ECE 568: Software Engineering II: Web Applications

Spring 2015

Final Project Report

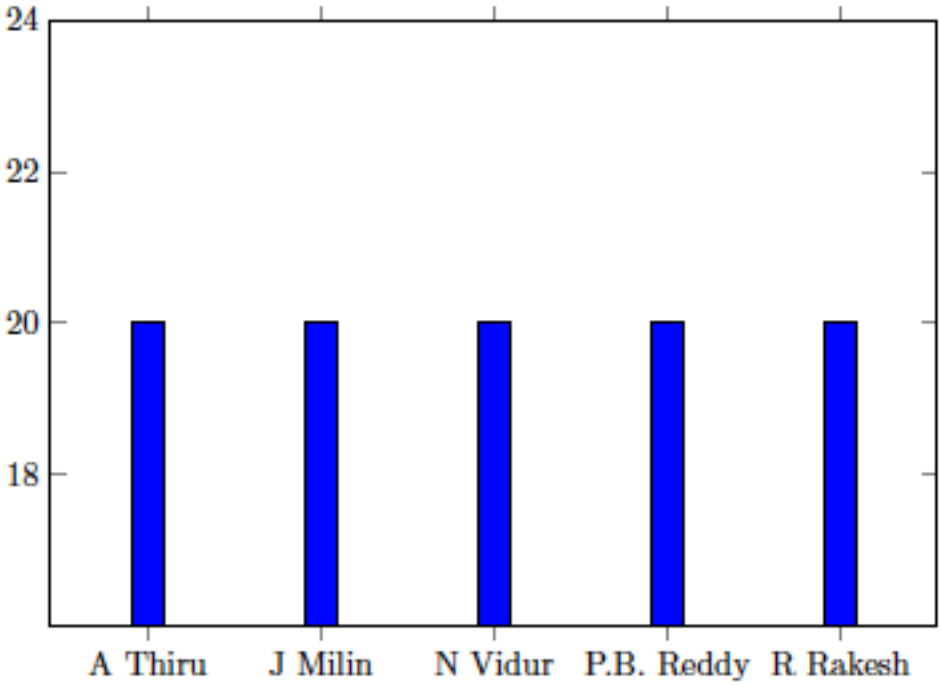
Group 13
Arunachalam, Thirunavakkarasu
Jaiswal, Milin
Nayar, Vidur
PappiReddy, Bhavya Reddy
Ravuru, Rakesh

Submitting to: Dr. Shiyu Zhou
Due on : May 1st 2015

Online Web Forecaster

Table of Contents

BREAK DOWN OF RESPONSIBILITY



Graph 1. Contribution Breakdown

All the members have equally contributed in making the project.

1.INTRODUCTION

The Stock Market is the market in which shares of publicly held companies are issued and traded either through exchanges or over-the-counter markets. Also known as the equity market, the stock market is one of the most vital components of a free-market economy, as it provides companies with access to capital in exchange for giving investors a slice of ownership in the company. The stock market makes it possible to grow small initial sums of money into large ones, and to become wealthy without taking the risk of starting a business or making the sacrifices that often accompany a high-paying career.

The idea of earning a huge sum of money by investing in the stock market attracts a lot of people to invest in the stock market. Most of the people invest in the stock market with little knowledge about the trends in the stock market and often misjudge the trends and end up losing huge sums of money. Even the stock moguls often make mistake in predicting the market trend due to human error.

Earlier all the dealings of buyers and sellers used to be done by the pension funds and brokerage firms. But with the introduction of computers in early 1990, the trade could be carried out online without involving the pension funds and brokerage firms. Soon they realized that it would be a good idea to not just use the computer to buy and sell the shares, but we could also use them to speculate the future values of the shares studying the random trends to accurately predict the future prices and manage the risks

In this project we attempt to develop a model that predicts the trend of the stock market with high precision and capable of aiding the investors to bag maximum profits from their investments. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. The successful prediction of a stock's future price could yield significant profit. To forecast these prices, two techniques are used:

- Fundamental Analysis
- Technical Analysis

Both Fundamental Analysis and Technical Analysis have their tools and back each other to predict the price of the stock accurately.

Fundamental Analysis

It seeks to identify the fundamental economic and political factors that determine a commodity's price. It is basically an analysis of the (current and future) demand for and supply of a commodity to determine if a price change is imminent, and in which direction and how prices are expected to change. This approach requires, gathering substantial amounts of economic data and political intelligence, assessing the expectations of market participants, and analyzing these information to predict futures price movement.

Technical Analysis

It involves the study of historical prices of the stock to predict its prices in the future. Technical analysts frequently utilize charts of past stock prices to identify historical price patterns. These price patterns are then used to predict the speculative future price of the stock. Prices show the fundamental information of trends, as well as other information, such as traders' expectations and the psychology of the market. The role of the technical analyst is to predict the best time to

buy and sell the shares to gain maximum profit from it. In technical analysis the candle stick chart is famous to map the change in price of the share when the market opened to the time when it closed also it shows the maximum value and the minimum value of that share during the day or week or year.

In the following sections we will describe the overall structure, behaviors and features of the fully integrated application, followed by some discussion on the algorithm, prediction modules, databases and web services used to implement the application, and finally we will show some results.

1.1 Project Overview

Our application uses few of the technical analysis tools to provide the stock trader with some basic stock trends and data to help make some informed decisions about a particular stock. We use the candlestick and head and shoulder analysis to improve the accuracy of our predictions. We allow the user to understand how the stock is doing on a short term as well as a long-term basis. Additionally, we provide some numerical metrics for the user so that he/she can make a decision to buy, sell, or hold a certain security either in the short term or long term. Numerical metrics that are considered in the application include, *short-term prediction using Bayesian estimation, and long-term shape prediction using support vector machine techniques (SVM) and artificial neural networks*. In the long-term prediction, the application detects for three specific cases of hold, sell and buy where we believe the stock is at a pivotal point and a reaction from the user is expected. Using the Google trend data to find out how many times the company was searched add some advantages of fundamental analysis to our project.

As the technical analysis uses the historical prices of the stock to predict its prices in the future, we collect the historical data of the stocks using the Yahoo Finance API. We use JAVA programming language to collect both the historic and real time data by querying the Yahoo Finance API. The collected real time and historic data of the stock is stored using PostgreSQL database.

Our project mainly focuses on the following aspects:

Data feed

Real time data is collected using Yahoo Finance API. For the purpose of short-term prediction, the real time data is collected every *30 seconds*. Long-term prediction is based on historical data, for this reason every day at 7 am the data is collected to accumulate sufficient historic data for the prediction algorithms to work on, thus making it a realistic prediction advisory.

Prediction

Two types of prediction strategies are used, short term and long term. For the purpose of short-term prediction, we use Bayesian predictor algorithm and for long-term algorithm we use Support Vector Machine (SVM) and artificial neural networks. Neural networks often exhibit

patterns similar to those exhibited by humans, which makes them a good fit for our project as stock prices fluctuate with human interaction. SVM is used for its high accuracy in predicting the stock price because of its theoretical guarantees of resistance towards over-fitting data.
With respect to the short-term predictions, *the application provides a countdown timer that shows how long the prediction is valid for. By default, the application polls information every 30 seconds from the financial sources, therefore the prediction is only valid for 30 seconds, time taken to make the next value available.*
The validity of long-term prediction is based on updating the stock values on daily basis. In other words, the java script pulls value at 7am each day that shows the close of day status of the said security of the previous day and adds it to the long-term table of the database.

Ease of use

The developed application strives to provide various tools to the end user to access valuable information and recommendations to aid him to make a wise decision with his stocks. The selected stock by the user would be added to his profile for easy viewing.

The user interface of our website is developed using Bootstrap, CSS and JS. We have tried to make the user interface as user friendly as possible so that even a novice in the field can access and take advantage of the application. The application runs superior algorithms to predict the price of the stocks and recommend the user whether to buy, sell or to hold the stock.

Recommendation Policy

As mentioned above the application provides recommendations to the user whether he should buy, sell or hold on to the stock. This decision is made using the predicted stock value using the long-term prediction algorithms. Google trend provides give a fair indication of the popularity of the company under review. Usually the more a company is searched, the popular it is and that makes it price to rise in the future.

2. SOFTWARE ENVIRONMENT

The data collection is done using Yahoo REST services - Yahoo query language accessed using a Java console application. The web application's front end is developed on JavaServer Pages (JSP). Most of the code is HTML/JQuery and hence can easily be ported to other platforms if necessary. The JSP front end uses highstocks for displaying graphs and DataTables JQuery plug in to display tables. The JSP front end access web services built on Java using the Jersey RESTful web services framework. These web services access the Postgre SQL database where all the data collected about the stocks is stored.

2.1 Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA, meaning that compiled Java code can run on all platforms that support Java without the need for recompilation [http://en.wikipedia.org/wiki/Java_(programming_language)].

We developed the application using java because of the easily availability of Yahoo Finance API in JAR format. Java is also widely used with great documentation that has been used for many years now. Wide availability of java frameworks and maturity of the platform makes java one of the most desirable programming language. The Java architecture is robust and reasonably elegant. It makes it possible to architect well-built applications. There are numerous free tools available for java are enterprise ready. For example, Apache / Tomcat / JBoss are a solid foundation to build a web-app from. There is a good availability of third-party (and open-source) libraries and Eclipse plug-ins for most of the additional functionality that might be needed but does not come in the core libraries.

2.2 Postgresql Backend

PostgreSQL, is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance. As a database server, its primary function is to store data securely, supporting best practices, and to allow for retrieval at the request of other software applications. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Just as there are many procedure languages supported by PostgreSQL, there are also many library interfaces as well, allowing various languages both compiled and interpreted to interface with PostgreSQL.[en.wikipedia.org/wiki/PostgreSQL]

An enterprise class database, PostgreSQL boasts sophisticated features such as Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, and write ahead logging for fault tolerance. PostgreSQL runs stored procedures in more than a dozen programming languages, including Java, Perl, Python, Ruby, Tcl, C/C++, and its own PL/pgSQL, which is similar to Oracle's PL/SQL. Included with its standard function library are hundreds of built-in functions that range from basic math and string operations to cryptography and Oracle compatibility.[http://www.postgresql.org/about/]

We decided to use PostGreSQL over MySQL because of the reasons given below:

Advantages of PostgreSQL

An open-source SQL standard compliant RDBMS: PostgreSQL is open-source and free, yet a very powerful relational database management system.

Strong community: A devoted and experienced community that can be accessed through knowledge bases and Q&A sites 24/7 for free supports PostgreSQL.

Strong third-party support: Regardless of the extremely advanced features, PostgreSQL is adorned with many great and open-source third-party tools for designing, managing and using the management system.

Extensible: It is possible to extend PostgreSQL programmatically with stored procedures, like an advanced RDBMS should be.

Objective: PostgreSQL is not just a relational database management system but an objective one - with support for nesting, and more.

Disadvantages of MySQL

Known limitations: By design, MySQL does not intend to do everything and it comes with functional limitations that some state-of-the-art applications might require.

Reliability issues: The way certain functionality gets handled with MySQL (e.g. references, transactions, auditing etc.) renders it a little-less reliable compared to some other RDBMSs.

Stagnated development: Although MySQL is still technical an open-source product, there are complaints regarding the development process since its acquisition. However, it should be noted that there are some MySQL-based, fully integrated databases that add value on top of the standard MySQL installations.

2.3 Scientific Computing Tools

2.3.1 MATLAB

MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or JavaTM. You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. For a million engineers and scientists in industry and academia, MATLAB is the language of technical computing.

2.3.2 **Spider SVM**

Spider is an object orientated environment for machine learning in MATLAB, for unsupervised, supervised or semi-supervised machine learning problems, and includes training, testing, model selection, cross-validation, and statistical tests. Implements SVM multi-class classification and regression. It aims to become a complete research/analysis toolbox: Includes training, testing, model selection, and statistical tests. Plugging objects together: e.g perform cross validation on the following system: greedy backward feature selection on a fast base algorithm, training on those features with an SVM for each output in a one-against-the-rest multi-class system, choosing all hyper parameters with a model selection method.

2.3.3 **Matlab Control Java API**

matlabcontrol is a Java API that allows for calling MATLAB from Java. You can eval, feval, as well as get and set variables. Interaction can be performed from either inside MATLAB or outside MATLAB.

2.4 Jersey Web Framework

Jersey RESTful Web Services framework is open source, production quality, framework, for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation.
Jersey framework is more than the JAX-RS Reference Implementation. Jersey provides it's own API that extend the JAX-RS toolkit with additional features and utilities to further simplify RESTful service and client development. Jersey also exposes numerous extension SPIs so that developers may extend Jersey to best suit their needs.[https://jersey.java.net]

2.5 Bitbucket Version Control

Bitbucket is a web-based hosting service for projects that use either the Mercurial or Git revision control systems. Bitbucket offers both commercial plans and free accounts. It is similar to GitHub. It is ad-free and provides a clean interface for archiving files, providing version control and web hosting. It has many other useful features like code review, bug tracking and release binaries that help manage multi-developer projects in an effortless way. [http://en.wikipedia.org/wiki/Bitbucket]

We chose to use git with bitbucket for source control because it gives us distributed revision control with access to all the source files from the cloud. Distributed revision control takes a peer-to-peer approach to version control, as opposed to the client-server approach of centralized systems. Rather than a single, central repository on which clients synchronize, each peer's working copy of the codebase is a complete repository. Distributed revision control synchronizes repositories by exchanging patches (sets of changes) from peer to peer. In a DVCS (such as Git, Mercurial, Bazaar or Darcs), clients' don't just check out the latest snapshot of the files: they fully mirror the repository. Thus if any server dies, and these systems were collaborating via it, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.

2.6 Amazon Web Services

Access to our web services would be required by the persons doing the development for the front end and the persons developing the prediction strategies. To make each person's work independent of each other, we decided to host our PostgreSQL database and web services on the Amazon cloud as it would help in true distributed development.

Details of the Amazon web services (AWS) instance:
EC2 instance running Windows Server 2012.
Amazon relational database service (RDS) hosting Postgre SQL 9.3.6.
ngrok tunnel to expose the web services publicly.

3. SYSTEM DESCRIPTION

3.1 System Blockdiagram

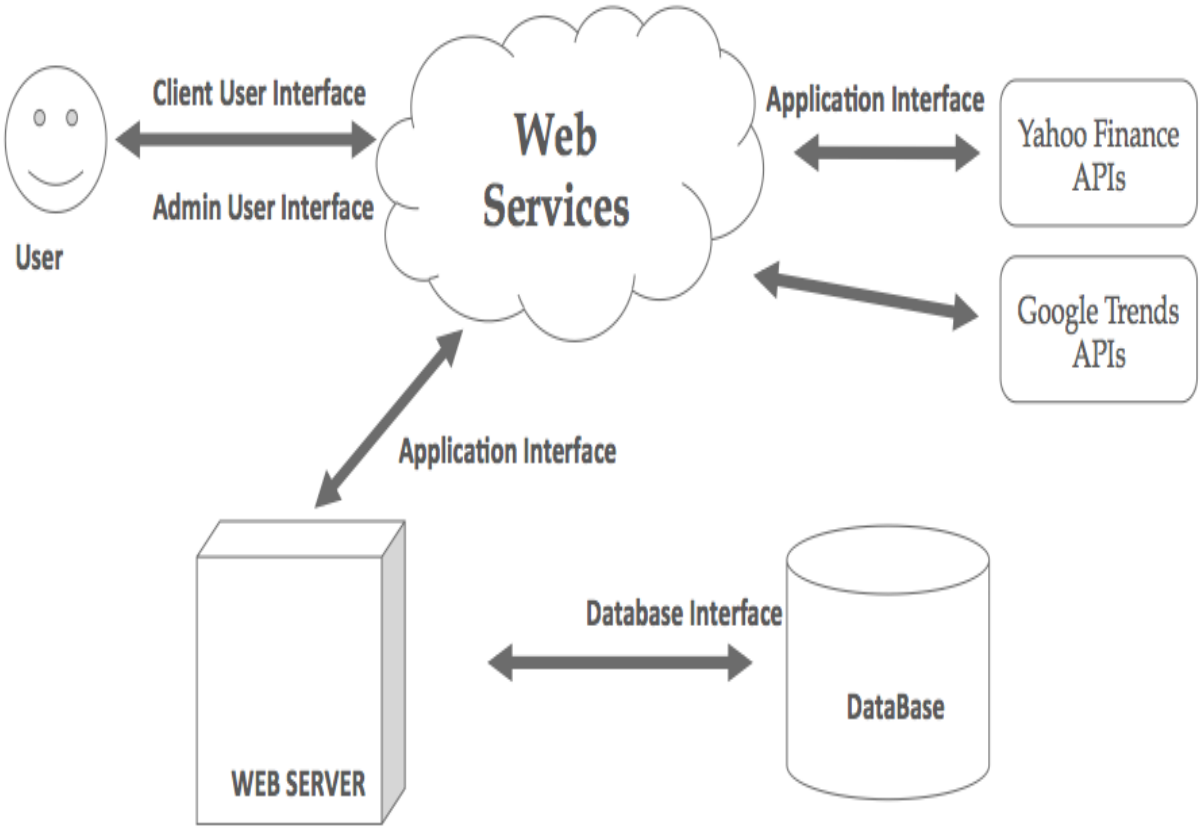


Figure 1. System Block Diagram

Figure 1 shows the basic system block diagram for our application. It can be seen how seamlessly all the components have been integrated to work in coherence. As mentioned above the system uses Java as the primary scripting language to query the Yahoo Finance and the Google trends APIs. The data storage is done entirely on the PostgreSQL database. The user interface webpage is developed using HTML, CSS, bootstrap and js.

We allow each user to register and create their own profile which they can access using a secure login. This gives each user an independent secure profile to view their stocks.

The application is capable of performing unbiased predictions from other web services by using our website.

3.2 System Description And Working

This section gives the functional description of the system.

The login screen is the first page the user lands on. He can either sign up for a new account on our system or can login with an existing account. The email id acts as the user name.

Once logged in, the user is presented with a list of the stocks we have data for on the left pane. When he chooses any of these, the right pane shows him 4 tabs - historical data, real time data, long term predictions and short term prediction. Each of these tabs show tabular data for the stock selected and the same data represented in the form of graphs. There is an additional tab for miscellaneous statistics and google trends.
The short predictions give the predictions for the next 5 minutes and the long term prediction gives the prediction for the next 10 days. We also recommend whether the user should buy or sell or hold depending on the long term prediction.

3.3 Use Case

3.3.1 Use Case Diagram

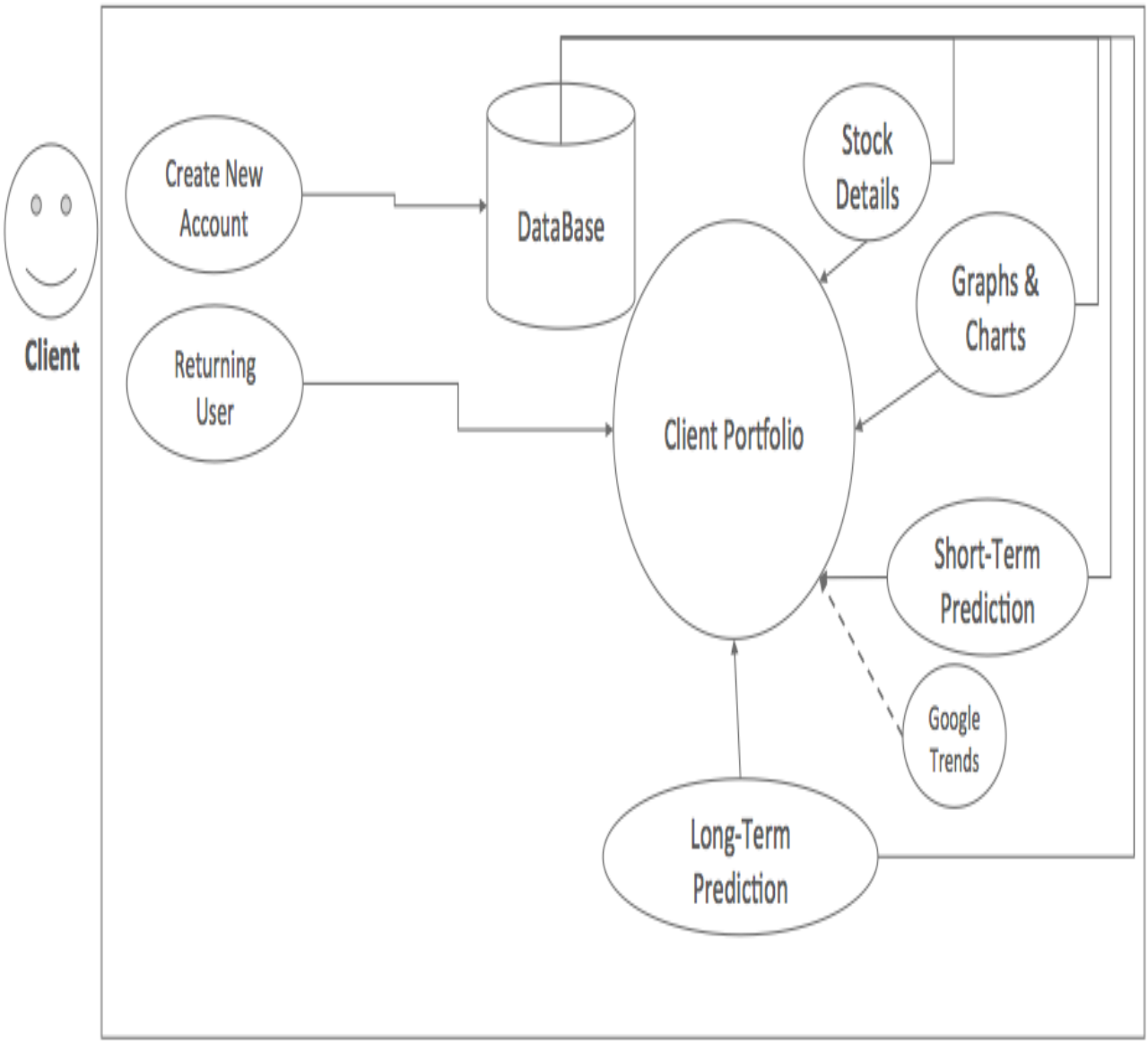


Figure 2. User Case Diagram

3.3.2 ***Actors And Goals***

Actor	Category	Goals
User	Initiating	Register for the application, login, Account management, stock search, request to get long-term and/or short-term prediction, request for generating graphs, request to get stock details.
Device system (Server)	Initiating and Participating	Forwards the request of the user to the database, automatically query's the yahoo API for stock prices, query's the Google trend for data, provides the user with decision on the stock.
Google Trends API	Participating	Collects the Google trend data.
Yahoo Finance API	Participating	Collects the stock prices from Yahoo Finance.
Database	Participating	Store the information for user authentication, stock values and the corresponding stock names.

3.3.3 ***Use Case Casual Description***

UC-1 Registration Allows the user to create an account on our website to use the services provided.

UC-2 Login Allows the authorized user to access the system by providing the correct ID password pair.

UC-3 Short-term prediction Allows the user to get the short-term prediction value of the stock selected.

UC-4 Long-term prediction Allows the user to get the long-term prediction value of the stock selected.

UC-5 Generate graphs Allows the user to access the graphical plots of the stock price of the selected stock.

UC-6 Stock decision The user will be provided a decision of whether it is a good idea to buy, sell or hold on to a stock based on long term prediction.

UC-7 Stock detail	The user will be allowed to access the stock price of the stock selected and also the historic price of the stock along with the short-term and long-term prediction of the stock.

USE CASE #1 DESCRIPTION

Use Case 1 Registration
Initiating Actor: User
Actors Goals: create a new account
Participating Actor: Device system (Server), Database
Preconditions: User does not have an account.
Success End Condition: User is registered and information is stored in the server.
Failed End Condition: Users id is already in use or password does not match the specifications.
Extension Point:
Flow of Events for Main Success Scenario: 1. User opens the webpage 2. User enters the personal information 3. Server checks if the id is taken in the Database and if the password is accepted 4. Server sends the newly added information to save it to the database 5. Server sends a message to display success case

USE CASE #2 DESCRIPTION

Use Case 2 Login
Initiating Actor: User
Actors Goals: To login the system by providing username and password
Participating Actor: Server, Database
Preconditions: User is already registered
Success End Condition: User is logged into the system
Failed End Condition: User enters incorrect username or password or the password does not correspond with the username.
Extension Point:
Flow of Events for Main Success Scenario: 1. User Opens the webpage 2. User enters the correct username and password 3. User logs in by selecting the login option

USE CASE #3 DESCRIPTION

Use Case 3 Short-term Prediction
Initiating Actor: User
Actors Goals: Get the short-term prediction price of the stock selected
Participating Actor: Server, Database
Preconditions: User is logged in and searched for a stock present in the database
Success End Condition: User is provided with the short-term prediction price of the stock selected.
Failed End Condition: N/A

Extension Point:
Flow of Events for Main Success Scenario: 1. User Opens the webpage and login 2. User searches for the stock he is interested in 3. User selects the short-term prediction

USE CASE #4 DESCRIPTION

Use Case 4 Long-Term Prediction
Initiating Actor: User
Actors Goals: Get the short-term prediction price of the stock selected
Participating Actor: Server, Database
Preconditions: User is logged in and searched for a stock present in the database
Success End Condition: User is provided with the long-term prediction price of the stock selected.
Failed End Condition: not enough historic data to perform a long term prediction of the stock.
Extension Point:
Flow of Events for Main Success Scenario: 1. User Opens the webpage and login 2. User searches for the stock he is interested in 3. User selects the long-term prediction

USE CASE #5 DESCRIPTION

Use Case 5 Generate Graphs
Initiating Actor: User
Actors Goals: View the charts for the selected stock symbols
Participating Actor: Server, Database
Preconditions: User is logged in and has selected the stock for which he wants the graph
Success End Condition: Graph is plotted for the values of the stock selected
Failed End Condition: Not enough stock data to generate graph
Extension Point:
Flow of Events for Main Success Scenario: 1. User Opens the webpage and login 2. User searches for the stock he is interested in 3. User selects the get graph option

USE CASE #6 DESCRIPTION

Use Case 6 Stock Decision
Initiating Actor: Server
Actors Goals: Decision to buy, sell or hold the stock to be displayed to the user
Participating Actor: Database, Google trends API
Preconditions: User is logged in and has selected the stock for which he wants the long-term prediction and opted for long-term prediction by electing the long-term prediction option.
Success End Condition: The decision based on the long-term prediction is displayed and Google trends is displayed for the user.
Failed End Condition: N/A
Extension Point:
Flow of Events for Main Success Scenario:

1. User Opens the webpage and login 2. User searches for the stock he is interested in 3. User selects the long-term prediction

USE CASE #7 DESCRIPTION

Use Case 7 Stock Details
Initiating Actor: User
Actors Goals: Get real time data and historic data of the selected stock
Participating Actor: Server, Database
Preconditions: User is logged in and has selected the stock for which he wants the details
Success End Condition: real time data and historic data of the selected stock is displayed
Failed End Condition: Current stock value present in database
Extension Point:
Flow of Events for Main Success Scenario: 1. User Opens the webpage and login 2. User searches for the stock he is interested in 3. User selects the get stock details option

3.3.4 ***Interactive Diagrams For Key Use Cases***

Use Case #1 Registration

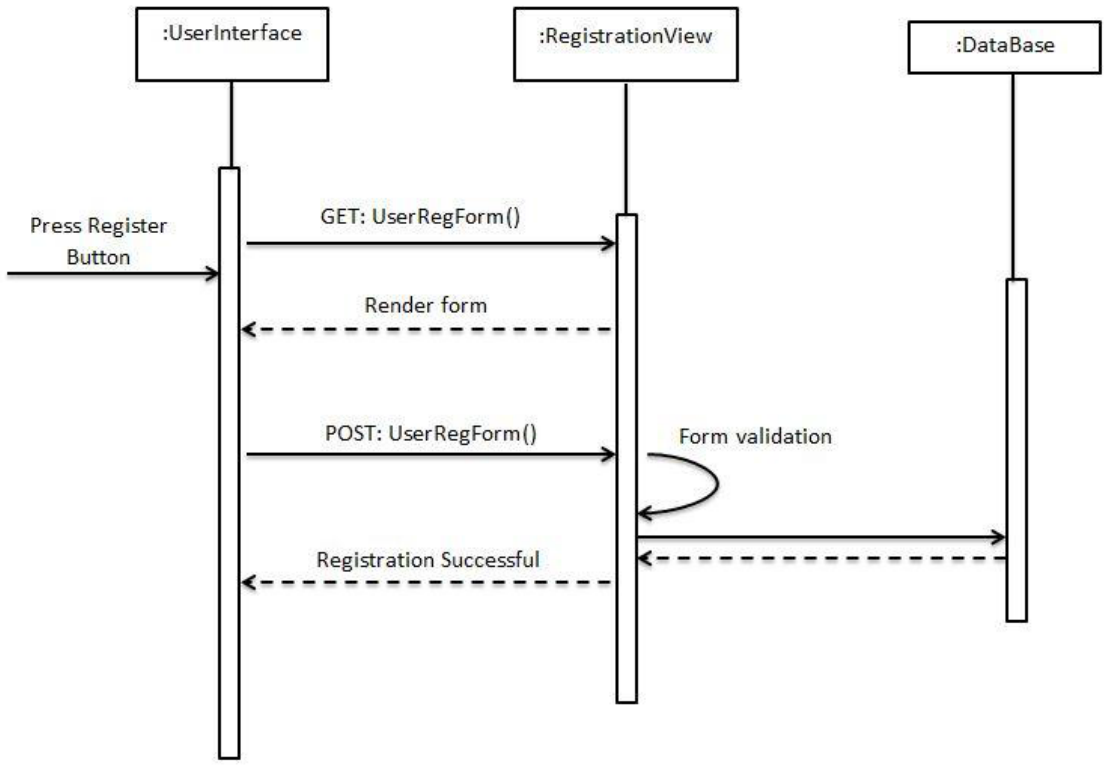


Figure 3. Interactive Diagram for Registration

The user needs to register to use the application. When the user generates the request to register, the server sends a registration form for the user. This registration form needs to be filled by the user according to the guidelines provided. When the user fills the registration form correctly authentication takes place and the data entered in the form is stored in the database.

Use Case #2 Login

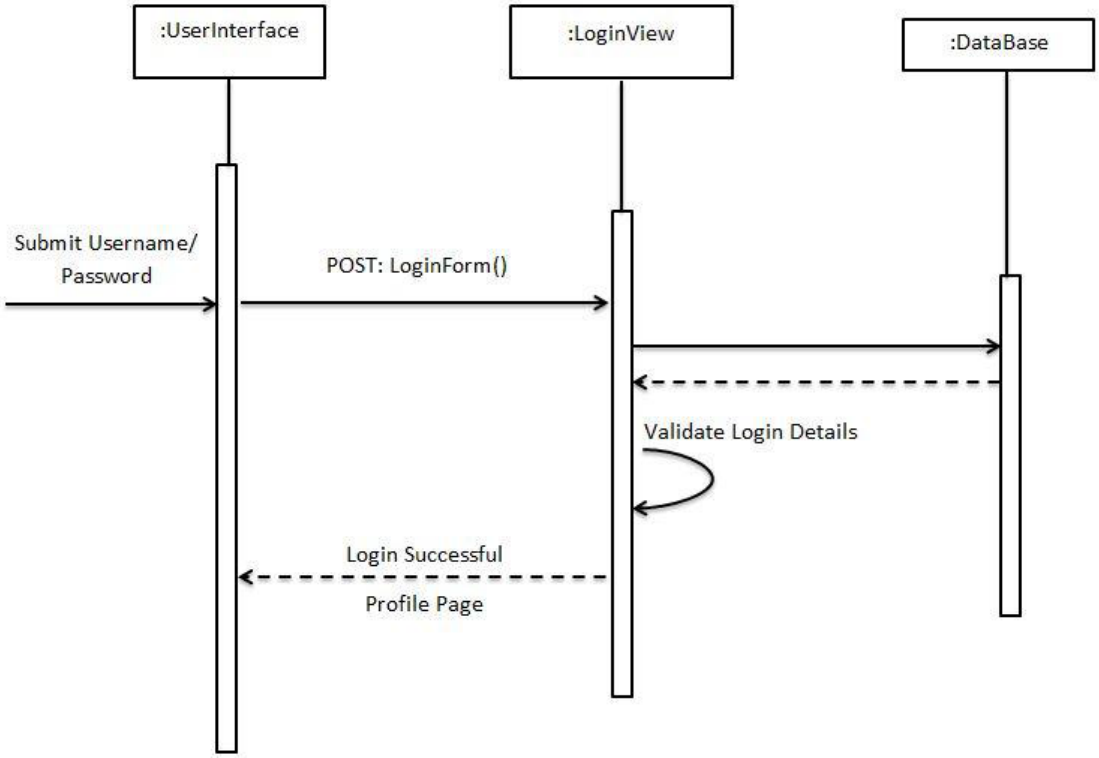


Figure 4. Interactive Diagram for Login

To be logged in the system, user is required to enter the correct username and password pair. When user provides the username and password pair, the pair is compared to the username and password pair stored in the database. If the pair matches with one of the pairs stored in the database, the user is successfully logged in to the system.

Use Case #5 Short-term Prediction

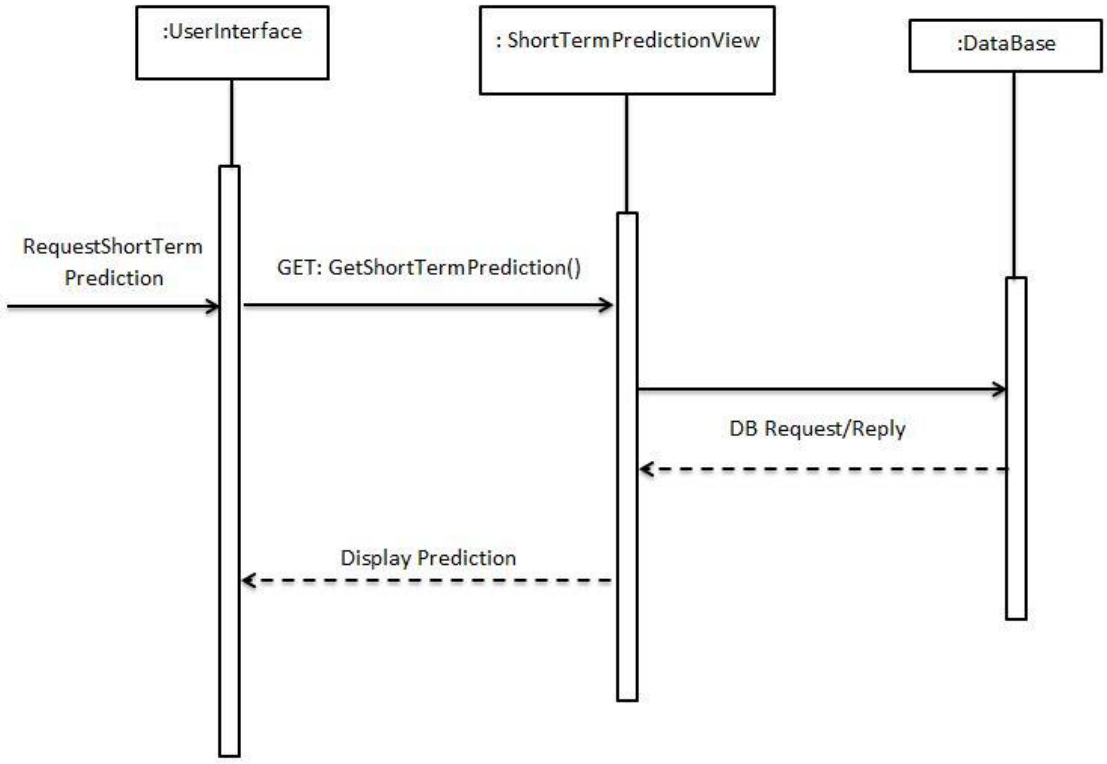


Figure 5. Interactive Diagram for Short-term Prediction

Short term prediction is done by us using Bayesian ppredictor and this prediction model takes short time to run. Hence we do it on demand. Whenever the user requests for the short term prediction, we read the latest values from the database and predict using Bayesian model. Then this prediction is given back to the user.

Use Case #6 Long-Term Prediction

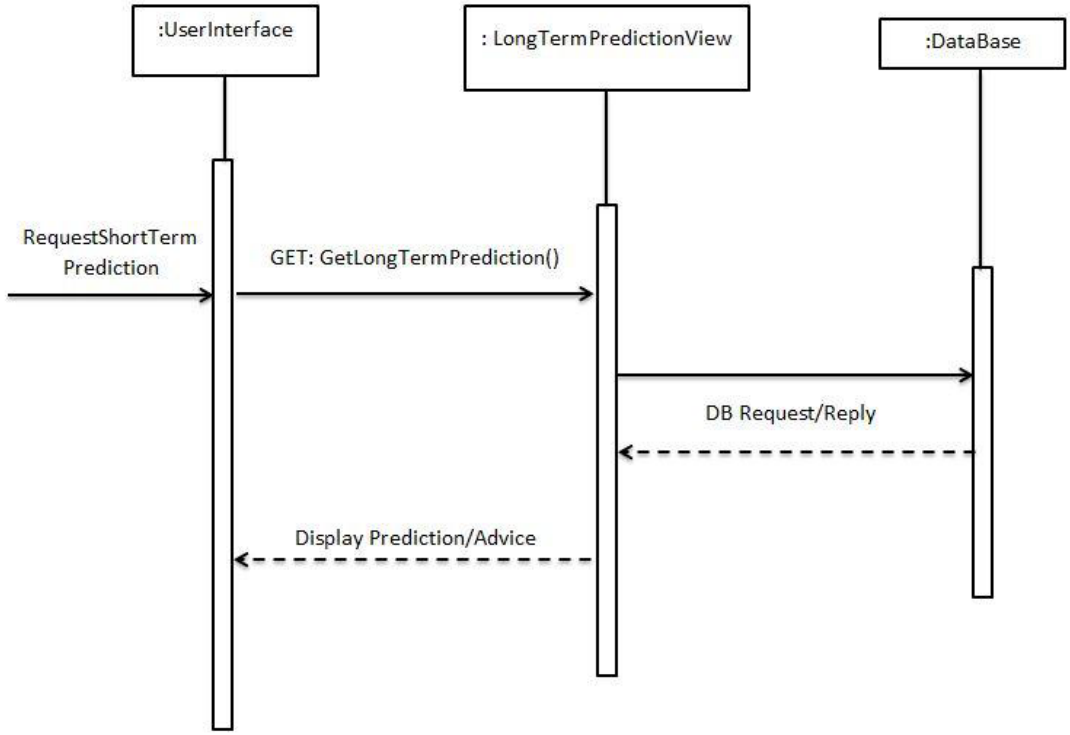


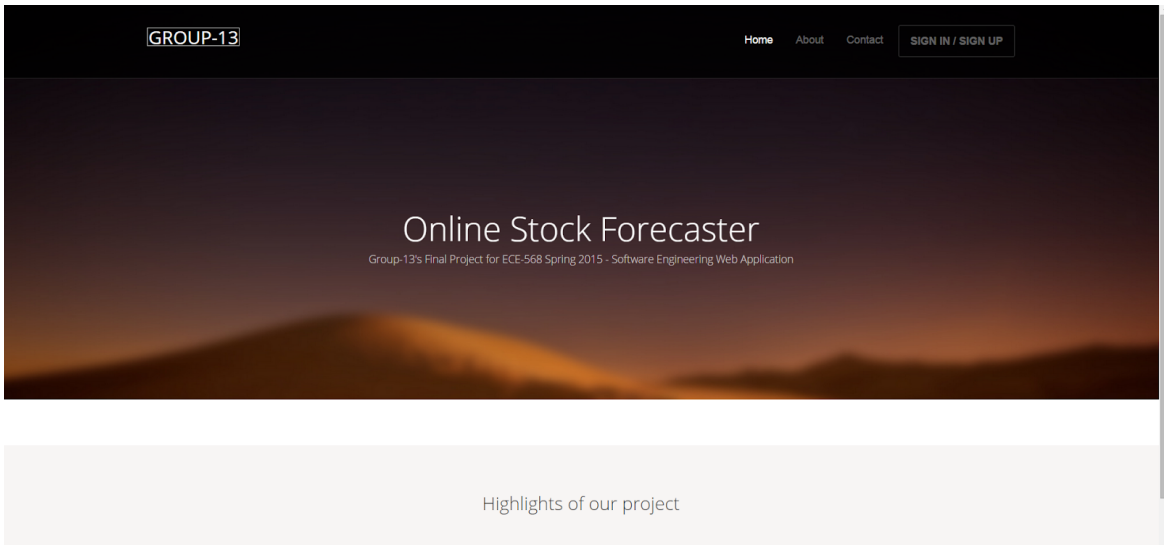
Figure 6. Interactive Diagram for Long-term prediction

Long term prediction is done by us using Artificial neural networks. This prediction model takes a long time to run. The input to this model is historical data which is collected once a day. Hence this model is also run once a day and preloaded into the database. Whenever the user requests for the long term prediction, the predictions are directly loaded from the database.

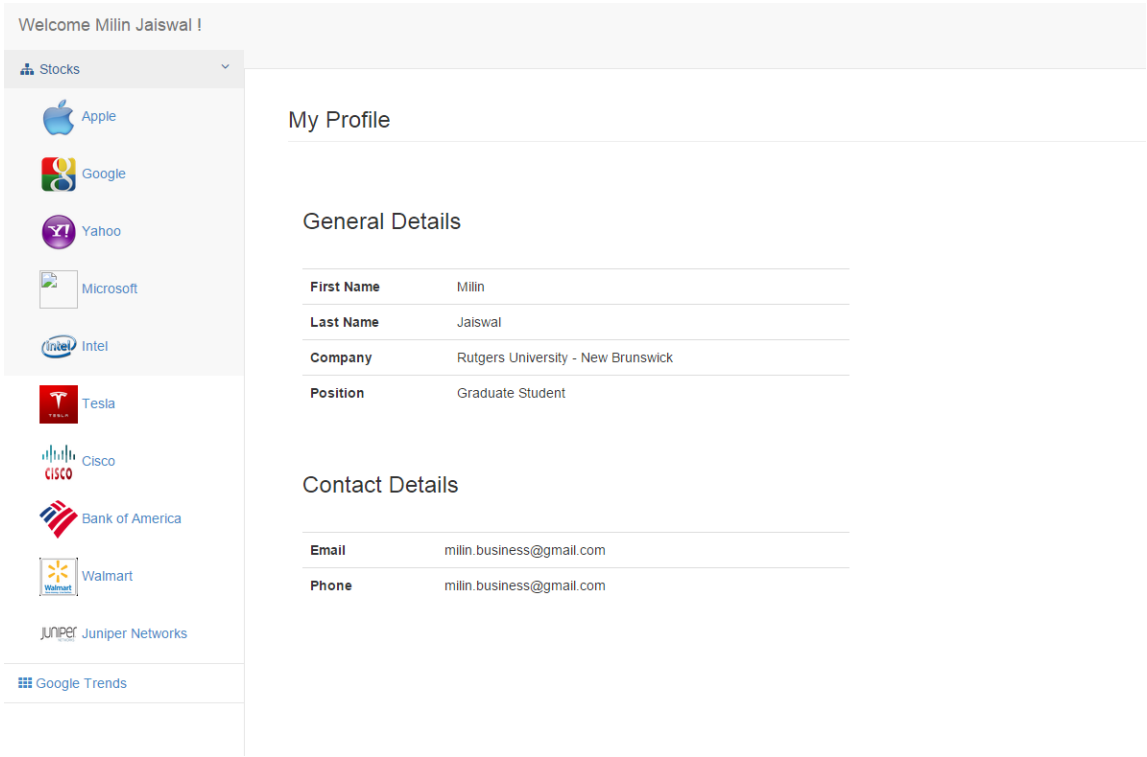
4. USER INTERFACE

This section gives a few screenshots of the user interface.

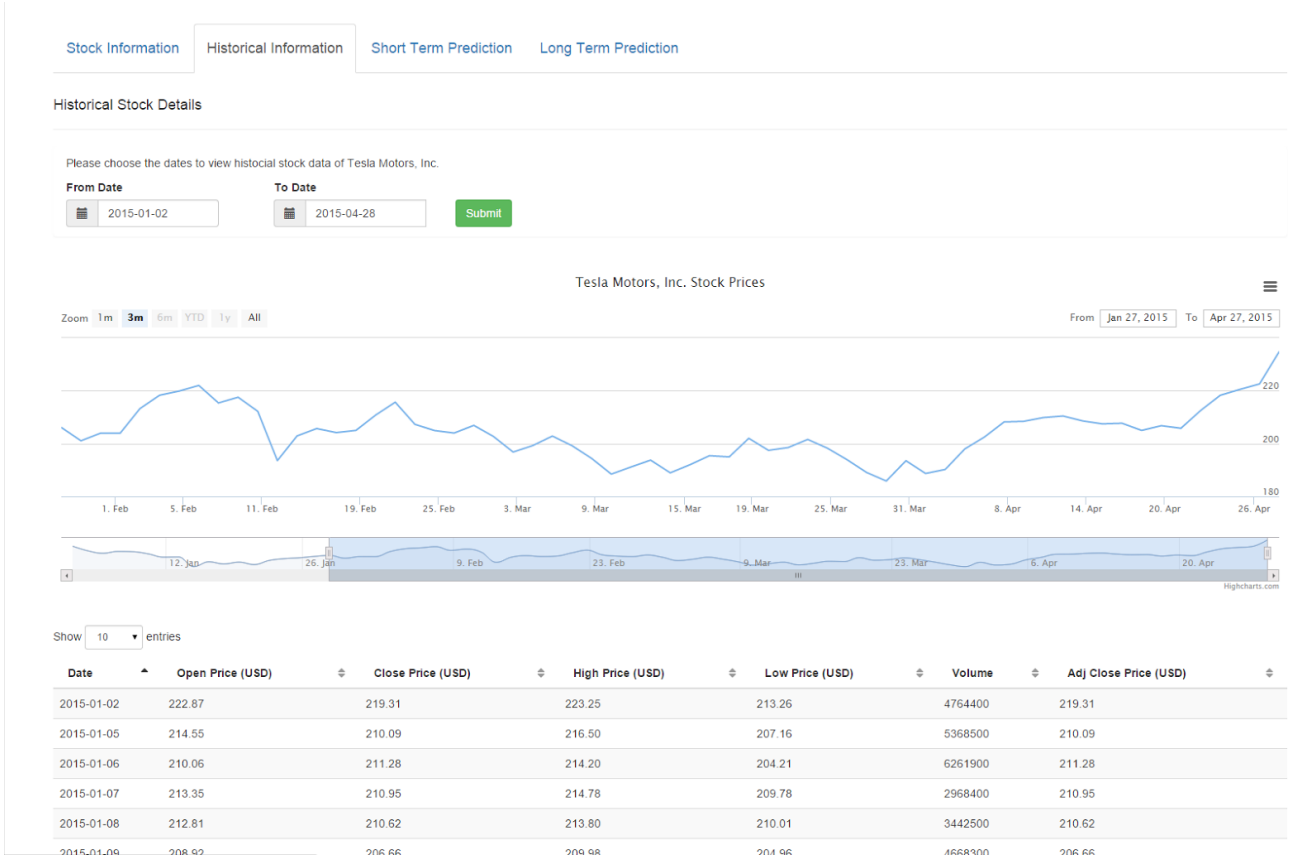
1. The landing page



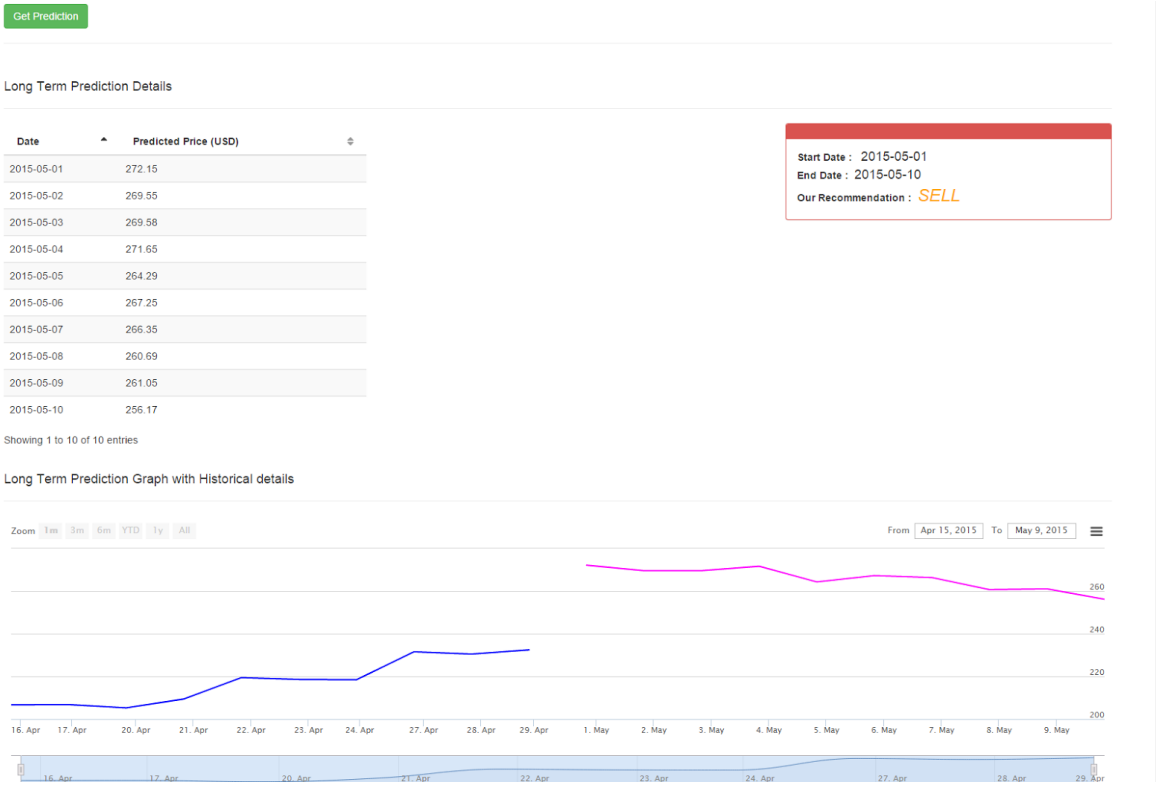
2. My Profile page



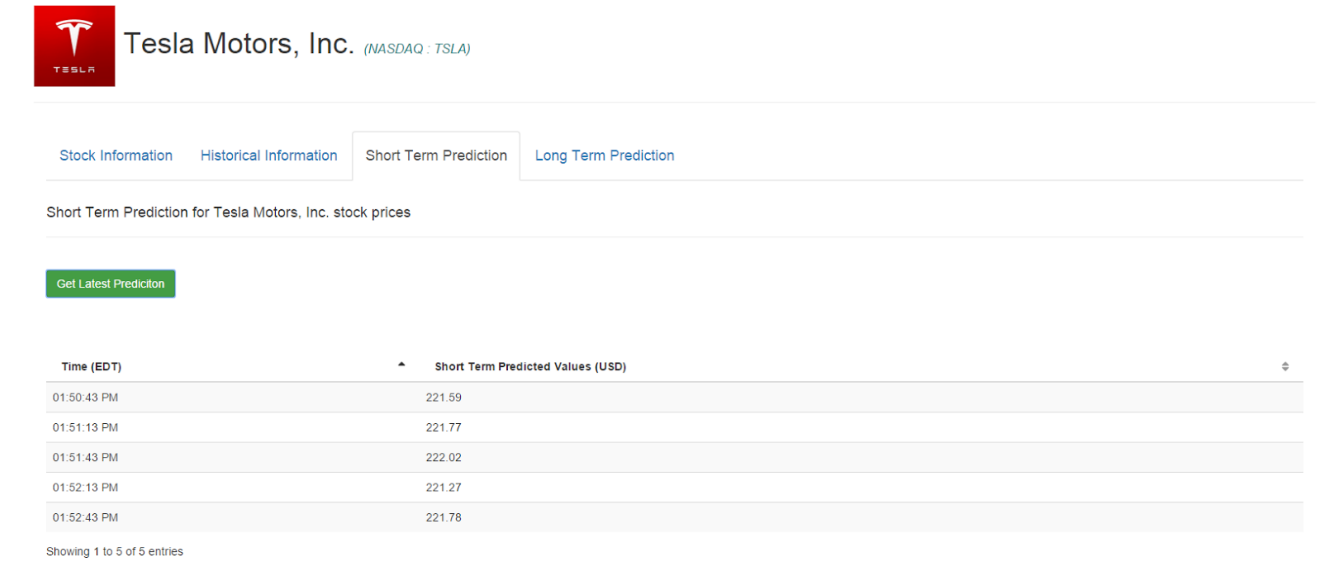
3. Historical information page



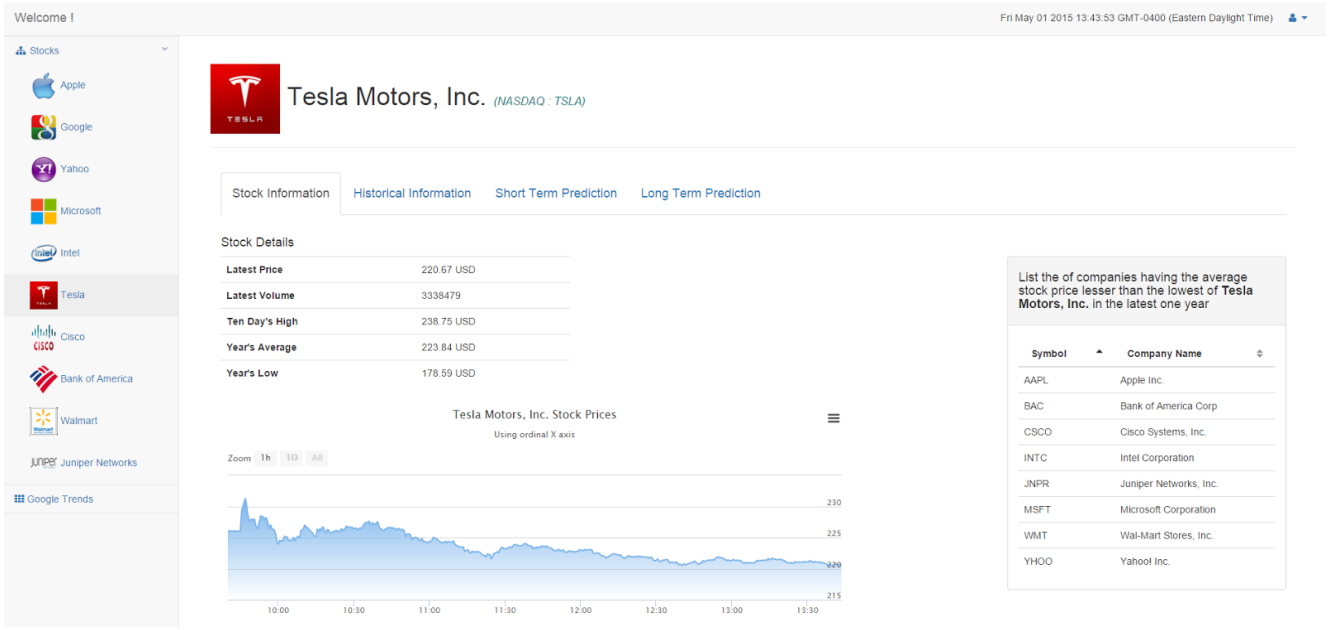
4. Long term prediction



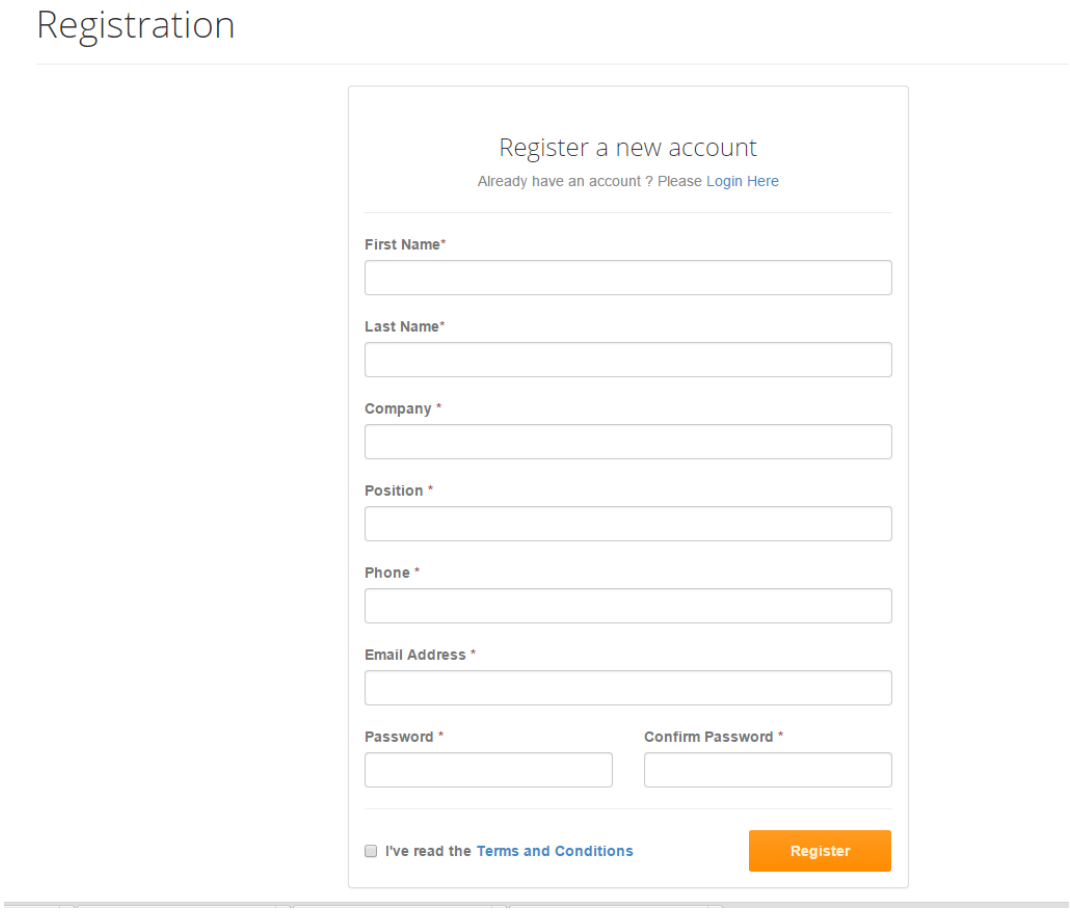
5. Short term prediction



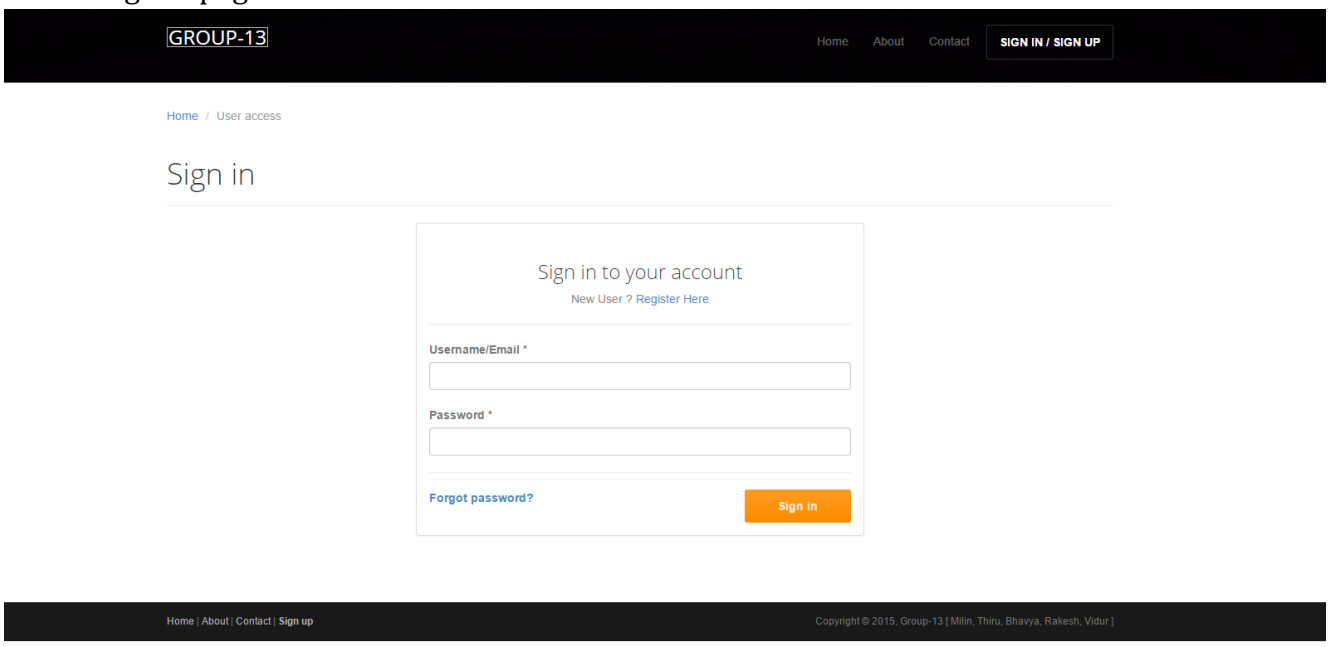
6. Stock Information page



7. User Registration page



8. User sign in page



5. PREDICTION STRATEGIES

The stock prices are very volatile and change frequently over time. There are ways to predict the future stock prices using few of the fundamental or technical analysis tools. A lot of work is going on in the field of using machine learning to take advantage of the technical analysis tools to predict the stock prices. This information equips the user with sufficient knowledge of the forecast of the stock prices to manage his shares wisely.

We use few of these techniques mentioned below for short term and long-term prediction of the stock prices.

5.1 Short Term Prediction

The developed application predicts the short-term variation in the stock prices to cater to the demands of the user who trades the shares on daily basis and would like to bank maximum profit from the fluctuations in the stock prices within a day. For the purpose of short-term prediction we use the Bayesian Estimator explained below.

5.1.1 Bayesian Estimator

Bayesian regression is one of the methods of linear regression where statistical analysis is performed based on Bayesian inference. When the regression model has errors that have a normal distribution, and if a particular form of prior distribution is assumed, explicit results are available for the posterior probability distributions of the model's parameters.

To implement Bayesian prediction, we make use of Bayesian curve fitting according to which, the training data x and t, along with a new test point x are provided. The end goal is used to predict the value of t, where x is the future time and t is the value taken on by the function at time x. The variables x and t signify two vectors constituting the training set for the prediction algorithm. Stated a different way, the problem becomes computing the distribution p(t|x, x, t). The expression is looking for the probability distribution of the (future value) given the future time, past times, and past values [Marsic, 2012].

The future stock price is given by the probability function

$$p(X \mid T, t, x) = N(X \mid m(x), s2(x)) \quad (1)$$

Where the mean and variance are given by

$$m(x) = \beta\varphi(x)^T S \sum_{n=1}^{N} \varnothing(x_n) t_n \quad (2)$$

$$s^2(x) = \beta^{-1} + \varphi(x)^T S \varnothing(x) \quad (3)$$

Here the matrix S is given by

$$S^{-1} = \propto I + \beta \sum_{n=1}^{N} \varnothing(x_n)\varnothing^T \qquad (4)$$

where I is the unit matrix, and we have defined the vector ϕ (x) with elements i(x) = xi for i = 0, . . . , M.

The above process is repeated for all values. From these the absolute error, average relative error are calculated. This process is repeated for the values of M from 0 to N-1 and M is selected at the value of N when the average relative error is least. β was chosen to be 11.1, while α was chosen to be 0.005 for implementing Bayesian estimation (Marsic, 2012).

5.2 Long Term Prediction

We provide long-term prediction of the stock prices for users who are willing to buy and hold on to their shares for a longer period of time. Long-term prediction is immune to the daily fluctuation of the stock price and follows a trend. For this purpose we use neural and machine-learning techniques to understand the trend and predict the future price of the stocks. There are various machine learning algorithms which are used for Stock prediction but amongst them the two most successful and commonly used are Support Vector Machines (SVMs) and Neural Networks (NNs). We carried out an initial literature survey of the two algorithms to figure out the best algorithm which suits our requirements. After a series of thoughts and calculations, we are using both the methods in long term prediction. Having said this, to make best use of the two we are using ANN for values prediction(ANN has proved results as discussed in the section 5.2.2 for predicting values) and SVM for predicting the stock decision(SVM is known for it's classification of data in machine learning).

5.2.1 No Free Lunch Theorem:

In machine learning, the **"No Free Lunch" theorem** states that there is no one model that works best for every problem. The assumptions of a great model for one problem may not hold for another problem, so it is common in machine learning to try multiple models and find one that works best for a particular problem. This is especially true in supervised learning or cross-validation is commonly used to assess the predictive accuracies of multiple models of varying complexity to find the best model. A model that works well could also be trained by multiple algorithms like **linear regression** could be trained by the **normal equations** or by **gradient descent**.

5.2.2 Artificial Neural Networks(ANN)

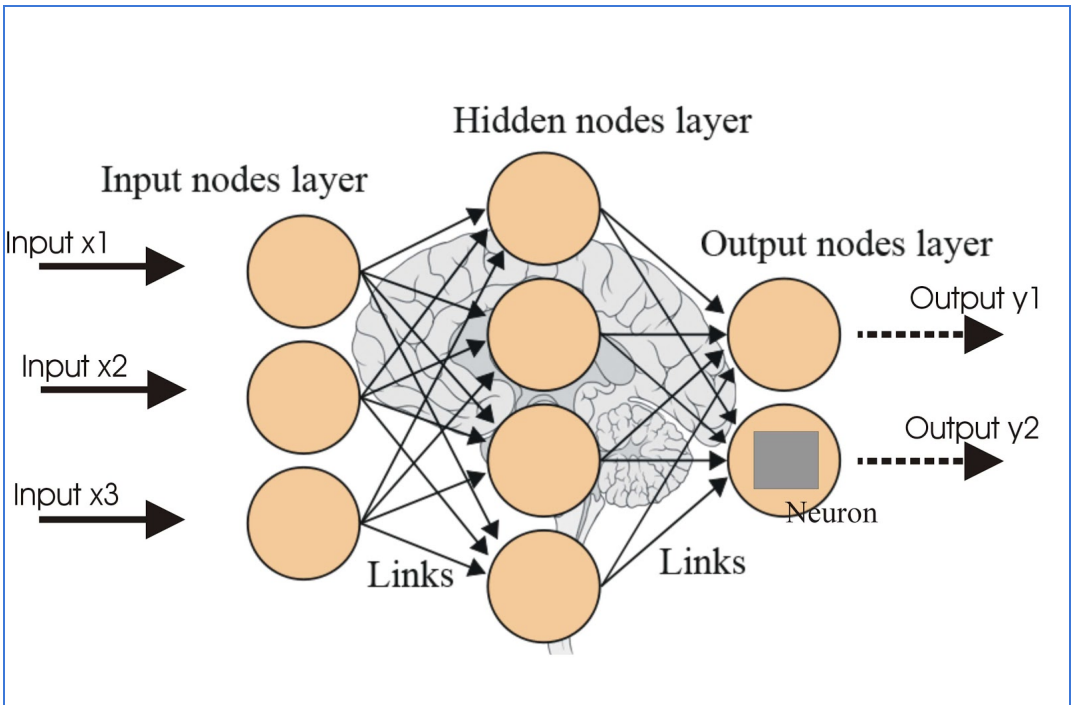


Figure 5.1

ANN is a computational model in light of organic neural systems. Neural Network comprises of interconnecting counterfeit neurons or hubs (programming builds that copy the properties of organic neurons) to order the information into characterized article class. By and large, NNs have no less than three layers; input(data), output(yield) and hidden layer (Fig.5.1). Every layer gets one or more inputs (object attributes) and sums them up to deliver an output that represents a defined object class. Generally, the sums at every node are weighted taking into account the interfacing branch. A NN is a versatile framework that changes its structure in view of outer or inward data that moves through the system amid the learning stage. During learning stage, the specimen records are handled and the yield is contrasted and the normal grouping of the record. The subsequent slip in the choice is utilized as a criticism to change weights related to phases. This is done iteratively until the framework effectively groups all the relevant information. The utility of NN models lies in the way that it can be utilized to infer a function (that classifies input data) based on training samples.

Feed Forward Networks:

A feedforward network is one in which the layer is made out of a set of inputs that feed data patterns to the system. Taking after the input layer there will be no less than one or more transitional layers, frequently called hidden layers. Hidden layers will be followed by an output layer, where the outcomes can be accomplished (Figure-2). In feedforward networks all connections are unidirectional.

Multi-Layered Processing Networks(MLP):

MLP systems are layered feed forward systems regularly prepared with static back-propagation. These systems, otherwise called backpropagation systems, are chiefly utilized for applications obliging static example characterization(shown in Figure 5.2). The back-propagation calculation chooses a preparation case, makes a forward and a regressive pass, and afterward rehashes until calculation joins fulfilling a prespecified mean squared errored

esteem. The fundamental point of interest of MLP systems is their usability and close estimation of any input/output map. The main disadvantage is that they train slowly and require lots of training data.

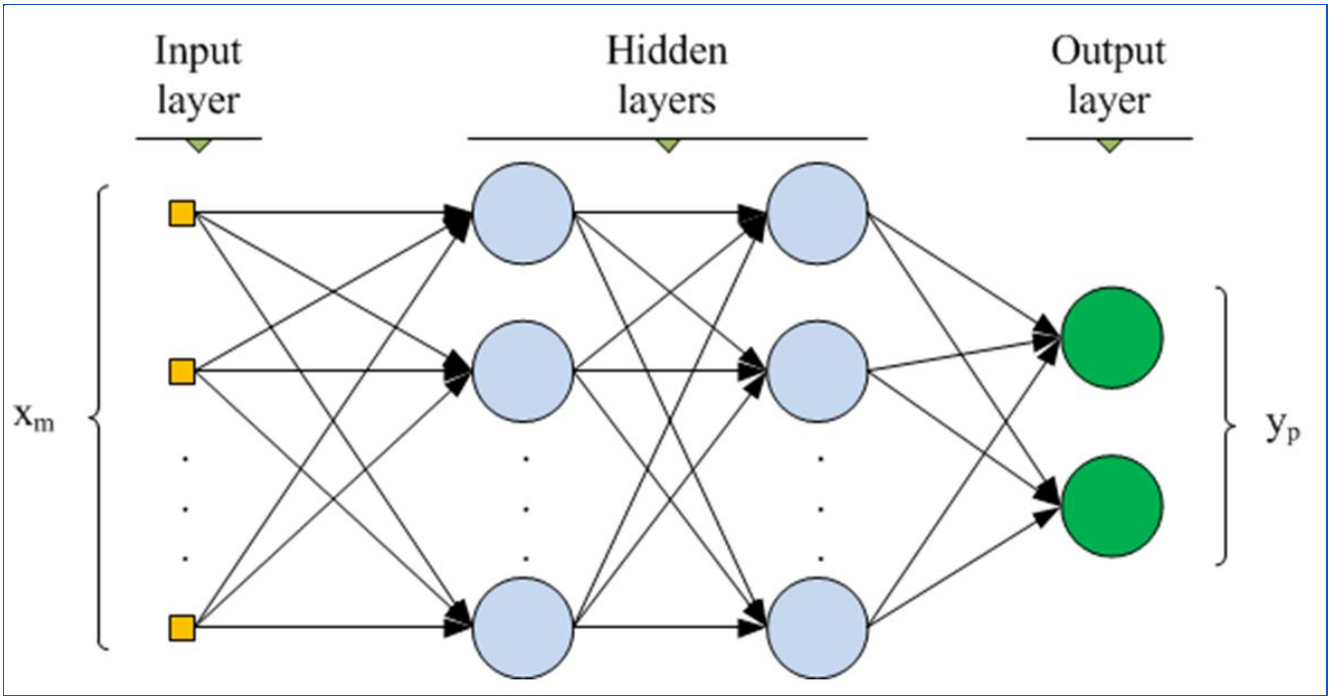


Figure 5.2 Multi Layered Processing Neural Network

Significance:

One of the first projects using ANN was by Kimoto and companions who had utilized ANN for the forecast of Tokyo stock exchange index. Mizuno and companions connected ANN again to Tokyo stock trade to foresee purchasing and offering signs with a general expectation rate of **63%**. Later in 1998, Sexton and companions deduced that use of momentum and start of learning at random points may solve the problems that may occur in training process. Phua and companions applied neural system with hereditary calculation to the stock trade business of Singapore and anticipated the business sector heading with a precision of 81%. In Turkey ANNs are basically utilized as a part of foreseeing financial failures. There has been no particular examination for forecast of Turkish stock exchange values.

Implementation:

After describing the significance of ANN in the above section, the current portion provides information about the ANN's implementation in our stock predictor project. We've followed the Multi Layered Processing(MLP) Neural Networks approach to estimate the future stock values of succeeding 10 days. ANN should be trained with previous or sample data before actually injecting the current dataset. After the NN learns the prediction pattern in the training phase the 'to-be' predicted data set should be passed. So, the second phase is known as testing phase. The project is implemented in Java with the neuroph.jar libraries support.

Training Phase:

- Input Data: For a given stock one year's data is available in the database. 256 historic data of the selected stock will serve as the raw input for the prediction.

- Input Layer: As shown in Figure 5.2, data should be passed as nodes(each node de-marked by x, a total of xm nodes). A group of **20** historic values are counted as entries in one node.
- Hidden Layers: There are no hard and fast rules in selecting the number of neurons or hidden layers. For a linearly separable data no hidden layers are actually required. But stock market data on a historical basis is less probabilistically linear. So, the hidden layer count is chosen by observing the training phase. In most of the stocks we predicted the common value for hidden layers is **15**.
- Output: Since, we are predicting future **10** stocks starting from the current date by considering previous 1 year's historical data, there are 10 output nodes in our system.
- The sample configuration looks as below:

```
public static void main(String[] args) throws ClassNotFoundException {

    int maxIterations = 10000;
    NeuralNetwork neuralNet = new MultiLayerPerceptron(20, 15, 10);
    ((LMS) neuralNet.getLearningRule()).setMaxError(0.001);
    ((LMS) neuralNet.getLearningRule()).setLearningRate(0.7);
    ((LMS) neuralNet.getLearningRule()).setMaxIterations(maxIterations);
    TrainingSet trainingSet = new TrainingSet();
```

Figure 5.3

- After creating the MLP layer, the training data set should be prepared by leaving the last 20 records(because we are interested in predicting these future 10 values from the last node so this becomes the test data) from the historic data .
- All the trained nodes should be learnt in the same thread. This step brings the training phase to an end.

Testing Phase:

- The last 20 records from the historic data extracted are created as a new learning set.
- Once the test data set is ready by calculating the neural network statistics with the calculate() method of Neuroph.jar's library the 10 output values will be fetched.
- The output values are stored into the ***'longterm_prediction_values'*** table of the database.
- Since the historic data gets updated only once a day, we're executing the ANN process one time a day.

5.2.2 Support Vector Machine (SVM)

SVM utilizes linear model to actualize nonlinear class limits through some nonlinear mapping the input vectors x into the high-dimensional feature space. A direct model developed in the new space can represent to a nonlinear choice limit in the original space. In the new space, an ideal differentiating hyperplane is developed. Hence, Thus, SVM is known as the algorithm that -nds a special kind of linear model, the maximum margin hyperplane as shown Figure 5.4. The most high edge hyperplane gives the greatest partition between the choice classes. The

training examples that are nearest to the high edge hyperplane are called support vectors. All other training examples are irrelevant for de-ning the binary class boundaries.

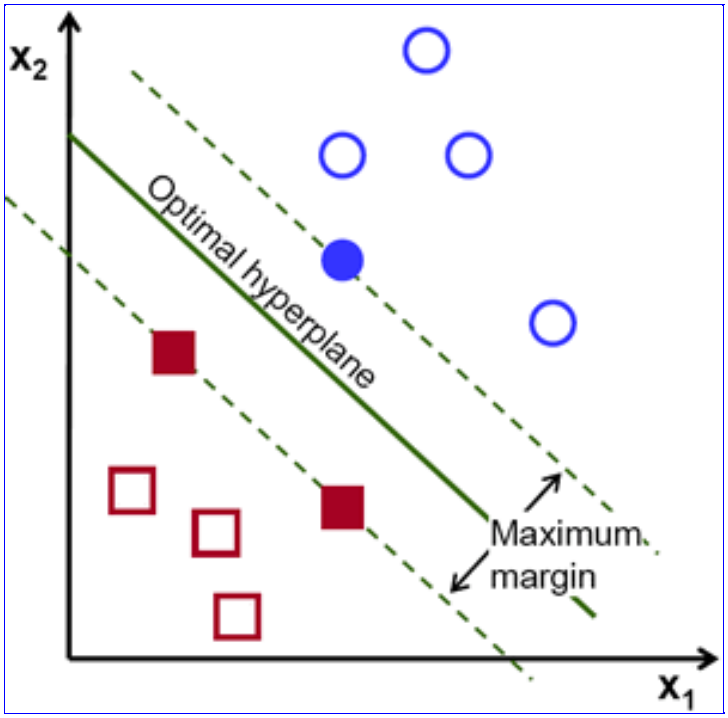


Figure 5.4

For the linearly separable case, a hyperplane separating the binary decision classes in the three-attribute case can be represented as the following equation:

$$y = w0 + w1x1 + w2x2 + w3x3; \quad (1)$$

where y is the outcome, xi are the attribute values, and there are four weights wi to be learned by the learning algorithm. In Eq. (1), the weights wi are parameters that determine the hyperplane. The maximum margin hyperplane can be represented as the following equation in terms of the support vectors: $y = b + iyix(i) \cdot x;$

Significance:

Technical examination of stocks and patterns has been utilized by stock brokers for quite a long time. In spite of the fact that it doesn't promise achievement and is not 100% precise, it is still one of the two key techniques for investigating stock costs, alongside central investigation. Specialized investigation of stocks and patterns utilizes outline examples, for example, head-and-shoulders arrangements and drop-base-arouses. SVM can be used as a tool to study these technical pattern.

Implementation:

We aim to predict the decision to buy/sell/hold for the users interested in long term stock investments by using SVM. The implementation of SVM based predictor is using MATLAB and SPIDER machine learning toolbox. SVM toolbox is used for the MATLAB numerical environment. The toolbox will be used for creating models for regression and classification using support vector machines. It has several machine learning algorithms and variations in SVMs. We will be training our Support Vector Machine using the train and test functionality provided by SPIDER. As we plan to predict 4 patterns in the data we use one_vs_rest algorithm(described in page 29) to find out the Support Vectors and the hyper plane as discussed in earlier sections.

Patterns Used:

Our system has been developed to detect various technical patterns for our long term prediction. They are as follows:

o Cup and Handle: A pattern on bar charts resembling a cup with a handle. The cup is in the shape of a "U" and the handle has a slight downward drift. The right-hand side of the pattern has low trading volume. It can be as short as seven weeks and as long as 65 weeks.

o Head and Shoulder: A technical analysis term used to describe a chart formation in which a stock's price:

1. Rises to a peak and subsequently declines.

2. Then, the price rises above the former peak and again declines.

3. And finally, rises again, but not to the second peak, and declines once more.

The first and third peaks are shoulders, and the second peak forms the head.

o Ascending Triangle: A bullish chart pattern used in technical analysis that is easily recognizable by the distinct shape created by two trendlines. In an ascending triangle, one trendline is drawn horizontally at a level that has historically prevented the price from heading higher, while the second trendline connects a series of increasing troughs. Traders enter into long positions when the price of the asset breaks above the top resistance.

o Descending Triangle: A bearish chart pattern used in technical analysis that is created by drawing one trendline that connects a series of lower highs and a second trendline that has historically proven to be a strong level of support. Traders watch for a move below support, as it suggests that downward momentum is building. Once the breakdown occurs, traders enter into short positions and aggressively push the price of the asset lower.

Multilabel Classification:To generate a decision (buy/sell/hold) the historical data of the stock selected should be analyzed technically based on the above 4 patterns using SVM. Classic SVM

divides the data into only two patterns. Now that we have 4 different patterns the data needs to be fitted into Multilabel classification serves the purpose.

Multilabel classification assigns to each sample a set of target labels. This can be thought as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A text might be about any of religion, politics, finance or education at the same time or none of these. A sample multilabel classification is shown in the below Figure 5.4.

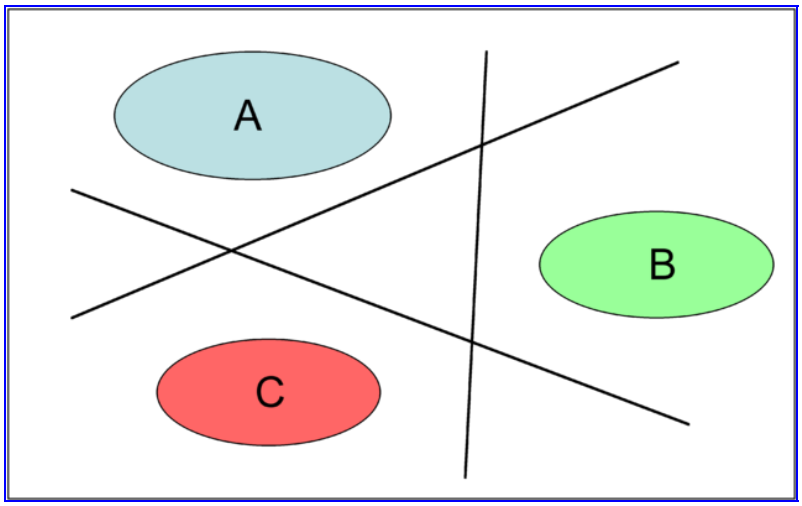


Figure 5.4

One-vs-Rest Classifier:

This strategy consists in fitting one classifier per class. For each classifier, the class is fitted against all the other classes. In addition to its computational efficiency (only n_classes classifiers are needed), one advantage of this approach is its interpretability. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy and is a fair default choice.

Structure of the implementation:

As mentioned in the above section, the classification code for the SVM application is written in MATLAB. To maintain the language standards across the application we have integrated Matlab with Java. So, the matlab scripts are called from the Java wrapper using Matlab Control API. As soon as the Control.java is executed for a particular stock it's previous 100 days data is loaded into a text file and this file will be read by stock predictor.m function. The data is nothing but the input for the analysis. Then, the input data is divided into chunks of 8 records and undergoes training phase(to find the relative pattern). If the input data fits into one of the 4-patterns the result will be buy or sell based on the pattern detected. Else, the decision is going to be hold. The decision is updated in the longterm_prediction_decision table by the Control.java class(shown in Figure 5.5) using JDBC.

```
public class Control {
public static void main(String[] args) throws MatlabConnectionException, MatlabInvocationException, IOException {

    MatlabProxyFactoryOptions options =
            new MatlabProxyFactoryOptions.Builder()
                .setUsePreviouslyControlledSession(true)
                .build();
        MatlabProxyFactory factory = new MatlabProxyFactory(options);
        MatlabProxy proxy = factory.getProxy();
        proxy.eval("addpath('/Users/Saibhavya/Documents/MATLAB')");
        proxy.eval("addpath('/Users/Saibhavya/Documents/MATLAB/spider')");
        proxy.feval("sctockpredictor_script");

        proxy.disconnect();
```

Figure 5.5

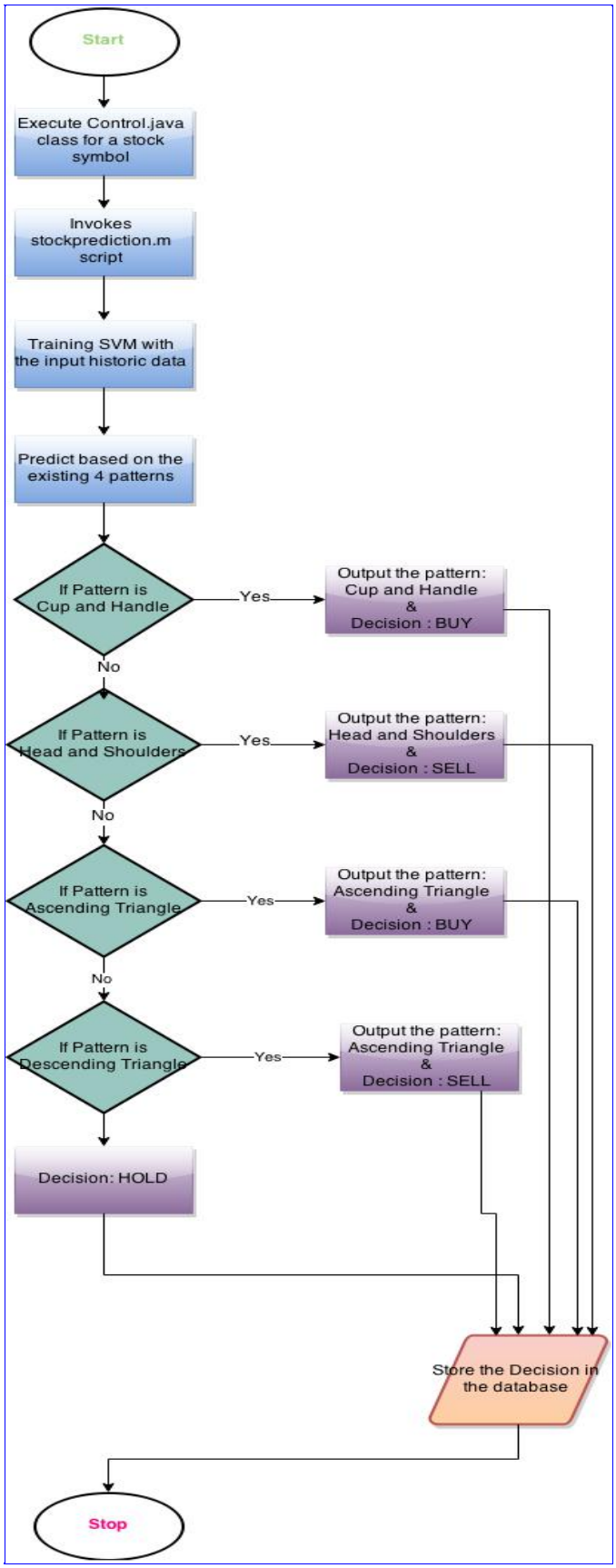
Matlab Scripts: stockpredictor_script.m : This is the main function which is called from the java file. It in-turn calls the rest of the functions.

traindata.m: This script contains sample data for the 4 patterns

trainstock.m: This script trains the data

test.m: This script test the input data for decision

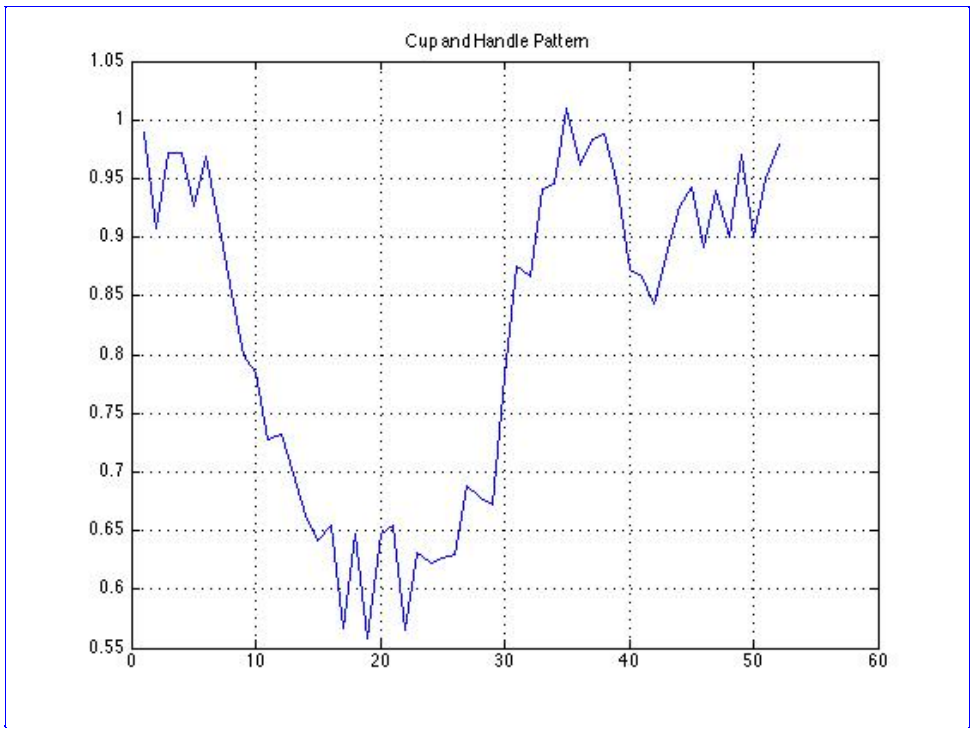
Sequential flow of SVM based predictor:



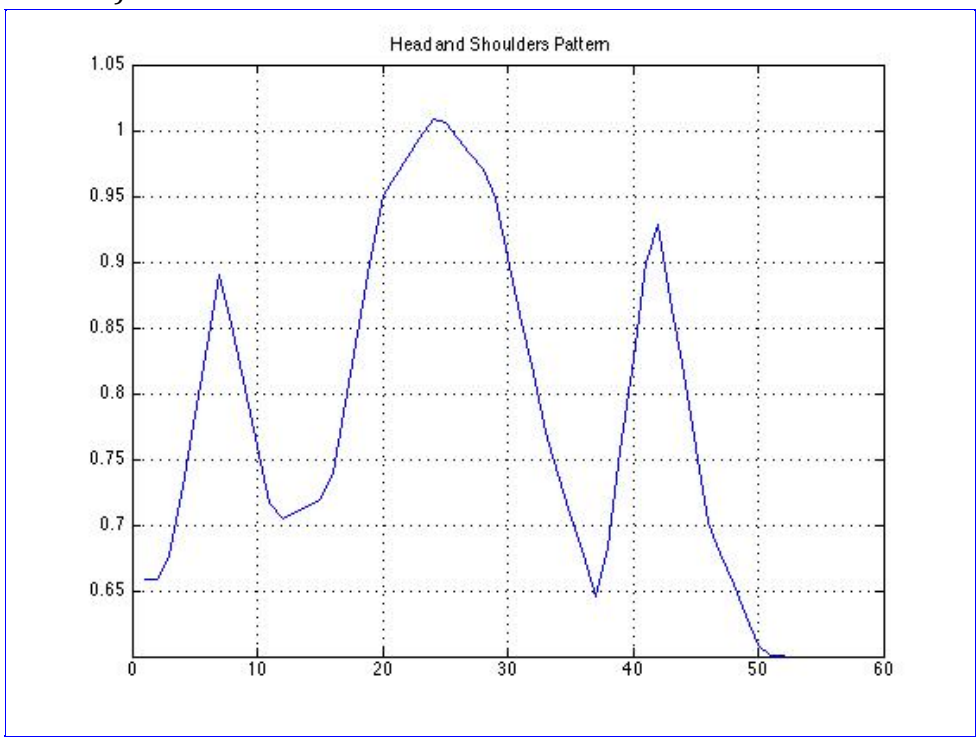
Sample patterns trained by SVM:

The following are the 4 training patterns plotted by our svmtrain.m script

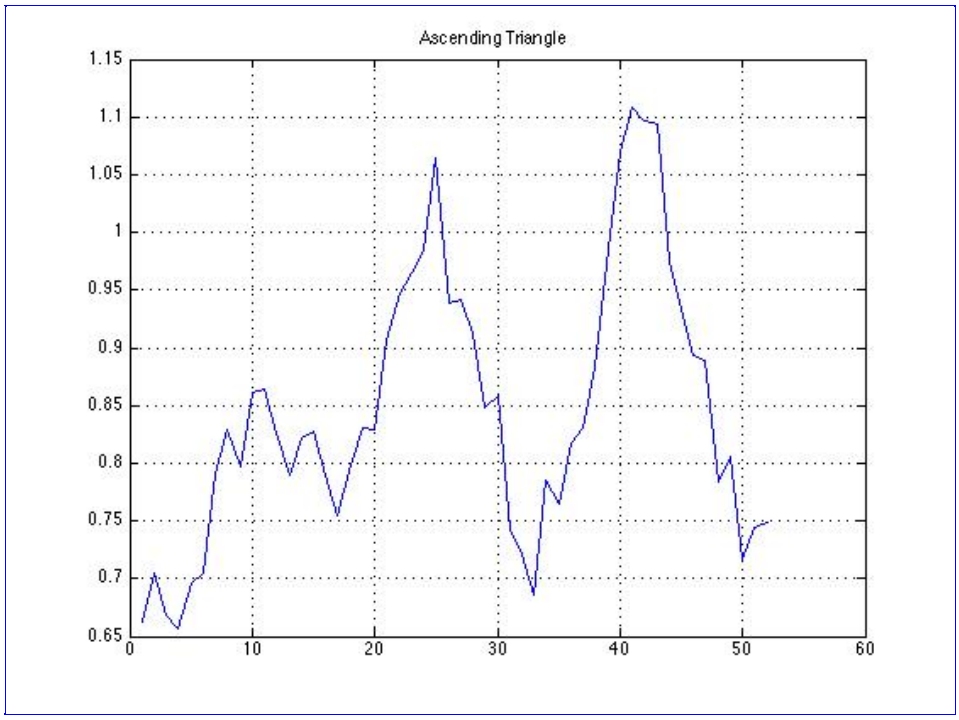
1) Cup and Handle Pattern:



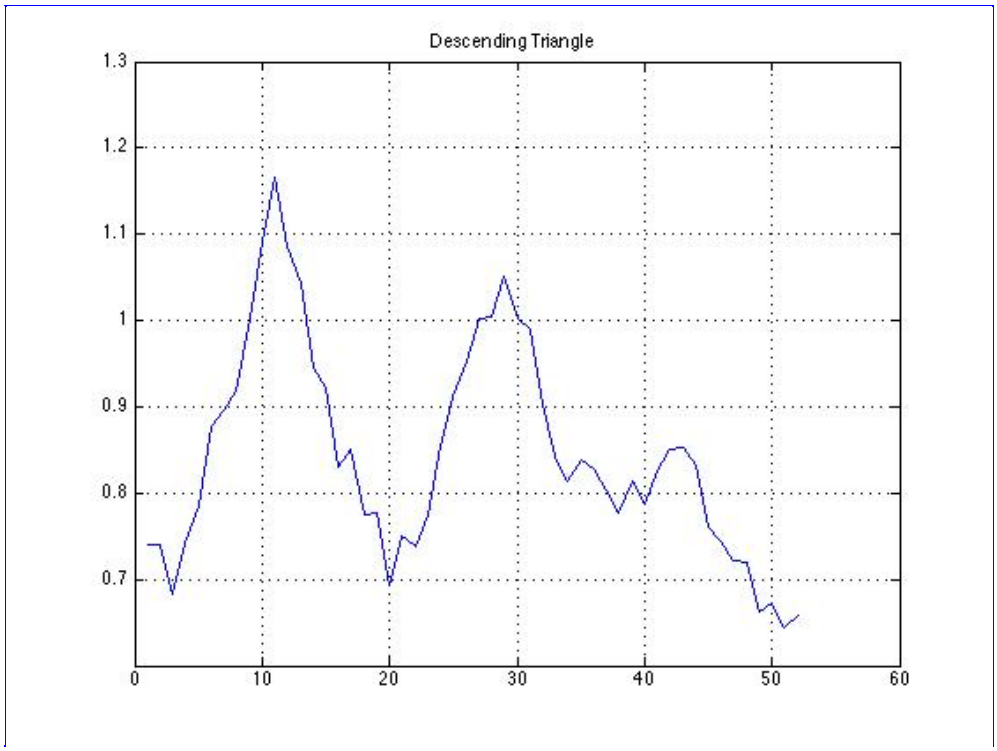
2) Head and Shoulders Pattern:



3) Ascending Triangle Pattern:



4) Descending Triangle Pattern:



Sample decision output in Matlab:

For any stock the Matlab's predicted output looks as below(Figure 5.6) based on the pattern detected:

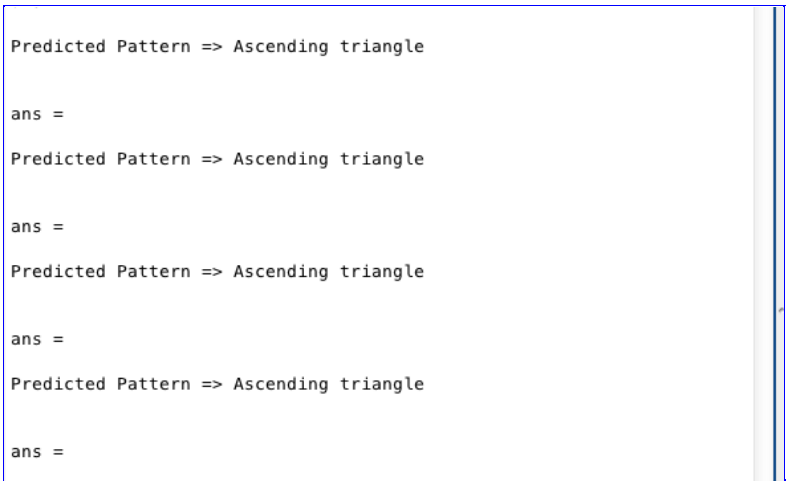

Predicted Pattern => Ascending triangle
ans =
Predicted Pattern => Ascending triangle
ans =
Predicted Pattern => Ascending triangle
ans =
Predicted Pattern => Ascending triangle
ans =

Figure 5.6

6. Overall Application Backend Details

6.1 Database Tables

This section shows the database tables that we have used in our project. We have used 7 tables.

stocks

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
symbol	varchar	10	NO	
name	varchar	30	NO	

real_time_data

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
symbol	varchar	10	NO	
date	date		NO	
time	time		NO	
price	numeric		YES	10
volume	int4		YES	32

historical_data

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
symbol	varchar	10	NO	
date	date		NO	
open_price	numeric		YES	10
close_price	numeric		YES	10
high_price	numeric		YES	10
low_price	numeric		YES	10
volume	int4		YES	32
adj_close_price	numeric		YES	10

longterm_prediction_decision

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
stock_symbol	varchar	10	YES	
stock_decision	varchar	10	YES	

longterm_prediction_values

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
stock_symbol	varchar	10	YES	
date	date		NO	
stock_value	float8		YES	53

user_login

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
first_name	varchar	20	YES	
last_name	varchar	20	YES	
email_id	varchar	50	NO	
phone_number	varchar	11	YES	
company	varchar	20	YES	
position	varchar	10	YES	
password	varchar	20	YES	

shortterm_prediction_values

column_name	data_type	character_maximum_length	is_nullable	numeric_precision
stock_symbol	varchar	10	NO	
time	timez		NO	
stock_value	float8		YES	53

6.2 Web Services

This section describes the various web services built by us. All our web services return JSON and some of them accept POST data in the form of JSON.

1. The real time data API (GET)
http://machine_name/com.stockprediction/rest/services/realtimedata?symbol=symbol
This API gives the real time data (time, price and volume)of the stock mentioned in the symbol argument of the query string for the present day from 9AM to the current time.

2. The stock details API (GET)
http://machine_name/com.stockprediction/rest/services/stockdetails?symbol=symbol
This API gives the details of the stock mentioned in the symbol argument of the query string. Details include year low, year average, latest price, latest volume and ten day high price.

3. List of stocks' details (GET)
http://machine_name/com.stockprediction/rest/services/list
This API gives the year low, year average, latest price, latest valume and ten day high price for all the stocks in our database.

4. List of stocks whose average is lesser than lowest of stock (GET)
http://machine_name/com.stockprediction/rest/services/listavglowthan?symbol=symbol
This API gives the list of stocks whose average price is less than the lowest of the stock metioned in the query string by the symbol.

5. Historical data (GET)
http://machine_name/com.stockprediction/rest/services/historicaldata?symbol=symbol
This API gives the historical data for the past 1 year for the stock mentioned in the query string by symbol.

6. Short term prediction (GET)
http://machine_name/com.stockprediction/rest/services/shortterm?symbol=symbol
This API gives the short term prediction for the next 5 minutes for the stock mentioned by symbol in the query string.

7. Long term prediction (GET)
http://machine_name/com.stockprediction/rest/services/longterm?symbol=symbol
This API gives the long term prediction for the next 10 days for the stock mentioned by symbol in the query string.

8. User Registration (POST)
http://machine_name/com.stockprediction/rest/services/userregistration
This API accepts details of the user - first name, last name, company, position, phone, email, password. It creates a record in the database.

9. User login (POST)
http://machine_name/com.stockprediction/rest/services/userlogin
This API accepts the email and the password of the user. If the password matches with the record in the database, it returns a JSON object with authenticated attribute as true along with the details of the user.

7. Fun Feature: **Google Trends**

As a fun feature the user can view the google trends chart for the past 5 years for any stock. The idea behind this feature is that, a high trending value for any name on Google is directly proportional to the popularity of the topic(interest of the topic over time). This concept can be made use as a factor for weighting the buy/sell/hold decision by the user. To illustrate how google trends help consider the below trending chart for the AAPL stock(Figure 7.1). The pointer in the figure shows a value of 58 and a hike marked with letter “B”. This is due to the release of Iphone 6plus(September,2014). The trend gradually decreased by December 2014. It’s a gradual decrease but not a sudden spike like the point “E” in the image. Proportionally, the stock price has reached all time high between Jan-2014 and October-2104 during September-2014. Hence, Google trends can be made use as one of the factors before deciding to buy or sell or hold a stock.

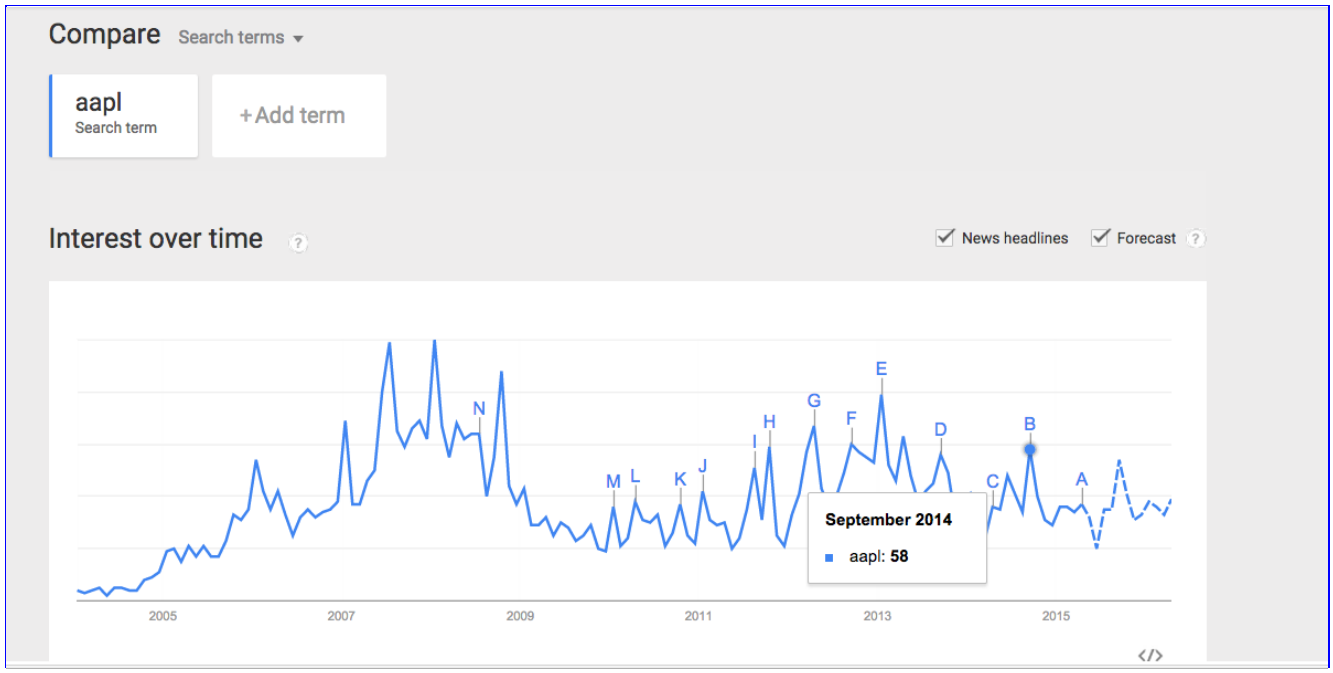


Figure: 7.1

In our application, a user can view a google trend for a stock by selecting the stock's name. Additionally, a user can compare the trending values of multiple stocks(as shown in the Figure: 7.2)

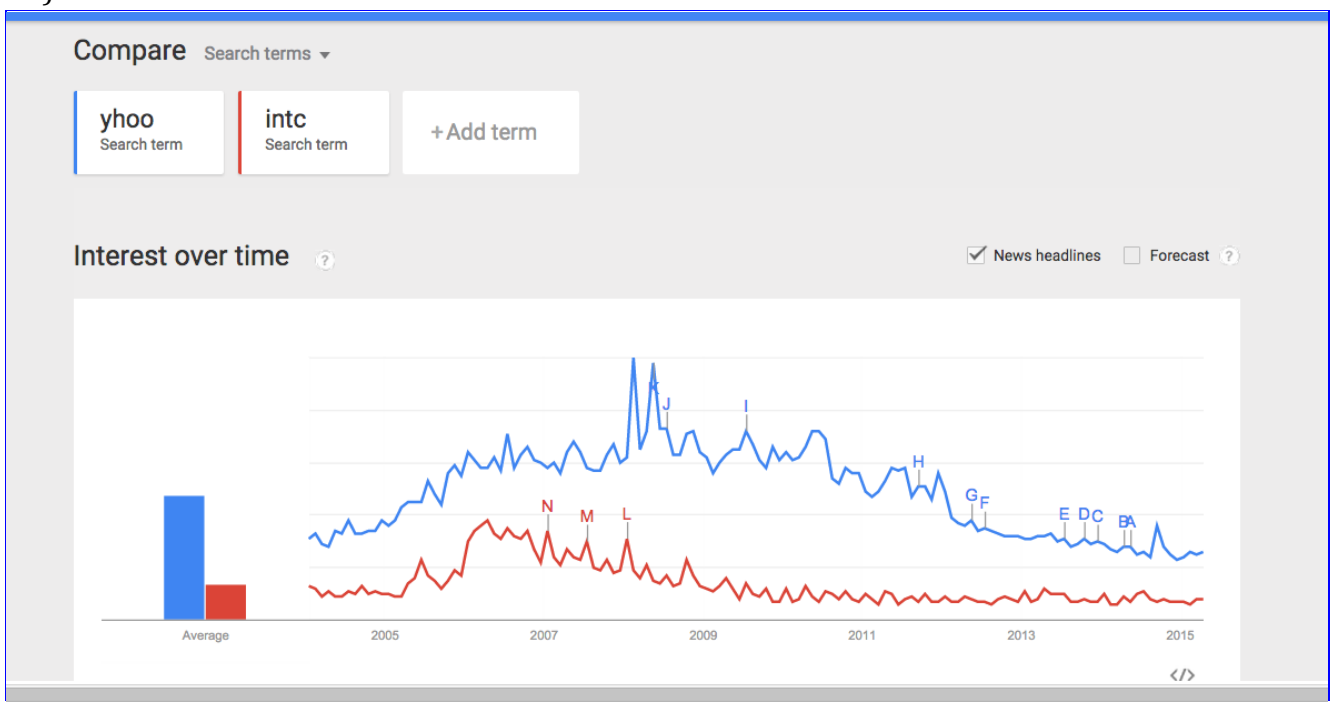


Figure 7.2: Comparing YHOO and INTC trends

8. CONCLUSION

Predicting stock markets has always been a topic under research. The simple reason being no investors will be willing to lose money by investing in stocks which trend down during the time frame the investor is interested in. We targeted to build a system that predicts the stock both in short term and long-term which is robust and has many salient features. In short-term, we are delivering the stock values for the next 5 minutes with a separation of 1 minute between each entry. In long-term, adding to the advantage of predicting the next 10 days stock prices, we are also suggesting the user with a decision. Following **no free-lunch theorem's** hypothesis we implemented two different algorithms for long-term prediction.

"A **picture** is worth a thousand **words"-Adrienne Carlson.** As this quote says, we believe that representation of the predicted data helps the user understand better about the predictions. So, the user can see and compare historical data(past 15 days) and predicted data(next 10 days) for any stock of their choice in our application. Analysis on stocks is a never ending task. We came up with a new idea of using google trends in providing insight to the user about the stock trending patterns. Collaboratively all these features together contributed in building a smarter prediction application.

9. FUTURE WORK

In extension of the current work we wish to achieve the following tasks in future:

- In the short term prediction, the values vary with a minor difference for the reason that data is collected at an interval of 30 seconds. To enhance the accuracy of the system along with the existing Bayesian Curve Fitting strategy we want to formulate a second strategy. Short term decisions are not suggested in the current system due to the above stated reason(short intervals and less variation of data). In the extension of work, we want to test and propose a prediction decision strategy in short term.
- Right now, Google trends are implemented as a fun feature in our application. Because of the capability of the trending in estimating stock price as mentioned under section 7, we would like to use it as a prediction strategy along with the existing systems in long term.
- Additional user features like to check the past history of predictions for a stock is also on our bucket list as a future task.