ECE:568

Software Engineering of Web Applications

Final Project Report

Group 16

Instructor: Shiyu Zhou

Contributions

All team members contributed equally.

Table of contents

1. Statement of Requirements

Problem Statement

Share market has always been green lands for businessmen to gain capitals and endorsements by selling the shares associated with their organization. It has also been the place for investors, all novices and matured to make good profit by selling these shares at the appropriate time. Although all of this sounds simple and uncomplicated, dealing with shares and their random varying behaviour in the market is a factor which even the most experts fail to predict accurately.

Motivation

For day to day trading, these risks are a major factor in keeping away new, eager yet unaccustomed, traders from the market. They lack the appropriate tools needed for the decision making process when compared to existing professional traders. This scenario is witnessed way too often and can range from the young wanting to have a steady stream of capital without the need to invest in a part time job whilst also being a full time student or even the old who are well established into life attempting to save up for retirement. In both of these cases, these are individuals who would gladly consider trading stocks but many a times shift away from the idea due to the lack of time, resources, and knowledge in the field.

To the college students amidst us, the idea of “gambling” in the stock market is fear inducing. Today, to turn this “gamble” into at most an informed estimate we would have to go through hours upon hours of research to first acquire the knowledge needed to even begin to understand how trading decisions are made. Understanding is only the first hurdle, however. Once one knows how to read charts, number crunch historical prices, or acquire and comprehend a company's balance sheets amongst many other details, the act of actually performing these tasks remains, something easier said than done. As students, the time and energy needed for such undertakings simply does not exist. This dilemma is one of which anyone foreign to the field of stocks can relate to.

The long-term investors among us, on the other hand, rely on a research heavy way to invest. There methods entail analyzing the financial performance of many companies and choosing the ones that appear to have the best growth potential. Since their prospects are long term, it may seem at first that the volatility of the market can be of no help here. Even these types of investors, however, are looking for as much profit as attainable and therefore do not overlook purchasing a stock at the lowest price possible. Thus, they too rely on the technical analysis techniques of day to day traders. Even with their acumen of the field, however, number

crunching and attempting to eyeball trends off of charts for these split second decisions can not only be time consuming and stressful but are also prone to human error.

Most investment advisors suggest maintaining a diverse investment portfolio in order to reduce the risk of investment loss. It is said we should not “put all of their eggs in one basket” and thus should increase the quantity of our investments over various industries. With ten times the stocks, however, arises the ten times the difficulties stated previously. Added on are the problem of organizing and keeping track of each and every investment. For the new and eager day trader among us, if the decision making of one investment seemed fearful how would he/she orchestrate his/her choices when suggested to acquire multiple shares? Without a proper way to declutter and systemize these decisions, these individuals may never move forward from owning one specific type of share at a time and thus be vulnerable to losses due to sudden and unforeseeable events even if their initial investment decision was deemed to be wise.

To summarize, the leading complications that exists with using the stock market as a viable stream of capital is the fact that many newcomers are not knowledgeable with the financial concepts needed to make an educated trade decision. Even then, the knowledgeable as well, may find that their techniques are time consuming and stress inducing. When deciding to swallow a set of data and, based on multiple patterns and trends, spit out a conscious commitment that can be the difference between a great profit and a terrible loss, human error is bound to occur. When a myriad of companies are further introduced into the equation, these obstacles accumulate exponentially and the need of quick, clear , and organized information is paramount.

Vision

Thus, there is a need for a system which puts forward all this information is a simple and crisp manner. In a field which has been depended on advice from experts who practice techniques based on their experience and relevant historical data and thus having possible factor of human errors, using computers to automate these is something that would certainly cut down on error rates. Computer systems have known to automate approaches which traditionally were done manually and have yielded good results. In today’s world where people use smartphones and tablets on the go, an application for prediction and suggestion for trading decisions is something that is highly desirable.

This system will be able to predict stock prices for a given share based on analysis of historical data and live data for that particular stock. All prediction methodologies fall into one of the two categories namely, fundamental analysis and technical analysis. Fundamental analysis of a stock is related to the study of the organization linked to that stock. The biggest part of fundamental

analysis involves delving into the financial statements. It involves looking at revenue, expenses, assets,etc. and all the other financial aspects of a company. Fundamental analysts look at this information to gain insight on a company's future performance. On the other hand, Technical analysis is a method of evaluating securities by analyzing statistics generated by market activity, such as past prices and volume. Technical analysts do not attempt to measure a security's intrinsic value, but instead use charts and other tools to identify patterns that can suggest future activity. Data used in fundamental analysis for a stock are generated much more slowly than the price and volume data used by technical analysts, thus the system will primarily make use of technical analysis as a means for making predictions.

Using technical analysis requires the dealing with multiple numbers as well as analyzing multiple detailed charts. These are complex tasks that we do not want to waste our time. The use of computer systems should not only automate this process but also produce prediction results with the accuracy of that greater than a human. Each of these results should also come with a reasonable confidence value asserting to the user how probable the given decision of “buy”, “sell”, or “hold out/sit in” is to be correct. There are various methods to gauge stock trends, and as many as possible should be used to increase prediction accuracy. The result of each model should be displayed to the user with up to date information regarding the current stock, as well as an overall decision based on a weighted average of all models. All of this should be easily accessible.

All of these features should be organized into an easy to use user interface not only accessible to desktop browsers. Through these methods, we will gain the organization a suitable environment needed to manage multiple stocks. The current method of tracking both owned and interesting stocks is simply by remembering and memorizing. An automated method would ultimately relieve us from this task.

Due to budget and technical limitations, not all existing stocks can be accounted for and thus, only a limited amount can be predicted by this service. These concerns also restrict the frequency of predictions as well as the rate of updating current stock prices. The system should display the user with charts depicting predictions for chosen set of stocks. The system will also make use of indicators for keeping track of predictions. Based on analysis of previous historical data and live data the system will also provide a suggestion regarding whether to hold on or buy the stock at hand. Also, the system will give user latest news related to the stock so that the user is always well informed about current trends regarding the stock and will help him in making an informed and sound decision.

Glossary of Terms

Algorithms: a step-by step procedure for calculations.

Artificial intelligence: the intelligence exhibited by a machine or software, the branch of computer science that develops machines and software intelligence. This intelligence can perceive its environment and take actions that maximize its chances for success.

Closing price: the final price at which a security is traded on a given trading day. It represents the most up-to-date valuation of a security until the next trading day.

Database: an organized collection of data that are typically organized to model relevant aspects of reality in a way that supports process rewiring this information.

Data mining: The computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning statistics, and data base systems.

Fundamental analysis: a method of evaluating stocks. It tries to measure the value relating to economic, financial and other qualitative and quantitative factors to produce a value that an investor can compare with current stock prices. With this they can decide what action to take with the security.

Long term investment: an account on the asset side of a company's balance sheet that represents the investments that a company intends to hold for more than a year. They may include stocks bonds, real estate and cash

Machine learning: A branch of artificial intelligence, concerned with the construction and study of systems that can learn from data.

Neural network: conceptually based off the central nervous system, it interconnects systems of neurons that can calculate values for inputs by feeding information through the network. Short term investment: An account in the current assets section of a company's balance sheet. This account contains any investments that a company has made that will expire within one year. For the most part, these accounts contain stocks and bonds that can be liquidated fairly quickly.

Stocks: A type of security that signifies ownership in a corporation and represents a claim on part of the corporation's assets and earnings.

Technical analysis: a method of evaluating securities by analyzing the statistics generated by the market today. It uses charts and other tools to identify patterns (such as past price and volume) that can suggest future activity of that security.

Technical indicators: look to predict the future price levels or simply the general price direction of a security by looking at past patterns.

Time Horizon: the length of time over which an investment is made or held before it is liquidated. It can range from seconds-decades.

User interface: the space where interaction between humans and machines occur. "Watch List": a list of tracked stocks stored within the system for a particular registered user. ! Web service: is a method of communicating between two electronic devices over the World Wide Web.

3. System Requirements

3.1 Enumerated Functional Requirements:

Identifier	Priority	Requirement
REQ 1	5	The System will store all the stock data from all the companies in any user portfolio at a regular interval.
REQ 2	5	The system will allow the user to add certain companies to the list of companies whose stock data is gathered at a regular interval.
REQ 3	5	The system will allow the user to search for stock prices of the companies in the database.
REQ 4	5	The system should perform predictive analysis on the stock data.
REQ 5	4	The system should display the stock data in the form of charts which is easy for users to understand.
REQ 6	4	The charts should contain price of stock at different time intervals.
REQ 7	4	The system should display the result of predictive analysis in the chart data i.e charts should also contain expected price of the stock in the coming future.
REQ 8	3	The system should provide the user with a quantifiable number on how likely the prediction is going to be true.
REQ 9	3	The system should show regular updates on different stock prices and companies in the form of news ticker.
REQ 10	2	The system should allow the user to register with an email id and store the email id and password for future authentication.

3.2 Enumerated NonFunctional Requirements:

Identifier	Priority	Requirement
REQ 11	5	The system should be available in the form of a website and accessible on all major web browsers
REQ 13	5	Provide customers with predictive data on the stock market to aid their trading decisions.
REQ 14	3	The system should provide extra information to user other than stock price.
REQ 15	3	The system should not allow unregistered users to use functionalities of the website.

3.3 On Screen Appearance Requirements:

Identifier	Priority	Requirement
REQ 16	5	The system should be fully functional on all major web browsers.
REQ 17	5	The webpage should be responsive and adapt to different screen sizes.
REQ 18	5	The system should have a fixed navigation bar that provides all the basic features required to navigate around the website
REQ 19	5	Function and purpose of each element on screen must be clear and direct by placing information in natural areas where natural will be defined based on other popular stock websites, as well as developer/designer intuition

4. Functional Requirements Specification

4.1 Stakeholders

Two Stakeholders can be identified:

1. Users: any user of the web service registered or not.
2. Administrator: maintains and updates website services.

4.2 Actors and Goals

Six actors can be identified:

1. Registered User: a registered user.
2. New User: any unregistered user
3. User: both a registered user and new user(will be used for diagrams and descriptions where both a registered user and a visitor can interact with the system)
4. Database: records of stock information (i.e historical prices, prediction results, confidence value, etc), user data (i.e. username, password, tracked stocks, email, etc), and system data (timers, search logs, prediction time logs).
5. Price Provider (i.e. Yahoo! Finance): Provides the current pricing of a stock of interest
6. Administrator: a special case User that maintains and updates website services.

4.3 Use Cases

4.3.1 Casual Description

The summary use cases are as follows:

UC-1: **Register** - allow a visitors to fill out the a registration form and become a user.

UC-2: **Login** - Allows user to access their account and view information for their tracked stocks.

UC-3: **Search Stock-** allows a user or visitor to search for a particular stock through keywords and
filter by sector or industry and view its corresponding prediction .

UC-4: **Add Stock**: Allows user to add new stocks to his portfolio.

UC-5: **Remove Stock**: Remove stocks from the portfolio.

UC-6: **Suggest** - allows a user to request for the system to suggest a stock that is predicted to have the highest gain.

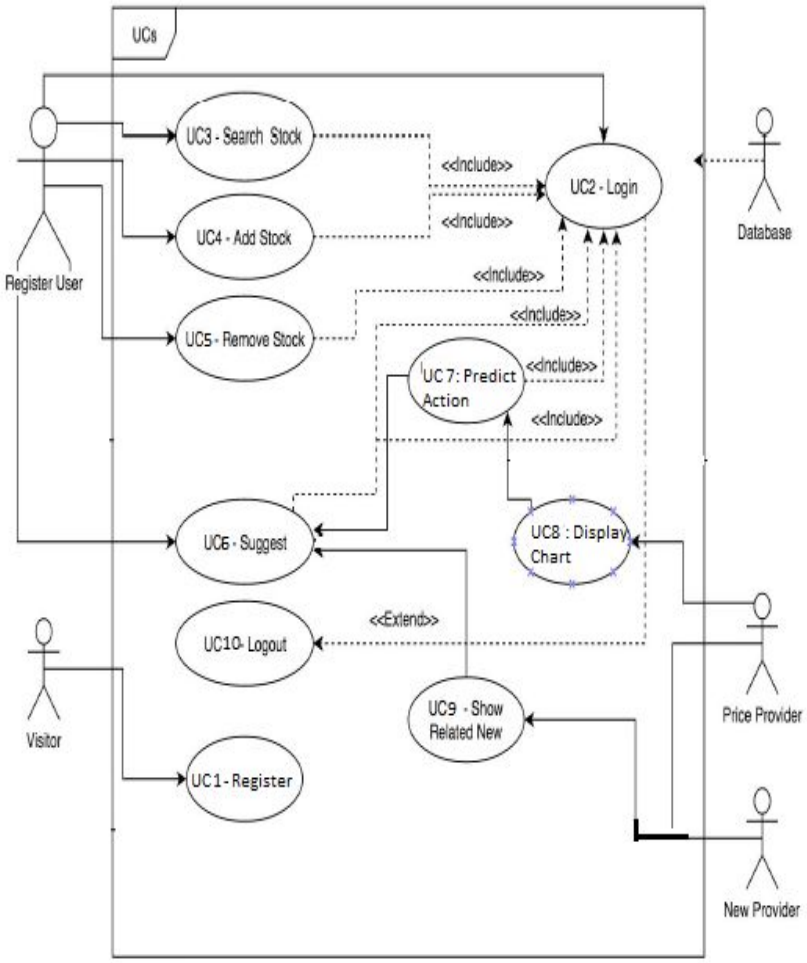
UC-7: **Predict Actions**: system gives the best possible suggestion the registered user

UC-8: **Display_Chart-** gives as output a chart showing movement of the stock over a long period of time. User can then check stock values at different time.

UC-9: **Show Stock Related News**: Users will have access to current events and news regarding for the given stocks in the portfolio.

UC-10: **Logout** - allow users to log out of the system.

4.3.2 Use Case Diagram



The use case diagram is shown in the above figure. As shown by the diagram a registered user can search, add or remove stock along with setting prediction parameters. He can also login to the database to perform all the above operations. A scheduler provides data acquisition updates. A grapher suggests the stock prices to the registered user in form of graphs and charts. A price provider predicts the stock prices based on the prediction parameters set by the registered user and suggests him to take the necessary action (buy, sell, hold). A new provider shows and suggests the related news to the registered users. A registered user can accomplish all that of a visitor but also is allowed to set prediction parameters. Tracking a stock is an extension of a search or suggestion because the registered user must first retrieve the stock before he/she can track it. A user must first get a prediction before he/she can learn the involved analysis behind it and thus it must be an extension of a search or suggestion. It should be noted that although an administrator has similar rights to that of a registered user (search, track, get a suggestion), the diagram does not show these connections because these are not ultimately the administrators goals. The database is a participant in all use cases, and thus its connections are not shown to reduce clutter.

5. Effort Estimation

Actor Name	**Description of relevant characteristics**	**Complexity**	**Weight**
User/Investor	User/Investor is interacting with the system via a graphical user interface	Complex	3
Profile	Profile is another system interacting through a protocol.	Average	2
Predictor	Predictor is another system which interacts with our system through a defined application	Simple	1
Database	Database is another system interacting through a protocol.	Average	2
API	API is another system which interacts with our system through a defined application programming interface	Simple	1

UAW = 2 X simple + 2 X average + 1 X complex = 2 X 1 + 2 X 2 + 1 X 3 = 9

Technical Complexity

TECHNICAL FACTOR	DESCRIPTION	PERCEIVED COMPLEXITY	WEIGHT	CALCULATED FACTOR
T1	Distributed system (running on multiple machines)	3	2	6
T2	Performance objectives (are response time and throughput performance critical?)	3	1	3
T3	End-user efficiency	3	1	3
T4	Complex internal processing	3	1	3
T5	Reusable design or code	0	1	0
T6	Easy to install (are automated conversion and installation included in the system?)	3	0.5	1.5
T7	Easy to use (including operations such as backup, startup, and recovery)	5	0.5	2.5
T8	Portable	2	2	4

T9	Easy to change (to add new features or modify existing ones)	1	1	1
T10	Concurrent use (by multiple users)	4	1	4
T11	Special security features	5	1	5
T12	Provides direct access for third parties (the system will be used from multiple sites in different organizations)	1	1	1
T13	Special user training facilities are required	0	1	0

TFC = C1+C2*TFT

= 0.6 + 0.01 × 34

= 0.94

Environment Complexity Factor (ECF)

ENVIRONMENTAL FACTOR	DESCRIPTION	PERCEIVED IMPACT	WEIGHT	CALCULATED FACTOR
E1	Beginner familiarity with the UML- based development	3	1.5	4.5
E2	Some familiarity with application problem	2	0.5	1
E3	Some knowledge of object-oriented approach	2	1	2
E4	Beginner lead analyst	3	1.5	4.5
E5	Highly motivated, but some team members occasionally slacking	4	1	4-
E6	Stable requirements expected	5	2	10
E7	No part-time staff will be involved	0	-1	0
E8	Programming language of average difficulty will be	3	-1	-3

	used			

Computations

ECF = C1+C2 × EFT

= 1.4 + -0.03 × 19

= 0.83

UCP = UUCP × TCF × ECF

= 163 × 0.94 × 0.83

=127.17 or 128 UseCasePoints

Duration =UCP × PF

= 128*28

= 3584 hours

Estimated time for complete development is **3584 hours**

6. Domain Analysis

6.1 Concept Definitions

The concepts and their definitions are discussed below.

Website

Definition: A hypertext document connected to the World Wide Web.

Responsibilities:

- Display HTML document that shows the actor the current context
- Shows what actions can be taken through buttons

Credentials

Definition: Specific user's username and associated password.

Responsibilities:

- Hold a user's username and password

Query

Definition: search query.

Responsibilities:

- Hold a specific search query

Controller

Definition: Directs or regulates the requests made from user or another concept.

Responsibilities:

- Access account creation
- Retrieves information from Data Renderer and passes to Website
- Coordinate decisions based on the specific use case

Portfolio

Definition: holds account information for a specific user.

Responsibilities:

- Holds account information for a specific user

Tracker

Definition: tracks stocks based on user's portfolio.

Responsibilities:

● Holds tracking information for specific users

Predictor
Definition: Generate stock predictions.
Responsibilities:
● Apply prediction algorithms to data

Stock Retriever
Definition: Collects data from Internet Fetch stock prices and re by querying the Yahoo Finance.
Responsibilities:
● Retains momentary stock data from external websites and passes to Data Handler

Stock Data Extractor
Definition: Extracts stock data to be stored within the database.
Responsibilities:
● Extracts stock data from a given file and stores it within the database

Databases
Definition: An organized collection of stock data, user data, and system data.
Responsibilities:
● Store user data
● Store stock data

Account Handler
Definition: A way to prove to a computer system that you really are who you are.
Responsibilities:
● Determine the validity of a logon request
● Provide help or tips to a user attempting a login

Stock Historical Prices
Definition: Holds historical prices for a given stock.
Responsibilities:
● Holds historical prices for a given stock

Stock Current Prices
Definition: Holds current price for a given stock.
Responsibilities:
● Holds current price for a given stock

Stock Information

Definition: Holds current stock information (stock symbol, company name, etc.) for a given stock.

Responsibilities:

- Holds current stock information for a given stock

Chart Connection

Definition: Retains a connection to the grapher.

Responsibilities:

- Retains a connection to the grapher
- Graph a given set of data

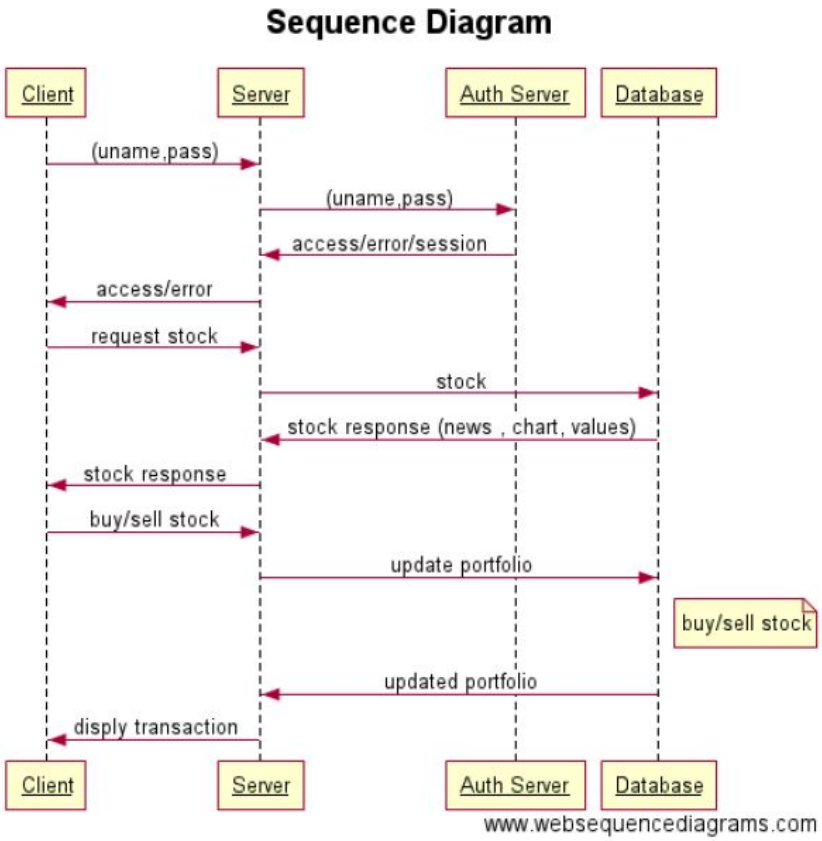
Searcher

Definition: Queries database for stocks.

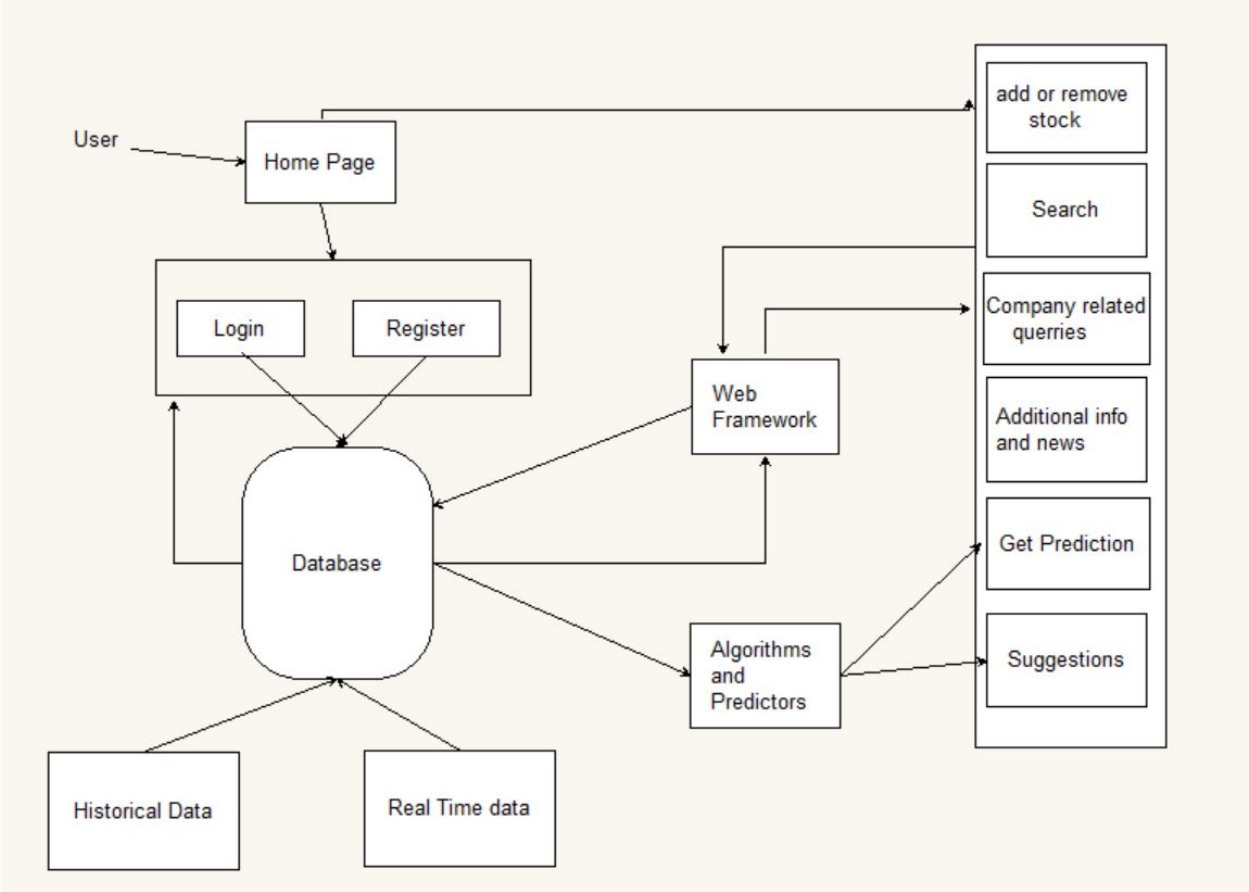
Responsibilities:

- Get stocks based on a keyword
- Get stocks based on predicted gain

7. Interaction Diagrams



8. System Architecture



Architectural Styles

Our software uses several architectural styles. They follow:

1) Client/Server
2) Event-driven
3) Rule-based system
4) Database-centric

Client/Server Architecture

Client/Server is our main architectural style; it separates our system requirements into two easily programmable systems. First, the client, which acts as the User Interface, requests data from the server, and waits for the server's response. Secondly, the server, which authorizes users and processes stock data into information the user can use. It then sends this processed

information to the client to display to the user.

Event-driven Architecture

Our system will only need to execute its functions after some major state change. It has no real time components like a video game. Instead, we'll have two event emitters, the user and the timer. Both will drive the application to execute relevant operations though the execution of different events. These events include login, adding new stocks, deleting stocks, and requesting an updated lists of stocks for the user; and a time-based update for the timer.

Rule-Based System

Our application will be rule-based. In other words, the system will use a set of rules that we determine it to analyze the stock information it gathers. These rules comprise a semantic reasoned which makes decisions for the application and the user. It uses a match cycle act cycle to deduct which stocks will be best to buy and which stocks would be best to sell. Then, it outputs these results to the user-interface.

Database-centric Architecture

Our system relies heavily on its database, both the store relevant stock data and to analyze the data we give it. The database-centric architecture offers:

1) A standard relational database management system. This means the data will be stored away from the client side application.
2) Dynamic table-driven logic. We need to update the tables every time stock prices change.
3) Stored procedures running on database servers to analyze our data.
4) A shared database for communication between parallel processes !

In short, it's a good way of managing a large amount of data.

Persistent Data Storage

The system-to-be requires data to be saved in order to outlive a single execution of the system. This data includes historical stock prices for all relevant stocks, related information for all relevant stocks, user account information, user search queries, and the timers dictating when the system should update current stock prices as well as begin calculating predictions. In order to organize this data, one must first elaborate on the different methods of storage.

A flat file database typically consists of multiple text files storing one record per line. Text files are simple and portable and can, in most cases, be used without requiring special software. They make it, however, difficult to search for specific information or to create reports that include only certain fields from each record. Thus, when creating new records, numerous redundancies occur as a portion of information in one file must be rewritten to all others.

A relational database, on the other hand, consists of multiple tables linked by “keys” certain pieces of information shared by two or more tables. This model takes advantage of the uniformity to build completely new tables out of required information from existing tables. In other words, it uses the relationship of similar data to increase the speed and versatility of the database.

Schema for the Historical Data

Column Name	Definition
id	bigserial NOT NULL
symbol	varchar
Time	date
open	double precision
close	double precision
high	double precision
low	double precision
volume	bigint

Schema for Real Time Data:

Column Name	Definition
id	serial NOT NULL
Symbol	varchar
Time	varchar NOT NULL
Price	real NOT NULL
Volume	bigint

Global Control Flow

Execution Orderliness:
The system can generally be defined as event-driven; it will wait for a user to make an action before processing data. The user's interaction will characterize their visit and the control structure will wait for the user's request, remaining idle until it receives such information. This allows for a user to sequence their actions upon a visit in different patterns without confusing the system. Some actions may require multithreading in order for updating to be accomplished thoroughly.

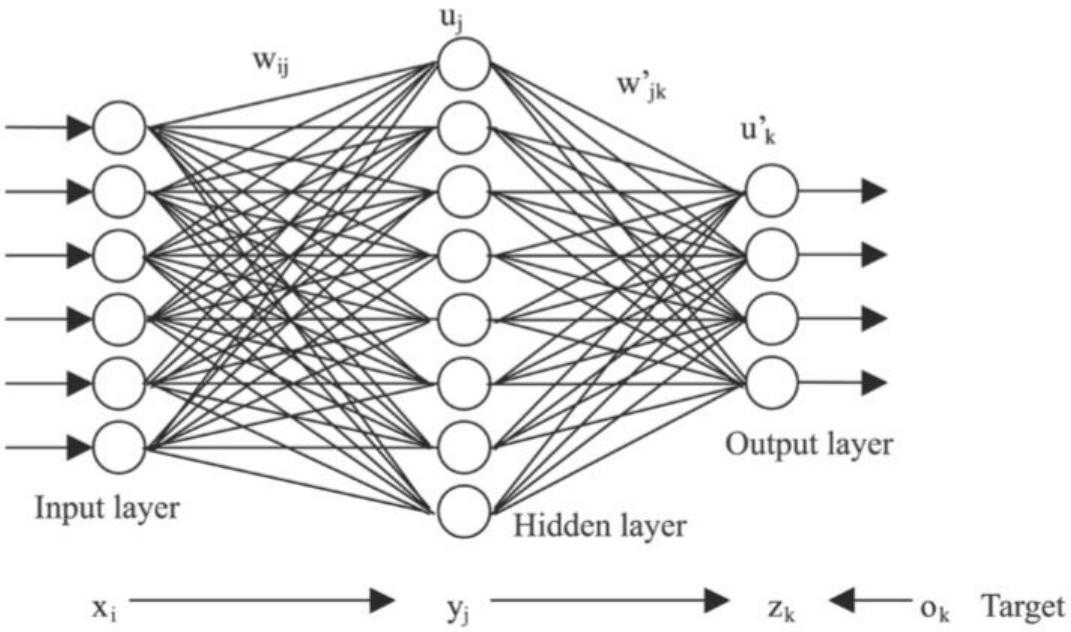
Time dependency:
The software will make use of timers to keep current, up-to-date, information regarding stocks in our database at all times. This is a real-time system that will update the database at exact defined times throughout a given day. !

Concurrency:
Our system has been implemented to function on the Web; this means that multithreading must be supported. We expect that at some instance there will be concurrent users accessing either the website or the database, so that will be accounted for using multiple threads.

9. Algorithms

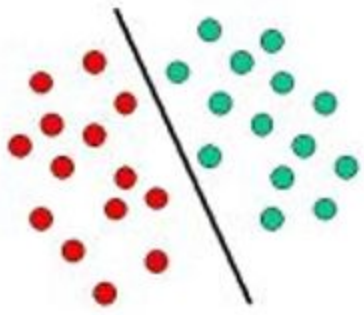
9.1 Artificial Neural Network (ANN)

Artificial neural networks (ANNs) are models inspired by the central nervous systems of animals and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. ANNs are composed of neurons which can exchange messages between each other. The neurons might be distributed in multiple layers. The connection between neurons is assigned with a weight. These numerical values of weights to these connections are updated on the fly as the system gains experience. This makes neural networks adaptable and capable of learning. The overall process of turning input into output is an emergent result of the programming of each individual neuron the data touches, and the starting conditions of the data itself. In the the brain, the “starting conditions” are the specific neural signals arriving from the spine, or elsewhere in the brain. In the case of an ANN, they’re whatever we’d like them to be, from the results of a search algorithm to randomly generated numbers to words typed out manually by researchers. Artificial neural networks are basically simulated brains. But it’s important to note that we can give our software “neurons” basically any programming we want; we can try to set up their rules so their behavior mirrors that of a human brain, but we can also use them to solve problems we could never consider before. The learning algorithm works as follows: send the same input into the network over and over and over, and every time it generates the correct output, record the time it took to do so. Some paths from A to B will be naturally more efficient than others, and the learning algorithm can start to reinforce neuronal behaviors that occurred during those runs that proceeded more quickly.



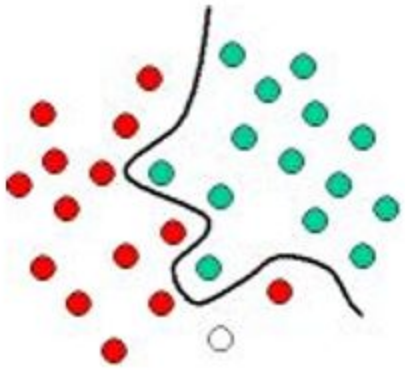
9.2 Support Vector Machines (SVM)

In this example, the objects belong either to class GREEN or RED. The separating line defines a boundary on the right side of which all objects are GREEN and to the left of which all objects are RED. Any new object (white circle) falling to the right is labeled, i.e., classified, as GREEN (or classified as RED should it fall to the left of the separating line).

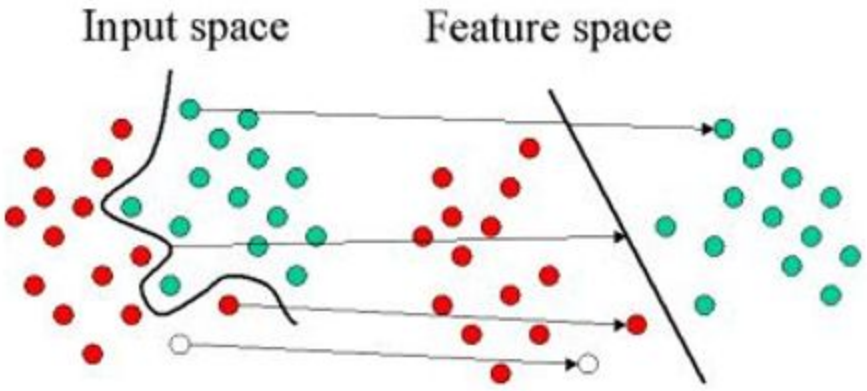


The above is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective groups (GREEN and RED in this case) with a line. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). This situation is

depicted in the illustration below. Compared to the previous schematic, it is clear that a full separation of the GREEN and RED objects would require a curve (which is more complex than a line). Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. Support Vector Machines are particularly suited to handle such tasks.



The illustration below shows the basic idea behind Support Vector Machines. Here we see the original objects (left side of the schematic) mapped, i.e., rearranged, using a set of mathematical functions, known as kernels. The process of rearranging the objects is known as mapping (transformation). Note that in this new setting, the mapped objects (right side of the schematic) is linearly separable and, thus, instead of constructing the complex curve (left schematic), all we have to do is to find an optimal line that can separate the GREEN and the RED objects.



9.3 Patterns For SVM

Moving Averages Model

Moving average is a statistical process that creates a set of averages for many small subsets of the full dataset. By taking the averages of mini-subsets of the dataset, it helps smoothen out the trend of the data by showing the long-term lengths. It usually shows the price fluctuation of a stock for a certain period, but can also be used for longer periods. The moving average is a lagging indicator since the predication is made on past prices. The two main ways to implement is the simple moving average model and the exponential moving average. The difference is that the exponential moving average gives more weight to more recent prices. In this project we used the simple moving average and used the period as 200 days. So, the average price is taken over the past 200 trading days. For instance, the first average price is taken from day 0 to day 200 the second average is taken from day 1 to day 201. This trend is continued until the current price is taken into account. The period is always kept constant at 200 days. This number was chosen from the research the group did which lead us to believe that 200 days will provide a good balance between placing emphasis on past prices as well as more recent prices.

Algorithm used for moving average:

$$SMA = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

$$Pm = prices\ of\ the\ stock\ at\ a\ certain\ point$$

$$n = period\ of\ days\ the\ data\ is\ across$$

ASCENDING TRIANGLE

An ascending triangle is 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. The chart below is an example of an ascending triangle:

An ascending triangle is generally considered to be a continuation pattern, meaning that it is usually found amid a period of consolidation within an uptrend. Once the breakout occurs, buyers will aggressively send the price of the asset higher, usually on high volume. The most common price target is generally set to be equal to the entry price plus the vertical height of the triangle.

An ascending triangle is the bullish counterpart of a descending triangle.

DESCENDING TRIANGLE

A bearish chart pattern used in technical analysis that is created by drawing a 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. The chart below is an example of a descending triangle:

This is a very popular tool among traders because it clearly shows that the demand for an asset is weakening, and when the price breaks below the lower support, it is a clear indication that downside momentum is likely to continue or become stronger. Descending triangles give technical traders the opportunity to make substantial profits over a brief period of time. The most common price targets are generally set to equal the entry price minus the vertical height between the two trendlines.

A descending triangle is the bearish counterpart of an ascending triangle.

9.4 BAYESIAN CURVE FITTING

In a fully Bayesian approach, we should consistently apply the sum and product rules of probability, which requires, as we shall see shortly, that we integrate over all values of w. Such marginalizations lie at the heart of Bayesian methods for pattern recognition.

In the curve fitting problem, we are given the training data x and t, along with a new test point x, and our goal is to predict the value of t. We therefore wish to evaluate the predictive distribution p(t|x, x, t). Here we shall assume that the parameters α and β are fixed and known in advance (in later chapters we shall discuss how such parameters can be inferred from data in a Bayesian setting). A Bayesian treatment simply corresponds to a consistent application of the sum and product rules of probability, which allow the predictive distribution to be written in the form:

p(t|x, x, t) = p(t|x, w)p(w|x, t) dw

Here p(t|x, w) is given by p(t|x, w, β) = N t|y(x, w), β −1 ! , and we have omitted the dependence on α and β to simplify the notation. Here p(w|x, t) is the posterior distribution over parameters. For problems such as the curve-fitting example, this posterior distribution is a Gaussian and can be evaluated analytically. The predictive distribution is given by a Gaussian of the form

p(t|x, x, t) = N(t|m(x), s^2 (x))

where the mean and variance are given by

m(x) = β ϕ(x) TS % N n=1 ϕ(xn)tn
s^ 2 (x) = β −1 + ϕ(x) TS ϕ(x)

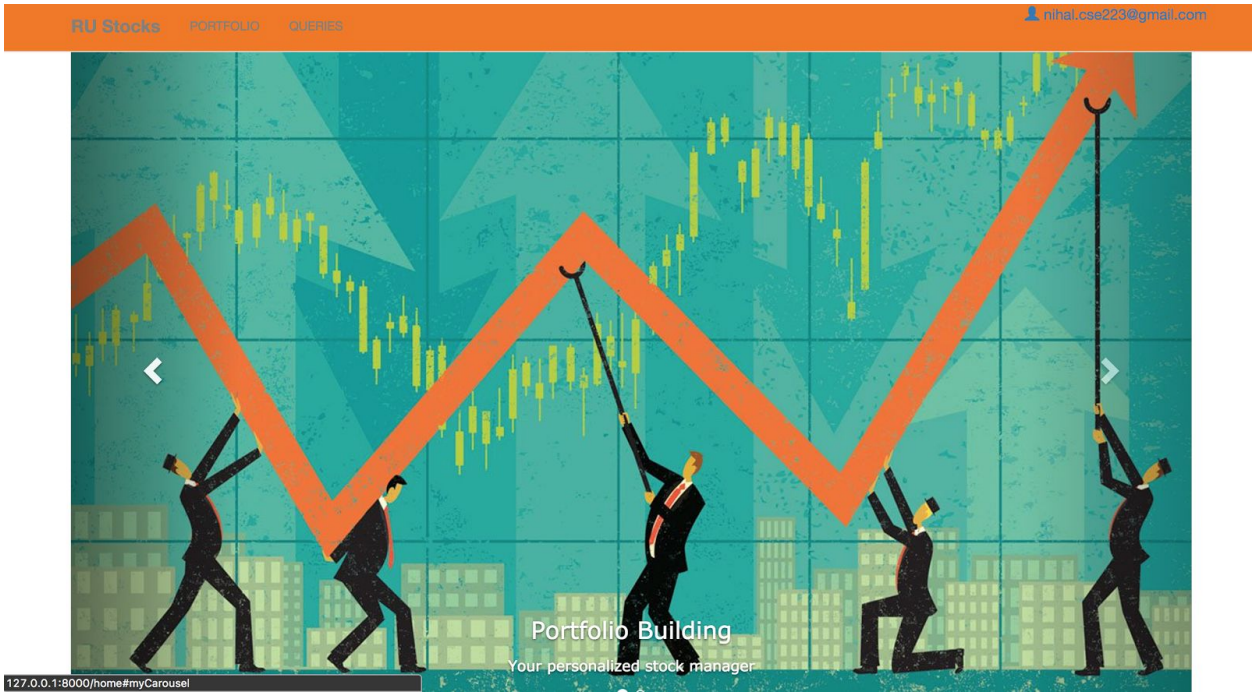
Here the matrix S is given by

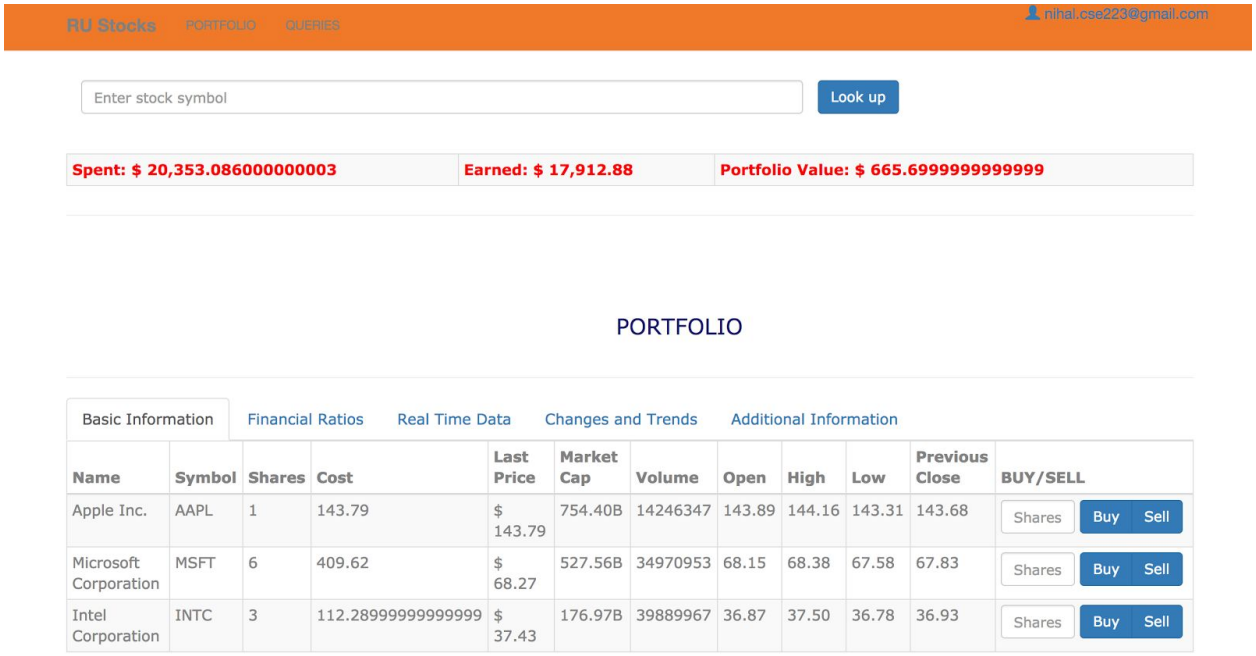
S^−1 = αI + β % N n=1 ϕ(xn) ϕ(x) T

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

We implemented Bayesian Curve Fitting by using the above procedure.

User Interface Design





RU Stocks PORTFOLIO QUERIES nihal.cse223@gmail.com

Enter stock symbol Look up

Spent: $ 20,353.086000000003	Earned: $ 17,912.88	Portfolio Value: $ 665.6999999999999

MICROSOFT CORPORATION MSFT

No. of Shares Buy MSFT

Basic Information | Financial Ratios | Real Time Data | Changes and Trends | Additional Information | News

Last Price	Market Cap	Volume	Avg Daily Volume	Open	High	Low
$ 68.27	527.56B	34970953	22057500	68.15	68.38	67.58

Check prediction for next 30 days for MSFT Get Recommendation for MSFT

RU Stocks PORTFOLIO QUERIES nihal.cse223@gmail.com

Last Price	Market Cap	Volume	Avg Daily Volume	Open	High	Low
$ 68.27	527.56B	34970953	22057500	68.15	68.38	67.58

Check prediction for next 30 days for MSFT Get Recommendation for MSFT

12. Future Work

In the future, we plan on adding further models to our overall predictions as well optimizing the current ones. We also intend to use twitter feeds and news feeds to make a sentiment analysis of the masses. A sentiment analysis will help the users get an idea about the sentiment towards a particular stock which will surely serve as a better indicator for making decisions. We also plan to make recommendations based on our sentiment analysis which we think will serve as an advice to novice investors regarding stock market trading.

13. REFERENCES

1. Software Engineering by Ivan Marsic, Rutgers University
http://www.ece.rutgers.edu/~marsic/books/SE/bookSE_marsic.pdf

2. Background knowledge of stock market prediction:
http://en.wikipedia.org/wiki/Stock_market_prediction

3. SVM:
http://www.vatsals.com/Essays/MachineLearningTechniquesforStockPrediction.pdf

4. ANN Neuroph Implementations
http://neuroph.sourceforge.net/sample_projects.html

5. Patterns for SVM:
http://www.investopedia.com