**Stock Market Analysis and Prediction**

A Minor Project Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY   
in   
COMPUTER SCIENCE AND ENGINEERING**

BY

**Yuvraj Yadav (EN16CS301300)**

**Utkarsh Jain (EN16CS301288)**

**Mangalam Tripathi(EN17CS3L1002)**

Under the Guidance of

**Pritesh Saklecha**



**Department of Computer Science and Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**NOVEMBER 2019**

**Report Approval**

The project work **“Stock Market Analysis and Prediction”** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name: Sachin Solanki

Designation: Project Coordinator

Affiliation:

External Examiner

Name:

Designation:

Affiliation:

**Declaration**

I/We hereby declare that the project entitled **“Stock Market Analysis and Prediction”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications in ‘Computer Science and Engineering Department’ completed under the supervision of Pritesh Saklecha, Assistant Professor in Computer Science and Engineering Department**,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Signature and name of the student(s) with date**

**Yuvraj Yadav Utkarsh Jain Mangalam Tripathi**

**Certificate**

I **Pritesh Saklecha** certify that the project entitled **“Stock Market Analysis and Prediction”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications by **Yuvraj Yadav, Utkarsh Jain and Mangalam Tripathi** is the record carried out by them under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pritesh Saklecha

Computer Science and Engineering Department

Medi-Caps University, Indore

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<Name of External Guide (If any)>

<Name of the Department>

Name of the Organization

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Suresh Jain

Head of the Department

Computer Science and Engineering

Medi-Caps University, Indore

**Acknowledgments**

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Sunil K Somani,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda,** Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Prof. (Dr.) Suresh Jain** for his/her continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide, Mr. Manoj Kumar**, Project Lead, Honeywell Technology Solutions Lab Pvt. Ltd as well as to my Internal Guide, Dr**. Debendra Kumar Panda,** Professor, Department of Electronics Engineering, MU, without whose continuous help and support, this project would ever have reached to the completion.

I would also like to thank to my team at Honeywell Mr. Kishore Sandrana, Mr .GauravVashisth, Mr. JayaprakashVendkadabathi, Ms. ShabnaVasudevan, Mr. Rajesh Kumar, Mr. GopiBadarla, Ms. Asha Kuriakose who extended their kind support and help towards the completion of this project.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

***Student may write as per their experience.***

**Yuvraj Yadav, Utkarsh Jain and Mangalam Tripathi**

B. Tech. IV Year

Department of Computer Science and Engineering

Faculty of Engineering

Medi-Caps University, Indore

**Abstract**

In this report we analyze existing and new methods of stock market prediction. We took two different approaches at the problem: Technical Analysis and the application of Machine Learning. We do provide a GUI based web application from where any user can get reports on any stock. We find evidence in support of the weak form of the Efficient Market Hypothesis, that the historic price does not contain useful information but out of sample data may be predictive. We show that Technical analysis and Machine Learning could be used to guide an investor’s decisions. Based on our findings, algorithmic trading programs are developed and a basic idea or decision of trading a stock can be strengthened.

In our project, Artificial neural networks and Machine learning algorithms are used to predict the stock moment.

**Keywords: Stock Market, Technical Analysis, Machine Learning, Web-Application, Artificial Neural Networks.**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |
|  | Report Approval | Ii |
|  | Declaration | Iii |
|  | Certificate | Iv |
|  | Acknowledgment | V |
|  | Abstract | Vi |
|  | Table of Contents | Vii |
|  | List of figures | Viii |
|  | List of tables | Ix |
|  | Abbreviations | X |
|  | Notations & Symbols | Xi |
| Chapter 1 | Introduction (Whichever is applicable) |  |
|  | 1.1 Introduction | 1 |
|  | 1.2 Objectives |  |
|  | 1.3 Significance |  |
|  | 1.4 Source of Data |  |
|  | 1.5 Problem in existing system and Justification |  |
|  | 1.6 Organization |  |
| Chapter 2 | System Requirement Analysis |  |
|  | 2.1 Information Gathering |  |
|  | 2.2 System Feasibility |  |
|  | 2.2.1 Economical |  |
|  | 2.2.2 Technical |  |
|  | 2.2.3 Behavioral |  |
|  | 2.3 Platform Specification (Development & Deployment) |  |
|  | 2.3.1 Hardware |  |
|  | 2.3.2 Software Implementation Language/ Technology |  |
| Chapter 3 | System Analysis |  |
|  | 3.1 Information Flow Representation |  |
|  | 3.1.1 Data Dictionary (if applicable) |  |
|  | 3.1.2 Data Flow |  |
|  | 3.1.3 Activity Diagram |  |
|  | 3.1.4 Use Cases/ |  |
|  | 3.1.5 Class Diagram |  |
| Chapter 4 | Design |  |
|  | 4.1Architectural Design |  |
|  | 4.1.1 Architectural Context Diagram |  |
|  | 4.1.2 Architectural Behavioral Diagram |  |
|  | 4.1.3 Description of Architectural Diagram |  |
|  | 4.1.4 Control Hierarchy |  |
|  | 4.2Procedural/Modular Approach |  |
|  | 4.2.1 Modules Used |  |
|  | 4.2.2 Internal Data Structures |  |
|  | 4.2.3 Algorithm Design for Operations |  |
|  | 4.3Data Design |  |
|  | 4.3.1 Data objects and Resultant Data Structures |  |
|  | 4.4Interface Design |  |
|  | 4.4.1 Human-Machine Interface Design Specification |  |
|  | 4.4.2 I/O Forms |  |
|  | 4.4.3 Reports |  |
| Chapter 5 | Testing |  |
|  | 5.1 Testing Objective |  |
|  | 5.2 Testing Scope |  |
|  | 5.3 Testing Principles |  |
|  | 5.4 Testing Methods Used |  |
|  | 5.5 Test Cases |  |
|  | 5.6 Sample Test Data & Results |  |
| Chapter 6 | Limitations |  |
| Chapter 7 | Future Scope |  |
| Chapter 8 | Conclusion |  |
| Chapter 9 | Bibliography and References |  |
| Chapter 10 | Appendices |  |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| **2.1** | **Deployment Diagram** |  |
|  |  |  |
|  |  |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
|  |  |  |

**Abbreviations**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
|  |  |

**Notations & Symbols**

|  |  |
| --- | --- |
| **Notation** | **Description** |
|  |  |

**Chapter-1 Introduction**

**1.1 Introduction**

Predicting the Stock Market has been the bane and goal of investors since its existence. Everyday billions of dollars are traded on the exchange, and behind each dollar is an investor hoping to profit in one way or another. Entire companies rise and fall daily based on the behaviour of the market. Should an investor be able to accurately predict market movements, it offers a tantalizing promises of wealth and influence. It is no wonder then that the Stock Market and its associated challenges find their way into the public imagination everytime it misbehaves. The 2008 financial crisis was no different, as evidenced by the flood of films and documentaries based on the crash. If there was a common theme among those productions, it was that few people knew how the market worked or reacted. Perhaps a better understanding of stock market prediction might help in the case of similar events in the future.

**1.2 Objectives**

Despite its prevalence, Stock Market prediction remains a secretive and empirical art. Few people, if any, are willing to share what successful strategies they have. A chief goal of this project is to add to the academic understanding of stock market prediction. The hope is that with a greater understanding of how the market moves, investors will be better equipped to prevent another financial crisis. The project will evaluate some existing strategies from a rigorous scientific perspective and provide a quantitative evaluation of new strategies.

Analysis and Predicting the Stock Market can be one of the biggest milestone that can be achieved using 3 different strategies of analyzing the stock market. Those three strategies are

* Fundamental Analysis
* Technical Analysis
* Application of Machine Learning

Main Objectives of this Project are as follows:

* To Provide a user friendly interface of stock analysis and prediction.
* Provide user current statistics and details of the stock.
* Creating a deep Analysis report of any stock.
* Searching for something new that can be used for analyzing and predicting future stock prices and plotting them on a graph.
* Predicting prices as much accurate as we can.

**1.3 Significance**

It is important here to define the scope of the project. Although vital to any investor operating in the real world, no attempt is made in this project at portfolio management. Portfolio management is largely an extra step done after an investor has made a prediction on which direction any particular stock will move. The investor may choose to allocate funds across a range of stocks in such a way to minimize his or her risk. For instance, the investor may choose not to invest all of their funds into a single company lest that company takes unexpected turn. A more common approach would be for an investor to invest across a broad range of stocks based on some criteria he has decided on before.

Main significance of our project is that it will help both investor and educators to look on to any stock. As our project generates two different reports of analysis and prediction, it makes easier for them to gather more knowledge and strengthen there hunches or ideas about any stock they wish to invest or teach upon.

**1.4 Source of data**

As stock data is one of the widely used data by both financial and educational sectors, it is widely available. But as financial data stock is collected using various sensor and programs or crawlers it is prone to errors. And good quality data costs a great deal of money that is out of our projects budget and hence we will be going for two data repositories one from Kaggle and another one from Alpha Vantage.

In this project we will be using data of US Stock Exchanges. Further explanation of data is provided below:

1. **US Stock Exchange data form Kaggle**: The full historical daily price and volume data for all US-based stocks and ETFs trading on the NYSE, NASDAQ, and NYSE MKT. It's one of the best data-sets of its kind you can obtain. The data (last updated 11/10/2017) is presented in CSV format as follows: Date, Open, High, Low, Close, Volume, OpenInt. Note that prices have been adjusted for dividends and splits.
2. **Alpha Vantage API:** Composed of a tight-knit community of researchers, engineers, and business professionals, Alpha Vantage Inc. is a leading provider of free APIs for realtime and historical data on stocks, forex (FX), and digital/crypto currencies.

**1.5 Problem in existing system and Justification:**

One of the greatest problem we encountered was in user friendliness. But there are another problems to that exists in previous works and some that aren’t a problem but needs to be solved. All the problems in existing system are as follows:

* Lack of user friendliness : In all the projects made prior to this one didn’t had any user interface.
* Lack of user interaction : Projects are made as a report and hence a user can’t change any thing in them directly. Either he/she must know programming to alter something or to interact with it.
* Analysis reports were not covering much: In some projects Analysis reports they generated were more on only on single or two/three stocks and were not providing deeper analysis.
* Trying to get better accuracy: It the most prominent problem that exist in every project and will be our project too because stock market is just unpredictable. And no one can actually predict by just writing a few lines of codes.

**Chapter-2 System Requirement Analysis**

Our project uses programming, scripting and styling languages. Several modules that are needed for proper execution of the project. As it does generate a web-based application, there will be a great cut in requirements thus the user can access our project by just using a web browser. But for executing our project by downloading it on a system various requirements must be full filled and they are as follows.

**2.1 Information Gathering**

Information is gathered from Internet using various blog post, websites, articles and repositories. Various blogs on stock market analysis such as Stock market technical analysis by MoneyControl. Websites such as Investopedia.com, Investing.com, Motilaloswal.in, etc,. Provides good techniques and tools for how to analyze any stock.

Various repositories like hirooku and github, provides practical implementations of the projects that are made on both the title, techniques, model and algorithms that are used in this project. Information on this models and algorithm is gathered from various websites like geeksforgeeks.com, tutorialspoint.com, there documentation pages, and python.pydata.com.

Various websites like analytics vidhya and medium’s machine learning page provide one of the best articles on using machine learning algorithms and deep learning models. Models such as LSTM, ARIMA and FBProphet are used in this project and hence a lot of information on there working and implementation can be found on this websites.

And at last Wikipedia.com the encyclopedia. Gives a lot of knowledge on everything that will be encountered in this project.

**2.2 System Feasibility**

A feasibility study is an assessment of the practicality of a proposed project or system. Feasibility of this project can be described in to various types of feasibility studies and are provided below:

2.2.1 Economical Feasibility:

In this project, the main constituent of the cost or economy that could have been is the financial data of various stocks. But as the data we used is freely available on Kaggle this gives this project a economic advantage.

Software’s and programs that are used in this project are all freeware. Jupyter notebook, python interpreter, pycharm community version and a web browser such as Firefox and Chrome are the main softwares that we will be using this project. And all of them are free to download.

Only costs that are applicable in this project are Internet charges and a computer system charges on which we will be executing the project. Therefore this project is economically feasible.

2.2.2 Technical Feasibility:

The main technology used in this project is Python. To some extent Flask, HTML and CSS are also used mainly in Front-end development that is in user interface. Project is built on Ubuntu Linux and hence works best on it.

Technical terms that are need to understand this project are Data Analysis as one part of the whole project is a Analysis report that provides stock data analysis, Machine Learning as all the algorithms used in the project for predicting the stock prices are done using that and Artificial Neural Networks(ANN) models that are used of increase the accuracy of project at predicting stock prices.

Man-Power that is needed in this software is 3 for developing and deployment and 1 for maintaining. On user side only one user is need with no prior knowledge of programming of any sort until and unless that user aim for changing the code written.

2.2.3 Behavioral Feasibility:

Project is inclusive of a user friendly web-application for taking the stock name , start and end dates for analysis. In this project, beautiful and interactive graph plots or visualization is done. As the reports generated are in the form of python notebooks, every cell in those reports have a description of what actually that cell’s function are and what work that cell is performing.

Beautiful and interactive interface and visualization puts this project at a great advantage. Though it increases quite a load on system.

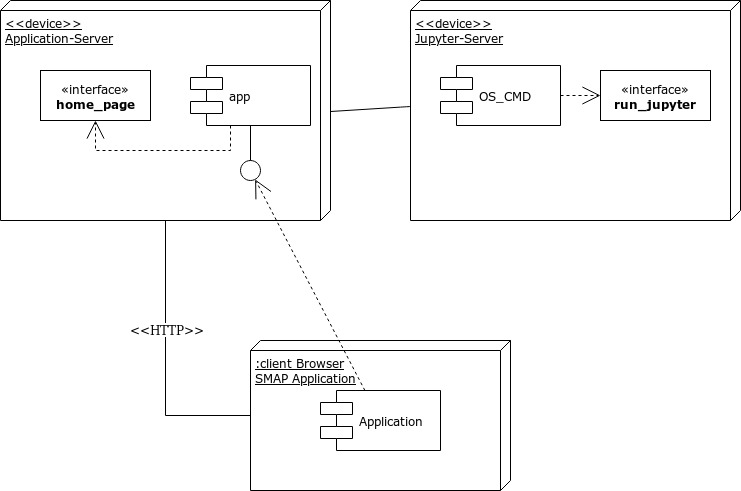
It takes quite a time to load as the data is large and processing it takes time. Two servers runs simultaneously and therefore increases the load on system and this are on the negative points that are encountered till the date.

**2.3 Platform Specification:**

Operating system used in development and testing is Ubuntu 18.04 LTS a linux distribution. Various tools, softwares and modules used in development process are:

* Python and its Packages like numpy, pandas, matplotlib, scikit-learn, keras, fbprophet, etc,.
* Jupyter notebooks
* Pycharm Community version IDE

For deployment, this project can be deployed on a web server and its functionalities can be accessed from a web browser. But for execution of this project on a system need python and its packages, jupyter notebook installed on that system.

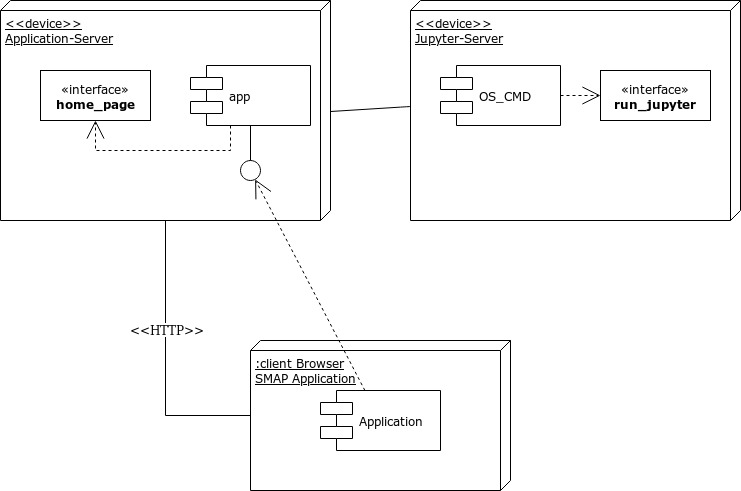
 2.1: Deployment Diagram

2.3.1 Hardware Requirements:

* + 2.5 Gz processor
  + 4 GB RAM (8GB- Recommended)
  + 4 GB HDD-Space
  + 1024x768 minimum screen resolution

2.3.2 Software Requirements:

* + Pycharm Community version
  + Jupyter Notebook
  + Python
  + Python packages: Numpy, Pandas, Keras, Scikit-learn, Matplotlib, Ggplot and Plotly
  + Web-Browser(ex: Mozilla Firefox, Chrome)
  + Operating System(ex: Linux, Windows 10 or Mac)

**Chapter-3 System Analysis**

System analysis of stock market analysis and prediction project gives a comprehensive result of how a each and every module is related or linked to each other. It also providing information of how actually data is flowing into and from the project. The main modules that are present in this project are app, stock\_analysis, stock\_prediction, file\_import and call\_api.

**3.1 Information Flow Representation**

In Information Flow Representation we will be using various UML diagrams such as dataflow diagram, ER diagram, control flow diagram, activity diagram, etc,. To provide a thorough explanation of internal working of the software. How analysis and prediction files work will be shown using below diagrams.

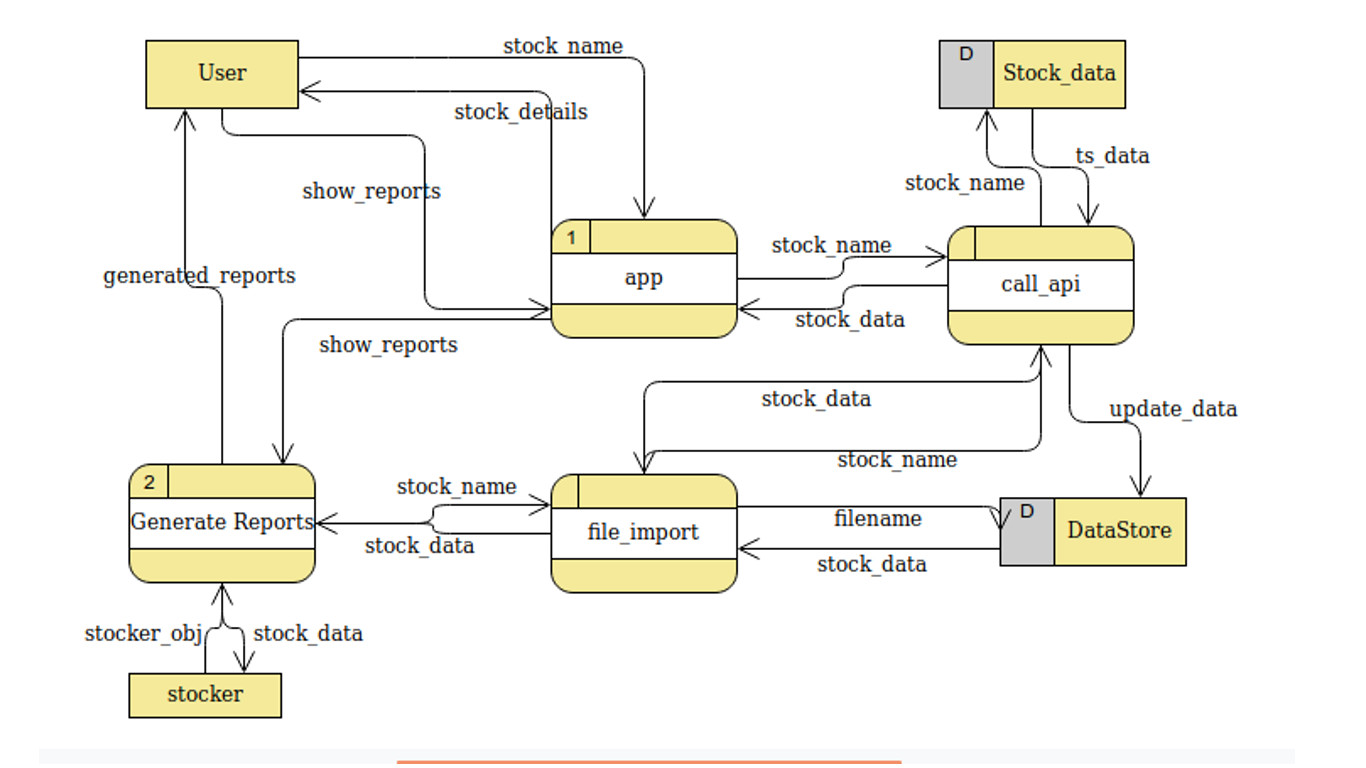
3.1.1 Data Dictionary:

All the data and there types are given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Description** |
| stock\_name | string | 4-5 | Symbolic name of stock is stored |
| start\_date | string | 10 | Start date used as initial date for analysis and prediction |
| end\_date | string | 10 | End date used as last date for analysis and prediction |
| data | dataframe | undefined | For storing stock data |
| plt | plot | 1 | For storing plots and |
| current\_stats | dictionary | 10 | For holding current stock data |
| title\_stats | dictionary | 10 | For holding details about stock |

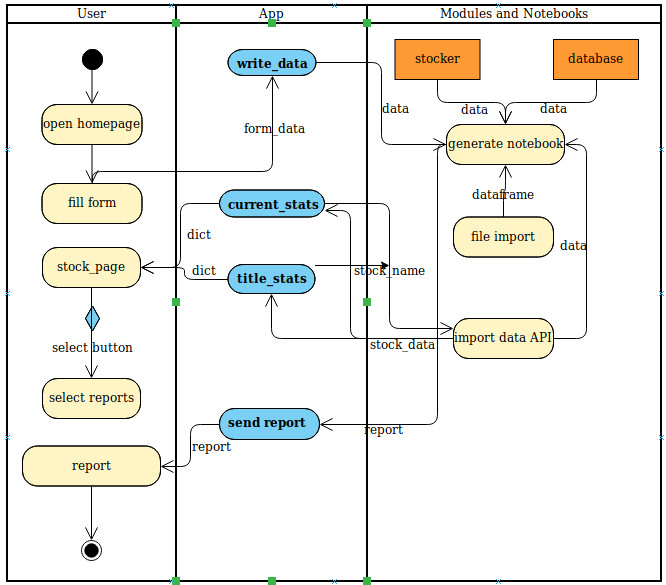
3.1.2 Data Flow:

For giving a thorough explanation on flow of data a dataflow diagram is as follows:

  
Fig 3.1: Dataflow diagram

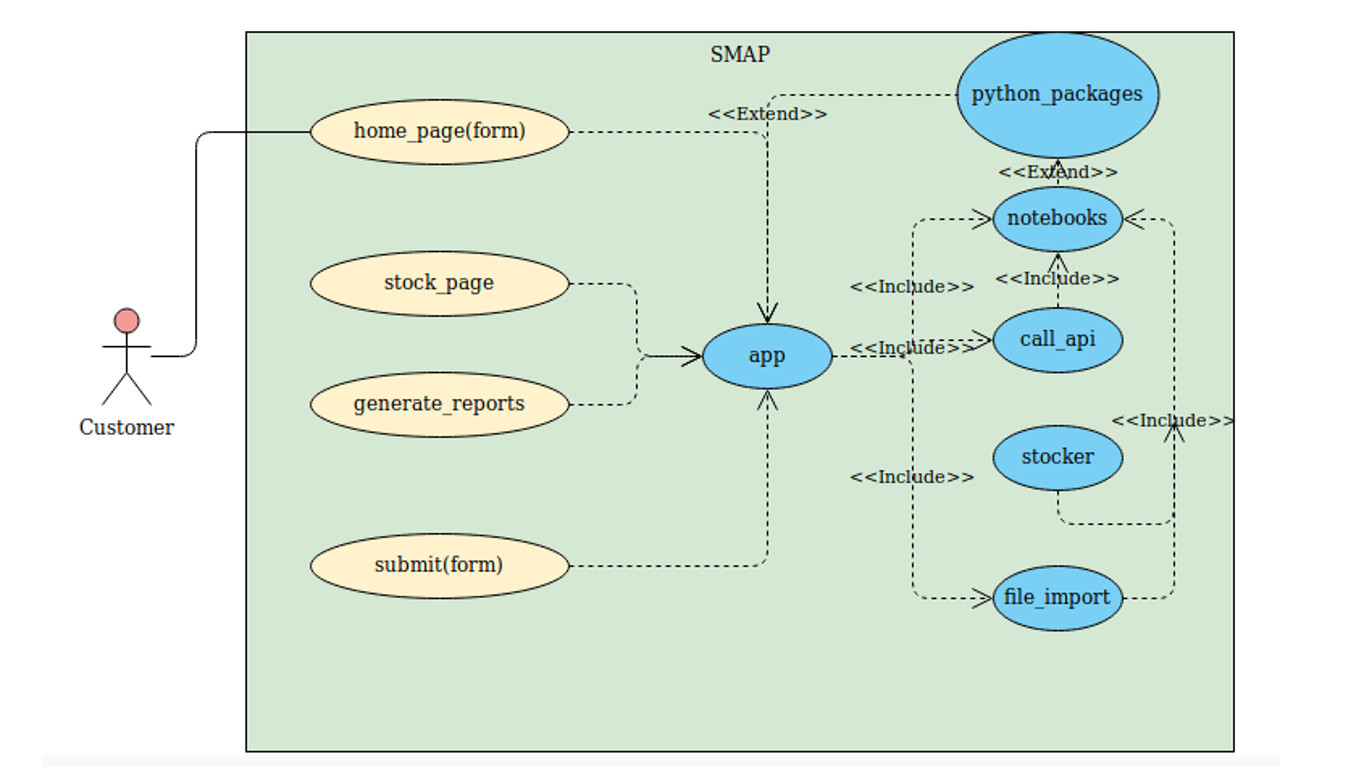
3.1.3 Activity diagram:

For showing what activities and process are performed in back-end.

  
Fig. 3.2: Activity Diagram

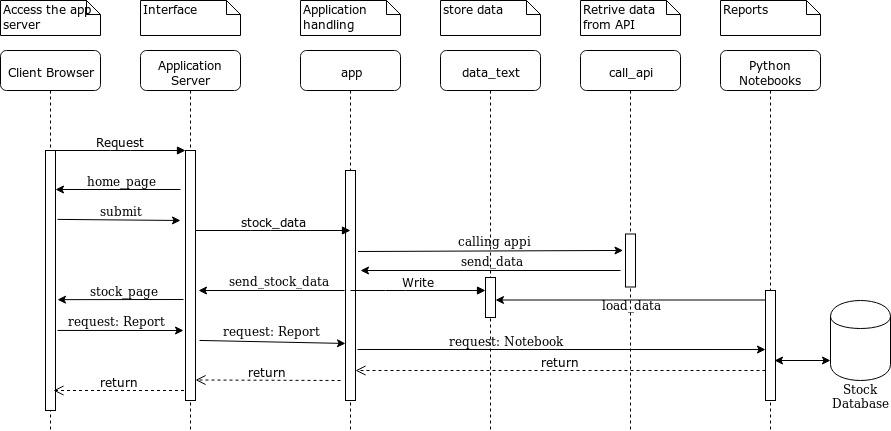
3.1.3 Usecase Diagram:

Use case diagrams are used in any project for showing the user interaction with the project. And it is as follows:

  
Fig. 3.3: Usecase Diagram

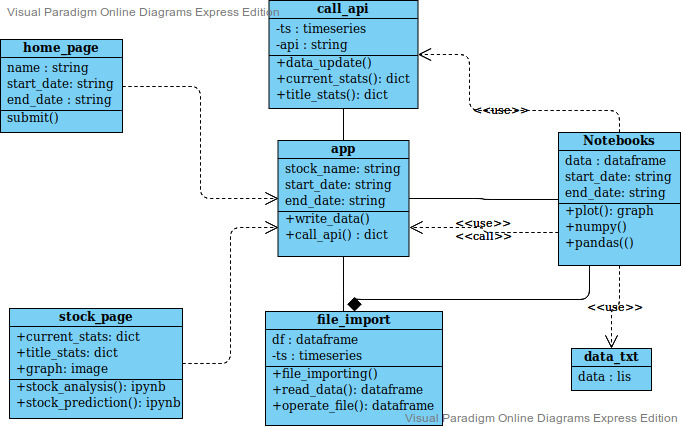
3.1.4 Sequence Diagram:

Sequence diagrams are used to represent how process and actions are performed in a sequential manner. Sequence diagram is as follows:

  
Fig. 3.4: Sequence Diagram

3.1.5 Class Diagram:

Class diagrams are used to describe a static structure of system by describing its classes. Class diagram is as follows:

  
Fig. 3.5: Class Diagram

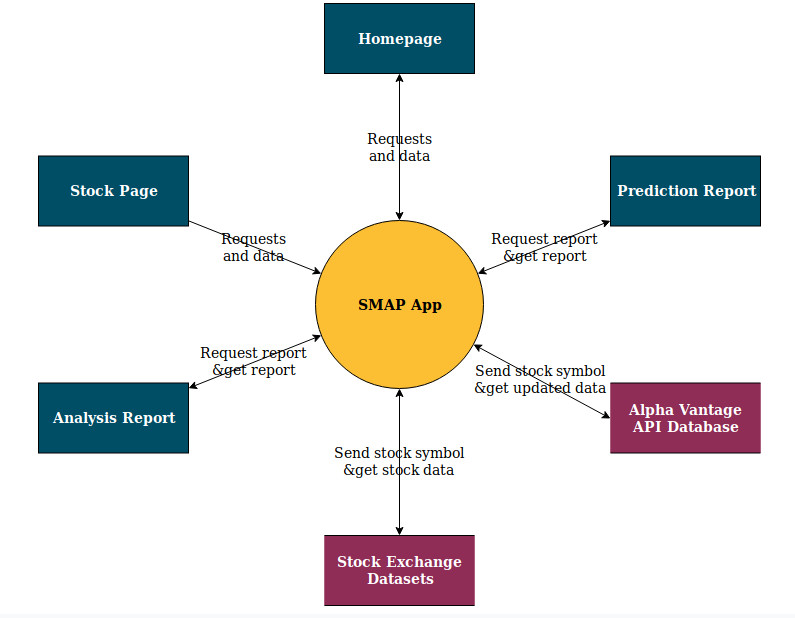
**Chapter-4 Design**

This chapter design includes all architectural design, approach, data design, interfaces that we used in this project that is Stock Market Analysis and Design Project.

**4.1 Architectural Design**

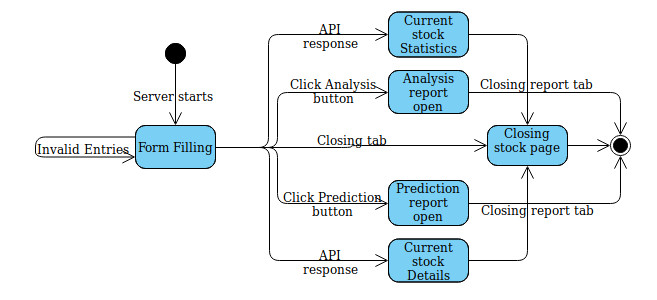
4.1.1 Architectural Context Diagram:

Architectural context diagram of stock market analysis and prediction project gives a overview of how project will look from the users side.

  
Fig. 4.1: Architectural Context Diagram

4.1.2 Architectural Behavioral Diagram:

Architectural behavioral diagram is used for describing the state of project at any specific instance of time while execution. There are many behavioral diagrams and here we will be using State machine diagram for showing the behavioral aspect of our project.

  
Fig. 4.2: State Machine Diagram

4.1.3 Description of Architectural Diagram:

Above architectural diagrams of this project infers the components of the project architecture. Architectural context diagram of the project shows the main architecture of the system with respect to user hiding the main component i.e. backend implementations of them.

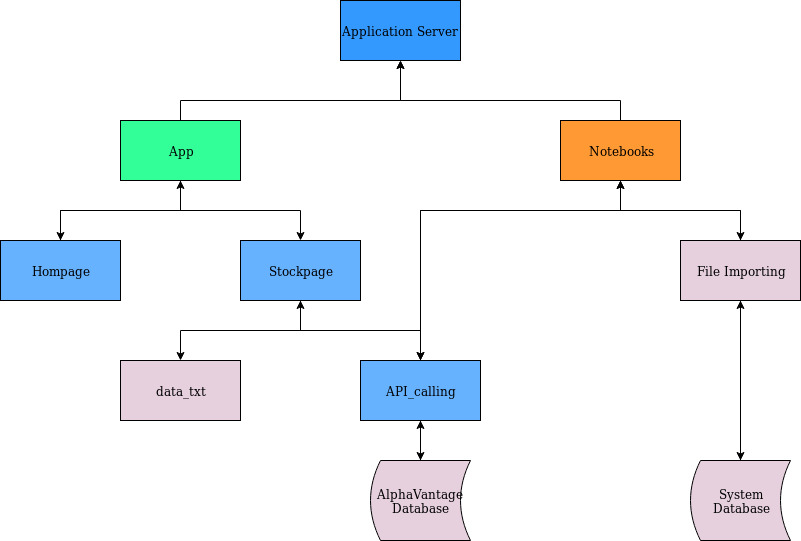
In context diagram the middle node SMAP App is the main system of the stock market analysis and prediction projects application, where all the data and request from user are accepted and transfers. And every other component present in diagram have there own work to performed

In second diagram, the state machine diagram of the projects shows the journey of project from its initialization to execution to the end. Initial step is running the application server that is opening ‘app’ file of the project it will provide a form. Filling that form and submitting it will open the stock page, where you will be seeing the parts of the project like current\_stats and all.

4.1.4 Control Hierarchy:

Control hierarchy’s initialization starts from starting the Application server through which a user access the application. This application the transfer the control to various other nodes that are present inside the whole application. This application uses two databases, first one is Alpha vantage stock data API’s database and another one is the database that is present inside the SMAP application. This database get update simultaneously with opening or loading of any stock data.

Control hierarchy explains the control flow from the starting node to the system end nodes. Explaining the architecture of the whole project. Below diagram explains it in a graphical manner.

  
Fig. 4.3: Control Hierarchy

**4.2 Both Modular and Procedural Approach**

In Stock Market Analysis and Prediction project we used modular approach in designing the application while for analysis and prediction we used procedural approach. As both analysis and predictions are in form of python notebooks the procedural approach suits them best. And in case of application, it is divided in multiple modules for proper organization of the application components.

4.2.1 Modules Used:

In this project we will be considering the notebooks that is reports of prediction and analysis as separate modules. All the modules that are present in this project

are given below:

| **Module Name** | **Description** |
| --- | --- |
| 1. app | App module contain flask codes that deploys application server and provide interface between user and system |
| 2. call\_api | call\_api module helps in connecting with alpha vantages stock database and retrieve data |
| 3. file\_import | file\_import module handles all the file operations done in whole project |
| 4. stocker | Stocker module contains codes and plotting styles for FBProphet model used in prediction |
| 5. analysis\_report | It is a python notebook in which stock analysis is performed |
| 6. prediction\_report | It is a python notebook in which stock prediction is performed using various modules. |
| 7. set\_get | In this module full details of any stock are stored and is used for transferring data |
| 8. var.txt | It is simple module used for transferring data from application to notebooks |
| 9. simple\_scaling | In this module functions and routine used for scaling of values and dataframe objects. |

4.2.2 Internal Data Structures:

As this is a machine learning project data structures mainly used in this project is pandas DataFrame. But there are other data structures besides dataframes and are given below with there description:

| Modules | Description |
| --- | --- |
| 1. DataFrame | Pandas dataframe is one of the most commonly used data structures for processing data in machine learning and data analysis projects |
| 2. NDArrays | Numpy’s ND arrays are most commonly used for operating on large number of values. |
| 3. List | List are pythons inbuilt heterogeneous data structure used for storing and processing values. |
| 4. Dictionary | It is key : value type data structure used for holding data in which key-value pairs exist. |

4.2.3 Algorithm Design for Operation:

In stock market analysis and prediction, various machine learning and deep learning algorithms are used for predicting the future prices of the stock. In this algorithm we try to fit our data into this models and get the predicted outcomes. Here artificial neural networks that we used in predicting the stocks are also present.

**4.2.3.1 KNN Algorithm:**

KNN stands for **K-Nearest Neighbors**. KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret output

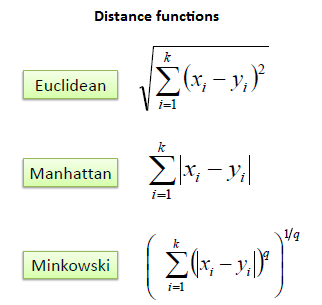
2. Calculation time

3. Predictive Power

KNN algorithm fairs across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time. Pseudo code for KNN is given below:

Load the data

1. Initialize the value of k
2. For getting the predicted class, iterate from 1 to total number of training data points
   1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it’s the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
   2. Sort the calculated distances in ascending order based on distance values
   3. Get top k rows from the sorted array
   4. Get the most frequent class of these rows
   5. Return the predicted class



**4.2.3.2 ARIMA Deep Learning Model:**

ARIMA, short for ‘Auto Regressive Integrated Moving Average’ is actually a class of models that ‘explains’ a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values. Any ‘non-seasonal’ time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models. An ARIMA model is characterized by 3 terms: p, d, q where,

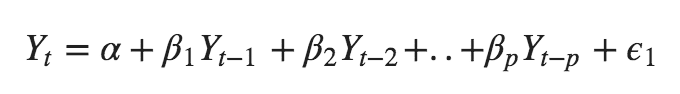
p is the order of the AR term

q is the order of the MA term

d is the number of differencing required to make the time series stationary

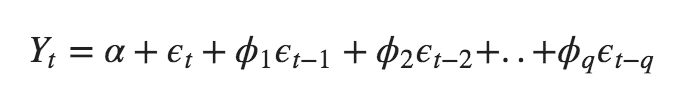
If a time series, has seasonal patterns, then you need to add seasonal terms and it becomes SARIMA, short for ‘Seasonal ARIMA’.

A pure Auto Regressive (AR only) model is one where Yt depends only on its own lags. That is, Yt is a function of the ‘lags of Yt’.

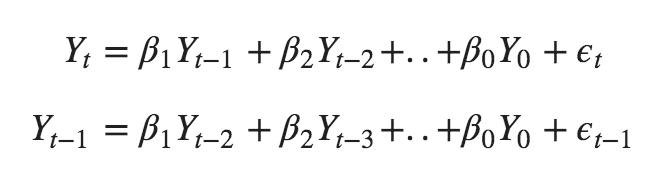


where, $Y{t-1}$ is the lag1 of the series, $\beta1$ is the coefficient of lag1 that the model estimates and $\alpha$ is the intercept term, also estimated by the model.

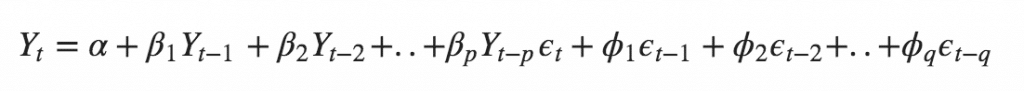
Likewise a pure Moving Average (MA only) model is one where Yt depends only on the lagged forecast errors.



where the error terms are the errors of the autoregressive models of the respective lags. The errors Et and E(t-1) are the errors from the following equations :



An ARIMA model is one where the time series was differenced at least once to make it stationary and you combine the AR and the MA terms. So the equation becomes:



**4.2.3.3 Long Short Term Memory (LSTM):**

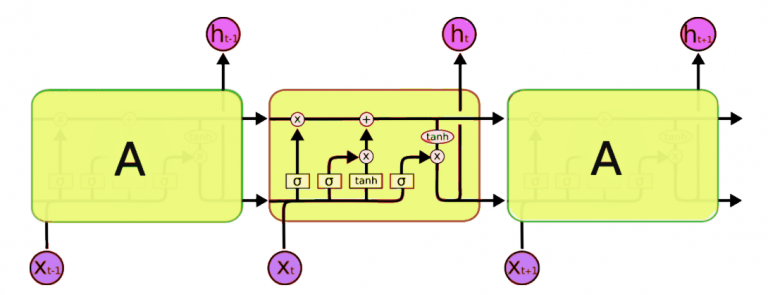
Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. Recurrent neural networks are different from traditional feed-forward neural networks. This difference in the addition of complexity comes with the promise of new behaviors that the traditional methods cannot achieve.

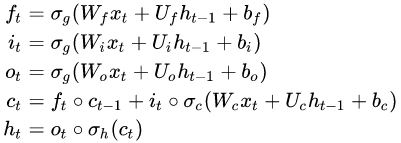
## Architecture of LSTMs:

A typical LSTM network is comprised of different memory blocks called **cells.** There are two states that are being transferred to the next cell; the **cell state** and the **hidden state**. The memory blocks are responsible for remembering things and manipulations to this memory is done through three major mechanisms, called **gates.**

### LSTM with a forget gate

  
Fig. 4.4: LSTM Architecture

The compact forms of the equations for the forward pass of an LSTM unit with a forget gate are:



where all the variable stands for:

: input vector to the LSTM unit

: forget gate's activation vector

: input/update gate's activation vector

: output gate's activation vector

: hidden state vector also known as output vector of the LSTM unit

: cell state vector



: weight matrices and bias vector parameters which need to be learned during training

Limitations of LSTMs:

LSTMs are a very promising solution to sequence and time series related problems. However, the one disadvantage that I find about them, is the difficulty in training them. A lot of time and system resources go into training even a simple model. But that is just a hardware constraint!

**4.2.3.4 FBProphet:**

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

Qualities of Prophet

* Fast and Accurate
* Fully automatic
* Tunable forecast

FBProphet uses decomposable time series model with 3 main components: seasonal, trends, holidays or events effect and error which are combined into this equation:

Here,

g(x) : seasonal component

s(x) : trends component

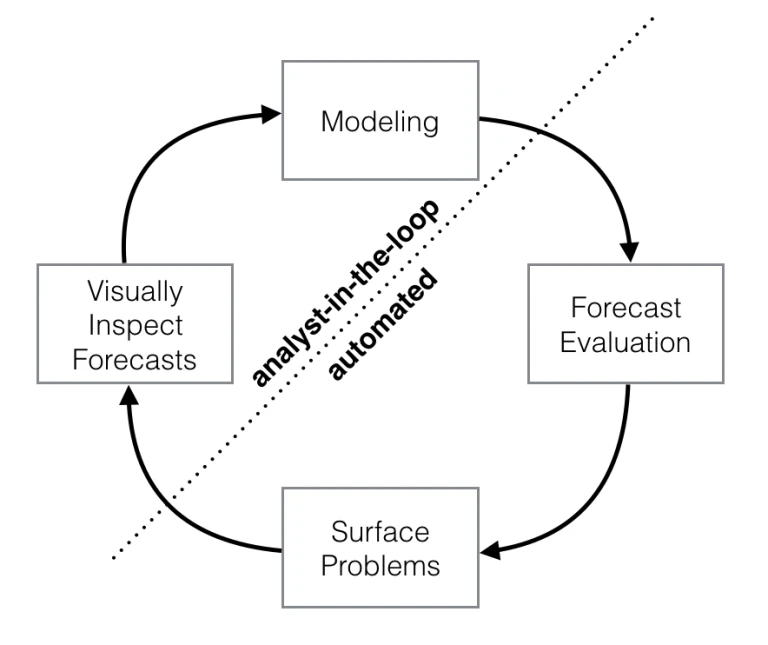
h(x) : event effect components

e(t) : error component

Working of Prophet

Prophet is optimized for the business forecast tasks we have encountered at Facebook, which typically have any of the following characteristics:

* hourly, daily, or weekly observations with at least a few months (preferably a year) of history
* strong multiple “human-scale” seasonalities: day of week and time of year
* important holidays that occur at irregular intervals that are known in advance (e.g. the Super Bowl)
* a reasonable number of missing observations or large outliers
* historical trend changes, for instance due to product launches or logging changes
* trends that are non-linear growth curves, where a trend hits a natural limit or saturates

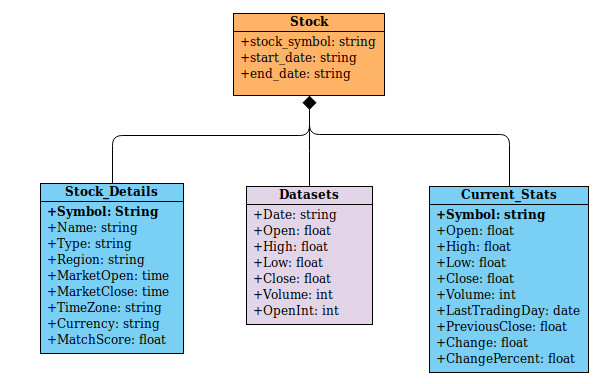
  
Fig. 4.5: FBProphet Diagram

**4.3 Data Design**

For data in this project is not in the form of database instead it is in the form of datasets stored in the form of text files. This text files contain data in a comma separated values format that is every line have values that are separated by commas. First line of this text files have following line in common:

Date, Open, High, Low, Close, Volume, OpenInt

This first line helps the data objects to determine the column names and all the below values are inserted in this columns respectively. This text files contain only two data types: 1) String and 2) Float. Dates are in form of strings while other values are float.

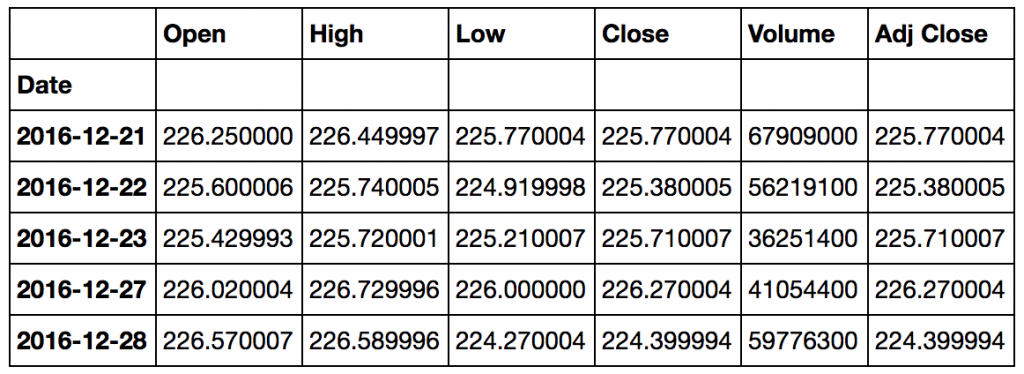
  
Fig. 4.6: Data Object Diagram

4.3.1 Data Objects and Resultant Data Structures:

Data object that are mainly used in this project are pandas DataFrame, ND arrays, list and dictionaries. Pandas Series object is also used in some case. All those data objects are listed with there details below:

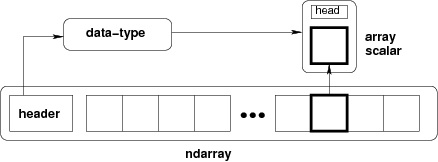
DataFrame

Pandas is a library in python that provides us to basic data structure which we will be mostly using. DataFrames are pandas data structure which is used to describe data in a tabular format. DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

  
Fig. 4.7: Pandas DataFrame Object

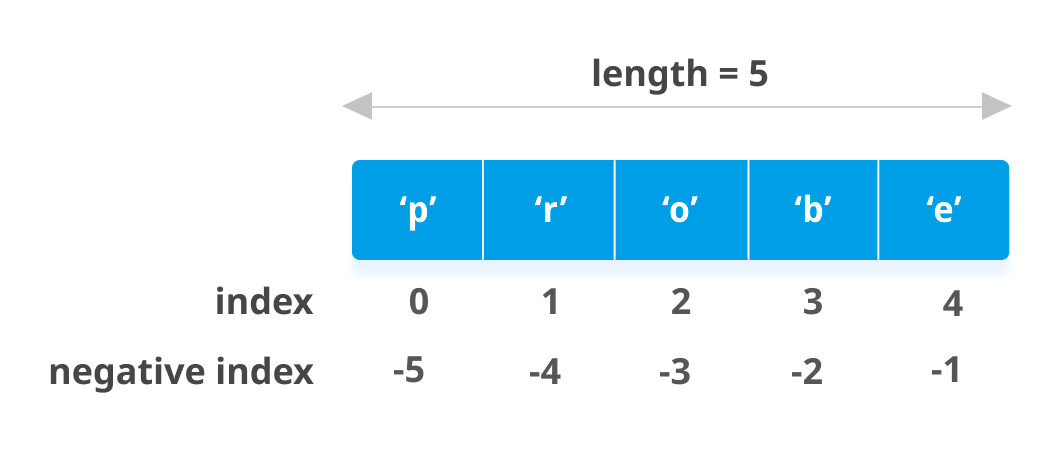
ND Arrays

ND in ndarrays stands for n dimensional arrays. Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In Numpy, number of dimensions of the array is called rank of the array. A tuple of integers giving the size of the array along each dimension is known as shape of the array. An array class in Numpy is called as ndarray.

  
Fig. 4.8: Numpy ND Array

Lists

Listsare just like the arrays, declared in other languages. Lists need not be homogeneous always which makes it a most powerful tool in Python. A single list may contain DataTypes like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation. The elements in a list are indexed according to a definite sequence and the indexing of a list is done with 0 being the first index.

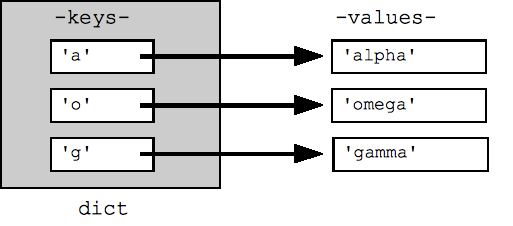
  
Fig. 4.9: Python Lists

Resultant Data Structures:

Resultant data structures are also in the form of DataFrames and dictionaries. As we have already discussed about DataFrame, lets us discuss dictionaries.

Dictionary

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

  
Fig. 4.10: Python Dictonary

**4.4 Interface Design**

In every project related to stock market analysis and prediction, user to machine interface is compromised but in this one we took care of that too. For a better user experience we developed a web application using Flask python. A simple and elegant design in which a user can just enter the stock symbol, initial and final dates of there analysis and they will get both analysis and prediction reports in form of python notebooks with current statistics of stock, details of the stock and a graph.

4.4.1 Human Machine Design Interface Specification:

Fig. 4.13: Stockpage Part-2

Human-machine design interface is built using Flask python. Flask is micro-web framework used for designing simple web application. It is classified as a micro-framework because it does not require particular tools or libraries. Specifications of interface are given below:

**Pages**

homepage(fevicon, title→ SMAP, form→ input(stock\_name, start\_date, end\_date), button(submit))

* Homepage has a elegant background image of mountains and sun.
* It has a heading that is name of the projects.
* It has a form to take stock symbol, initial and final date.
* Clicking on submit button will submit the form and send it to the application system

stockpage(fevicon, title→ stock\_name, current\_stats, stock\_details, graph, button(Analysis\_report, Prediction\_report))

* Stockpage also have a elegant background image of mountains and sun.
* It consists of stock details table that shows company name, type, currency traded in, etc,. In this frame.
* Next one is current stats table that shows company’s current open, high, close, etc,. In this frame.
* Then the graph, which shows chart of open, close, high and low of a stock.
* Two buttons and the main part of the project, analysis report and prediction report which open in Jupyter in another tab.

jupyter-notebook-tab(notebook-name)

* In this tab stock analysis or prediction is done.
* Jupyter itself has an elegant structure and works efficiently.
* Jupyter notebook supports inline plots through which whole analysis and prediction reports can be shown completely in just a single window and also in a single tab.

4.4.2 I/O Forms:

This application consist of a single form, present on homepage of the web application. It has three attribute and one submit button. This form is used for taking the symbol of the stock, whose analysis and price prediction has to be done. Specification of the forms are given below:

**form**

* name→ homepage
* size→ 60% of screen width
* attributes numbers→ 3
* number of buttons→ 1
* functions→ to send attributes to the back-end

4.4.3 Reports:

* Human- machine interface design is simple but elegant.
* It is provided in the form of web application built using Flask python.
* It consist of 4 pages Homepage, stockpage (dynamic page), analysis report and prediction report page which is type of python notebook.
* A single form is present to take user inputs that is stock symbol, initial and final date.

**Chapter-5 Testing**

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product. Types of testing we will be using:

**5.1 Testing Objective**

The objective of testing in this project report is to test the software on various aspects, detect errors and correct them and make project error free. Another objective of testing is that we can get as many software defects as they are in it. We will ensure software is bug free before release.

**5.2 Testing Scope**

5.2.1 In Scope

* Functionality testing
* Unit testing
* Usability testing
* Interface testing
* API testing
* Database testing
* Compatibility testing
* Black-box testing

5.2.2 Out of Scope:

* Hardware testing
* Integration testing
* System testing
* Sanity testing
* Smoke testing

**5.3 Testing Principles**

* All the tests be traceable to customer requirements
* Tests should be planned long before testing begins
* Testing should begin “in the small” and progress toward testing “in the large”
* Exhaustive testing is not completely possible
* To be most effective, testing should be conducted by an independent third party
* Absence of errors fallacy
* Defect clustering that is defect are found only in small number of modules

**5.4 Testing Methods Used:**

Software Testing Strategies: A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. As important, a software testing strategy provides a road map. Testing is a set of activities that can be planned in advance

and conducted systematically.

Various strategies are given below:

Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design of module. This is also known as “Module Testing”. Acceptance of package is used for computerization of module. Machine Utilization was prepared and approved by the project leader. In this testing step, each module is found to be working satisfactory as regards to the expected output from the module. The suggested changes were incorporated into the system. Here each module in the Machine Utilization has been tested.

Functionality Testing

Functionality testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

Usability Testing

Usability testing is a way to see how easy to use something is by **testing** it with real users. Users are asked to complete tasks, typically while they are being observed by a researcher, to see where they encounter problems and experience confusion.

Interface Testing

Interface Testing is defined as a software testing type which verifies whether the communication between two different software systems is done correctly. A connection that integrates two components is called interface. This interface in a computer world could be anything like API's, web services, etc.

API Testing

API testing is a type of software testing that involves testing application programming interfaces (APIs) directly and as part of integration testing to determine if they meet expectations for functionality, reliability, performance, and security. Since APIs lack a GUI, API testing is performed at the message layer.

Compatibility Testing

Compatibility testing is a part of the non-functional tests that a software is put through. It is conducted to test a software's compatibility with the computing environment and its elements, such as operating systems, browsers, hardware, mobile devices, databases, computer peripherals, and other system applications.

Black-box Testing

Black box testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications.

**5.5 Test Cases**

For test we took three stocks as test case, namely aapl, msft, wmt which corresponds to Apple, Microsoft and Walmart corporations, respectively,. On the basis of these three stocks we follow all the testing methods and find bugs and errors in the project.

Selenium software testing tool is used for testing the web application part of this project. Selenium is a command line based tool that is written in python and can be used for usability and compatibility testing of the web application. Through Selenium we got perfect results no error was generated. As the web application consists of only a small part of the program it is made bug free.

And rest of the project is tested manually as it is notebook format and for the best results manual testing is much help full. All the results that we got are in the table below. A single stock test is also provided below with full functionality of the application with analysis and prediction report.

**5.6 Sample Test Data & Results:**

Table for results found by automated selenium testing of web application:

| **Simulated user Utterance** | **Confidence score** | **Matched Intent(s)** | **Unrecognized intent(s)** |
| --- | --- | --- | --- |
| Form Filling | 1.00 | 1.00 | 0.00 |
| Form submission | 1.00 | 1.00 | 0.00 |
| Calling stock details | 1.00 | 1.00 | 0.00 |
| Calling current statistics | 1.00 | 1.00 | 0.00 |
| Showing graph | 1.00 | 1.00 | 0.00 |
| Opening reports | 1.00 | 1.00 | 0.00 |

Manual testing of the Analysis reports:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part Being Tested** | **Confidence Score** | **Error(s) Encountered** | **Response Time** |
| Importing modules | 0.90 | 0.10 | 6.46 s |
| Load data | 0.80 | 0.20 | 4.13 s |
| Updating datasets | 0.70 | 0.30 | 36 ms |
| Calling API | 0.70 | 0.30 | 36 ms |
| Plotting all graphs | 0.90 | 0.10 | 56.30 s |
| Working with stocker | 0.90 | 0.10 | 12.80 s |
| Loading indexers | 0.70 | 0.30 | 1.13 s |

Manual Testing of Prediction Report:

| **Part Being Tested** | **Confidence Score** | **Error(s) Encountered** | **Response Time** |
| --- | --- | --- | --- |
| Importing Modules | 0.90 | 0.10 | 23.10 s |
| Load data | 0.80 | 0.20 | 4.13 s |
| Updating datasets | 0.70 | 0.30 | 36 ms |
| Calling API | 0.70 | 0.30 | 36 ms |
| KNN algorithm | 1.00 | 0.00 | 2.50 s |
| Scaled KNN algorithm | 1.00 | 0.00 | 3.50 s |
| ARIMA model | 0.95 | 0.05 | 450.5 s |
| LSTM epoch = 1 | 1.00 | 0.00 | 15.4 s |
| LSTM epoch = 4 | 1.00 | 0.00 | 28.8 s |

**Chapter-6 Limitations**

Well beside, the provided solution above there are many important limitation of this stock market analysis and prediction project. And we will be discussing them below:

As we all know, stock market is known to be one of the most unpredictable thing in world. But as we are trying to predict it here, we are trying to get close to something though not very accurately. In this project we are trying to strengthen the idea or hunch that a investor has in his mind about the stock.

This project does not cover fundamental analysis of a stock. Fundamental analysis of stock is one major type of analysis that must be before investing in any stock. As fundamental analysis can’t be taught to a machine it remains a limitation.

Price of a stock depends on a large number of things and a lot of them are uncertain. And even if we somehow get that data it won’t be feasible to process that amount of data and it will certainly be out of the project scope.

Since for updating the datasets, we call Alpha Vantage API, which requires internet. Therefore without internet our project won’t work properly.

Our project covers only US stock data and therefore can’t be used for the companies or stocks that are listed in US stock exchanges. Cryptocurriences or the digital currencies are also out of the project scope currently.

**Chapter-7 Future Scope**

Due to limitation in technical assistance, time and data we are not able to perform all the aspects that a stock market predictor can have. But besides that, there is lot more features and small changes that can be made in project to get best of it. And we will be discussing them below:

First of all, the range stocks can be increased that is, stocks of other countries can be added to project. Such that, this project can be used for analysis of stock data of companies other than those listed on US stock exchanges.

The cryptocurriences or digital currencies can also be analyzed and there prices could be predicted.

Accuracy of this project can be increased more by adding Twitter sentiment analysis to the project. Though by it we mean, adding two projects SMAP and Twitter Sentiments together to get better predictions.

In our research we found that, for some stocks, KNN was working better for some it was LSTM and for some stocks ARIMA worked better. So, we want to convey that more models and algorithms can be added such that more stocks can be predicted correctly to some extent. We do noticed that for smaller time intervals used for analysis and prediction, KNN worked best and hence adding more model, we will have more insights on which stock fits best with which model.

This project can be deployed to a server, so that anyone from anywhere can use this project for analyzing the stocks.

**Chapter-8 Conclusion**

With a lot of research in stock market and its prediction, we gained a lot of knowledge on how a stock is analyzed and why this stock should be traded. We utilized machine learning, deep learning and data visualization concepts in this project to make it more accurate and to search something new in it. Various plots in this project helps us in visualizing different components and performance of any stock over the time.

The Analysis reports gives a thorough explanation on the performance and past movements of the price of any stock. The volume traded tells us about which stocks were or or in trend. Analysis report uses various techniques and methods through which technical analysis of any stock is performed and with great visualization reading and understanding it becomes much easier.

The Prediction report gives us various aspects on how a stock will be moving in the future. Stock prediction is one of the trending works in this time and various deep learning model gives us an overview of how a stock will be moving and performing in the future. Usage of multiple deep learning model and machine learning algorithms is done here as we found in our research that some stocks fit only in some certain models and hence to increase the number of stocks we add extra models and algorithms to it.

Last but not the least, the web application provides a good user interface as in other projects that were built before it used a certain number of stocks and if a user wants to analyze stocks other than that he/she must alter the code. But in our project, we provide a web application by using which any stock listed in US stock exchanges can be analyzed and its future prices and movements can be predicted.

Real world applications of our project:

* It can be used to generate automated trading algorithms
* We can use this project both for analyzing any stock and its performance in any time interval.
* Our datasets are large enough that they hold data of the stocks from the day they were listed on any of the stock exchanges.
* Prediction part of the project can be used for measuring the performance of any stock in future upto 30days.

**Chapter-8 Bibliography and References**

*Thawornwong, S, Enke, D.* Forecasting Stock Returns with Artificial Neural Networks, Chap. 3. In: Neural Networks in Business Forecasting, Editor: *Zhang, G.P. IRM Press, 2004.*

Jacaruso, Lucas Cassiel (2018-12-08). "A method of trend forecasting for financial and geopolitical data: inferring the effects of unknown exogenous variables". Journal of Big Data.

Aishwarya Singh, (2018-10-25). “Stock Prices Prediction Using Machine Learning and Deep Learning Techniques”. A blog on AnalyticsVidhya. URL:<https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/>

*Jacob O. Wahlgren, Matthew D. Rosi, Steven Chris Marshall (Aug 2016).* “Stock Analysis and Simulation”. Project accepted in Worcester Polytechnic Institute.

*Wikipedia.org* . “Stock Market Prediction”. Last edited : 25 Oct 2018.

*Towardsdatascience.com.* “Machine Learning and Deep Learning”. URL:<https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>

*Machinelearningplus.com (2018-09-03). “*ARIMA Model – Complete Guide to Time Series Forecasting in Python*”* URL:<https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

*Investopedia.com*  Aug 2019 URL:<https://www.investopedia.com/about-us>

*Investing.com* Sep 2019 URL:<https://www.investing.com/about-us/>

*2019 DataCamp Inc.* “Learning Python and becoming data scientist” URL:<https://www.datacamp.com/>

*Online Stock Trading Guide.* Head and shoulders pattern, March 2015.

*Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan.* “Predicting

direction of stock price index movement using artificial neural networks

and support vector machines”: The sample of the istanbul stock exchange.

Expert systems with Applications, 38(5):5311–5319, 2011.

**Appendices**