

Project Report On
VISUALISING AND FORECASTING STOCK USING DASH BOARD

Submitted by
Y.ASHOK

Under The guidance of
R.VENKATESHWAR RAO



Department of Mathematics
Dr. V. S. Krishna Govt. Degree & P.G. College(A),
Visakhapatnam-530013, India

CERTIFICATE

This is to certify that the project report entitled “**VISUALISING AND FORECASTING STOCK USING DASH BOARD**”

Submitted By

Y.ASHOK

record of work carried out by him in partial fulfilment of the requirement for the award of the Degree of Bachelor of Science (Computer Science), as prescribed by the Dr. V.S. Krishna Govt. Degree & P. G. College(A) in the Academic Year 2019 –2020.

Guide

R.VENKATESHWAR RAO

Head of Department

ACKNOWLEDGEMENT

We take this opportunity to thank those who helped us immensely throughout this project and whose efforts we will always appreciate and remember.

We thank to MR. R, VENKATESHWAR RAO for their valuable suggestions, their indefatigable efforts to review our work, inspiring us and helping us in all possible ways at any time.

We would also like to extend our sincere appreciation to all staff members of Computer Department, without their help and co-operation the Project report would not have been a success.

Last but not the least we express our extreme gratitude to the Almighty, without whose blessing nothing is possible.

ABSTRACT:

In modern financial market, the most crucial problem is to find essential approach to outline and visualising the predictions in stock markets to be made by individuals attain maximum profit by investment. The stock market is a transformative, non-straight dynamical and complex system. Long term investment is one of the major investment decisions.

Though, evaluating shares and calculating elementary values for companies for long term investment is difficult. In this paper we are going to present comparison of machine learning aided algorithms to evaluate the stock prices in the future to analyse market behaviour. Our method is able to correctly analyse supervised algorithms and compare which algorithm performs the best to predict the future stock market prices in the market.

OBJECTIVE:

You will be creating a single page web application using Dash (a python framework) and some machine learning models which will show company information (logo, registered name and description) and stock plots based on the stock code given by the user. Also, the ML model will enable the user to get predicted stock prices for the date inputted by the user.

ACKNOWLEDGMENT:

This paper could not have been completed without the support of many people, some of whom have helped directly and Some of them indirectly such as by publishing their research paper online which helped us to recognize with this concept Easily.

We have deepest gratitude and feeling of appreciation to our Project mentor Citronade Choubey Ma'am (Assistant Professor-CSE), her presence at any time throughout the Semester, important guidance, opinion, encouragement, and support greatly enhanced this project work. We are grateful to the college for providing the necessary infrastructure and technical support for the completion of this project. At last, we would like to thank our family and friends for their Unconditional support.

Table of Contents

ABSTRACT

OBJECTIVE

ACKNOWLEDGEMENT

CHAPTER 1 INTRODUCTION

1.1. Background of Study

1.2. Problem Statement

1.3. Scope of Study

CHAPTER 2 LITERATURE REVIEW

2.1. Stock Market Analysis

2.2. Stock Market Prediction

2.3. Time Series

2.3.1. Autoregressive Integrated Moving Average (ARIMA)

2.4. Existing Financial System

CHAPTER 3 RESEARCH METHODOLOGY

3.1. Phase 1: Problem Identification & Knowledge Discovery

3.1.1. Interview

3.1.2. Initial Forecasting Design

3.1.3. Tools for Development

3.1.4. Equipment and Materials

3.2. Phase 2: Prototyping Methodology

3.2.1. Requirement Gathering and Planning

3.2.2. Prototyping Phase

3.2.3. Testing Phase

3.2.4. Deployment Phase

3.3. Phase 3: Proposed Framework

3.3.1. Flow Chart

3.3.2. System Architecture

CHAPTER 4 RESULT AND DISCUSSION

- 4.1. Introduction
- 4.2. Qualitative method: Interview
 - 4.2.1. Stock Price Fundamental
- 4.3. Experimental

CHAPTER 5

- 5.1. Proposed System
- 5.2. Methodology
- 5.3. Result
 - 5.3.1. Style Your Application
 - 5.3.2. Generating a company's Information and Graphs
 - 5.3.3. Creating the machine Learning Model

CHAPTER 6

- 6.1. What is the stock Market
- 6.2. Understanding The Stock Market
- 6.3. How The Stock Market Works
- 6.4. Functions of Stock Market
- 6.5. Efficient Price Discovery
- 6.6. Support All Eligible Types of Participants
 - 6.6.1. Investor Protection
 - 6.6.2. Balanced Regulation
 - 6.6.3. Regulating The Stock Market
 - 6.6.4. Stock Market Participants
- 6.7. How Stock Exchanges Make Money
 - 6.7.1. Competition For Stock Markets
 - 6.7.2. Significance of The Stock Market

6.7.3. Examples of Stock Market

CHAPTER 7

7.1. Existing System

7.2. Data Source for Market Prediction

7.3. Market Mimicry

7.4. Time Series aspect Structuring

CHAPTER 8

8.1. Proposed System

8.1.1. Downloading The Data

8.2. Splitting Data into a Training set and a test set

8.3. Standard Average

8.4. Exponential Moving Average

8.5. Data Generator

8.6. Data Augmentation

CHAPTER 9

9.1. Code

CHAPTER 10

10.1. Conclusion

10.2 References

CHAPTER-1
INTRODUCTION

INTRODUCTION

Investment companies and individual investors have been utilizing money models to possess a higher understanding of the market and build a profitable investment. Plenty of data regarding stock data fluctuations is at hand for analysis and processing. Investors take calculated guesses by analysing data. They browse the news, study the corporate history, trade trends and numerous variables that go into creating a prediction. The prevailing theories are that stock costs are quite random and unpredictable. This raises the question that why prime companies like Morgan Stanley and Citigroup rent quantitative analysts to create predictive models.

This paper seeks to utilize Deep Learning models, LSTM Neural Networks, to predict stock costs. As for data with time-frames RNNs are available handy however recent researches have shown that LSTM networks are the foremost sought-after and helpful variants of RNNs.

A business could become prone to market fluctuations on the far side your management together with market sentiment, economic conditions or developments in your sector. Exchanging the stocks on money markets is the one of the significant speculation exercises. Already, scientist developed different stock examination system that could empower to envision the bearings of stock esteem development. Predicting and foreseeing significant worth future cost, in perspective of the present cash related information and news, is of colossal use to the financial pros. financial masters need to know whether some stock will get higher or lower over particular time period. To obtain the accurate output, the approach used to implement in machine learning along with supervised learning algorithms. Results are tested using different types of supervised learning algorithms with a different set of features.

Stock market is characterized as active, unforeseeable and nonlinear in universe. Prediction of stock prices is a very difficult task as it depends on various things including so many things, global economy, financial reports of company and performance etc.

Traditionally, two important methods have been made for prediction of the stock price of an company.

Technical analysis method uses recorded price of stocks like closing and opening price, volume trade, adjacent closing values etc. of the stock for predicting the coming price of the stock. The second type of analysis is qualitative

analysis, which is performed on the basis of exterior things like company's bio, market circumstance, profitable and political factors, descriptive information in the form of financial new articles, social media and even blogs by economical analyst. The stock market datasets are changing every day, therefore we need to have the latest or real-time to support data analysis. Of course, in the market, there are a lot of APIs provided to extract the dataset. Some of the APIs may incur some costs due to providing the real-time data on time. In this post, I will use Python third-party module to extract the data from Yahoo-Finance. Pandas will be used for data processing, while PowerBI use for data visualization. In this article, I will guide how to use Python. In Power Bi to build Stock Market Analysis and data visualization in a faster way.

The Measures for the Suitability Management of Securities and Futures Investors (Hereinafter refer to as the Measures) of China has come into force in July 27, 2017. As the first special regulation of investor protection in China's securities and futures market, this act is an important foundation law of the capital market. The act will regulate the behaviour of the brokers and also provide evidence for the settlement of disputes in the future. The Measures requires that when a staff of a securities or insurance company sells financial products and signs cooperation documents, the company should that complies with certain rules for verification of possible.

1.1 BACKGROUND OF STUDY:

VISUALISING AND FORECASTING STOCKS USING DASH:

Developing this simple project idea using the Dash library (of Python), we can make dynamic plots of the financial data of a specific company by using the tabular data provided by finance python library. On top of it we can use a machine learning algorithm to predict the upcoming stock prices.

STOCK VISUALISATION:

Stock Chart (or Stock Graph) is a powerful tool for the data visualization and analysis of price movements over time in stock, derivative, currency, bond (credit/debt), commodity, and any other markets.

FORECASTING:

Forecasting refers to the practice of predicting what will happen in the future by taking into consideration events in the past and present. Basically, it is a decision-making tool that helps businesses cope with the impact of the future's

uncertainty by examining historical data and trends. It is a planning tool that enables businesses to chart their next moves and create budgets that will hopefully cover whatever uncertainties may occur.

PROCESS OF FORECASTING:

Forecasting is the process of making predictions based on past and present data. Later these can be compared (resolved) against what happens. For example, a company might estimate their revenue in the next year, then compare it against the actual results. Prediction is a similar, but more general term.

STOCK FORECAST:

Stock Forecast can refer to: Stock forecasts based on human experience: Human traders based on their experience in terms of stock price patterns, volume changes, and market news/rumours.

Regarding a particular stock. In finance, forecasting is used by companies to their experience in terms of stock price patterns, volume changes, and market news/rumours regarding a estimate earnings.

DASHBOARD:

A dashboard is a visual display of all of your data. While it can be used in all kinds of different ways, its primary intention is to provide information at-a-glance, such as KPIs.

A dashboard usually sits on its own page and receives information from a linked database. In many cases it's configurable, allowing you the ability to choose which data you want to see and whether you want to include charts or graphs to visualize the numbers.

IMPORTANCE OF DASHBOARDS:

Dashboards allow all kinds of professionals the ability to monitor performance, create reports and set estimates and targets for future work.

Other benefits include:

- A visual representation of performance, such as with charts and graphs.
- The ability to identify trends.
- An easy way of measuring efficiency.
- The means to generate detailed reports with a single click.
- The capacity to make more informed decisions.
- Total visibility of all systems, campaigns, and actions.

- Quick identification of data outliers and correlations.

PROJECT CONTEXT:

Investment companies and individual investors have been utilizing money models to possess a higher understanding of the market and build a profitable investment. Plenty of data regarding stock data fluctuations is at hand for analysis and processing.

Investors take calculated guesses by analysing data. They browse the news, study the corporate history, trade trends and numerous variables that go into creating a prediction.

The prevailing theories are that stock costs are quite random and unpredictable. This raises the question that why prime companies like Morgan Stanley and Citigroup rent quantitative analysts to create predictive models.

This paper seeks to utilize Deep Learning models, LSTM Neural Networks, to predict stock costs. As for data with time-frames RNNs are available handy however recent researches have shown that LSTM networks are the foremost sought after and helpful variants of RNNs.

A business could become prone to market fluctuations on the far side your management together with market sentiment, economic conditions or developments in your sector. Exchanging the stocks on money markets is the one of the significant speculation exercises. Already, scientist developed different stock examination system that could empower to envision the bearings of stock esteem development. Predicting and foreseeing significant worth future cost, in perspective of the present cash related information and news, is of colossal use to the financial pros. financial masters need to know whether some stock will get higher or lower over particular time period. To obtain the accurate output, the approach used to implement in machine learning along with supervised learning algorithms. Results are tested using different types of supervised learning algorithms with a different set of features.

Stock market is characterized as active, unforeseeable and nonlinear in universe. Prediction of stock prices is a very difficult task as it depends on various things including so many things, global economy, financial reports of company and performance etc.

1.2. 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 is 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 investors+ directly trust their capital to hedge funds and asset management company to manage their money and expect return in the future.

Forecast Factor or Stock Price Forecasting Dashboard with Data Mining is attempting the entire problem explained before. The dashboard is aim to provide a user-friendly application that can be used by all types of investors 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 the 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 and lastly scope of study.

1.3. Scope of Study:

This project aims are to predict public listed companies stock price listed in open market all over the world. Because the prediction tool is using data mining method, historical data of the price is very important. The project will use existing historical data that distributed freely in internet and provided by trustable provider to make sure the desired result is fully trusted and reliability. This project could be adapted into large capital market around the world. However, certain of limitation is necessary to make sure that the scope will not be large and become time consuming, and decided to only focus on 10 stocks of the highest market capitalization of companies publicly listed in Bursa Malaysia.

Furthermore, the implementation of several methods needs to be tested into different environment where the excellent desired output will be chosen as the object of this project. The historical data is downloaded from search engine and historical data provider 'Yahoo Financial'. Several developer and time series experts have used yahoo finance historical data in order to calculate the price prediction with different approaches. Other than that, the data would also be

tested and analyse to ensure the accuracy and reliability of data. The historical data that will be used in this project would be naturally come from ‘closed price’ without look into volume of transaction and other variable in the historical data. This is because the historical closed price is already adequate to find the pattern or plot and to predict the next pattern of future stock price.

The scope of the study would further assist in creating the accuracy and reliability that the system proposes. The historical data downloaded from notnative provider such as ‘Yahoo Financial’ will always be look into the real data provided by the regulators in particular country. The make sure accuracy of data, this project will also try to find the best model approach that can be used to provide an output that become solution for large communities.

CHAPTER-2

LITERATURE REVIEW

2.1 STOCK MARKET ANALYSIS:

There are numerous methods currently use by the investment managers and brokers to evaluate stocks. The evaluation covers many elements i.e., the performance of the stock to be market index as a whole, company news and action, return analysis made by the experts. All of the evaluation later leads towards predicting the future value of the stock in short and long period of time fundamental and technical analysis are the most widely used for stock valuation analysis method (Smart, Gutman, & Hoehn, 2014). Those analyses are looking into different attributes and class of measurement of the company

Fundamental analysis or economic analysis focuses on the financial performance and economic situation. This method is mutually exclusive monitored the financial performance of the particular company towards the stock price in the security market. One most assured that the indication of the stock value whether it is overvalued or undervalued is reflected through the fundamental value of the company.

In conclusion, fundamental analysis is important to investors to help get insight of the economic value of and the potential reflection to the share price (Smart et al., 2014).

In conclusion, fundamental analysis is the company evaluation of the financial performance and their current competition state in which influence the stock price in the market. Hence, it will automatically change the investor behaviour into the stock price.

The situation of undervalued is arbitrated when fundamental or the economic value of the company is lower to value of stock price, which eventually strive the line of graph to down to adjust the value consecutively.

As in the case of overvalued the share price value is much higher than the economic value, and led the investor to bullish the stock and lower down the price until the equilibrium achieved.

On the other hand, technical analysis focuses onto the stock behaviour in the market rather than the economic value occur in corporation.

(Yu & Cheng, 2007) explained that technical analysis method deals with determination of the stock price based on the past pattern of the stock (using time-series analysis), performed by technical analysis.

2.2. Stock Market Prediction:

As investing in stock has become one of the common tools of investors to gain massive return, stock price prediction has become a common way to attain maximum return. Technical analysis has been widely known of its superiority to other analysis to predict stock prices. Numerous attempts have influence the breadth of the study of stock price prediction, this attempt only aims to create an accurate and proven model that provide a valid result. Forecasting with technical analysis visualized in graphical pattern or table, which explained a list down the probabilities of future trend of stock market. Time series analysis is the main branch of forecasting techniques. Time series is a trend chart of historical data that visualize the trend in series of time. There is a huge variety of ways in forecasting economy indices like stock prices, most of which are time series analysis (TSA) methods based on structured data (e.g., stock prices table) (Pai & Lin, 2005).

2.3. Time Series:

Definition Time series analysis is the main branch of forecasting techniques. Time series is a trend chart of historical data that visualize the trend in series of time. (Devi, B, Sundar, & Alli, P, 2013) explained that time series is the best technique to forecast the trend of the stock price in indices, die to its volatility movement. Time series features forecasting the share price in seasonal

and long-term period. It provides a trend image of flow of the movement of share price, the growth, and the variations. The trend visualized to the predictors will eventually all-inclusive view and decision of stock in the market.

The general trend towards stock market among the society is that it is highly risky for investment or not suitable for trade (Devi, B et al., 2013). The seasonal variance and steady flow of any index will help both for existing and naïve investor to understand and make a decision to invest in the stock market. To solve these types of problems, the time series analysis will be the best tool for forecast and also to predict the trend (Brockwell & Davis, 1991). Time series technique is insufficient to be use when predictors want to define between demands of particular stock and other factors such as economic situation and changes in the price of a company to the others, but it is better when it is used to visualize volatility in time desired (Moon, Mentzer, Smith, & Garver, 1998). Nevertheless, time series analysis is still considered as the most relevant tools to forecast stock price (Brockwell & Davis, 1991).

Time series visualize single or many data series at the homogenous distance of time sequence. Time series are widely use to visualize financial history or its prediction weather, temperature, etc. Typically, time series plotted in line chart, time series given the user the output of trend based on the data given. Time series chart is as shown in Figure 1.

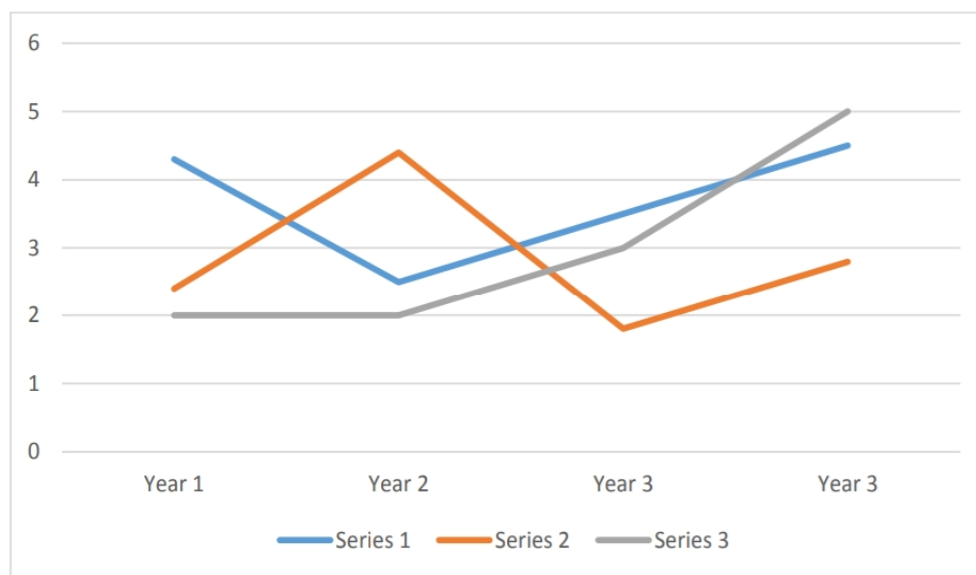


Figure 2.3 Time Series Chart

According to Shekar, Kannan, Sarthak, & Arumugam (2010) prediction of financial time series is the same difficult as stock market prediction. It is important to consider both of the prediction at the same time to prevent work repetition and work burden.

Financial time series forecasting is using the historical data for later forecast the future based on the time interval that the predictor wants. After the prediction visualized, the predictor will eventually could see the trend and relate the historical and future each other. Later they will use the trend to make financial decision-making onto buy or sell final decision.

2.3.1. Autoregressive Integrated Moving Average (ARIMA):

Mondal *et al.* (2014) argued that stock prices could be forecasted due to they are not random generated values relatively they are a time series model and can be plotted in trend perspective. Forecasting usually performed by the analyst to make financial plans. Since having an excellent knowledge about share price in the future serves the confidence of the professional traders and investors. The forecaster is forecasting to opt and conclude either enter or exit the market. This to make sure a high return without compromising cents of loss during the process.

Since the aim to provide a necessary information and data for investor to make decision, a model of forecast to show the trend is important. ARIMA is an algorithm approach to transform the series, which is better compare to forecasting directly, and gives more accurate results (Al Wadi, Ismail, Altherr, & Karim, 2010; Devi, B *et al.*, 2013).

ARIMA model is classifies as ARIMA (p, d, q) where p signifies autoregressive of the data, d signifies to the integer of the data, lastly q as the moving average of the data set.

Equation 2-1 ARIMA Model (Devi, B *et al.*, 2013)

$$Y_t = C + \theta_1 Y_{t-1} + \theta_2 Y_{t-2} + \dots + \theta_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

- C – Constant term
- AR – Non seasonal Auto Regressive coefficients ($\theta_1, \dots, \theta_p$)
- MA - Non seasonal Moving Average coefficient ($\theta_1, \dots, \theta_q$)
- AR Lags - Lags corresponding to nonzero, non-seasonal AR coefficients
- MA Lags – Lags corresponding to nonzero, nonseasonal MA coefficients
- D - Degree of no seasonal differencing, D [if D has value 0 meaning no seasonal integration]
- Variance – Scalar variance of the innovation process (σ^2)
- Distribution - Distribution of the innovation process

2.4. Existing Financial System:

Bases on the findings, there are few existing products of dashboard offering to large community easily access in internet or standalone application that require

user to install in personal computer. There are (2) existing dashboard chosen for this comparison study, that all of them come from developer origin from United States of America (USA). The dashboard is FactSet and Bloomberg Forecast by Bloomberg Terminal.

CHAPTER-3

RESEARCH METHODOLOGY

In conducting this project, various processes and activities required before delivering the project and the complete application proposed. **Figure 3-1** illustrate phases involved and deliverables.

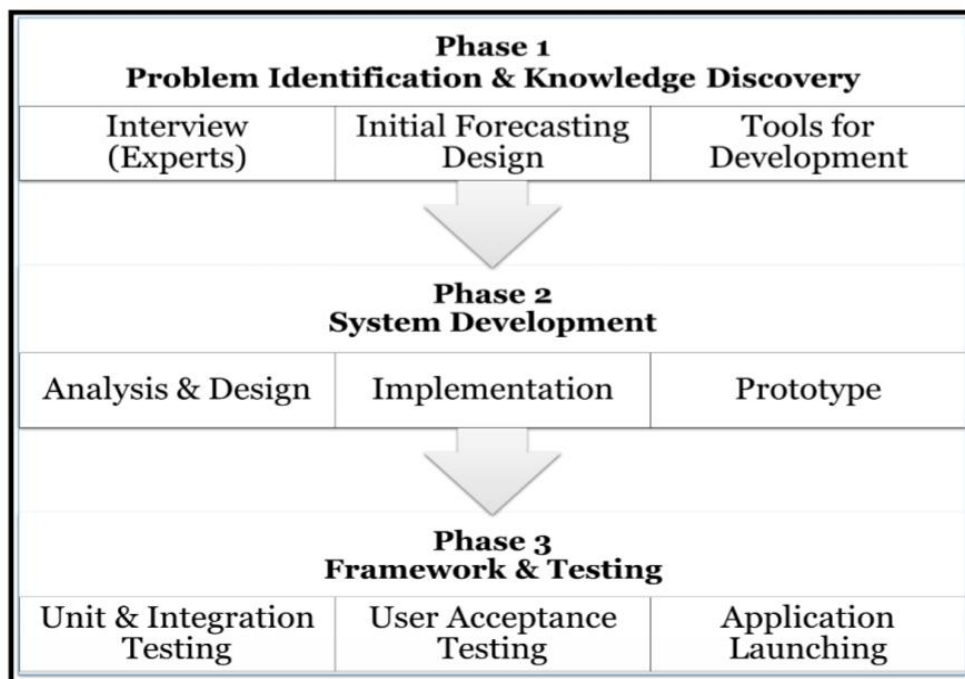


Figure 3-1 Application Development Phases

3.1. Phase 1: Problem Identification & Knowledge Discovery

Early before these phases started, a problem identification and objective are gathered and discussed. Various data and current state of the forecasting methodology is compiled and selected. Those data are grouped based on the

group of stock price investment as general, investment, stock price prediction, and data mining technique and model.

3.1.1. Interview:

Interview to the expert is conducted to find related findings and analyse the need of the application being proposed. The expert selected for this study is Ms. Nor Abiah Zainal Abidin, who has high experience on stock price forecasting and data mining technique.

Above terms during her job at one of the Malaysia Government Linked Company.

3.1.2. Initial Forecasting Design:

Forecasting design phase is the process to determine tools, resources, and data that will be useful for the project development and deliverables. Research and interview to the expert conducted to receive inputs regarding methodology and design that will be proper to implement in the dashboard.

3.1.3. Tools for Development:

i.) R Programming (R Studio)

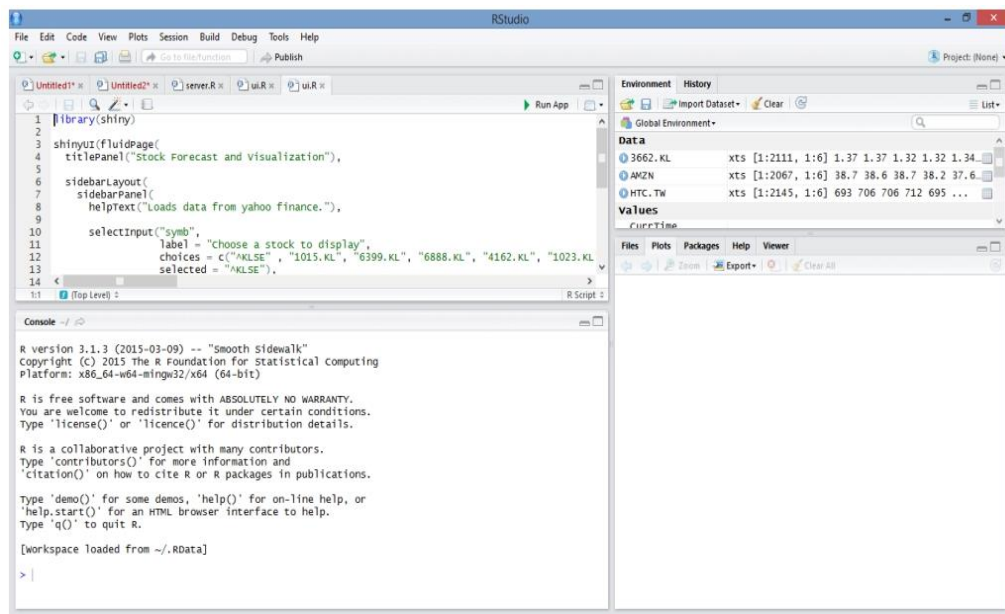


Figure 3-2 R Studio Interface

R is a programming language and software environment for statistical computing graphics. The R language is widely used among statisticians and data miners for developing statistical software. R programming is a very common tool to be used for forecasting, as many financial institutions currently required their employees to fully understand this software. R programming is previously only

widely used for statistical studies, but now it has been widely used for financial application to have some plotting and forecast.

Other tools that can be used than R Programming is to utilized the same function tools such as Mat Lab, Python, etc. R is a very important tool to develop this project. R is an open-source software that full with capabilities and packages to ease the user in doing several works with it.

This project utilized several packages such as “quant mod”, “ggplot2”, “TTR”, and “shiny”. Package “quant mod” used to extract data from Yahoo Finance directly. “ggplot2” and “TTR” utilized to plot the graph and time series of the historical data and the forecasted data. Lastly, “shiny” is being used to create a User Interface (UI) from R Studio.

ii.) HTML Web Developer

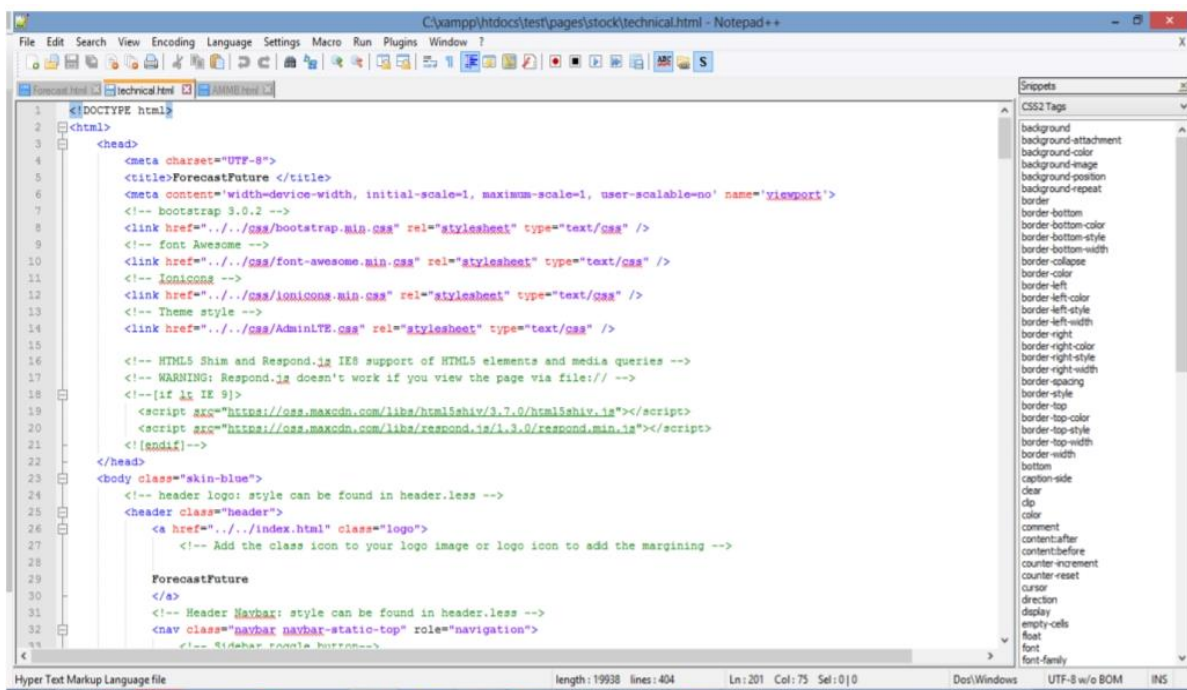


Figure 3-3 HTML Web Developer - Notepad++

HTML Web Developer is tools that being used to develop the dashboard and functions. HTML chosen due to its ability to launch online and interconnected to the cyber world of internet. Notepad++ used to code the HTML and gather all of the resources of back end to the dashboard. XAMPP utilized to reveal the HTML code and API widgets connected directly to internet. To make the widget of the Google News and Yahoo News newsfeed and twitter Tweets, several APIs from Twitter Dev, Yahoo Finance, Yahoo Pipes, etc. applied and setting into necessary requirements.

3.1.4. Equipment and Materials:

This project requires information and data that will be using in the planning and development phase. The most important of materials is stock price historical data, a computerized stock quote from the market that has been group into raw data. Currently, there are many ways to locate and extract the historical data from internet.

Such websites Financial Content, Yahoo Finance, Google Finance, and AOL are providing a real time historical data for paid subscribing and free. This project utilizes a feature from R Programming with one of its toolboxes to take real time data from Yahoo Finance. Furthermore, as explained before Twitter newsfeed used for fundamental analysis for user decision making after user successfully analyse the price and predict the quote. The algorithm of Twitter will be carried by only allowing user to input the search criteria based on the company's name or quote stock such as Google GOOG or Apple AAPL. CNBC (2013) reported that some of the big hedge funds and investors are now using twitter and other social media for their trade strategies or only to maintain watching on corporate action.

3.2. Phase 2: Prototyping Methodology:

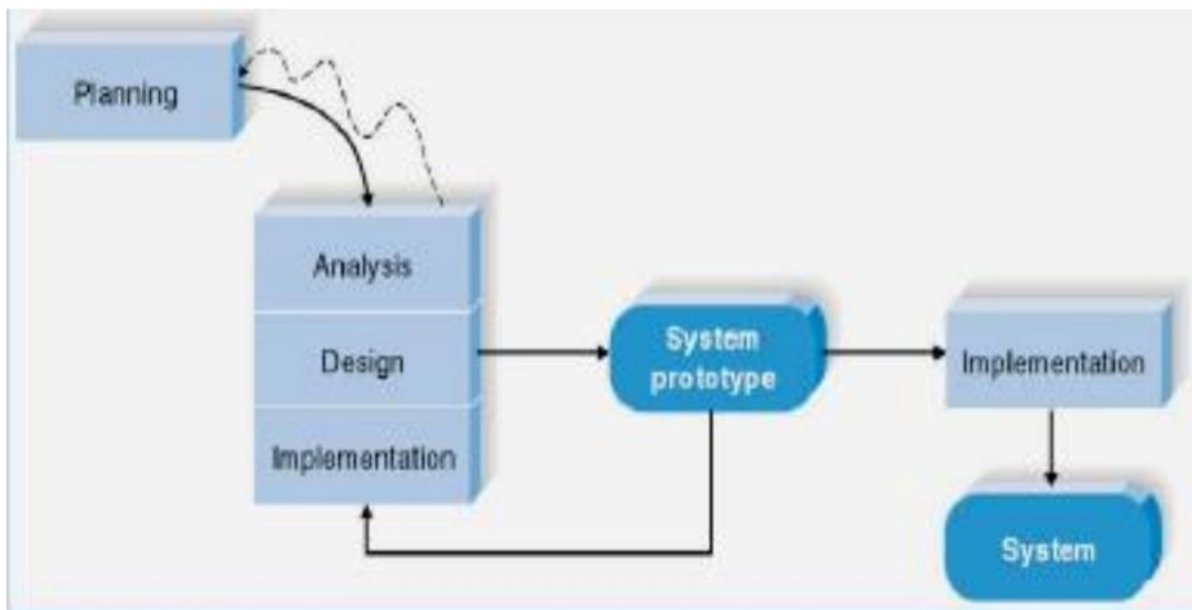


Figure 3-4 Prototyping Methodology (Box, Jenkins, & Reince, 2013)

Prototyping-based Methodology is a development lifecycle designed to give much faster development and higher-quality results than those achieved with traditional lifecycle. Prototyping has been widely known to make developers experience of software development is easier than other methodology. Provides a system for the users to interact with the product although it may not finish yet.

Prototyping picked as the method of system lifecycle to conduct this project due to its abilities to adapt with developer time that has been set previously.

Although the major steps such as analysis and quick design, testing, and implementation blocked into time. Prototyping helps developer to take care any problem in testing phase because of substantial prototyping cycles.

These cycles can help the developer to make sure the analysis is correct and enable to re-build the analysis without compromising the final phase and product.

3.2.1. Requirement Gathering and Planning:

The project topic and title to the final year project committee with the endorsement from supervisor proposed has been completed in the second week of FYP 1. After the topic, approved students are allowed to continue making the project and discuss with respective supervisor to conduct a requirement and 22 scope analysis. Phases occurred during the completion of Final Year Project are displayed in Gantt chart and key milestone so that students can easily monitor the process and key deliverables.

3.2.2. Prototyping Phase:

The prototyping phase starts with gathering all equipment, materials, and resources required. The historical data extracted from Yahoo Finance, the statistical software selected, R Programming, and models to be implemented for forecasting the stock quote. The historical price is either extracted automatically by using one of the tool boxes from R, or by downloading it from the Yahoo Finance site. Furthermore, the forecasting method is designed by passively state the level of confidence of risk made by the developer. The prototype made by combining the method use in the R Programming and the HTML Developer, which is expected to, completed at the same time. The building of the statistical algorithm will be the widely held in the phase of FYP one, while the combining method will be briefly done in FYP 2. Several attempts have been done towards find the algorithm and proper model that will be chosen. Analysing the stock price by plotting it into time series as shown in Figure 3-2, is one of the steps to complete the project.

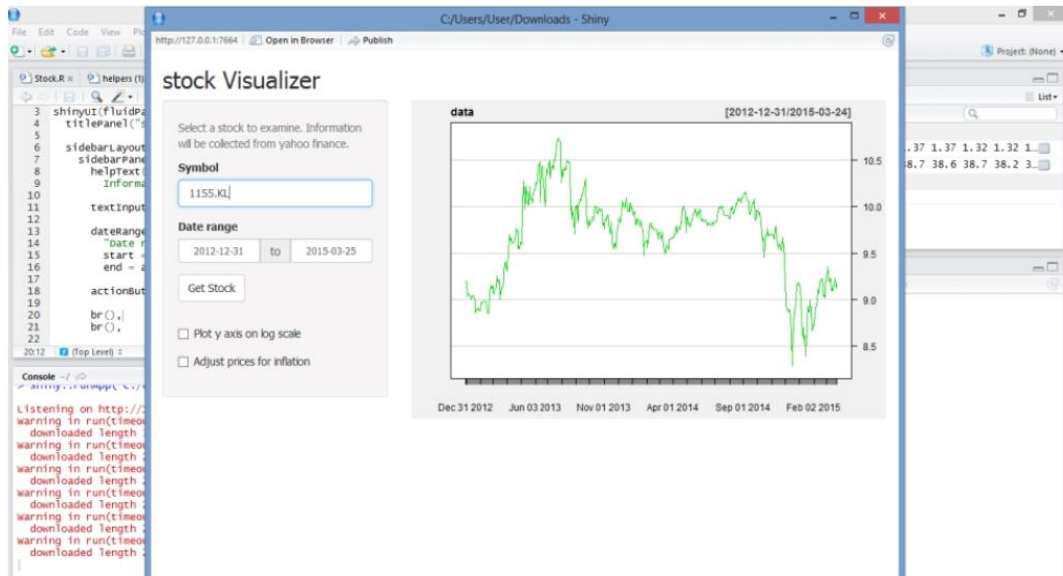


Figure 3-5 Stock Visualization

3.2.3. Testing Phase:

Testing will be performed after the entire project modules have been completed and ready to test. Testing is expected to be done by the series of attempt by the experts and several students who interested to study this project. The testing will eventually result to some kind of record and feedback from the users those benefits to the form. The evaluation is also given and recorded for getting insight and feedback from the users as well as any shortcomings in requirements or project deliverables.

3.2.4. Deployment Phase:

The final phase of the prototyping is to deploy the project after completing series of testing and improvement in terms of design and functional capabilities. The dashboard will later be ready to be installed and use by the users for their own investing strategy.

3.3. Phase 3: Proposed Framework:

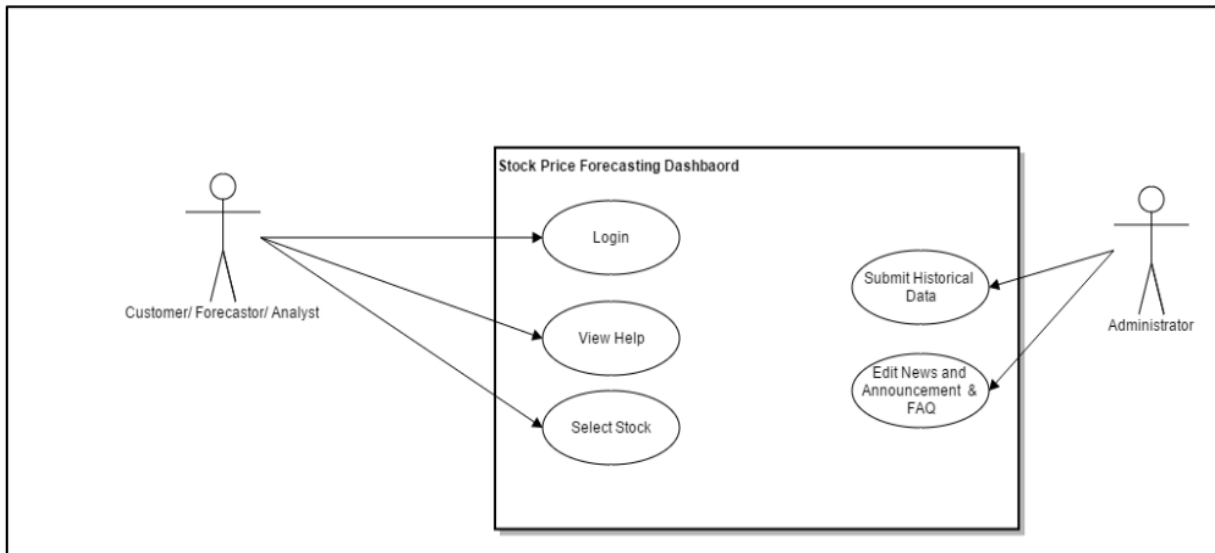


Figure 3-6 Use Case Diagram

Use Case above explained the user and admin ability towards the application. User is able to login to the application to make sure the security of the application guaranteed. View Help is the pop-up window to explain the application brief and guidelines of ‘how-to’ when they open to the dashboard.

The help window is also filled with the explanation of the copyright, the dashboard creator details and the contact details. Lastly, the main function of the dashboard is that user is able to select stock or write the stock quote of the market to the tool box that will be searched and forecast by the machine learning.

The administrator is the owner to the application. The admin has authority to submit the historical data to the database or submit the new list of quotes and edit the help windows.

3.3.1. Flow Chart:

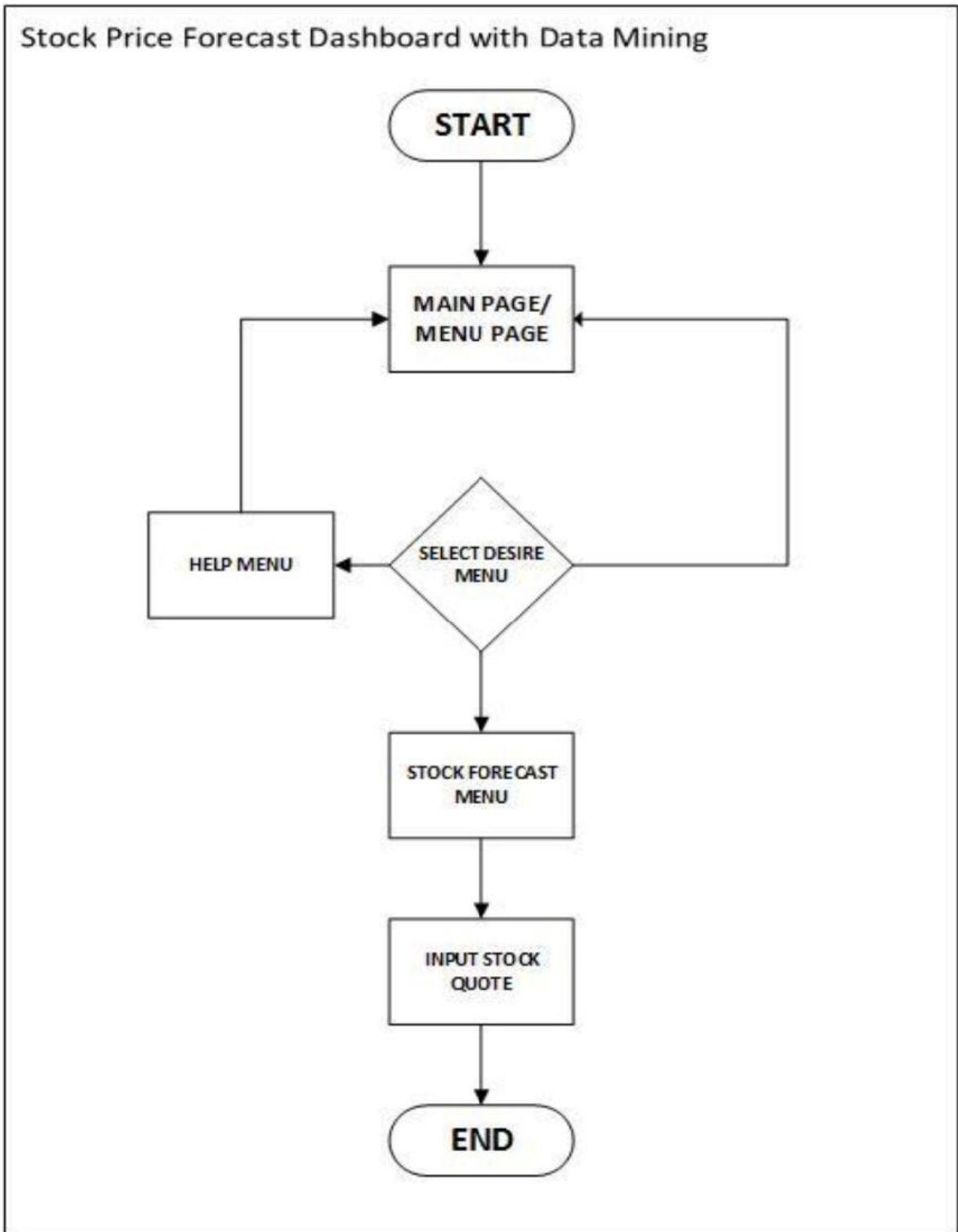


Figure 3-7 Flow Chart 3

3.3.2. System Architecture:

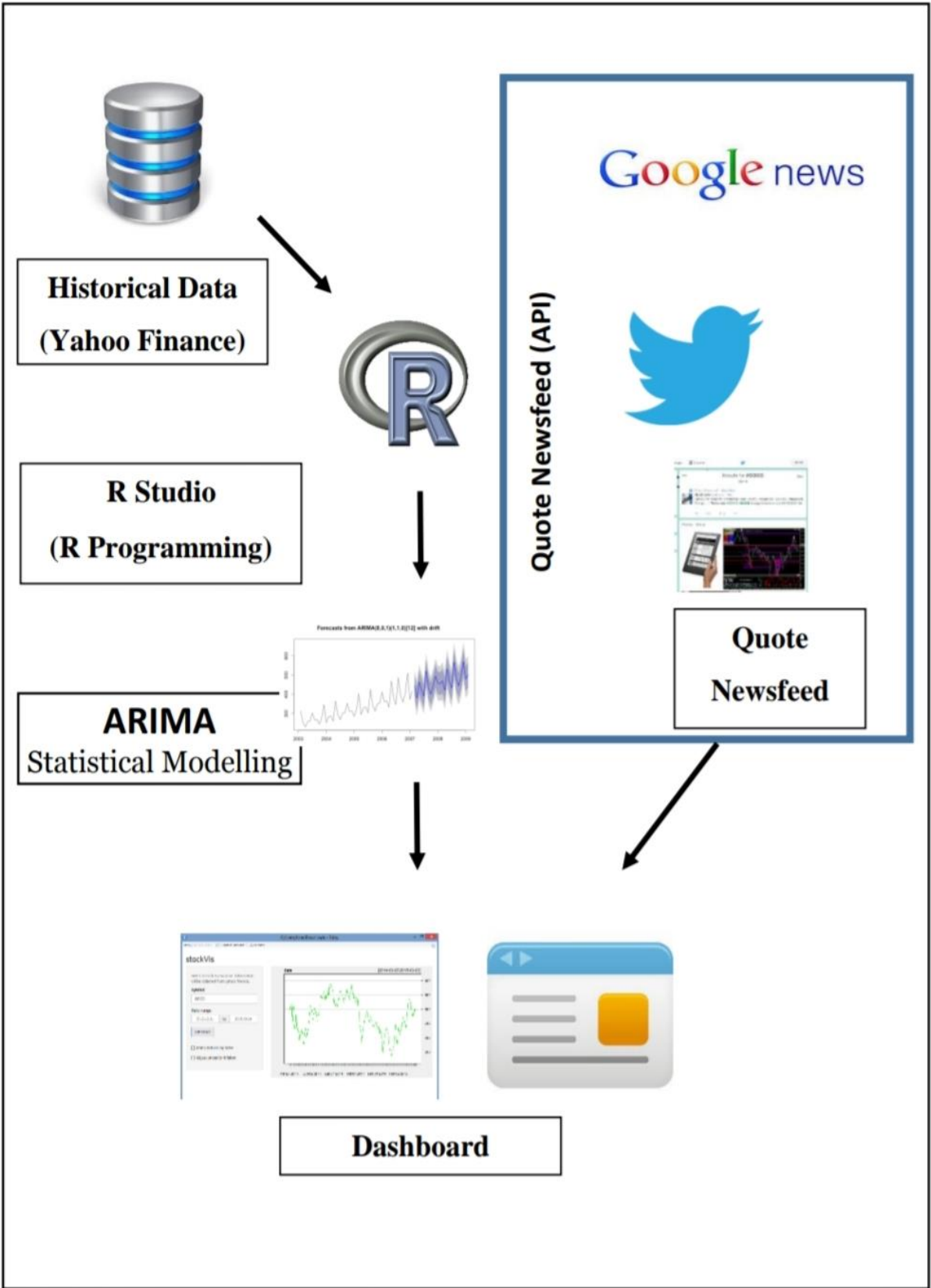


Figure 3-8 System Architecture

CHAPTER-4
RESULT AND DISCUSSION

4.1. Introduction:

This chapter explained the process and result of research performed to uphold the project objectives. The method of research conducted are expert interview and experimental. An interview was conducted by interviewing one of the experts that has this study background and perform day-to-day basis on statistical occupation and depiction to R Programming. Furthermore, experimental method conducted by attempt series of experiment of stock price forecasting with model chosen to investigate and decide the best implementation for the project deliverables.

4.2. Qualitative Method: Interview

The main objective to conduct interview to the experts is to gain deep understanding on the stock price forecast and existing forecasting tools, which use by the expert or by the company. Furthermore, by interviewing the expert, several inputs and suggestion towards the project and the dashboard. Nineteen questions were delivered to the expert that consist of stock price forecasting fundamental, the forecasting model, and the dashboard.

4.2.1. Stock Price Fundamental:

Several questions asked to the experts to receive necessary point of information to accomplish the project. The experts conveyed her opinion that stock forecasting is an attempt to learn and analyse the movement of the price trend with either fundamental or technical or quantitatively approach. Most of the investors are using technical approach, as it is only required short time and lesser resources compare to the other. 32 She said that fundamental approach usually looks into the balance sheet of the company or the financial statement released by the company. The company's movement of Return on Investment (ROI), management direction and economic situation can also be used to predict the company's stock performance in the future.

Moreover, quantitative approach is trying to avoid fundamental factors and solely treat current and historical price as the subject of method by using statistical or other technique. Next, the expert agrees that all forecaster need to understand the technical of the forecasting method. The expert believes that the forecaster needs to understand the model and the technique so that when they start to do some forecasting, they can easily interact with the model if they want to change the parameters to find the proper and fit to their profiles without affect the whole method applied. On the topic about who the most party benefits to stock forecasting, she explained that the person who most likely do forecasting for stock price range from equity analysts, equity investment, sales representative or broker and personal or group investor.

Onto the current issue that the broker or security companies are currently only provide technical paper of bundle of report of technical analysis which not included with from the commission to the broker or trading fees the experts believes that it should not be supposed to.

She believes that the companies pay researchers and analysts' salary to analyses and forecast the stock has already come together and not need to add separate cost to consumers. Therefore, it is not justifiable to add certain of money for the investor if they want to subscribe the forecasting report.

The expert later added that there are numbers of method currently use to forecast share price, such as ARIMA, Smart Beta, econometrics, and machine learning. She said she prefers to apply ARIMA as the method to forecast stock price, if the forecaster is still new in these activities.

She said ARIMA package is mostly available, easy to get, and do not require the user to do some math. When author was trying to proposed the application that want to developed, she personally agrees and encourage the developer to also make it in subscribe basis to keep the value exit.

She responded that if the dashboard is distributed to 33 everyone especially beginner investor, it may increase number of investors in the future, as the forecasting is become easier. Based on the interview stated above, it can be concluded that the dashboard proposed is in line with the demand in the market.

As more people are currently starting to trade in stock market, more knowledge and instant update of information is truly important. The dashboard method to utilize ARIMA as its model is also in a row to the analyst and statistician use right now, as it is ways and suitable for beginner predictors.

4.3. Experimental:

Several attempts done to find the most suitable model being use for this project.

Overall, most of the attempt are using ARIMA model and use necessary package that is free distributed in the application. From the perspective of forecasting ability, author has been successfully implemented ARIMA model to forecast the price.

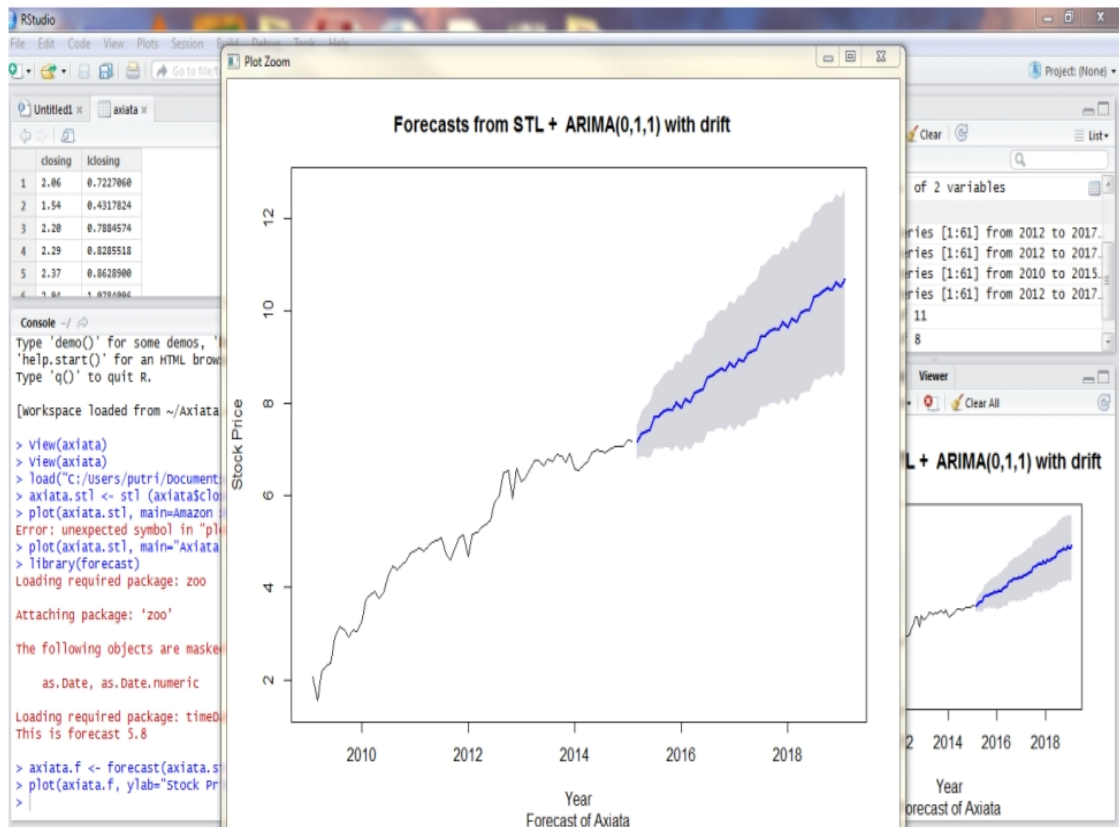


Figure 4-1 Stock Price Forecasting with ARIM

Figure - shown proposed design of the dashboard that will be developed during developing phase. The proposed design has been comparing to certain of example of existing dashboard in the market. The dashboard will be using full HTML web page by combining result from R Programming and Twitter and Google News API as the objects.

Twitter API will be fascinated from the technique of bootstraps that offers free-cost to developers especially in HTML platform.

These user interfaces are presented on the dashboard that being live on localhost.

The application developed to forecast stock price of an index or a company in Bursa Malaysia.

The main idea of this dashboard is to become an alternative for the early or ready-to-enter investor to analyses information and forecast the stock price before entering the market.

There are has been the technique of bootstraps that offers free-cost to developers comparing to certain of example of existing dashboard mainly 4 activities which are “Dashboard” or homepage, “Forecast”, “Technical Analysis”, and “Companies Profile”.

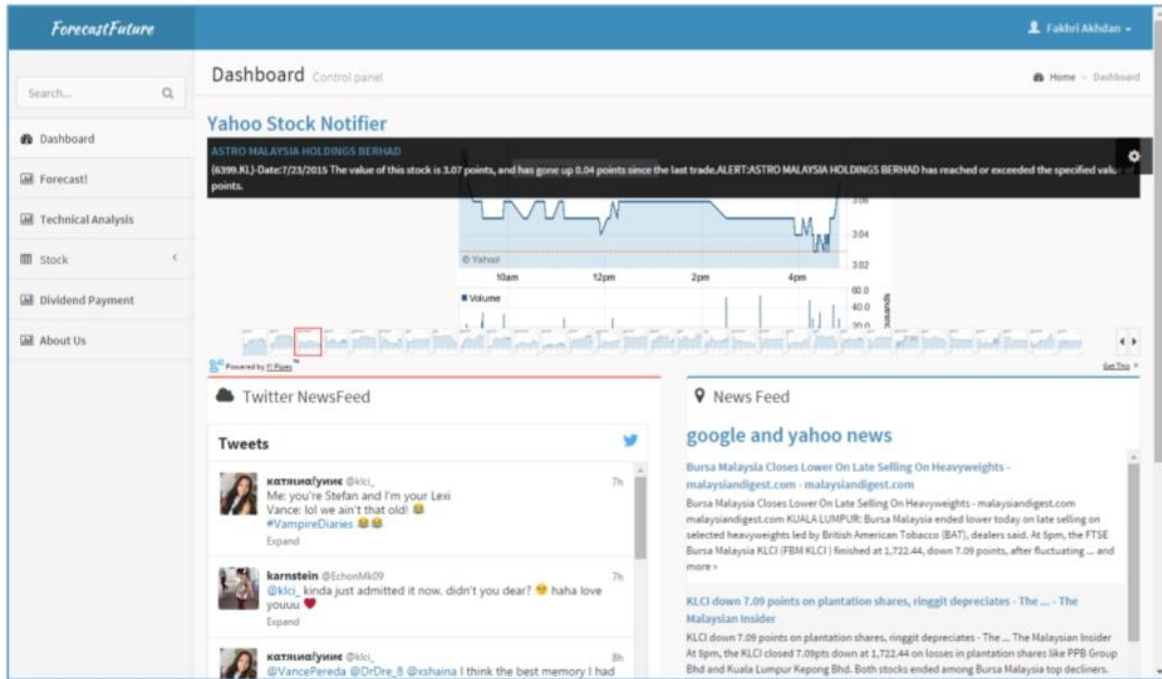


Figure 4-2 Homepage Page

Home Page or Dashboard, as in Figure 3, consists of options “Forecast!” “Technical Analysis”, “Stock Profile”, “About Us”. For “Forecast!” button, the system will proceed to forecasting tool page. “Technical Analysis” will open to the page of technical analysis. “Stock Profile” button will show and display the stock profile by options of the tree-view. Lastly, “About Us” will proceed to the dashboard disclaimer and profile. On the dashboard or the homepage, it is displaying a live Stock Yahoo Notifier from the Yahoo Finance of top 30 largest companies on the Bursa Malaysia by market capitalization called as Kuala Lumpur Composite Index (KLCI) (Bursa Malaysia, n.d.). Added with the twitter and google newsfeed portraying live tweet and news quote of the KLCI and market subject.

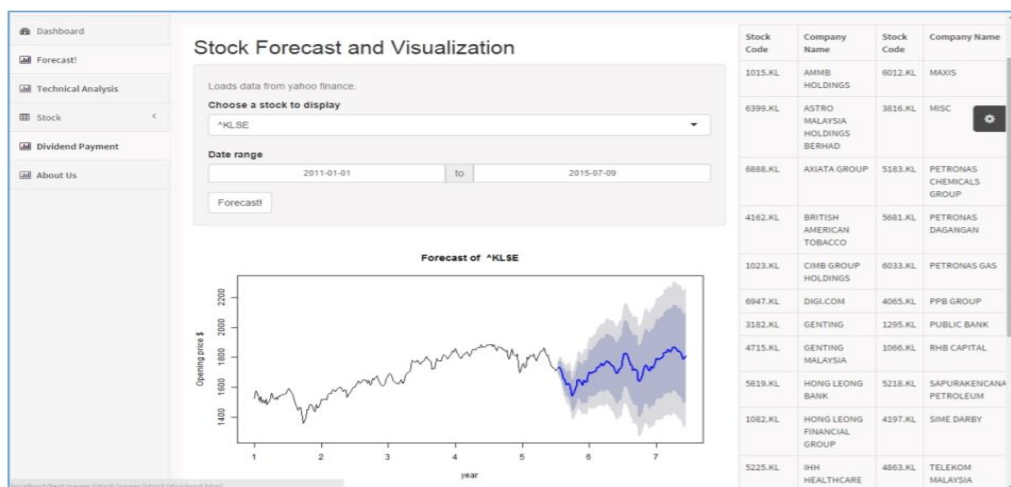


Figure 4-3 Homepage Page

Home Page or Dashboard, as in Figure 3, consists of options “Forecast!” “Technical Analysis”, “Stock Profile”, “About Us”. For “Forecast!” button, the system will proceed to forecasting tool page. “Technical Analysis” will open to the page of technical analysis. “Stock Profile” button will show and display the stock profile by options of the tree-view. Lastly, “About Us” will proceed to the dashboard disclaimer and profile. On the dashboard or the homepage, it is displaying a live Stock Yahoo Notifier from the Yahoo Finance of top 30 largest companies on the Bursa Malaysia by market capitalization called as Kuala Lumpur Composite Index (KLCI) (Bursa Malaysia, n.d.). Added with the twitter and google newsfeed portraying live tweet and news quote of the KLCI and market subject.

At the “Forecast!” page, user is able to pick any of the KLCI’s 30 biggest market capitalization in Bursa Malaysia by choosing the stock code in the drop-down menu. The user could refer to the table on the right to find the desired company or index code to the names of the stock company. After the code has been chosen, a date of parameters can be an optional choose if the user wants to analyse the historical price less or more than 5 years.

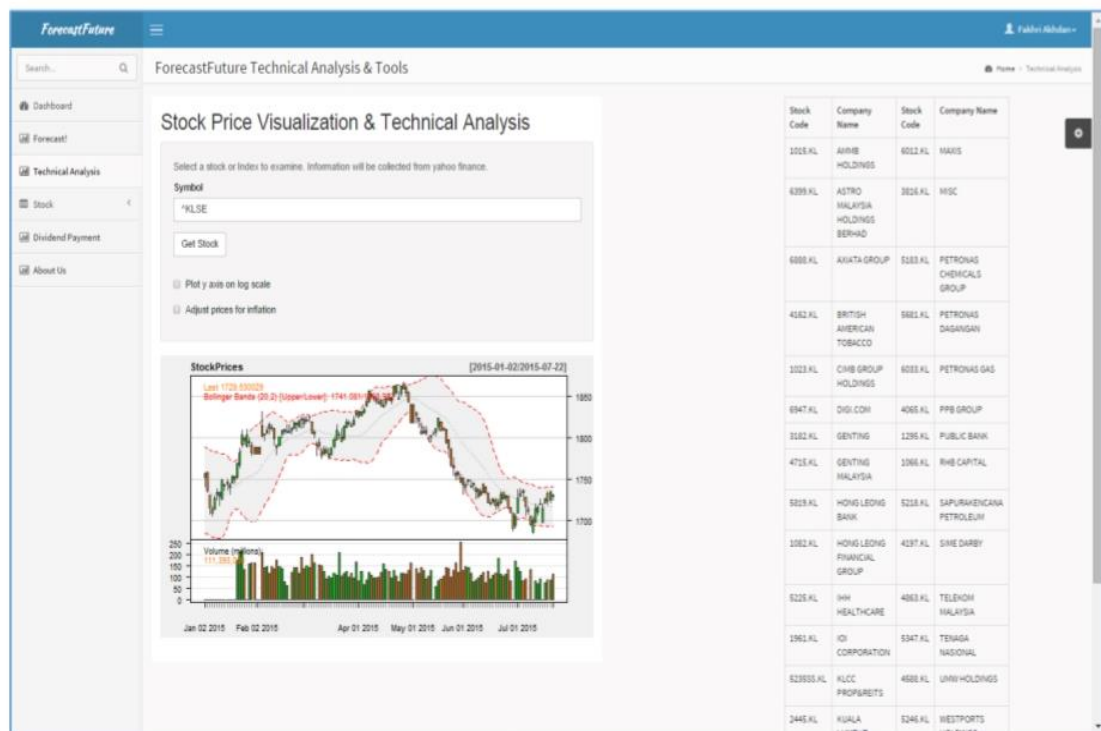


Figure 4-4 Stock Price Visualization & Technical Analysis

“Technical Analysis” page is able to displaying a technical analysis-using graph and chart technique that usually use by professional trader to analyze pattern and predict the price. In this page, user is not only limited to list 30 companies in KLCI on the next to the right table of information, but also enable user to analyse every stock quote that registered in Yahoo Finance. The date parameters are also changeable to analyse lesser and longer time of period.

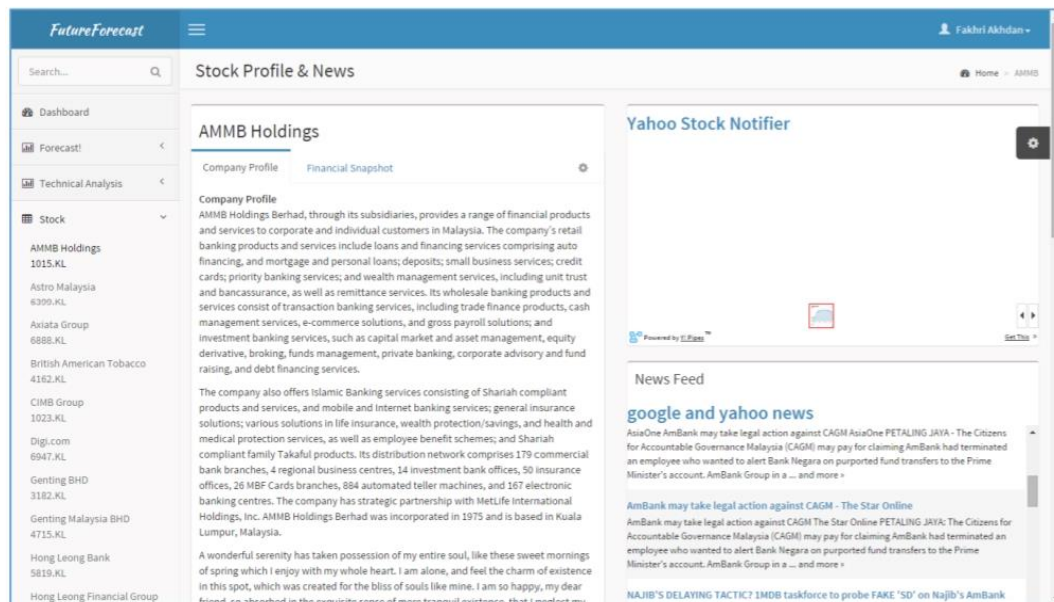


Figure 4-5 Stock Profile & News

At “Stock Profile” page, user is served with the stock of company profile and financial standing of the particular company. Yahoo Stock Notifier is also added to provide an updated stock price while reading the stock profile. Under the notifier, the newsfeed from Google news and Yahoo is helping the user to analyse the summary of business action related to the company.

CHAPTER-5
PROPOSED SYSTEM

PROPOSED SYSTEM

We have created a single page web application using dash and machine learning models. The main goal of this project is to precisely predict the future closing price of stock over a period of time in future. In this project we have used dash html components and dash core components to create website's structure and for enhancing the site's UI we have used CSS for styling.

Plots of data are generated by using the matplotlib library of python and the data is fetched using finance. Machine learning models are implemented for predicting the stock price for the dates requested by the user. The project is deployed on Heroku to host it live.

METHODOLOGY:

We saw that the new user was afraid to invest in the share market because he did not have the knowledge and did not have any tool with the help of which he could do this work.

So, then we created a tool with the help of machine learning and deep learning, which can tell in very precise way the market will move by analysing things.

We did this work by taking data from finance. We imported this data through library and analysed with the help of machine learning model and we did it with a machine learning model because it does this thing very precisely.

In this we have used the model of Deep Learning LSTM, we have trained the data in its own way, we have learned to use it because it is very advanced and performs this task very accurately and the result is more accurate.

We used this method because in this we can train the data, due to which it can do its work very accurately and get close to the real result with great accuracy.

There is a drawback in this that it will work the way we trained it machine learning model and we did it with a machine learning.

Machine learning models are implemented for predicting the stock price for the dates requested by the user.


We have to face a lot of difficulty in trained the data because the model cannot make any changes in itself like the data is transcribed, it works the same way.

RESULT:

Create Basic Website Layout Expected Outcome. By now you should have the basic web page setup as shown below in the second image which can be seen by starting the server locally as shown below in the first image.

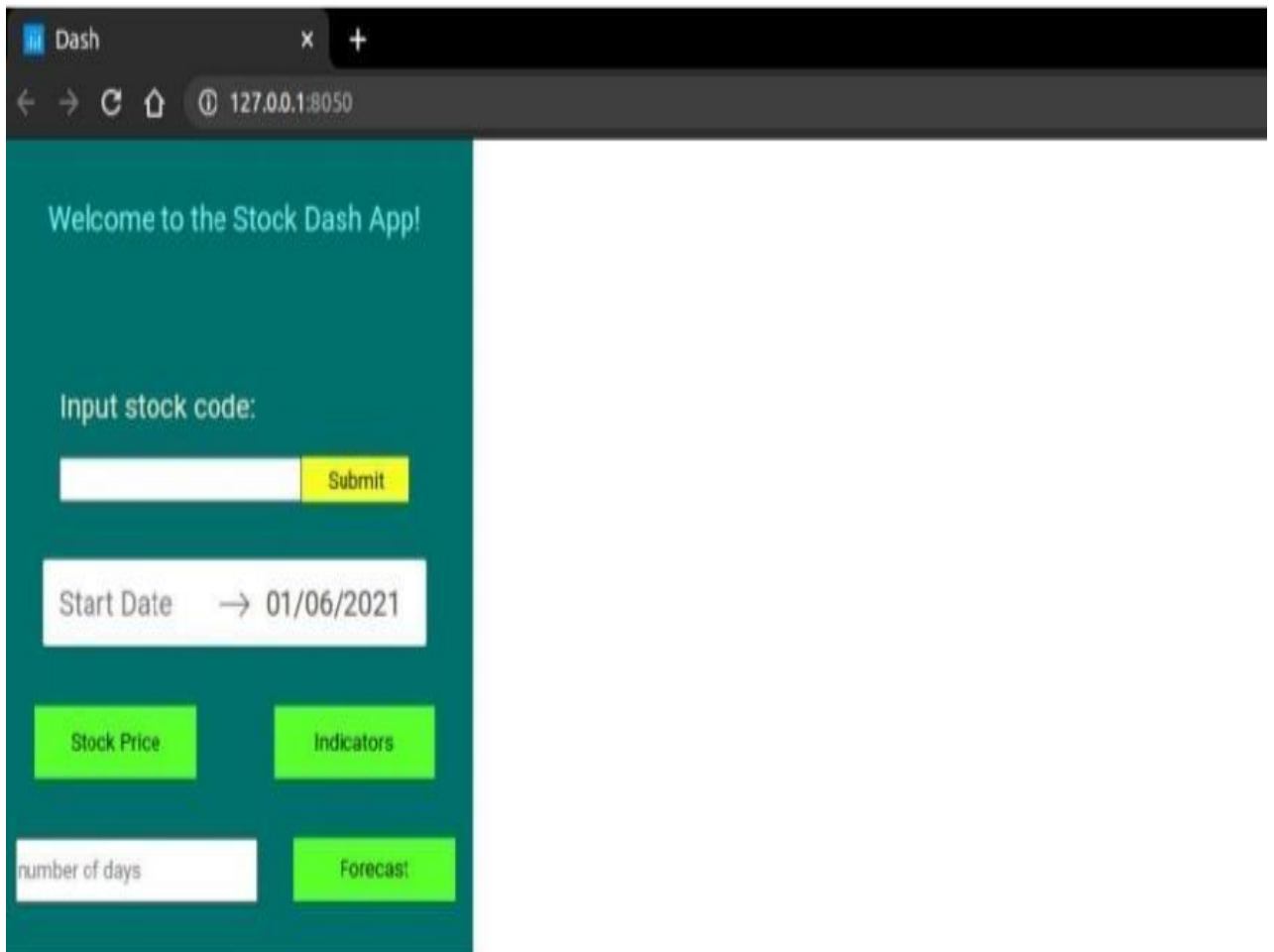
```
sujoy@sujoy-VivoBook-ASUSLaptop-X412FAC-X412FA ~/D/M/stock-dash (master)>
/usr/bin/python3 "/home/sujoy/Documents/My Projects/stock-dash/app.py"
Dash is running on http://127.0.0.1:8050/

* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
```



Style Your Application:

At this stage, your webpage should have a much better look than before. You may use the below web page as your starter



Generating a Company's Information and Graphs:

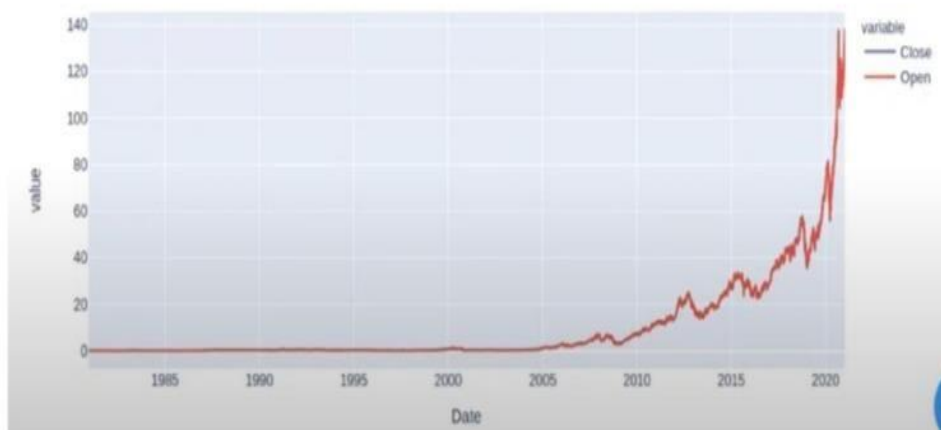
We are going to use the finance python library to get company information (name, logo and description) and stock price history. Dash's call back functions will be used to trigger updates based on change in inputs.

Start Date → 01/02/2021

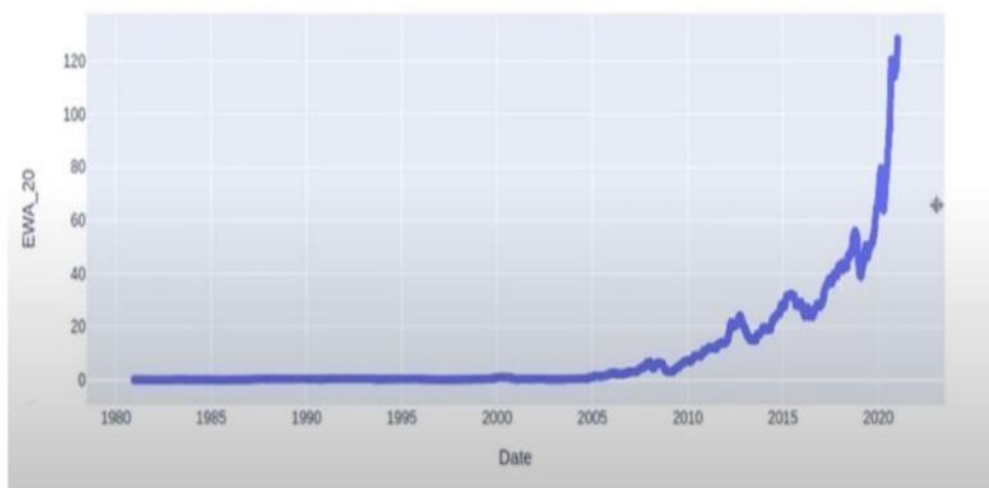
 **Apple Inc.**

Apple Inc. designs, manufactures, and markets smartphones, personal computers, tablets, wearables, and accessories worldwide. It also sells various related services. The company offers iPhone, a line of smartphones; Mac, a line of personal computers; iPad, a line of multi-purpose tablets; and wearables, home, and accessories comprising AirPods, Apple TV, Apple Watch, Beats products, HomePod, iPod touch, and other Apple-branded and third-party accessories. It also provides AppleCare support services; cloud services store services; and operates various platforms, including the App Store, that allow customers to discover and download applications and digital content, such as books, music, video, games, and podcasts. In addition, the company offers various services, such as Apple Arcade, a game subscription service; Apple Music, which offers users a curated listening experience with on-demand radio stations; Apple News+, a subscription news and magazine service; Apple TV+, which offers exclusive original content; Apple Card, a co-branded credit card; and Apple Pay, a cashless payment service, as well as licenses its intellectual property. The company serves consumers, and small and mid-sized businesses; and the education, enterprise, and government markets. It sells and delivers third-party applications for its products through the App Store. The company also sells its products through its retail and online stores, and direct sales force; and third-party cellular network carriers, wholesalers, retailers, and resellers. Apple Inc. was founded in 1977 and is headquartered in Cupertino, California.

Closing and Opening Price vs Date



Exponential Moving Average vs Date

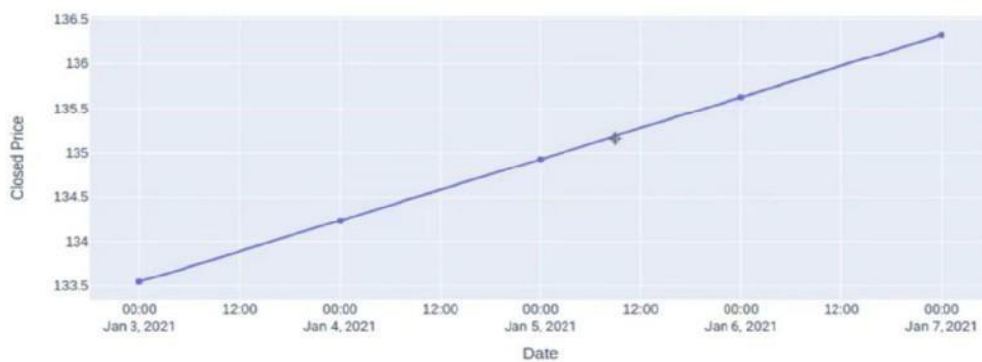


Creating the Machine Learning Model:

After finishing all this, we should have a complete project where user input - stock, code can give us the company information and relevant plots. Also, user input number of days can give us a forecast plot.



Predicted Close Price of next 5 days



Exponential Moving Average vs Date



Deploying the project on Heroku Finally, our web app is deployed and can be accessed by anyone in the world.

CHAPTER-6
Stock Market

6.1. What is the Stock Market?

The stock market refers to the collection of markets and exchanges where regular activities of buying, selling, and issuance of shares of publicly-held companies take place. Such financial activities are conducted through institutionalized formal exchanges or over-the-counter (OTC) marketplaces which operate under a defined set of regulations. There can be multiple stock trading venues in a country or a region which allow transactions in stocks and other forms of securities.

While both terms - stock market and stock exchange - are used interchangeably, the latter term is generally a subset of the former. If one says that she trades in the stock market, it means that she buys and sells shares/equities on one (or more) of the stock exchange(s) that are part of the overall stock market. The leading stock exchanges in the U.S. include the New York Stock Exchange (NYSE), Nasdaq, and the Chicago Board Options Exchange (CBOE). These leading national exchanges, along with several other exchanges operating in the country, form the stock market of the U.S.

Though it is called a stock market or equity market and is primarily known for trading stocks/equities, other financial securities - like exchange traded funds (ETF), corporate bonds and derivatives based on stocks, commodities, currencies, and bonds - are also traded in the stock markets.

6.2. Understanding the Stock Market:

While today it is possible to purchase almost everything online, there is usually a designated market for every commodity. For instance, people drive to city outskirts and farmlands to purchase Christmas trees, visit the local timber market to buy wood and other necessary material for home furniture and renovations, and go to stores like Walmart for their regular grocery supplies.

Such dedicated markets serve as a platform where numerous buyers and sellers meet, interact and transact. Since the number of market participants is huge, one is assured of a fair price. For example, if there is only one seller of Christmas trees in the entire city, he will have the liberty to charge any price he pleases as the buyers won't have anywhere else to go. If the number of tree sellers is large in a common marketplace, they will have to compete against each other to attract buyers.

The buyers will be spoiled for choice with low- or optimum-pricing making it a fair market with price transparency.

Even while shopping online, buyers compare prices offered by different sellers on the same shopping portal or across different portals to get the best deals, forcing the various online sellers to offer the best price.

A stock market is a similar designated market for trading various kinds of securities in a controlled, secure and managed environment. Since the stock market brings together hundreds of thousands of market participants who wish to buy and sell shares, it ensures fair pricing practices and transparency in transactions. While earlier stock markets used to issue and deal in paper-based physical share certificates, the modern-day computer-aided stock markets operate electronically.

6.3. How the Stock Market Works?

In a nutshell, stock markets provide a secure and regulated environment where market participants can transact in shares and other eligible financial instruments with confidence with zero- to operationalism. Operating under the defined rules as stated by the regulator, the stock markets act as primary markets and as secondary markets.

As a primary market, the stock market allows companies to issue and sell their shares to the common public for the first time through the process of initial public offerings (IPO). This activity helps companies raise necessary capital from investors. It essentially means that a company divides itself into a number of shares (say, 20 million shares) and sells a part of those shares (say, 5 million shares) to the common public at a price (say, \$10 per share).

To facilitate this process, a company needs a marketplace where these shares can be sold. This marketplace is provided by the stock market. If everything goes as per the plans, the company will successfully sell the 5 million shares at a price of \$10 per share and collect \$50 million worth of funds.

Investors will get the company shares which they can expect to hold for their preferred duration, in anticipation of rising share price and any potential income in the form of dividend payments. The stock exchange acts as a facilitator for this capital raising process and receives a fee for its services from the company and its financial partners.

Following the first-time share issuance IPO exercise called the listing process, the stock exchange also serves as the trading platform that facilitates regular buying and selling of the listed shares. This constitutes the secondary market. The stock exchange earns a fee for every trade that occurs on its platform during the secondary market activity.

The stock exchange shoulders the responsibility of ensuring price transparency, liquidity, price discovery and fair dealings in such trading activities. As almost all major stock markets across the globe now operate electronically, the exchange maintains trading systems that efficiently manage the buy and sell orders from various market participants.

They perform the price matching function to facilitate trade execution at a price fair to both buyers and sellers.

A listed company may also offer new, additional shares through other offerings at a later stage, like through rights issue or through follow-on offers. They may even buy back or delist their shares. The stock exchange facilitates such transactions.

The stock exchange often creates and maintains various market-level and sector-specific indicators, like the S&P 500 index or Nasdaq 100 index, which provide a measure to track the movement of the overall market. Other methods include the Stochastic Oscillator and Stochastic Momentum Index.

The stock exchanges also maintain all company news, announcements, and financial reporting, which can be usually accessed on their official websites. A stock exchange also supports various other corporate-level, transaction-related activities.

For instance, profitable companies may reward investors by paying dividends which usually comes from a part of the company's earnings. The exchange maintains all such information and may support its processing to a certain extent.

6.4. Functions of a Stock Market:

A stock market primarily serves the following functions:

Fair Dealing in Securities Transactions: Depending on the standard rules of demand and supply, the stock exchange needs to ensure that all interested market participants have instant access to data for all buy and sell orders thereby helping in the fair and transparent pricing of securities.

Additionally, for example, there may be three buyers who have placed orders for buying Microsoft shares at \$100, \$105 and \$110, and there may be four sellers who are willing to sell Microsoft shares at \$110, \$112, \$115 and \$120.

The exchange (through their computer operated automated trading systems) needs to ensure that the best buy and best sell are matched, which in this case is at \$110 for the given quantity of trade.

Stock exchange needs to ensure that all interested market participants have instant access to data processing to a certain extent.

6.5. Efficient Price Discovery:

Stock markets need to support an efficient mechanism for price discovery, which refers to the act of deciding the proper price of a security and is usually performed by assessing market supply and demand and other factors associated with the transactions.

Say, a U.S.-based software company is trading at a price of \$100 and has a market capitalization of \$5 billion. A news item comes in that the EU regulator has imposed a fine of \$2 billion on the company which essentially means that 40 percent of the company's value may be wiped out.

While the stock market may have imposed a trading price range of \$90 and \$110 on the company's share price, it should efficiently change the permissible trading price limit to accommodate for the possible changes in the share price, else shareholders may struggle to trade at a fair price.

Liquidity Maintenance: While getting the number of buyers and sellers for a particular financial security are out of control for the stock market, it needs to ensure that whosoever is qualified and willing to trade gets instant access to place orders which should get executed at the fair price.

Security and Validity of Transactions: While more participants are important for efficient working of a market, the same market needs to ensure that all participants are verified and remain compliant with the necessary rules and regulations, leaving no room for default by any of the parties.

Additionally, it should ensure that all associated entities operating in the market must also adhere to the rules, and work within the legal framework given by the regulator.

6.6. Support All Eligible Types of Participants:

A marketplace is made by a variety of participants, which include market makers, investors, traders, speculators, and hedgers. All these participants operate in the stock market with different roles and functions.

For instance, an investor may buy stocks and hold them for long-term spanning many years, while a trader may enter and exit a position within seconds.

A market maker provides necessary liquidity in the market, while a hedger may like to trade in derivatives for mitigating the risk involved in investments.

The stock market should ensure that all such participants are able to operate seamlessly fulfilling their desired roles to ensure the market continues to operate efficiently.

6.6.1. Investor Protection:

Along with wealthy and institutional investors, a very large number of small investors are also served by the stock market for their small number of investments. These investors may have limited financial knowledge, and may not be fully aware of the pitfalls of investing in stocks and other listed instruments.

The stock exchange must implement necessary measures to offer the necessary protection to such investors to shield them from financial loss and ensure customer trust.

For instance, a stock exchange may categorize stocks in various segments depending on their risk profiles and allow limited or no trading by common investors in high-risk stocks. Exchanges often impose restrictions to prevent individuals with limited income and knowledge from getting into risky bets of derivatives.

6.6.2. Balanced Regulation:

Listed companies are largely regulated and their dealings are monitored by market regulators, like the Securities and Exchange Commission (SEC) of the U.S. Additionally, exchanges also mandate certain requirements – like, timely filing of quarterly financial reports and instant reporting of any relevant developments - to ensure all market participants become aware of corporate happenings.

Failure to adhere to the regulations can lead to suspension of trading by the exchanges and other disciplinary measures.

6.6.3. Regulating the Stock Market:

A local financial regulator or competent monetary authority or institute is assigned the task of regulating the stock market of a country. The Securities and Exchange Commission (SEC) is the regulatory body charged with overseeing the U.S. stock markets.

The SEC is a federal agency that works independently of the government and political pressure. The mission of the SEC is stated as: "to protect investors, maintain fair, orderly, and efficient markets, and facilitate capital formation."

The Securities and Exchange Commission (SEC) is the regulatory body charged with overseeing the U.S. stock markets.

6.6.4. Stock Market Participants:

Along with long-term investors and short-term traders, there are many different types of players associated with the stock market. Each has a unique role, but many of the roles are intertwined.

Depend on must also adhere to the rules, and work within the legal framework given by the regulator players associated with the stock marketing each other to make the market run effectively.

- Stockbrokers, also known as registered representatives in the U.S., are the licensed professionals who buy and sell securities on behalf of investors. The brokers act as intermediaries between the stock exchanges and the investors by buying and selling stocks on the investors' behalf. An account with a retail broker is needed to gain access to the markets.
- Portfolio managers are professionals who invest portfolios, or collections of securities, for clients. These managers get recommendations from analysts and make the buy or sell decisions for the portfolio. Mutual fund companies, hedge funds, and pension plans use portfolio managers to make decisions and set the investment strategies for the money they hold.
- Investment bankers represent companies in various capacities, such as private companies that want to go public via an IPO or companies that are involved in pending mergers and acquisitions. They take care of the listing process in compliance with the regulatory requirements of the stock market.
- Custodian and depot service providers, which are institutions holding customers' securities for safekeeping so as to minimize the risk of their theft or loss, also operate in sync with the exchange to transfer shares to/from the respective accounts of transacting parties based on trading on the stock market.
- **Market maker:** A market maker is a broker-dealer who facilitates the trading of shares by posting bid and ask prices along with maintaining an inventory of shares.
- He ensures sufficient liquidity in the market for a particular (set of) share(s), and profits from the difference between the bid and the ask price he quotes.
- Mutual fund companies, hedge funds, and pension plans use portfolio managers to make decisions and set the investment.

6.7. How Stock Exchanges Make Money?

Stock exchanges operate as for-profit institutes and charge a fee for their services. The primary source of income for these stock exchanges are the revenues from the transaction fees that are charged for each trade carried out on its platform. Additionally, exchanges earn revenue from the listing fee charged to companies during the IPO process and other follow-on offerings.

The exchange also earns from selling market data generated on its platform - like real-time data, historical data, summary data, and reference data – which is vital for equity research and other uses. Many exchanges will also sell technology products, like a trading terminal and dedicated network connection to the exchange, to the interested parties for a suitable fee.

The exchange may offer privileged services like high-frequency trading to larger clients like mutual funds and asset management companies (AMC), and earn money accordingly. There are provisions for regulatory fee and registration fee for different profiles of market participants, like the market maker and broker, which form other sources of income for the stock exchanges.

The exchange also makes profits by licensing their indexes (and their methodology) which are commonly used as a benchmark for launching various products like mutual funds and ETFs by AMCs. Many exchanges also provide courses and certification on various financial topics to industry participants and earn revenues from such subscriptions.

6.7.1. Competition for Stock Markets:

While individual stock exchanges compete against each other to get maximum transaction volume, they are facing threat on two fronts.

Dark Pools:

Dark pools, which are private exchanges or forums for securities trading and operate within private groups, are posing a challenge to public stock markets. Though their legal validity is subject to local regulations, they are gaining popularity as participants save big on transaction fees.

Blockchain Ventures:

Amid rising popularity of blockchains, many crypto exchanges have emerged. Such exchanges are venues for trading cryptocurrencies and derivatives associated with that asset class.

Though their popularity remains limited, they pose a threat to the traditional stock market model by automating a bulk of the work done by various stock market participants and by offering zero- to low-cost services.

6.7.2. Significance of the Stock Market:

The stock market is one of the most vital components of a free-market economy. It allows companies to raise money by offering stock shares and corporate bonds. It lets common investors participate in the financial achievements of the companies, make profits through capital gains, and earn money through dividends, although losses are also possible. While institutional investors and professional money managers do enjoy some privileges owing to their deep pockets, better knowledge and higher risk-taking abilities, the stock market attempts to offer a level playing field to common individuals. The stock market works as a platform through which savings and investments of individuals are channelized into the productive

investment proposals. In the long term, it helps in capital formation & economic growth for the country.

6.7.3. Examples of Stock Markets:

The first stock market in the world was the London stock exchange. It was started in a coffeehouse, where traders used to meet to exchange shares, in 1773. The first stock exchange in the United States of America was started in Philadelphia in 1790. The Buttonwood agreement, so named because it was signed under a buttonwood tree, marked the beginnings of New York's Wall Street in 1792.

CHAPTER-7
Existing System

7.1. EXISTING SYSTEM

Machine learning:

With the advent of the digital computer, stock market prediction has since moved into the technological realm. The most prominent technique involves the use of artificial neural networks (ANNs) and Genetic Algorithms (GA). Scholars found that the bacterial chemotaxis optimization method may perform better than GA. ANNs can be thought of as mathematical function approximators. The most common form of ANN in use for stock market prediction is the feed forward network utilizing the backward propagation of errors algorithm to update the network weights. These networks are commonly referred to as Backpropagation networks. Another form of ANN that is more appropriate for stock prediction is the time recurrent neural network (RNN) or time delay neural network (TDNN).

Examples of RNN and TDNN are the Elman, Jordan, and Elman-Jordan networks. (See the Elman and Jordan Networks). For stock prediction with ANNs, there are usually two approaches taken for forecasting different time horizons: independent and joint. The independent approach employs a single ANN for each time horizon, for example, 1-day, 2-day, or 5-day. The advantage of this approach is that network forecasting error for one horizon won't impact the error for another horizon—since each time horizon is typically a unique problem.

The joint approach, however, incorporates multiple time horizons together so that they are determined simultaneously. In this approach, forecasting error for one time horizon may share its error with that of another horizon, which can decrease performance. There are also more parameters required for a joint model, which increases the risk of overfitting. Of late, the majority of academic research groups studying ANNs for stock forecasting seem to be using an ensemble of independent ANNs methods more frequently, with greater success. An ensemble of ANNs would use low price and time lags to predict future lows, while another network would use lagged highs to predict future highs.

The predicted low and high predictions are then used to form stop prices for buying or selling. Outputs from the individual "low" and "high" networks can also be input into a final network that would also incorporate volume, intermarket data or statistical summaries of prices, leading to a final ensemble output that would trigger buying, selling, or market directional change. A major finding with ANNs and stock prediction is that a classification approach (vs. function approximation) using outputs in the form of buy($y=+1$) and sell($y=-1$) results in better predictive reliability than a quantitative output such as low or high price.

Since NNs require training and can have a large parameter space; it is useful to optimize the network for optimal predictive ability.

7.2. Data sources for market prediction:

Tobias Preiss et al. introduced a method to identify online precursors for stock market moves, using trading strategies based on search volume data provided by Google Trends. Their analysis of Google search volume for 98 terms of varying financial relevance, published in Scientific Reports, suggests that increases in search volume for financially relevant search terms tend to precede large losses in financial markets. Out of these terms, three were significant at the 5% level ($|z| > 1.96$). The best term in the negative direction was "debt", followed by "colour".

In a study published in Scientific Reports in 2013, Helen Susannah Moat, Tobias Preiss and colleagues demonstrated a link between changes in the number of views of English Wikipedia articles relating to financial topics and subsequent large stock market moves.

The use of Text Mining together with Machine Learning algorithms received more attention in the last years, with the use of textual content from the Internet as input to predict price changes in Stocks, Stocks and other financial markets. The collective mood of Twitter messages has been linked to stock market performance.

The study, however, has been criticized for its methodology. The activity in stock message boards has been mined in order to predict asset returns. The enterprise headlines from Yahoo! Finance and Google Finance were used as news feeding in a Text mining process, to forecast the Stock's price movements from the Dow Jones Industrial Average.

7.3. Market mimicry:

Using new statistical analysis tools of complexity theory, researchers at the New England Complex Systems Institute (NECSI) performed research on predicting stock market crashes. It has long been thought that market crashes are triggered by panics that may or may not be justified by external news. This research indicates that it is the internal structure of the market, not external crises, which is primarily responsible for crashes.

The number of different stocks that move up or down together were shown to be an indicator of the mimicry within the market, how much investors look to one another for cues. When the mimicry is high, many stocks follow each other's movements - a prime reason for panic to take hold. It was shown that a dramatic increase in market mimicry occurred during the entire year before each market crash of the past 25 years, including the financial crisis of 2007–08.

7.4. Time series aspect structuring:

Aspect structuring, also referred to as Jacaric Aspect Structuring (JAS) is a trend forecasting method which has been shown to be valid for anticipating trend changes on various stock market and geopolitical time series datasets. The method addresses the challenge that arises with high dimensional data in which exogenous variables are too numerous or immeasurable to be accounted for and used to make a forecast.

The method identifies the single variable of primary influence on the time series, or "primary factor", and observes trend changes that occur during times of decreased significance in the said primary variable. Presumably, trend changes in these instances are instead due to so-called "background factors". Although this method cannot elucidate the multivariate nature of background factors, it can gauge the effects they have on the time-series at a given point in time even without measuring them. This observation can be used to make a forecast.

CHAPTER-8
Existing System

8.1. PROPOSED SYSTEM

8.1.1. Downloading the Data:

We will be using data from the following sources:

1. Alpha Vantage. Before you start, however, you will first need an API key, which we can obtain for free here. After that, we can assign that key to the `spikey` variable.

2. Use the data from this page. we will need to copy the Stocks folder in the zip file to your project home folder. Stock prices come in several different flavours. They are,

- Open: Opening stock price of the day
- Close: Closing stock price of the day
- High: Highest stock price of the data
- Low: Lowest stock price of the day

Getting Data from Alpha vantage

We will first load in the data from Alpha Vantage. Since you're going to make use of the American Airlines Stock market prices to make your predictions, you set the ticker to "AAL". Additionally, you also define a URL string, which will return a JSON file with all the stock market data for American Airlines within the last 20 years, and a `Filatovs`'ve, which will be the file to which you save the data.

You'll use the ticker variable that you defined beforehand to help name this file. Next, you're going to specify a condition: if you haven't already saved data, you will go ahead and grab the data from the URL that you set in URL string; You'll store the date, low, high, volume, close, open values to a pandas Data Frame of and you'll save it to `Filatovs`'ve. However, if the data is already there, you'll just load it from the CSV.

Getting Data from Kaggle Data found on Kaggle is a collection of csv files and you don't have to do any pre-processing, so you can directly load the data into a Pandas data Frame.

Data Exploration Here you will print the data you collected in to the data Frame. You should also make sure that the data is sorted by date, because the order of the data is crucial in time series modelling.

8.2. Splitting Data into a Training set and a Test set:

You will use the mid-price calculated by taking the average of the highest and lowest recorded prices on a day. Now you can split the training data and test data. The training data will be the first 11,000 data points of the time series and rest will be test data.

Normalizing the Data Now you need to define a scaler to normalize the data. Minmax Scalar scales all the data to be in the region of 0 and 1. You can also reshape the training and test data to be in the shape [data size, numerators].

Due to the observation, you made earlier, that is, different time periods of data have different value ranges, you normalize the data by splitting the full series into windows. If you don't do this, the earlier data will be close to 0 and will not add much value to the learning process. Here you choose a window size of 2500.

You can now smooth the data using the exponential moving average. This helps you to get rid of the inherent raggedness of the data in stock prices and produce a smoother curve.

- Note that you should only smooth training data.
- One-Step Ahead Prediction via Averaging

Averaging mechanisms allow you to predict (often one time step ahead) by representing the future stock price as an average of the previously observed stock prices. Doing this for more than one time step can produce quite bad results. You will look at two averaging techniques below; standard averaging and exponential moving average. You will evaluate both qualitatively (visual inspection) and quantitatively (Mean Squared Error) the results produced by the two algorithms. The Mean Squared Error (MSE) can be calculated by taking the Squared Error between the true value at one step ahead and the predicted value and averaging it over all the predictions.

8.3. Standard Average:

You can understand the difficulty of this problem by first trying to model this as an average calculation problem. First you will try to predict the future stock market prices (for example, x_{t+1}) as an average of the previously observed stock market prices within a fixed size window (for example, a_{t-N}, \dots, x_t .) (Say previous 100 days). Thereafter you will try a bit fancier "exponential moving average" method and see how well that does. Then you will move on to the "holy-grail" of time-series prediction; Long Short-Term Memory models.

In other words, you say the prediction at $t+1$ is the average value of all the stock prices you observed within a window of t to $t-N$.

It seems that it is not too bad of a model for very short predictions (one day ahead). Given that stock prices don't change from 0 to 100 overnight, this behaviour is sensible. Next, you will look at a fancier averaging technique known as exponential moving average.

8.4. Exponential Moving Average:

You might have seen some articles on the internet using very complex models and predicting almost the exact behaviour of the stock market. But beware! These are just optical illusions and not due to learning something useful.

You will see below how you can replicate that behaviour with a simple averaging method. In the exponential moving average method, you calculate x_{t+1} as, $x_{t+1} = EMA_t = \gamma \times EMA_{t-1} + (1-\gamma) x_t$ at where $EMA_0 = 0$ and EMA is the exponential moving average value you maintain over time.

The above equation basically calculates the exponential moving average from $t+1$ time step and uses that as the one step ahead prediction. γ decides what the contribution of the most recent prediction is to the EMA. For example, a $\gamma=0.1$ gets only 10% of the current value into the EMA. Because you take only a very small fraction of the most recent, it allows to preserve much older values you saw very early in the average.

If Exponential Moving Average is this Good, why do You Need Better Models? You see that it fits a perfect line that follows the True distribution (and justified by the very low MSE). Practically speaking, you can't do much with just the stock market value of the next day. Personally, what I'd like is not the exact stock market price for the next day, but would the stock market prices go up or down in the next 30 days. Try to do this, and you will expose the incapability of the EMA method. You will now try to make predictions in windows (say you predict the next 2 days window, instead of just the next day). Then you will realize how wrong EMA can go.

Here is an example:

Predict More Than One Step into the Future

To make things concrete, let's assume values, say $x_t=0.4$, $EMA_t=0.5$ and $\gamma=0.5$

- Say you get the output with the following equation $x_{t+1} = EMA_t = \gamma \times EMA_{t-1} + (1 - \gamma) x_t$
- So you have $x_{t+1} = 0.5 \times 0.5 + (1 - 0.5) \times 0.4 = 0.45$
- So $x_{t+1} = EMA_t = 0.45$

One solution you have that will output useful information is to look at momentum-based algorithms. They make predictions based on whether the past recent values were going up or going down (not the exact values). For example, they will say the next day price is likely to be lower, if the prices have been dropping for the past days, which sounds reasonable. However, you will use a more.

Complex Model:

An LSTM models. These models have taken the realm of time series prediction by storm, because they are so good at modelling time series data. You will see if there actually are patterns hidden in the data that you can exploit.

Introduction to LSTMs:

Making Stock Movement Predictions Far into the Future Long Short-Term Memory models are extremely powerful time-series models. They can predict an arbitrary number of steps into the future. An LSTM module (or cell) has 5 essential components which allows it to model both long-term and short-term data.

- **Cell state (ct)** - This represents the internal memory of the cell which stores both short term memory and long-term memories Hidden state (ht) - This is output state information calculated w.r.t. current input, previous hidden state and current cell input which you eventually use to predict the future stock market prices. Additionally, the hidden state can decide to only retrieve the short or long-term or both types of memory stored in the cell state to make the next prediction.
- **Input gate (it)** - Decides how much information from current input flows to the cell state
- **Forget gate (ft)** - Decides how much information from the current input and the previous cell state flows into the current cell state
- **Output gate (ot)** - Decides how much information from the current cell state flows into the hidden state, so that if needed LSTM can only pick the long-term memories or short-term memories and long-term memories and the equations for calculating each of these entities are as follows.
- $i_t = \sigma(W\{ix\}x_t + W\{ih\}h_{t-1} + b_i)$
- $\tilde{c}_t = \sigma(W\{cx\}x_t + W\{ch\}h_{t-1} + b_c)$
- $f_t = \sigma(W\{fx\}x_t + W\{fh\}h_{t-1} + b_f)$
- $c_t = f_t c_{t-1} + i_t \tilde{c}_t$
- $o_t = \sigma(W\{ox\}x_t + W\{oh\}h_{t-1} + b_o)$
- $h_t = \tanh(ct)$

For a better (more technical) understanding about LSTMs you can refer to this article. TensorFlow provides a nice sub-API (called RNN API) for implementing time series models. You will be using that for your implementations.

8.5. Data Generator:

You are first going to implement a data generator to train your model. This data generator will have a method called `unroll_batches(...)` which will output a set of `num_unrollings` batches of input data obtained sequentially, where a batch of data is of size `[batch size, 1]`. Then each batch of input data will have a corresponding output batch of data. For example, if `num_unrollings=3` and `batch size=4` a set of unrolled batches it might look like,

- Input data: $[x_0, x_{10}, x_{20}, x_{30}], [x_1, x_{11}, x_{21}, x_{31}], [x_2, x_{12}, x_{22}, x_{32}]$ $[x_0, x_{10}, x_{20}, x_{30}], [x_1, x_{11}, x_{21}, x_{31}], [x_2, x_{12}, x_{22}, x_{32}]$
- Output data: $[x_1, x_{11}, x_{21}, x_{31}], [x_2, x_{12}, x_{22}, x_{32}], [x_3, x_{13}, x_{23}, x_{33}]$ $[x_1, x_{11}, x_{21}, x_{31}], [x_2, x_{12}, x_{22}, x_{32}], [x_3, x_{13}, x_{23}, x_{33}]$

8.6. Data Augmentation:

Also, to make your model robust you will not make the output for x_{t+1} . Rather you will randomly sample an output from the set.

$x_{t+1}, x_{t+2}, \dots, x_{t+N}$ where N is a small window size.

CHAPTER-9
Existing System

9.1. CODE:

Make sure that you have all these libraries available torn the code successfully from pandas' data reader import data import matplotlib. Request import or import NumPy as np import tensor flow as ft.

```
# This code has been tested with TensorFlow 1.6 from sclera. pre-processing
import Memescape data source='Kaggle'# alpha vantage or Kaggle if data
source=='alpha vantage' plot as plot import as pd import datetime as dt import
rule. # ===== Loading Data from Alpha Vantage
=====
```

```
url_string=
```

```
"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symb
ol=%s&outputsize=full&apikey=%s"
```

```
s" %tkinter, api_key # Save data to
this file_to_save=
'stock_market_data-%s.csv %ticker
```

```
# If you haven't already saved data,
```

```
# Go ahead and grab the data from the url
```

```
# And store date, low, high, volume, close, open values to a
Pandas Data Frame if not os.path.exists(file_to_save): with
urllib.request.urlopen(url_string) as url:
```

```
data = json.loads(url.read().decode()) # extract
stock market data data = data['Time Series
(Daily)'] df =
pd.DataFrame(columns=['Date','Low','High','Cl
ose','Open']) for k,v in data.items():
```

```
date = dt.datetime.strptime(k, '%Y-%m-%d')
```

```
data_row = [date.date(),float(v['3. low']),
float (v ['2. high']), float (v ['4. close']),
float(v['1. open'])] df.loc [-1, :] = data_row
df.index = df.index + 1 print('Data saved to
: %s'%file_to_save)
df.to_csv(file_to_save)
```

```
# If the data is already there, just load it
from the CSV else:
```

```

print('File already exists. Loading
data from CSV') df =
pd.read_csv(file_to_save) else:

# ===== Loading Data from Kaggle
===== # You will be using HP's
data. Feel free to experiment with other data.

# But while doing so, be careful to have a large enough dataset and also pay
attention to the data normalization

df =

pd.read_csv(os.path.join('Stocks','hpq.us.txt'),delimiter=',',usecols=['Date','Open
','High','Low','Close']) print('Loaded data from the Kaggle repository')

# Sort DataFrame by date df =
df.sort_values('Date') # Double check the result
df.head() plt.figure(figsize = (18,9))
plt.plot(range(df.shape[0]),(df['Low']+df['High'])/2
.0)
plt.xticks(range(0,df.shape[0],500),df['Date'].loc[:,
500],rotation=45) plt.xlabel('Date',fontsize=18)
plt.ylabel('Mid Price',fontsize=18) plt.show()

# First calculate the mid prices from the
highest and lowest high_prices =
df.loc[:, 'High'].as_matrix() low_prices =
df.loc[:, 'Low'].as_matrix() mid_prices =
(high_prices+low_prices)/2.0 train_data =
mid_prices[:11000] test_data =
mid_prices[11000:]

# Scale the data to be between 0 and 1

# When scaling remembers! You normalize both test and train data with respect
to training data

# Because you are not supposed to have
access to test data scaler = MinMaxScaler()
train_data = train_data.reshape(-1,1)
test_data = test_data.reshape(-1,1)

# Train the Scaler with training data
and smooth data

```

```

smoothing_window_size = 2500 for di
in
range(0,10000,smoothing_window_size):
scaler.fit(train_data[di:di+smoothing_
window_size,:])
train_data[di:di+smoothing_window_size,:] =
scaler.transform(train_data[di:di+smoothing_window_size,:])

# You normalize the last bit of remaining data
scaler.fit(train_data[di+smoothing_window_size:,:])
train_data[di+smoothing_window_size:,:] =
scaler.transform(train_data[di+smoothing_window_size:,:])

# Reshape both train and test
data train_data =
train_data.reshape(-1)

# Normalize test data test_data =
scaler.transform(test_data).reshape(-
1)

# Now perform exponential moving average smoothing

# So the data will have a smoother curve than the
original ragged data EMA = 0.0 gamma = 0.1 for ti
in range(1 1000):

EMA = gamma*train_data[ti] + (1-
gamma)*EMA train_data[ti] = EMA

# Used for visualization and test purposes
all_mid_data =
np.concatenate([train_data,test_data],axis=0)
window_size = 100 N = train_data.size
std_avg_predictions = [] std_avg_x = []
mse_errors = [] for pred_idx in
range(window_size,N): if pred_idx >= N:

date = dt.datetime.strptime(k, '% Y-%m-%d').date() +
dt.timedelta(days=1) else:

```

```

date = df.loc[pred_idx,'Date']
std_avg_predictions.append(np.mean(train_data[pred_idx-
window_size:pred_idx])) mse_errors.append((std_avg_predictions[-1]-
train_data[pred_idx])**2) std_avg_x.append(date) print('MSE error for
standard averaging: %.5f'%(0.5*np.mean(mse_errors))) plt.figure(figsize =
(18,9)) plt.plot(range(df.shape[0]),all_mid_data,color='b',label='True')
plt.plot(range(window_size,N),std_avg_predictions,color='orange',label='Predi
ction')

```

```

#plt.xticks(range(0,df.shape[0],50),df['Date'].loc[:
:50],rotation=45) plt.xlabel('Date') plt.ylabel('Mid
Price') plt.legend(fontsize=18) plt.show()
window_size = 100 N = train_data.size
run_avg_predictions = [] run_avg_x = []
mse_errors = [] running_mean = 0.0
run_avg_predictions.append(running_mean)
decay = 0.5 for pred_idx in range(1,N):
    running_mean = running_mean*decay + (1.0-
decay)*train_data[pred_idx-1]
    run_avg_predictions.append(running_mean)
    mse_errors.append((run_avg_predictions[-1]-
train_data[pred_idx])**2) run_avg_x.append(date)
    print('MSE error for EMA averaging:
%.5f'%(0.5*np.mean(mse_errors))) plt.figure(figsize =
(18,9))
    plt.plot(range(df.shape[0]),all_mid_data,color='b',label='
True')
    plt.plot(range(0,N),run_avg_predictions,color='orange',
label='Prediction')

```

```

#plt.xticks(range(0,df.shape[0],50),df['Date'].loc[:,50],rotation=45)

plt.xlabel('Date') plt.ylabel('Mid Price')
plt.legend(fontsize=18) plt.show() class
DataGeneratorSeq(object): def
__init__(self,prices,batch_size,num_unroll): self._prices
= prices self._prices_length = len(self._prices) -
num_unroll self._batch_size = batch_size
self._num_unroll = num_unroll self._segments =
self._prices_length //self._batch_size self._cursor =

```

```

[offset * self._segments for offset in
range(self._batch_size)] def next_batch(self):

batch_data =
np.zeros((self._batch_size),dtype=np.float32)
batch_labels =
np.zeros((self._batch_size),dtype=np.float32) for b
in range(self._batch_size): if
self._cursor[b]+1>=self._prices_length:
#self._cursor[b] = b * self._segments self._cursor[b]
= np.random.randint(0,(b+1)*self._segments)
batch_data[b] = self._prices[self._cursor[b]]
batch_labels[b]=
self._prices[self._cursor[b]+np.random.randint(0,5)
] self._cursor[b] =
(self._cursor[b]+1)%self._prices_length return
batch_data,batch_labels def unroll_batches(self):

unroll_data,unroll_labels
= [],[] init_data, init_label
= None,None for ui in
range(self._num_unroll):
data, labels =
self.next_batch()
unroll_data.append(data)
unroll_labels.append(labe
ls) return unroll_data,
unroll_labels def
reset_indices(self): for b
in
range(self._batch_size):

self._cursor[b] =
np.random.randint(0,min((b+1)*self._segments,self._prices_lengt
h-1)) dg = DataGeneratorSeq(train_data,5,5) u_data, u_labels =
dg.unroll_batches() for ui,(dat,lbl) in
enumerate(zip(u_data,u_labels)):

print('\n\nUnrolled
index %d'%ui) dat_ind
= dat lbl_ind = lbl

```

```
print('\tInputs: ',dat )
print('\n\tOutput:',lbl)
```

D = 1 # Dimensionality of the data. Since your data is 1-D this would be 1
num_unrollings = 50 # Number of time steps you look into the future.

```
batch_size = 500 # Number of samples in a batch num_nodes = [200,200,150]
# Number of hidden nodes in each layer of the deep LSTM stack we're using
n_layers = len(num_nodes) # number of layers dropout = 0.2 # dropout amount
tf.reset_default_graph() # This is important in case you run this multiple times
```

```
# Input data.
```

```
train_inputs,
train_outputs = [],[]
```

```
# You unroll the input over time defining placeholders
for each time step for ui in range(num_unrollings):
```

```
train_inputs.append(tf.placeholder(tf.float32,
shape=[batch_size,D],name='train_inputs_%d'%ui))
train_outputs.append(tf.placeholder(tf.float32, shape=[batch_size,1], name =
'train_outputs_%d'%u)
lstm_cells = [
tf.contrib.rnn.LSTMCell(num_units=num_nodes[li], state_is_tuple=True,
initializer= tf.contrib.layers.xavier_initializer()
)
```

```
for li in range(n_layers)] drop_lstm_cells =
[tf.contrib.rnn.DropoutWrapper( lstm,
input_keep_prob=1.0,output_keep_prob=1.0-dropout,
state_keep_prob=1.0-dropout
```

```
) for lstm in lstm_cells] drop_multi_cell =
tf.contrib.rnn.MultiRNNCell(drop_lstm_cells) multi_cell =
tf.contrib.rnn.MultiRNNCell(lstm_cells) w =
tf.get_variable('w',shape=[num_nodes[-1], 1],
initializer=tf.contrib.layers.xavier_initializer()) b =
tf.get_variable('b',initializer=tf.random_uniform([1],-0.1,0.1))
```

```
# Create cell state and hidden state variables to maintain the state of the LSTM
```

```
c, h = [],[]
initial_state = []
for li in
```

```

range(n_layers)
:
c.append(tf.Variable(tf.zeros([batch_size, num_nodes[li]]), trainable=False))
h.append(tf.Variable(tf.zeros([batch_size, num_nodes[li]]), trainable=False))
initial_state.append(tf.contrib.rnn.LSTMStateTuple(c[li], h[li]))

# Do several tensor transformations, because the function dynamic_rnn
requires the output to be of # a specific format. Read more at:
https://www.tensorflow.org/api\_docs/python/tf/nn/dynamic\_rnn
all_inputs =
tf.concat([tf.expand_dims(t,0) for t in train_inputs],axis=0)

# all_outputs is [seq_length, batch_size, num_nodes]
all_lstm_outputs,
state = tf.nn.dynamic_rnn( drop_multi_cell, all_inputs,
initial_state=tuple(initial_state), time_major = True, dtype=tf.float32)
all_lstm_outputs = tf.reshape(all_lstm_outputs,
[batch_size*num_unrollings,num_nodes[-1]])
all_outputs =
tf.nn.xw_plus_b(all_lstm_outputs,w,b)
split_outputs =
tf.split(all_outputs,num_unrollings,axis=0)

# When calculating the loss you need to be careful about the exact form,
because you calculate

# loss of all the unrolled steps at the same time

# Therefore, take the mean error or each batch and get the sum of that over
all the unrolled steps
print('Defining training Loss')
loss = 0.0
with tf.control_dependencies([tf.assign(c[li], state[li][0]) for li in
range(n_layers)]+
[tf.assign(h[li], state[li][1]) for li in
range(n_layers)]):
for ui in range(num_unrollings):
loss += tf.reduce_mean(0.5*(split_outputs[ui]-
train_outputs[ui])**2)
print('Learning rate decay
operations')
global_step = tf.Variable(0,
trainable=False)
inc_gstep =
tf.assign(global_step,global_step + 1)
tf_learning_rate =
tf.placeholder(shape=None,dtype=tf.float32)
tf_min_learning_rate =
tf.placeholder(shape=None,dtype=tf.float32)
learning_rate = tf.maximum(

```

```

tf.train.exponential_decay(tf_learning_rate, global_step,
decay_steps=1, decay_rate=0.5, staircase=True),
tf_min_learning_rate)

# Optimizer. print('TF Optimization
operations') optimizer =
tf.train.AdamOptimizer(learning_rate)
gradients, v =
zip(*optimizer.compute_gradients(loss))
gradients, _ =
tf.clip_by_global_norm(gradients, 5.0)
optimizer = optimizer.apply_gradients(
zip(gradients, v)) print('\tAll done')
print('Defining prediction related TF
functions') sample_inputs =
tf.placeholder(tf.float32, shape=[1,D]) #
Maintaining LSTM state for prediction
stage sample_c, sample_h,
initial_sample_state = [],[],[] for li in
range(n_layers):
sample_c.append(tf.Variable(tf.zeros([1,
num_nodes[li]]), trainable=False))
sample_h.append(tf.Variable(tf.zeros([1,
num_nodes[li]]), trainable=False))
initial_sample_state.append(tf.contrib.rnn
n.LSTMStateTuple(sample_c[li],sample
_h[li]))

reset_sample_states = tf.group(*[tf.assign(sample_c[li],tf.zeros([1,
num_nodes[li]])) for li in range(n_layers)],

*[tf.assign(sample_h[li],tf.zeros([1, num_nodes[li]])) for li in
range(n_layers)]) sample_outputs, sample_state =
tf.nn.dynamic_rnn(multi_cell, tf.expand_dims(sample_inputs,0),
initial_state=tuple(initial_sample_state), time_major = True,
dtype=tf.float32) with
tf.control_dependencies([tf.assign(sample_c[li],sample_state[li][0]) for li
in range(n_layers)]+
[tf.assign(sample_h[li],sample_state[li][1]) for li in range(n_layers)]):

```

```

sample_prediction =
tf.nn.xw_plus_b(tf.reshape(sample_outputs,[1,-1]), w, b)
print('\tAll done')

epochs = 30 valid_summary = 1 # Interval you
make test predictions n_predict_once = 50 #
Number of steps you continuously predict for
train_seq_length = train_data.size # Full length of
the training data train_mse_ot = [] # Accumulate
Train losses test_mse_ot = [] # Accumulate Test
loss predictions_over_time = [] # Accumulate
predictions session = tf.InteractiveSession()
tf.global_variables_initializer().run() # Used for
decaying learning rate loss_nondecrease_count =
0

loss_nondecrease_threshold = 2 # If the test error hasn't increased in this many
steps, decrease learning rate print('Initialized') average_loss = 0

# Define data generator data_gen =
DataGeneratorSeq(train_data,batch_size,num_unrol
lings) x_axis_seq = []

# Points you start your test predictions
from test_points_seq =
np.arange(11000,12000,50).tolist() for ep
in range(epochs):

# ===== Training ===== for
step in range(train_seq_length//batch_size): u_data,
u_labels = data_gen.unroll_batches() feed_dict = {} for
ui,(dat,lbl) in enumerate(zip(u_data,u_labels)):
feed_dict[train_inputs[ui]] = dat.reshape(-1,1)
feed_dict[train_outputs[ui]] = lbl.reshape(-1,1)
feed_dict.update({'tf_learning_rate': 0.0001,
tf_min_learning_rate:0.000001})

_, l = session.run([optimizer, loss],
feed_dict=feed_dict) average_loss += l

# ===== Validation
===== if (ep+1) % valid_summary == 0:

average_loss = average_loss/(valid_summary*(train_seq_length//batch_size))

```

```

# The average loss if
(ep+1)%valid_sum
mary==0:

print('Average loss at step %d: %f' %
(ep+1, average_loss))
train_mse_ot.append(average_loss)
average_loss = 0 # reset loss
predictions_seq = [] mse_test_loss_seq = []

# ===== Updating State and Making Predicitons =====
for w_i in test_points_seq:
mse_test_loss = 0.0

our_predictions = []
if (ep+1)-
valid_summary==0
:

# Only calculate x_axis values in the first
validation epoch x_axis=[]

# Feed in the recent past behavior of
stock prices # to make predictions
from that point onwards for tr_i in
range(w_i-num_unrollings+1,w_i-
1):

current_price = all_mid_data[tr_i]
feed_dict[sample_inputs] =
np.array(current_price).reshape(1,1) _ =
session.run(sample_prediction,feed_dict=feed_d
ict) feed_dict = { } current_price =
all_mid_data[w_i-1] feed_dict[sample_inputs] =
np.array(current_price).reshape(1,1)

# Make predictions for this many steps

# Each prediction uses previous prediciton as it's
current input for pred_i in
range(n_predict_once):

pred =

session.run(sample_prediction,feed_dict=fe

```

```

ed_dict)

our_predictions.append(np.asscalar(pred))

feed_dict[sample_inputs] =
np.asarray(pred).reshape(-1,1) if (ep+1)-
valid_summary==0:

# Only calculate x_axis values in the first
validation epoch
x_axis.append(w_i+pred_i) mse_test_loss
+= 0.5*(pred-
all_mid_data[w_i+pred_i])**2
session.run(reset_sample_states)
predictions_seq.append(np.array(our_predi
ctions)) mse_test_loss /= n_predict_once
mse_test_loss_seq.append(mse_test_loss)
if (ep+1)-valid_summary==0:
x_axis_seq.append(x_axis)

current_test_mse = np.mean(mse_test_loss_seq)

# Learning rate decay logic if
len(test_mse_ot)>0 and current_test_mse >
min(test_mse_ot):

loss_nondecrease_count += 1
else:

loss_nondecrease_count = 0 if
loss_nondecrease_count >
loss_nondecrease_threshold :

session.run(inc_gstep) loss_nondecrease_count = 0 print('\tDecreasing learning
rate by 0.5') test_mse_ot.append(current_test_mse) print('\tTest MSE:
%.5f'%np.mean(mse_test_loss_seq))
predictions_over_time.append(predictions_seq) print('\tFinished Predictions')
best_prediction_epoch = 28 # replace this with the epoch that you got the best
results when running the plotting code plt.figure(figsize = (18,18))
plt.subplot(2,1,1) plt.plot(range(df.shape[0]),all_mid_data,color='b')

# Plotting how the predictions change over time

```

```

# Plot older predictions with low alpha and newer predictions
with high alpha start_alpha = 0.25 alpha =
np.arange(start_alpha,1.1,(1.0-
start_alpha)/len(predictions_over_time[:,3])) for p_i,p in
enumerate(predictions_over_time[:,3]): for xval,yval in
zip(x_axis_seq,p):

plt.plot(xval,yval,color='r',alpha=alpha[p_i])
plt.title('Evolution of Test Predictions Over
Time',fontsize=18)
plt.xlabel('Date',fontsize=18) plt.ylabel('Mid
Price',fontsize=18)

plt.xlim(11000,12500)
plt.subplot(2,1,2)

# Predicting the best test prediction you got
plt.plot(range(df.shape[0]),all_mid_data,color='b') for
xval,yval in
zip(x_axis_seq,predictions_over_time[best_prediction_epo
ch]):

plt.plot(xval,yval,color='r')
plt.title('Best Test Predictions Over
Time',fontsize=18)
plt.xlabel('Date',fontsize=18)
plt.ylabel('Mid Price',fontsize=18)
plt.xlim(11000,12500) plt.show()

```

CHAPTER-10
Existing System

10.1. CONCLUSION:

In this, we learnt how difficult it can be to devise a model that is able to correctly predict stock price movements. We started with a motivation for why we need to model stock prices. This was followed by an explanation and code for downloading data. Then we looked at two averaging techniques that allow you to make predictions one step into the future. We next saw that these methods are futile when you need to predict more than one step into the future. Thereafter we discussed how you can use LSTMs to make predictions many steps into the future. Finally, we visualized the results and saw that your model (though not perfect) is quite good at correctly predicting stock price movements.

Here, I'm stating several takeaways.

1. Stock price/movement prediction is an extremely difficult task. Personally, I don't think any of the stock prediction models out there shouldn't be taken for granted and blindly rely on them. However, models might be able to predict stock price movement correctly most of the time, but not always.
2. Do not be fooled by articles out there that shows predictions curves that perfectly overlaps the true stock prices. This can be replicated with a simple averaging technique and in practice it's useless. A more sensible thing to do is predicting the stock price movements.
3. The model's hyperparameters are extremely sensitive to the results you obtain. So, a very good thing to do would be to run some hyperparameter optimization technique (for example, Grid search / Random
4. search) on the hyperparameters. Below I listed some of the most critical hyperparameters
 - o The learning rate of the optimizer
 - o Number of layers and the number of hidden units in each layer
 - o The optimizer. I found Adam to perform the best
 - o Type of the model. You can try GRU/ Standard LSTM/ LSTM with Peepholes and evaluation performance difference
5. In this we did something faulty (due to the small size of data)! That is we used the test loss to decay the learning rate. This indirectly leaks information about test set into the training procedure. A better way of handling this is to have a separate validation set (apart from the test set) and decay learning rate with respect to performance of the validation set.

10.2. REFERENCES:

- file:///C:/Users/Visitor/Desktop/Paper_109.pdf
- <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119197652.oth1>
- <http://thestockmarketwatch.com/learn/stocks-basicsconclusion/>
- <https://www.datacamp.com/community/tutorials/lstm-python-stock-market>
- <https://link.springer.com/content/pdf/bbm%3A978-81-322-1590-5%2F1.pdf>
- <https://www.scribd.com/document/110661247/Conclusion>
- <https://www.pantechsolutions.net/stock-market-prediction-using-machinelearning>
- <https://www.ijeat.org/wp-content/uploads/papers/v8i4/D6321048419.pdf>
- <https://markdunne.github.io/public/mark-dunne-stock-market-prediction.pdf>
- <https://www.investopedia.com/terms/s/stock-analysis.asp>