle, you can see Python and SAS. SAS is a dedicated tool to run a statistical analysis for business, but it is not free. SAS is a click and run software. Python, however, is a language with a monotonous learning curve. Python is a fantastic tool to deploy Machine Learning and AI but lacks communication features. With an identical learning curve, R is a good trade-off between implementation and data analysis.

When it comes to data visualization (DataViz), you'd probably heard about Tableau. Tableau is, without a doubt, a great tool to discover patterns through graphs and charts. Besides, learning Tableau is not time-consuming. One big problem with data visualization is you might end up never finding a pattern or just create plenty of useless charts. Tableau is a good tool for quick visualization of the data or Business Intelligence. When it comes to statistics and decision-making tool, R is more appropriate.

Stack Overflow is a big community for programming languages. If you have a coding issue or need to understand a model, Stack Overflow is here to help. Over the year, the percentage of question-views has increased sharply for R compared to the other languages. This trend is of course highly correlated with the booming age of data science but, it reflects the demand of R language for data science.

In data science, there are two tools competing with each other. R and Python are probably the programming language that defines data science.

## 2.4 Should you choose R?

Data scientist can use two excellent tools: R and Python. You may not have time to learn them both, especially if you get started to learn data science. Learning statistical modeling and algorithm is far more important than to learn a programming language. A programming language is a tool to compute and communicate your discovery. The most important task in data science is the way you deal with the data: import, clean, prep, feature engineering, feature selection. This should be your primary focus. If you are trying to learn R and Python at the same time without a solid background in statistics, its plain stupid. Data scientist are not programmers. Their job is to understand the data, manipulate it and expose the best approach. If you are thinking about which language to learn, let's see which language is the most appropriate for you.

The principal audience for data science is business professional. In the business, one big implication is communication. There are many ways to communicate: report, web app, dashboard. You need a tool that does all this together.

**2.5 SHINY DASHBOARD**

Data visualization plays a vital role in life of a Data Scientist. It is easier to visualize complex data and relationships than deciphering them from spreadsheets / tables.

There are several tools for visualizing data such as Tableau, Qlik, Dygraphs, Kibana etc. If I talk specifically about R, it provides three plotting systems:

* The Base Plotting System.
* The Lattice System.
* The ggplot2 System.

But, writing codes for plotting graphs in R time & again can get very tiring. Also, it is very difficult to create an interactive visualization for story narration using above packages. These problems can be resolved by dynamically creating interactive plots in R using Shiny with minimal effort.

If you use R, chances are that you might have come across Shiny. It is an open package from RStudio, used to build interactive web pages with R. It provides a very powerful way to share your analysis in an interactive manner with the community. The best part about shiny is that you don’t need any knowledge of HTML, CSS or JavaScript to get started.

Today, I will walk you through all the steps involved in creating a shiny app as well as deploying it online to make it accessible to everyone. This article will provide you a good understanding n how shiny apps work and how they can be useful. To provide you a hands on experience on creating Shiny Apps on your own I will be using the Loan Prediction III Practice Problem. And am sure by the end of this article you will be able to create Shiny apps yourself.

## 1. Shiny: Overview

Shiny is an open package from RStudio, which provides a web application framework to create interactive web applications (visualization) called “Shiny apps”. The ease of working with Shiny has what popularized it among R users. These web applications seamlessly display R objects (like plots, tables etc.) and can also be made live to allow access to anyone.

Shiny provides automatic reactive binding between inputs and outputs which we will be discussing in the later parts of this article. It also provides extensive pre-built widgets which make it possible to build elegant and powerful applications with minimal effort.

Any shiny app is built using two components:

**1.UI.R:** This file creates the user interface in a shiny application. It provides interactivity to the shiny app by taking the input from the user and dynamically displaying the generated output on the screen.

**2. Server.R:** This file contains the series of steps to convert the input given by user into the desired output to be displayed.

## 3. Setting up shiny

Before we proceed further you need to set up Shiny in your system. Follow these steps to get started.

1. Create a new project in R Studio

2. Select type as Shiny web application.

3. It creates two scripts in R Studio named ui.R and server.R.

4. Each file needs to be coded separately and the flow of input and output between two is possible.

5. Writing “ui.R”

If you are creating a shiny application, the best way to ensure that the application interface runs smoothly on different devices with different screen resolutions is to create it using fluid page. This ensures that the page is laid out dynamically based on the resolution of each device.

The user interface can be broadly divided into three categories:

* Title Panel: The content in the title panel is displayed as metadata, as in top left corner of above image which generally provides name of the application and some other relevant information.
* Sidebar Layout: Sidebar layout takes input from the user in various forms like text input, checkbox input, radio button input, drop down input, etc. It is represented in dark background in left section of the above image.
* Main Panel: It is part of screen where the output(s) generated as a result of performing a set of operations on input(s) at the server.R is / are displayed.

**4. Writing SERVER.R**

This acts as the brain of web application. The server.R is written in the form of a function which maps input(s) to the output(s) by some set of logical operations. The inputs taken in ui.R file are accessed using $ operator (input$InputName). The outputs are also referred using the $ operator (output$OutputName). We will be discussing a few examples of server.R in the coming sections of the article for better understanding.

5. Deploying the Shiny app on the Web

The shiny apps which you have created can be accessed and used by anyone only if, it is deployed on the web. You can host your shiny application on “Shinyapps.io”. It provides free of cost platform as a service [PaaS] for deployment of shiny apps, with some restrictions though like only 25 hours of usage in a month, limited memory space, etc.  You can also use your own server for deploying shiny apps.

## Chapter -: 3

## INTRODUCTION TO PROJECT

Sales Forecasting is the process of using a company’s sales records over the past years to predict the short-term or long-term sales performance of that company in the future. This is one of the pillars of proper financial planning. As with any prediction-related process, risk and uncertainty are unavoidable in Sales Forecasting too. Hence, it’s considered a good practice for Sales forecasting teams to mention the degree of uncertainties in their forecast. Sales Forecasting is a globally-conducted corporate practice where a number of objectives are identified, action-plans are chalked out as well as budgets and resources are allotted to them. The first step to proper Sales Forecasting is to know the things that fall within your domain directly as a salesperson. This usually relates to your sales staff, clients and prospects. Other factors to consider during the setup of a forecast are the negative ones like – uncertainty, abrupt changes in consumer shopping patterns, etc. One of the most common yet basic challenges that the management of companies face in making business sales forecasts is that their usual approach is a “top to down” one. This approach leaves very little scope for interaction with the sales manager and the salespersons during the data collection process.

For a successful and accurate Sales Forecasting, it’s necessary to take into consideration the direction from significant departments of the organization, comprising of seniors, managers, sales teams and finally – your own gut feeling. Let’s list down these sources of instructions and how they contribute towards designing a reliable sales forecast. Directions from Top-level Seniors – It may be initially necessary for you to increase your sales by 10%, however your seniors, being wiser, may ask you to reconsider your target depending on promises made to outside investors as well as stockholders. Directions from one’s own manager – These kind of directions are mostly integrated along with the direction from the top level, but their expectations are generally little more conservative and realistic. If the top management gives you a target of 15% sales growth, your manager will tell you what the real expectations are. Direction from Sales Teams – For instance, if the Sales Teams may project a growth of 10% over the management's forecast figure of 20%; this extra-conservative number is a cushion, so that they could increase their chances to beat the sales forecast. Direction from other Entities – Many other entities also take part in Forecasting. Chief among them are the Research and Development department, Human Resource department, Marketing department finance team, manufacturing unit, etc. Once you are done taking feedback and inputs from all thesepeople, the final question to ask is – what is your interpretation of all these factors? Most often a person’s gut feeling is more accurate than all the numbers put in front of him. Although it’s not advisable to go against the company’s decision, it is always a good policy to do further research till the negative hunch.

**Role of External Factors**

While participating in a sales forecast, it is crucial for you to answer after considering both the corporate and the departmental viewpoints that may arise. This will provide the real balance between the expectations of the management and the real-case scenarios that different departments project. External factors have a very significant role to play in Sales Forecasting. This is mainly because they are not dependent on the organizations’ functioning; the organization is dependent on theirs. Organizations study external factors with great detail because they cannot control or influence them. Just as a forecasting can only inform you about the weather but cannot change it. The most influential factor is the competition, where the competition stands in terms of market share, new line of products, recognition of brand, expansion or contraction of the sales force, etc. Also, whether there is a new competitor in the market or if any competitor is losing out in business

**3.1** **Problem Statement**

Today, there are numerous tools to predict stock price used by investment managers, corporate or individual investors. Some of the tools that utilize sophisticated methodology and statistical model have implemented to the system or application. However, those kinds of applications are usually costly to be acquired by individual investor or beginner player in stock market. Thus, an application that utilizes an open source tools and data from fast and rich of content from internet can be suggested for the solution of the problem. The application focused on user friendliness features in context to deliver the best forecasting method and result for the users. Inexperienced stock traders or called as early investor is usually entered the market with minimum of principle. Commonly, early investor are using salary remainder, grants, and family capital trust to start their first capital to enter the market. This resulted a high tendency of soft approach and be very careful put their money into respected stock according into investor personal risk profile. Thus, many early investor directly trust their capital to hedge funds and asset management company to manage their money and expect return in the future. However, not all early investors opted out from directly trading in the market. Some of them are doing the trading activities by their own to evade several fees and commission to asset management companies that is renowned to be highpriced. To expose this early investor with sufficient knowledge about stock trading mechanism and information, a content management tool is importantly required. Yet, most tools relevant to stock market information or stock price forecasting come with expensive subscription fees, which excluded from the trading software fees. This situation could hardly discouraging potential traders to start trading and investing in the market. As many factors influence the fluctuate moving stock price of a company or sector of a company, it is also affected by the incoming news articles and updates (Godbole&Srinivasaiah, 2007; Herath, 2007). This news varies from latest earning from the company, announcement related to management, dividends announcement by the company, analysis by editorial news and professional traders, and domestic and international news that related to the company. This 4 news has made it an important factor to be use for each trader to analyze stock of a company before make any decision. Unfortunately, some of the current forecasting tools is not providing a relevant and most recent information about the company to use by the tools subscribers, which hinders the investors to make fundamental analysis. Forecast Factor or Stock Price Forecasting Dashboard with Data Mining is attempt the entire problem explained before. The dashboard is aim to provide a user-friendly application that can be used by all types of investor with different objectives. An advance and proven statistical model is utilized to plot the historical and prediction of the price to ensure an accurate and reliable result to te users. The rest of this chapter will be organized as follows: the next section will put forward the problem statement of this project. This will be followed with the objectives section .

**3.2 OBJECTIVE OF PROJECT**

The main objectives of this making this project are:

To create app that is able to forecast stock price movement fluctuation in the future.

In essence, this project is to help investor to make decision to buy or sell the stock rather than need them to read long writing report that is complicated. This project deliverables would be:

∙Analyze time series model approach for prediction .

∙ A system that would able to generate forecast plot and newsfeed for decision-making purposes.

There are numerous instances where two financially unsound companies enter into Mergers and Acquisitions. Often, these companies form a strong partnership and emerge as a challenging competitor. Managers need to check whether any of their competitors are involved in any such mergers or acquisitions, and if they are, then what is their collective strength and which minuses of each other they are cancelling out. Some people might say that being a salesperson, you should abide by a philosophy similar to all other staff members, i.e., “winning over the numbers is the game”. In fact, the reality is that winning numbers only proves to the clients that you can perform. Getting numbers is fine, however the individual contributions of team-mates is a significant factor in correlation to the culture of the corporate world. You need to consider many things, such as the economic status of the environment you are operating in, whether the spot of business is going through growth, recession, etc. You would also need to check if there are any government-implemented hikes in the interest rates, pricing of commodities and what is the current rate of unemployment. Foreign and domestic regulatory bodies implement policies from time to time, which also dramatically influences your business

**3.3 Source Code :**

library(shiny)  
library(shinydashboard)  
library(ggplot2)  
library(tseries)  
library(forecast)  
library(magrittr)  
library(timeDate)  
library(dplyr)  
library(zoo)  
header <- dashboardHeader(title = "Basic Dashboard")  
sidebar <- dashboardSidebar(  
    sidebarMenu(  
        menuItem("Dashboard", tabName = "dashboard", icon = icon("dashboard")),  
        selectInput("ohlcv", "Select a column:",  
                    choices = c('select','open','high','low','close','volume'),  
                    selected = 'select'),  
          
        selectInput("graphs", "Select a graph:",  
                    choices = c('Select','TimeSeries','DecomposedSeries','DifferencedSeries','HoltWinter'),  
                    selected = 'Select'),  
        selectInput("acfpcfval", "Select VALUES:",  
                    choices = c('Select','ACF','PACF'),  
                    selected = 'Select'),  
        numericInput("pval", "P", value=""),  
        numericInput("dval", "D", value=""),  
        numericInput("qval", "Q", value=""),  
        selectInput("arima", "Select a model:",  
                    choices = c('Select','ARIMA','Auto Arima'),  
                    selected = 'Select'),  
          
        numericInput("ndays", "Number of days to forecast:", value=0),  
          
        selectInput("forecast", "Select to Forecast:",  
                    choices = c('Select','Forecast-Plot','Forecast-Values'),   
                    selected = 'Select'),  
          
        selectInput('holtwinterforecast','select year:',  
                    choices = c('select','1','2','3','4','5'),  
                    selected='select')  
          
        ))  
  
frow1 <- fluidRow(  
    valueBoxOutput("value1"),  
    valueBoxOutput("value2"),  
    valueBoxOutput("value3")  
)  
frow2 <- fluidRow(   
    box(  
        title = "Diffrents Graphs"  
        ,status = "primary"  
        ,solidHeader = TRUE   
        ,collapsible = TRUE,   
        tabBox(id="firsttab",title = "GRAPHS",width = 100,  
               div(style = 'height:50vh ;overflow-x:auto;overflow-y:auto',  
                   tabPanel(plotOutput("graphs"),title="Graphs"))  
        )),  
      
    box(  
      title = "ACF/PACF Graphs"  
      ,status = "primary"  
      ,solidHeader = TRUE   
      ,collapsible = TRUE,  
      div(  
        tabBox(  
          id = "acfval",  
          side = "right",  
          title = "Graphs",  
          width = 100,  
          tabPanel(id="acff",plotOutput("calval"),title="ACF Graph"),  
          tabPanel(id="pacff",plotOutput("calvall"),title="PACF Graph")  
        ))),  
    box(  
        title = "Arima model Values"  
        ,status = "primary"  
        ,solidHeader = TRUE   
        ,collapsible = TRUE,   
        verbatimTextOutput('arimaPlot'),  
        verbatimTextOutput('arimaPlott')  
    ),  
    box(  
      title = "Forecasting Graphs Graphs"  
      ,status = "primary"  
      ,solidHeader = TRUE   
      ,collapsible = TRUE,  
      div(  
        tabBox(  
          id = "acfval",  
          side = "right",  
          title = "Graphs",  
          width = 100,  
          tabPanel(id="acff",plotOutput("forecastPlot"),title="Arima Graph"),  
          tabPanel(id="pacff",plotOutput("fautoarimaaa"),title="Auto Arima Graph")  
        )))  
)  
recommendation <- read.csv('/Users/jaswinder/Desktop/ashokstock.csv',stringsAsFactors = F,header=T)  
body <- dashboardBody(frow1,frow2)  
ui <- dashboardPage(title = 'This is my Page title', header, sidebar,body,skin="blue")   
  
server <- function(input, output) {  
    
  maxopen=max(openn)  
  minopen=min(openn)  
  maxclose=max(closee)  
  minclose=min(closee)  
  output$value1 <- renderValueBox({  
    valueBox(  
      formatC(maxopen, format="d", big.mark=',',width = 1)  
      ,paste("Max ↑"," Min Open :",minopen)  
      ,icon = icon("stats",lib='glyphicon')  
      ,color = "purple")    
  })  
  output$value2 <- renderValueBox({  
    valueBox(  
      formatC(maxclose, format="d", big.mark=',',width = 1)  
      ,paste("Max ↑"," Min Close :",minclose)  
      ,icon = icon("gbp",lib='glyphicon')  
      ,color = "green")    
  })  
  output$value3 <- renderValueBox({  
    valueBox(  
      formatC(paste(round(recommendation$high[263]),"-",round(recommendation$low[263])), format="d", big.mark=',',width = 1)  
      ,paste('Last week High and Low Value')  
      ,icon = icon("menu-hamburger",lib='glyphicon')  
      ,color = "yellow")     
  })  
    output$graphs <- renderPlot ({   
          
        if(input$graphs == 'TimeSeries'){  
          openn=c(recommendation$open)  
          tym=ts(openn,start = c(2014,7), end =c(2019,7) , frequency = 7)  
          plot.ts(tym, main = "Time-Series plot", col = "blue")  
          }  
         else if(input$graphs == 'DecomposedSeries'){  
            stldecomp = stl(tym, s.window="periodic")  
            plot(stldecomp)  
        } else if(input$graphs == 'DifferencedSeries'){  
            #plot.ts(differncedSeries,col = "blue")  
        }else if(input$graphs=='HoltWinter')  
        {  
          if(input$holtwinterforecast=='1')  
          {  
            a\_ts\_hw1=HoltWinters(tym,beta = F,gamma= F)  
            a\_ts\_hw\_fcst1= forecast:::forecast.HoltWinters(a\_ts\_hw1,h=1)  
            plot(a\_ts\_hw\_fcst1)  
          }else if(input$holtwinterforecast=='2')  
          {  
            a\_ts\_hw2=HoltWinters(tym,beta = F,gamma= F)  
            a\_ts\_hw\_fcst2= forecast:::forecast.HoltWinters(a\_ts\_hw2,h=2)  
            plot(a\_ts\_hw\_fcst2)   
          }else if(input$holtwinterforecast=='3')  
          {  
            a\_ts\_hw3=HoltWinters(tym,beta = F,gamma= F)  
            a\_ts\_hw\_fcst3= forecast:::forecast.HoltWinters(a\_ts\_hw3,h=3)  
            plot(a\_ts\_hw\_fcst3)  
          }else if(input$holtwinterforecast=='4')  
          {  
            a\_ts\_hw4=HoltWinters(tym,beta = F,gamma= F)  
            a\_ts\_hw\_fcst4= forecast:::forecast.HoltWinters(a\_ts\_hw4,h=4)  
            plot(a\_ts\_hw\_fcst4)  
          }else if(input$holtwinterforecast=='5')  
          {  
            a\_ts\_hw5=HoltWinters(tym,beta = F,gamma= F)  
            a\_ts\_hw\_fcst5= forecast:::forecast.HoltWinters(a\_ts\_hw5,h=5)  
            plot(a\_ts\_hw\_fcst5)  
          }}  
    
    })  
      
    output$calval <- renderPlot({  
      if(input$acfpcfval=='ACF'){  
        acff=acf(tym,lag.max = 20)  
      }  
        
        
    })  
    output$calvall <- renderPlot({   
      if(input$acfpcfval=='PACF'){  
        pacff=pacf(tym,lag.max = 20)  
      }  
        
    })  
    output$value <- renderText({ input$pval })  
    output$value <- renderText({ input$qval })  
    output$value <- renderText({ input$dval })  
      
    output$arimaPlot <- renderText({   
      if(input$arima == 'ARIMA'){  
        amv <<- arimafunc()  
        paste("AIC:",amv$aic," AICc:",amv$aicc," BIC:",amv$bic)  
      }  
      else if(input$arima == 'Auto Arima'){  
        am <<- autoarimaa()  
        paste("AIC:",am$aic," AICc:",am$aicc," BIC:",am$bic)  
      }  
    })  
    output$arimaPlott <- renderText({   
       if(input$arima == 'Auto Arima'){  
        am <<- autoarimaa()  
        paste("AIC:",am$aic," AICc:",am$aicc," BIC:",am$bic)  
      }  
    })  
    arimafunc <- function(){  
      M\_arima <- Arima(tym, order=c(input$pval,input$dval,input$qval))  
      return(M\_arima)  
    }  
    output$forecastPlot <- renderPlot({  
        if(input$forecast == 'Forecast-Plot'){  
            if(input$ndays == 0){  
                error()  
            } else {  
                fc <<- forecastfunc()  
                plot(fc, col = "darkgreen")  
            }  
        }  
    })  
    output$fautoarimaaa <- renderPlot({  
      if(input$forecast == 'Forecast-Plot'){  
        if(input$ndays == 0){  
          error()  
        } else {  
          af <<- forecastfun()  
          plot(af,col = "darkgreen")  
        }  
      }  
    })  
      
    forecastfunc <- function(){  
      Mforecasts <- forecast(amv, h = input$ndays)  
      return(Mforecasts)  
    }  
    forecastfun <- function(){  
      forcast <- forecast(am, h = input$ndays)  
      return(forcast)  
    }  
      
    autoarimaa <- function(){  
      aarima <- auto.arima(tym)  
      return(aarima)  
        
    }  
      
    getColOpt <- function(cv){  
        if(cv == 'select'){  
            print("Choose a column")  
        } else if(cv == 'open'){  
            return("open")  
        } else if(cv == 'high'){  
            return("high")  
        } else if(cv == 'low'){  
            return("low")  
        } else if(cv == 'close'){  
            return("close")  
        } else if(cv == 'volume'){  
            return("volume")  
        }  
    }  
    # output$tp <- function(){  
    #   datee=c(recommendation$date)  
    #   op=c(recommendation$open)  
    #   val=lm(op ~ datee)  
    #   a<-data.frame(date="2024-07-19")  
    #   res=as.integer(predict.lm(val,a,data=recommendation))  
    #   return(res)  
    #     
    # }  
      
}  
  
  
  
shinyApp(ui = ui, server = server)

##### **Chapter -: 4**

##### **OUTPUT**

