SOFTWARE ENGINNERING OF WEB APPLICATIONS

WEB-BASED STOCK FORECASTER

Team: Group #10

Localhost/code/run/homepage.html

Team Member: Rong Zhang, Feng Rong, Xiaoyi Tang, Zhe Chang, Jingxuan Chen, Xiuqi Ye

April 27, 2017

Table of Contents

1. INDIVIDUAL CONTRIBUTION BREAKDOWN

We all contribute ourselves in this project, and our project goes well under the strength of our efforts. Each one contributes equally and we have finished a great project.

2. CUSTOMER STATEMENT OF REQUIREMENTS

2.1 Motivation

Nowadays, people are more and more likely to look for an effective and reliable way to keep their wealth growing by personal financial investment. When it comes to personal financial investment, investing in stock market seems to be one of the most popular way people prefer to choose because of its high return. With the popularity of the personal computer and the Internet, more and more people can easily make their individual investing in stock market without the help from pension funds and brokerage firms. People can conveniently sit in front of their computer and do all kinds of researches and comparisons about various companies, in a quicker and cheaper way. It is appealing that the individual investor entering the stock markets without the help of brokerage firms. However, high return comes with high risk, people need to make a cautious and wise decision to reduce the risk. Computers can help people to achieve that goal. Computers can become such an "assistant" for individual investors which can provide the automation of trading in financial markets for individual investors. As far as what is concerned above, we plan to implement such an "assistant", a stock application, giving people useful help in investing in the stock market. Some web applications have existed which can provide functions showing the stock price trend and graphs such as Yahoo Finance[1].And those applications also provide a little bit prediction for the users. However, the prediction part of those existing applications seems not so helpful in

predicting in both short term and long term and not providing specific trade suggestions, while some of the individual investors may be looking for such kind of application functions that could help them make their choice easier and more reliable when trading in stock market. Therefore, we would like to implement such a stock predicting web application[2] that can use some meaningful algorithms and index to calculate and predict the stock price in both short term and long term, meanwhile, give users some trade recommendation according to the result from the prediction.

Vision

The first and foremost purpose of the system is to predict future stock prices. There are three levels of the whole system: interface level, service level and data level.[3] The user input the username and password to login the system (If this is the first time using this Web, the user need to register for a username and password). The user can choose the name of the company and desired service, then the user will get the required output through the interface in the presentation level. The services level performs various actions on this data using Web Service and Indicator algorithms. Web Services are employed to provide connectivity between the prediction module and end-user. The storage level lies at the bottom of other levels with a database that stores and records the entire data. The user open the Home page and will be required a username and passcode. If this is the first time of user to login the web, the web will ask for the user to register a username and passcode. After creating an account, the user input the correct pairs of the username and passcode, the user will go to the Service page. In the service page, there are few key option for user to choose: Stock List, Price Data, Email, Indicators and Price Prediction.

1) Stock List : This service shows the name and real time price of the 10 stocks. These prices are ticked every 1 minute from http://finance.yahoo.com

2) Price Data: This service shows the graph of historical and real time price of the stock.

3) Email: This service let user to choose a stock, when the price lower than the price user set preciously, user will get an Email notification.

4) Indicators: In this service, we implemented three indicators to provide some advises to user.

5) Price Prediction: This service we have short term and long term prediction.

Google Finance (http://finance.google.com) is being used for fetching the historical data for our system. Our project where our PHP script get historical stock data which includes the following: High, Low, Close, Open and volume etc for a list of companies and stores it in the database (MySQL)[4].

3. GLOSSARY OF TERMS

Algorithms: a step-by step procedure for calculations.

Artificial Neural Network: Neural networks is the development of a computer system that models the human brain and its nervous system. also see (Artificial intelligence)

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.

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.

Web service: is a method of communicating between two electronic devices over the World Wide Web

4. SYSTEM REQUIREMENTS

4.1 Enumerated Functional Requirements

	PW	REQUIREMENT
REQ1	5	The system shall allow the administrator to add and remove stocks from the list of stocks that predictions are made on.
REQ2	5	The system shall continuously gather the time series of the current market data (stock prices, trading volumes, etc) for a set number of companies
REQ3	5	The system shall periodically apply prediction algorithms or models on the obtained data and store the results to a central database.
REQ4	5	The system shall allow users to request a stock that is predicted to have a large gain in the near future.
REQ5	4	The system shall obtain and display a confidence value for each prediction given to the user.
REQ6	3	The system shall allow a method for users to learn the professional analysis used for each prediction method if he/she chooses to.
REQ7	3	The system shall allow for the administrator to alter the rate at which current stock prices are updated and the rate at which predictions are made.
REQ8	3	The system must pull data from the server when the user requests it.
REQ9	1	The system must store a list of stocks that the user chooses.
REQ10	3	The system must allow new users to create an account.

FIGURE 1

4.2 On-Screen Appearance Requirements

IDENTIFIER	REQUIREMENT
REQ11	This brings you back to the home screen of the application when it is clicked
REQ12	The system shall be able to run through all prediction models for each and every listed stock within 1 hour.

FIGURE 2

5. FUNCTIONAL REQUIREMENTS SPECIFICATION

5.1 Stakeholders

Three Stakeholders can be identified:

1. User: any user could log in the system and get web services.
2. Administrator: maintains and updates website services.

5.2 Actor and Goals

Our system has both human and non-human actors.

User: a registered user.

Administrator: The manager that is in charge or keeping the system updated and in working order.

User: both a registered user and visitor (will be used for diagrams and descriptions where both a registered user and a visitor can interact with the system)

Prediction Algorithm: The algorithm(s) that will calculate the prediction.

Database: The database will hold all the user data and information as well as all the stock information. All the user information and their portfolios will be stored in the database.

Yahoo! Finance API: The API is where we will pool the real time stock data .

Graph: Provide visual charts from raw data.

Google Finance: This is where we will pool out the historical stock data.

5.3 Use Cases

Use Case	Name	Participant
Case 1	Management for Web-service	Administrator
Case 2	Sign Up	User
Case 3	Log In	User
Case 4	Get Query	User
Case 5	Get Quotes	User
Case 6	Get Graph	User
Case 7	Get Stock Indicator	User
Case 8	Get Prediction Price	User
Case 9	E-mail Notification	User
Case 10	Account Management	User

FIGURE 3

Use case 1: Management for Web-service

As the administrators, they need to analysis whether the website is work normally. And they also need to manage and maintain the web serving the customers continually. As an important part of the whole system, the administrator can create the database, update the data in the database (send a request to Google or Yahoo for the data and read them to import them to the database), and manage the users' information.

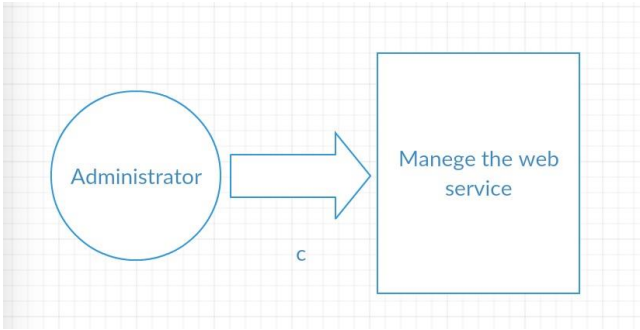


FIGURE 4

Use case 2: Sign up

The system needs the user to create an account to get access. The visitor can only access the homepage. The user of the website can create a new account by click the button “Sign up” and be brought to the sign up screen to create their new account. The user need to fill out the required information: user name, password, email and click “Sign up”. Once completed and submitted, the information will be sent to the database. The database will then check the availability of the user name and insert the user’s information into the database. Once the user signed up successfully, the user will be brought to the log in page. The user need to log in to get the access to the index page of the system.

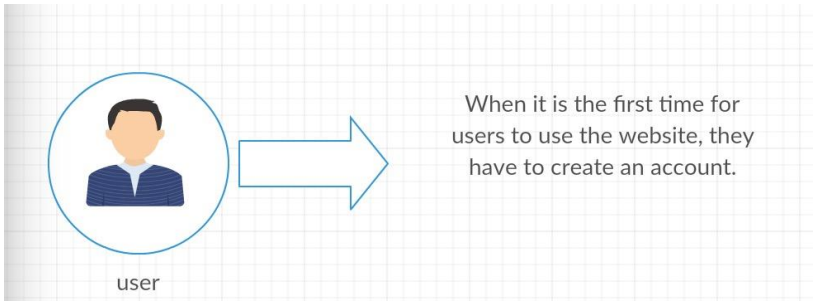


FIGURE 5

Use case 3: Log in

The system needs the user to have an account to get access. Once you signed up successfully or already having an account, you can click the “log in” button to access the log in page. The user need to fill the username and password and click “log in” button. Once completed and submitted, the information will be sent to the database, then the database will check the match of the password and username. If match, the user will log in successful and be brought to the index page of the system.

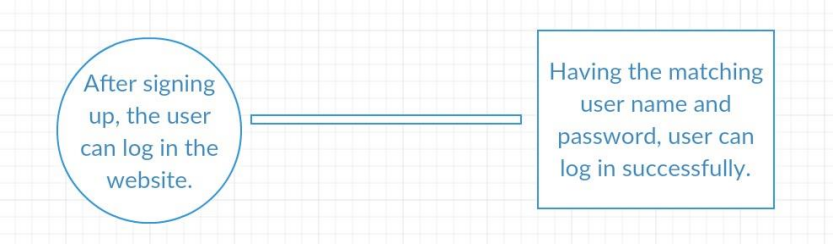


FIGURE 6

Use case 4: Get Query

For each main option (interested, hold), there are several stocks stored inside.

The customers can get a lot of information of these stocks, such as the real-time price, the historical price, the recent price graphs (days, months, one years, two years) and so on. This provides user with information about the company and the last year’s all quarter results and the current years’ quarter results.

Use case 5: Get quotes

After logging in the system, the user can get the quotes of the stocks. At the index page of the system, the user can click the “stock list” to access the stock list page. There will be the stock symbols with their real-time price. In this case, the website will send a request to the database and catch the current price and show them on the screen.

Use case 6: Get graphs

As a presentation of history price and current price, the user can notice the flow and trend of the stocks in the graph. The user can click “PriceData” then “historical price” or

“real time” to access the result page, then choose a stock to check the specific graph of the stock. The page will call a function to get the data from the database then show them on the screen as a chart.

Use case7: Get Stock Indicator

After analyzing the history data and prediction, the server should return back a guiding suggestion to the user. Based on the prediction result, the server can analyze the user’s performance and market situation to get some guiding result and help the user to do the decision.

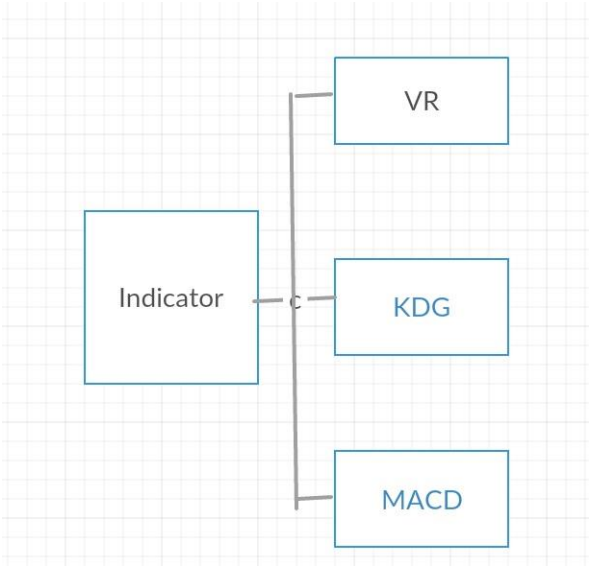


FIGURE 7

Use case 8: Get Prediction Price

For the prediction of each stock, users do not care how you predict the prices, the only thing they care about is the accuracy of your prediction. Hence, it is necessary to provide the data accuracy for each stock’s prediction to enable users know the whether they should trust a specific predication.

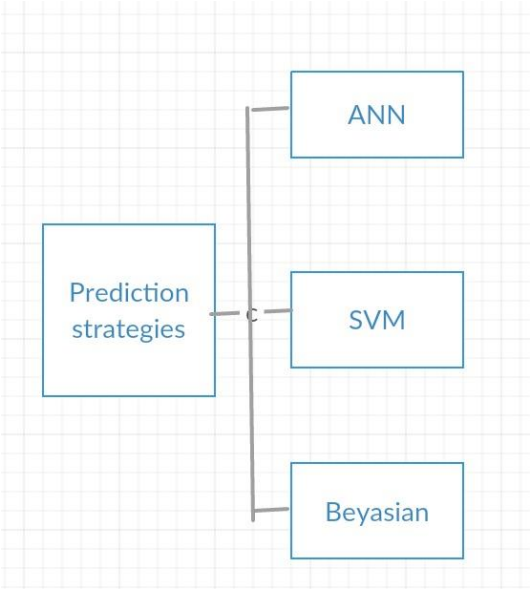


FIGURE 8

Use case 9: E-mail Notification

Once the user created an account or log in the website, the user can add a favorite stock and a stop loss price. Once the stock price goes lower than the stop loss price, the system will send a notification to the user's E-mail.

Use case 10: Account Management

User can choose the stock and the stop loss price of the stock the prefer to get E-mail notification.

6.EFFORT ESTIMATION

6.1 Unadjusted Actor Weight

ACTOR NAME	DESCRIPTION	COMPLEXITY	WEIGHT
User	a registered user.	Complex	3
Database	records of stock information, user data, and system data	Average	2

Price Provider	Provides the current pricing of a stock of interest	Average	2
Administrator	special case User that maintains and updates website services.	Complex	3
Graph	Provide visual charts from raw data.	Average	2

FIGURE 9

6.2 Unadjusted Use Case

Use Case	Name	COMPLEXITY	WEIGHT
Case 1	Management for Web-service	Average	2
Case 2	Sign Up	Simple	1
Case 3	Log In	Simple	1
Case 4	Get Query	Average	2
Case 5	Get Quotes	Average	2
Case 6	Get Graph	Complex	3
Case 7	Get Stock Indicator	Complex	3
Case 8	Get Prediction Price	Complex	3
Case 9	E-mail Notification	Average	2
Case 10	Account Management	Average	2

FIGURE 10

7. DOMAIN ANALYSIS

7.1 Concept Definitions

The concept 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.

Query：

Definition: Search query.

Responsibilities:

- Hold a specific search query.

PageMaker:

Definition: Generates display inputs ultimately for website

Responsibilities:

- Must be able to display text, numbers and graphics for website environment

Motivation: Data and images simply cannot come to website in a quick and easy fashion. There must be a transformation or parsing of “raw” local data that can be manipulated to fit the website environment. The reason for doing this dates to the fact that a web-site is necessary to accommodate a large spectrum of use

Account:

Definition: holds account information for a specific user.

Responsibilities:

- holds account information for a specific user.

Predictor:

Definition: Generate stock predictions.

Responsibilities:

- Apply prediction algorithm to data.

UpdateHistorical:

Definition: Send a request to the data provider and fetch the historical price data..

Responsibilities:

- Retains momentary stock data from external websites and passes to Data Handler.

DB Connection:

Definition: An organized collection of stock data, user data, and system data.

Responsibilities:

- Store times.
- Store user data.
- Store stock data.

Email Sender:

efinition: A service to send email to user when the stock price lower than expectation.

Responsibilities:

- Send email to user .
-

7.2 Attributes

Email Sender hold a database which contains the stock that user concerns the most. When the price goes lower than the user's expectation, this service will send a Email

8. Class Diagram and Interface Specification

8.1 Class Diagram

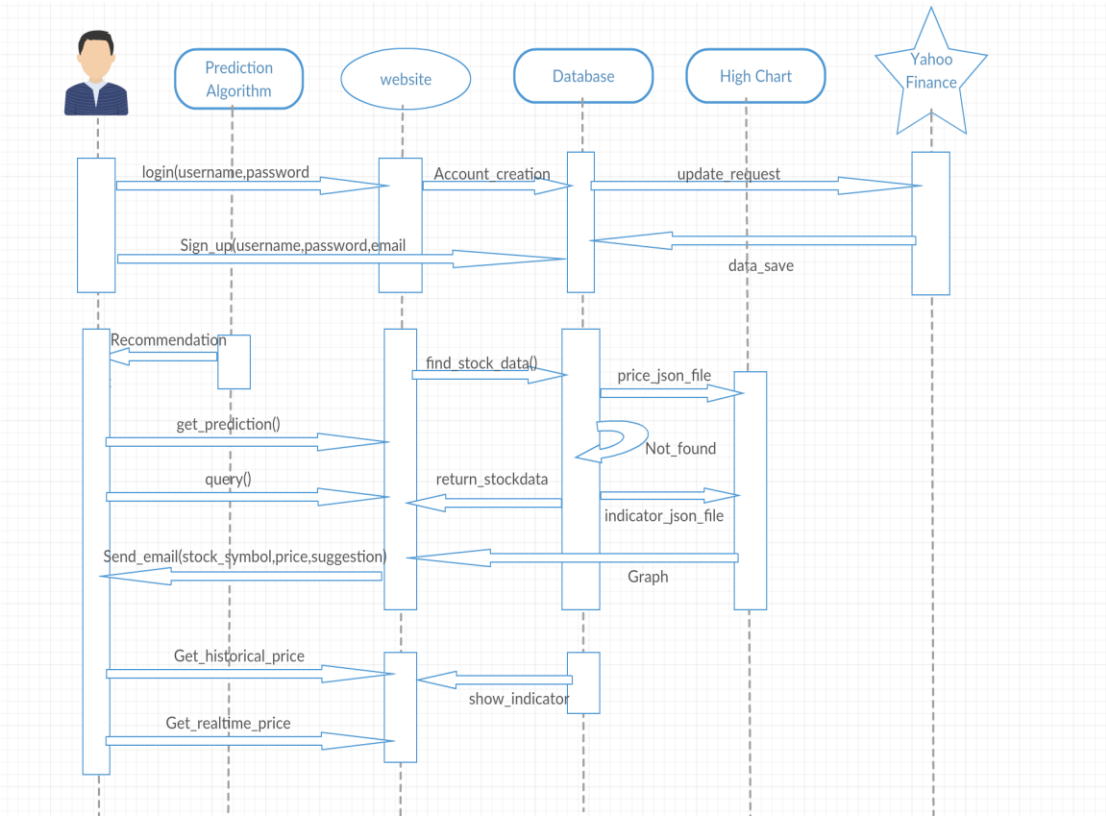


FIGURE 11

8.2 Data Types and Operation Signatures

User

+sign_up(string username, string password, string email): void

Users must create a personal account and sign up when they first browse and decide to use our website.

+login(string username, string password): void

Users must login on our website before using our website each time.

+send_email(string stock_symbol, string username, string email): void

Users can select one stock as their most interested and our website will help them keep track of the stock information and keep the user informed of all relevant data.

Predictor

+prediction_bayesian(string date, string stock_symbol): double

help user predict the price of the stock they want with Bayesian Linear Regression.

+prediction_neuralnetwork(string date, string stock_symbol): double

help user predict the price of the stock they want with Neural Network.

+prediction_svm

help user predict the price of the stock they want with Support Vector Machine.

+get_recommendation(string stock_symbol): boolen

After providing the predicted price of each stock, we will also show our suggestion of whether the user should hold the stock which means we return a yes/no result.

Database

+db_create(): void

create a database for the website to store the user and stock information.

+db_connected(string dbname): void

connect our database with our website to realize query or some other function.

+db_save(): void

get information from yahoo finance and write it to the database.

+update_historical_price(): void

extract historical price of each stock from yahoo finance and update the price in the database

+update_realtime_price(): void

extract realtime price of each stock from yahoo finance and update it in the database.

Query

+get_historical_price(string stock_symbol): void

get the history price information of each stock from starting date and show the curve graph of price.

+get_realtime_price(string stock_symbol): void

get the realtime price information of each stock during one day and show the curve graph of price.

+get_highest_price(string stock_symbol): void

select the highest price of one stock in the whole year from the database and show it on the website.

+get_average_price(string stock_symbol): void

select the average price of one stock in the whole year from the database and show it on the website.

+get_lowest_price(string stock_symbol): void

select the lowest price of one stock in one year from the database and show it on the website.

Indicator

+show_VR(int volume, string startdate, string enddate): void

calculate the reference and show the VR index to indicate the trend of stock.

+show_KDJ(int volume, string startdate, string enddate): void

calculate the reference and show the KDJ index to indicate the trend of stock.

+show_MACD(int volume, string startdate, string enddate): void

calculate the reference and show the MACD index to indicate the trend of stock.

9. System Architecture and System Design

9.1 Architecture Styles

Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only. In the design of the user interface, Bootstrap is widely used to keep the web page dynamically showing suitable to the screen of the user.

jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications. jQuery is used to let the client easier to pass the data and create the chart.

9.2 Data Storage

In data storage part, we use MySQL to create database and store the data. There are three tables in our database, they are Historicalprice, RealtimePrices, and Userinfo.

1.HistoricalPrice

Historical is used to store the historical price data of the stocks and their indicators' parameter. Here is a figure to show the structure of the table.

#	Name	Type	Collation	Attributes	Null	Default
1	**Symbol**	varchar(11)	latin1_swedish_ci		No	*None*
2	**Date**	varchar(20)	latin1_swedish_ci		No	*None*
3	**Open**	decimal(7,2)			No	*None*
4	**High**	decimal(7,2)			No	*None*
5	**Low**	decimal(7,2)			No	*None*
6	**Close**	decimal(7,2)			No	*None*
7	**Volume**	int(11)			No	*None*
8	**AdjClose**	decimal(7,2)			No	*None*
9	**K**	decimal(5,2)			Yes	*NULL*
10	**D**	decimal(5,2)			Yes	*NULL*
11	**J**	decimal(5,2)			Yes	*NULL*
12	**EMA12**	decimal(5,2)			Yes	*NULL*
13	**EMA26**	decimal(5,2)			Yes	*NULL*
14	**DIF**	decimal(5,2)			Yes	*NULL*
15	**DEM**	decimal(5,2)			Yes	*NULL*
16	**MACD**	decimal(5,2)			Yes	*NULL*
17	**VR**	decimal(11,10)			Yes	*NULL*

FIGURE 12

2.RealtimePrices

Realtimeprices is a table which is used to store the real-time price and volume data of the stocks. It will be updated automatically every minute and insert a new value into the table.

Here is the structure of the table.

#	Name	Type	Collation	Attributes	Null	Default
1	**Symbol**	varchar(11)	latin1_swedish_ci		No	*None*
2	**Date**	varchar(20)	latin1_swedish_ci		No	*None*
3	**Time**	varchar(10)	latin1_swedish_ci		No	*None*
4	**Price**	decimal(7,2)			No	*None*
5	**Volume**	int(11)			No	*None*

FIGURE 13

3.UserInfo

UserInfo is a table to store the users' data. Their username, password, email and their chosen stock and stop loss price.

Here is the structure of this table.

#	Name	Type	Collation	Attributes	Null	Default
1	**Email**	varchar(30)	latin1_swedish_ci		No	*None*
2	**username**	varchar(15)	latin1_swedish_ci		No	*None*
3	**password**	varchar(15)	latin1_swedish_ci		No	*None*
4	**stock**	varchar(10)	latin1_swedish_ci		Yes	*NULL*
5	**eprice**	decimal(7,2)			Yes	*NULL*

FIGURE 14

9.3 Web Service

SOAP

SOAP (Simple Object Access Protocol or Service Oriented Architecture Protocol) is the communication protocol for Web services. It is intended for exchanging structured information (based on XML) and is relatively simple (lightweight). Most commonly it runs over HTTP (Appendix C), but it can run over a variety of underlying protocols. It has been de- signed to be independent of any particular programming model and other implementation-specific semantics. A key advantage of SOAP is that, because it is XML based, it is programming-language, platform, and hardware independent.

Developing Web Services with Axis

Server-side Development with Axis

At the server side (or the Web service side), the steps are as follows:

- Define Java interface of the Web service (and a class that implements this interface)
- Generate the WDSL document from the service Java interface (Java WSDL)
- Generate the skeleton Java class (server-side proxy) from the WSDL document(WSDL → Java)
- Modify the skeleton proxy to interact with the Java class that implements the Java interface (both created in Step 1 above)

Client-side Development with Axis

At the client side (or the service consumer side), the steps are as follows:

- Generate the stub Java class (server-side SOAP proxy) from the WSDL document
- Modify the client code to invoke the stub (created in Step 1 above)

10. Algorithms and Data Structure

10.1 Bayesian Theorem

Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{\mathrm{P(B|A)}P(A)}{\mathrm{P(B)}} \quad (1)$$

where A and B are events.

P(A) and P(B) are the probabilities of A and B independent of each other. P(A | B), the conditional probability of A given that B is true. P(B | A) is the probability of B given that A is true.

Using Bayes' theorem, the posterior distribution for w is proportional to the product of the prior distribution and the likelihood function

$$p(w|x,t,\beta) \propto \mathrm{p(t|x,w,\beta)p(w|\alpha)} \quad (2)$$

10.2 Curve fitting theorem

Real data sets typically have underlying regularity that we are trying to learn. Observed data is those given n points (x,y), The goal in the curve fitting problem is to do predictions for the target variable t given some new value of the input variable x on the basis of a set of $x(x_1, x_2, \ldots\ldots, x_n)^T$, training data comprising N input values 1 N and their $t(t_1, t_2, \ldots\ldots, t_n)^T$ corresponding target values 1 N. We can express our uncertainty over the value of the target variable using a probability distribution. For this purpose, we shall assume that, given the value of x, the corresponding value of t has a Gaussian distribution with a mean equal to the value y(x, w) of the polynomial curve given by (2). Thus we have:

$$\mathrm{(t|w,x,\beta)} = N(t|y(x,w),\beta^{-1}) \quad (3)$$

where, for consistency with the notation in later chapters, we have defined a precision parameter Is corresponding to the inverse variance of the distribution.

In the curve fitting problem, we are given the training data x and t, along with a new test point x_t , and our goal is to predict the value of t. We therefore wish to evaluate the predictive distribution $p(t|x_t, x, t)$. Here we shall assume that the parameters Is

and Is are fixed and known in advance (later we would discuss how many parameters can be inferred from data in a Bayesian setting).

10.3 Bayesian curve fitting

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

$$p(t|x_t, x, t) = \int p(t|x_t, w)p(w|x, t)dw \quad (4)$$

Here $p(t|x_t, w)$ is given by (3), 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. This posterior distribution is a Gaussian and can be evaluated analytically. Similarly, the integration can also be performed analytically with the result that the predictive distribution is given by a Gaussian of the form.

$$p(t|x_t, x, t)N(t|m(x_t), s_2(x_t)) \quad (5)$$

where the mean and variance are given by

$$m(x_t) = \beta\varphi(x_t)^T * S * \sum_{n=1}^{N} \varphi(x_n)t_n \quad (6)$$

$$s^2 = \beta^{-1} + \varphi(x_t)^T * S * \varphi(x_t) \quad (7)$$

Here the matrix S is given by

$$S^{-1} = \alpha I + \beta \sum_{n=1}^{N} \varphi(x_n)\varphi(x_t)^T \quad (8)$$

where I is the unit matrix, and we have defined the vector φ(x) with elements $\varphi_i(x) = x^i \ \ for \ \ i = 0,1,2 \ldots\ldots, M$.

We see that the variance, as well as the mean, of the predictive distribution in (5) is dependent on x. The first term in (7) represents the uncertainty in the predicted value of t due to the noise on the target variables and was expressed already in the maximum likelihood predictive distribution (3) through β_{ML}^{-1} . However, the second term arises from the uncertainty in the parameters w and is a consequence of the Bayesian treatment.

Programming Technologies

We use C++ under Windows to do Bayesian curve fitting for the next day stock price prediction. The strategy is below:

- Read the dataset collected by PHP which contains several tables, each table contains different stock price information.
- Trying to find the last date in that dataset.
- Using recent price saved in tables to do Bayesian curve fitting(the last one year's price) to predict the price tomorrow.

10.4 Support vector machine (SVM)

Definition

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite- dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $k(x, y)$ selected to suit the problem. The hyper planes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is

constant. The vectors defining the hyper planes can be chosen to be linear combinations with parameters αi of images of feature vectors xi that occur in the data base. With this choice of a hyper plane, the points x in the feature space that are mapped into the hyper plane are defined by the relation:
$\sum_i \alpha_i k(x_i, x) = constant$. Note that if $k(x, y)$ becomes small as y grows further away from x, each term in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i . In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyper plane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space.

10.5 Support vector regression (SVR)

The classic SVM is for classification, but SVM can used for regression too. Then we can use it to predict the time series. It is also called SVR (Support vector regression). A version of SVM for regression was pro- posed in 1996 by Vladimir N. Vapnik, Harris Drucker, Christopher J.C. Burges, Linda Kaufman and Alexander J. Smola. This method is called support vector regression (SVR). The model produced by support vector classification (as described above) depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction. Another SVM version known as least squares support vector machine (LS-SVM) has been proposed by Suykens and Vandewalle.

Training the original SVR means solving minimize $\frac{1}{2}\left|\left|w\right|\right|_2$ subject to

$$\begin{cases} y_i - < w, x_i > - b \leq \varepsilon \\ < w, x_i > + b - y_i \leq \varepsilon \end{cases}$$

where x_i is a training sample with target value yi . The inner product plus intercept $< w, x_i > + b$ is the prediction for that sample, and ε is a free parameter that serves as a threshold: all predictions have to be within an ε range of the true predictions. Slack variables are usually added into the above to allow for errors and to allow approximation in the case the above problem is infeasible.

Programming Technologies

Because the price of the stock is kind of time series, so we can use support vector regression to predict it. We use Libsvm and Matlab to implement the SVR to do it. LIBSVM and LIBLINEAR are two popular open source machine learning libraries, both developed at the National Taiwan University and both written in C++ though with a C API. LIBSVM implements the SMO algorithm for kernelized support vector machines (SVMs), supporting classification and regression. LIBLINEAR implements linear SVMs and logistic regression models trained using a coordinate descent algorithm. The SVM learning code from both libraries is often reused in other open source machine learning toolkits, including GATE, KNIME, Orange and scikit-learn. Many bindings to it exist for programming languages such as Java, MATLAB and R.11 Our SVM predict program will read the data from the dataset, and then do the prediction. After getting the predict value, it will write back the value to dataset. The web can just writing data and reading result from the dataset.

10.6 Artificial neural network

Network function

The word network in the term 'artificial neural network' refers to the interconnections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons, some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations.

An ANN is typically defined by three types of parameters:

- The interconnection pattern between the different layers of neurons.
- The learning process for updating the weights of the interconnections
- The activation function that converts a neuron's weighted input to its output activation.

Mathematically, a neuron's network function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is the nonlinear weighted sum, where $f(x) = K(\sum_i w_i g_i(x))$, where K (commonly referred to as the activation function) is some predefined function, such as the hyperbolic tangent. It will be convenient for the following to refer to a collection of functions g_i as simply a vector $g = (g_1, g_2, \ldots\ldots, g_n)$.

Supervised learning

In supervised learning, we are given a set of example pairs $(x, y), x \in X, y \in Y$ and the aim is to find a function f :X →Y in the allowed class of functions that matches the examples. In other words, we wish to infer the mapping implied by the data; the cost

function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain.

A commonly used cost is the mean-squared error, which tries to minimize the average squared error between the network's output, f (x), and the target value y over all the example pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called multilayer perceptrons (MLP), one obtains the common and well-known backpropagation algorithm for training neural networks.

Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher", in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

Programming Technologies

We totally have 4 layers which means we set two hidden layer together with input layer and output layer. For input layer, we choose continuous 10 days' prices as 10 inputs, 5 hidden neurons for each hidden layer and one output neuron. We choose sigmoid function for our activation function which really performs well.

Backpropagation is our way to adjust weights. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function.

Our training set about 300 days' prices before the days we want to predict. In each batch, we have 100 compositions of 7 inputs. Then, when we have already trained, we choose the last 10 days to forecast the next day's price.

10.9 KDJ Indicator

Definition

KDJ indicator is also called the random index, by George. Blue with Dr. (George Lane) was first proposed, is a very innovative, practical and technical analysis indicators, which at first used in futures markets analysis, was widely used for short-term stock market trend analysis, futures and stock market technical analysis tools most commonly used. The J line represents the divergence of the %D value from the %K . The value of J can go beyond [0, 100] for %K and %D lines on the chart. Stochastic KDJ generally based on the principles of statistics, through a specific period (usually 9, 9 weeks, etc.) appeared within the high, low, and the last closing price and computing cycles of these three the ratio between the relations to calculate the final calculation of random values immature period RSV, and then smoothed moving average method to calculate the K values, D values and J values, and plotted graphs of determining the stock trend.

Computation Method

Firstly, we should calculate the RSV value of a specific period(N days, N weeks etc.), which means the immature random parameter value, then calculate the K value, D value, J value. Take KDJ with period of N days as example, its formulas is:

$$N\ \ day's\ \ RSV = \frac{C_n - L_n}{H_n - L_n} * 100 \qquad (10)$$

In this formula, C_n is the price of the Nth day; L_n is the lowest price during N days; H_n is the highest price during the N days. Then, to calculate the K value and D value:

$$that\ \ day's\ \ K = \frac{2}{3} * the\ \ previous\ \ day's\ \ K + \frac{1}{3} * that\ \ day's\ \ RSV$$

$$that\ \ day's\ \ D = \frac{2}{3} * the\ \ previous\ \ day's\ \ D + \frac{1}{3} * that\ \ day's\ \ K$$

$$J = 3 * that\ \ day's\ \ K - 2 * that\ \ day's\ \ D$$

If lack of the previous day's K and D value, set them to 50

In our application, we choose 9 days as the period of the KDJ, and we would use the following formula:

$$9\ \ day'sRSV = \frac{C-L_9}{H_9-L_9} * 100 \qquad (11)$$

In this formula, C is the closing price of the 9th day; L9 is the lowest price during 9 days; H9 is the highest price during the 9 days.

$$K = \frac{2}{3} * 8th\ \ day's\ \ K + \frac{1}{3} * 9th\ \ day's\ \ RSV$$

$$D = \frac{2}{3} * 8th\ \ day's\ \ D + \frac{1}{3} * 9th\ \ day's\ \ K$$

$$J = 3 * 9th\ \ day's\ \ K - 2 * 9th\ \ day's\ \ D$$

If lack of the previous day's K and D value, set them to 50.

Working process:

According to the general standards, if K, D, J these three values are below 20 as oversold zone, it is a buy signal;

If K, D, J these three values are above 80 as the overbought area, it is a sell signal;

If K, D, J these three values are between 20-80, it represents hold or sit- out.

A negative value of J combined with %K and %D at the bottom range indicates a strong over sold signal.

Likewise, when the J value goes above 100, combined with %K and %D at the top range, it will indicate a strong over bought signal.

In an increasing trend, when the K value is bigger than D value and the K line is increasing to break through the D line, it is the buying signal;

In a decreasing trend, when the K value is smaller than the D value and the K line is decreasing to break through the D line, it's a selling signal. The KDJ indicator is not suitable for the inactively trading stock with a small circulation. But high accuracy for the popular stock with large circulation.

When the speed of the K and D being increasing or decreasing is reducing, the gradient becoming flat is a reversal signal in short term.

Take sony historical data as an example:

Programming Technologies

We use PHP, SQL and JavaScript to implement the prediction algorithm. First, we set default value of K and D to 50 and J to 0 of specific stock for each day which are saved in the MySQL database. Second, we get the stock data from the MySQL and use PHP to implement the formulas mentioned above to calculate the K, D and J. Then, we update the K, D and J value saved in the MySQL database. Third, we use JavaScript to plot the line chart of K, D and J showing the trend.

10.10 Volume ratio (VR)

Definition

The Up/Down Volume Ratio is a powerful technical tool that identifies stocks that have a high probability of experiencing either a prolonged up or down move. This indicator identifies stocks that are either under accumulation (Bullish) or experiencing distribution (Bearish). The Up/Down Volume Ratio is computed by totaling the stock's volume on days when it closes up and divide that total by the volume traded on days when the stock closed down. The assumption is that if a stock closes up for the day, the volume was buying induced and thus the stock is under accumulation. Conversely, if a stock closes down for the day, the trading activity is deemed to be selling induced, a sign of distribution.

Calculation method

$$VR = \frac{total\ \ volume\ \ at\ \ rising\ \ days\ \ of\ \ N\ \ days}{total\ \ volume\ \ at\ \ declining\ \ days\ \ of\ \ N\ \ days}$$

And the application of the law:

- VR fell to below 40%, the market is very easy to form the bottom.
- The VR value of the most general distribution in around 150%, once over 250%, the

market is very easy to produce a bull market.

- The VR of more than 450%, should be high-grade crisis consciousness, pay attention to the inversion.

Programming Technologies

We use PHP, SQL and JavaScript to implement the prediction algorithm.

Moving average convergence/divergence(MACD)

Definition

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of prices. The MACD series is the difference between a "fast" (short period) exponential moving average (EMA), and a "slow" (longer period) EMA of the price series. The average series is an EMA of the MACD series itself.

Calculation method

- First, calculate the close price of SHORT term's day index smooth moving average and LONG term's day smooth moving average, representing EMA (SHORT) and EMA (LONG).
- Then find the difference between the smooth moving average of the two indices, the formula is like:

$$DIFF = EMA(short) - EMA(long)$$

- Calculate the DIFF's M day average index smooth moving average, recorded as DEA.
- Finally, we can easily get the value of MACD, formula:

$$MACD = 2 * DIFF - DEA$$

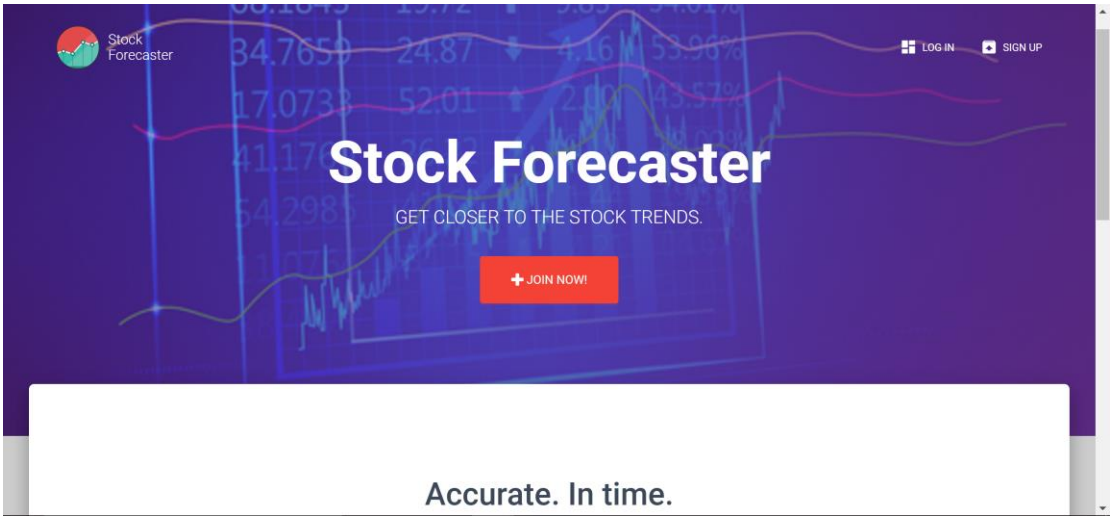
- MACD is usually drawn as the zero axis fluctuations in the column chart.

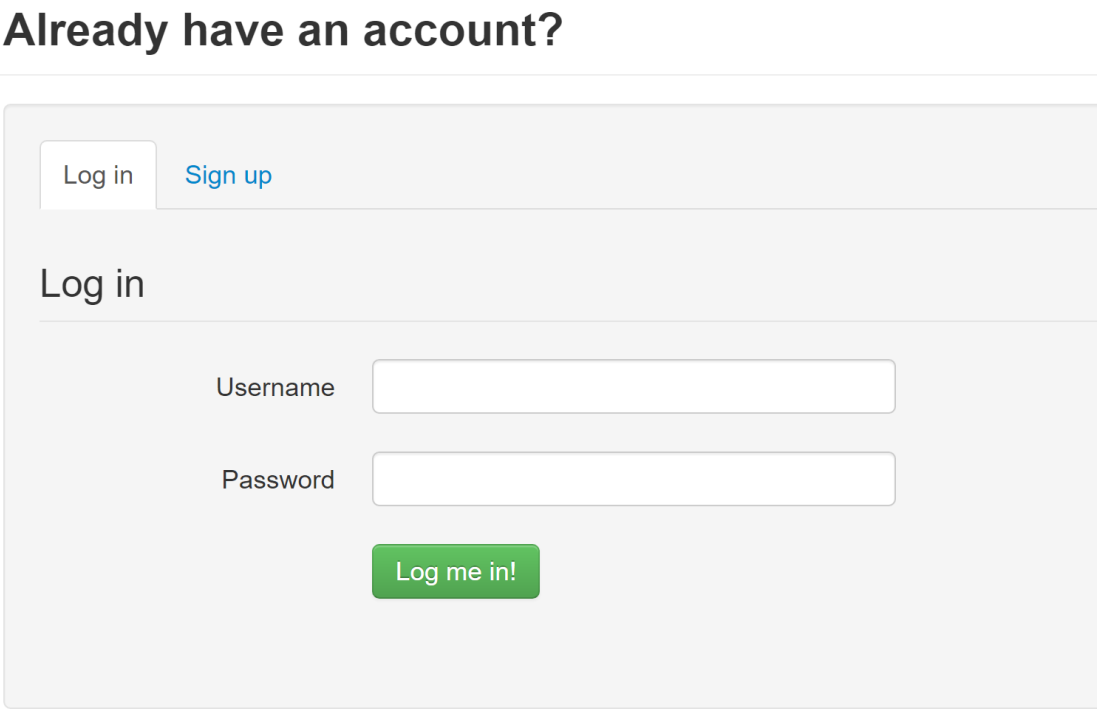
Programming Technologies

We use PHP, SQL and JavaScript to implement the prediction algorithm.

11 USER INTERFACE DESIGN AND IMPLEMENTATION

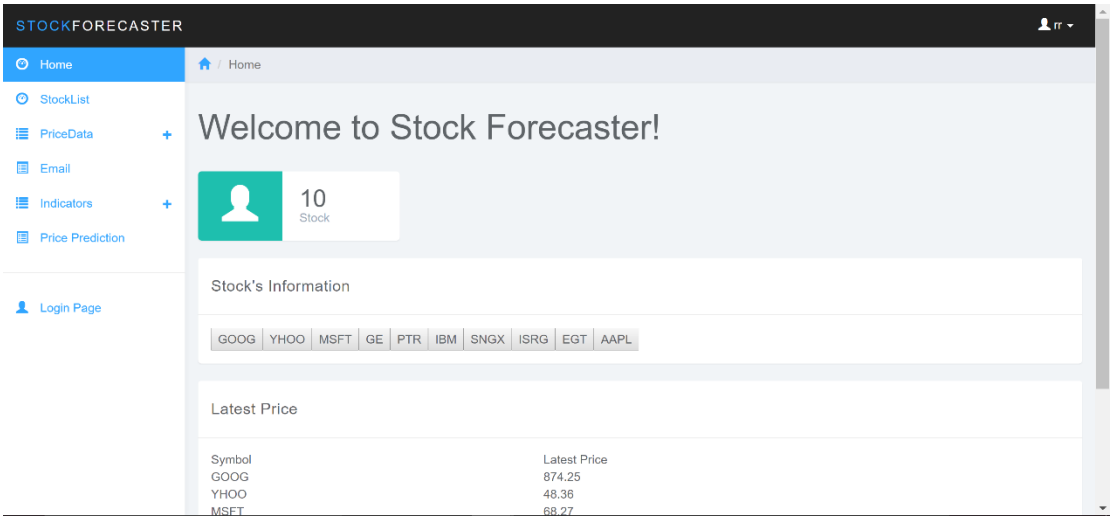
11.1 Overview





This is our log in page.

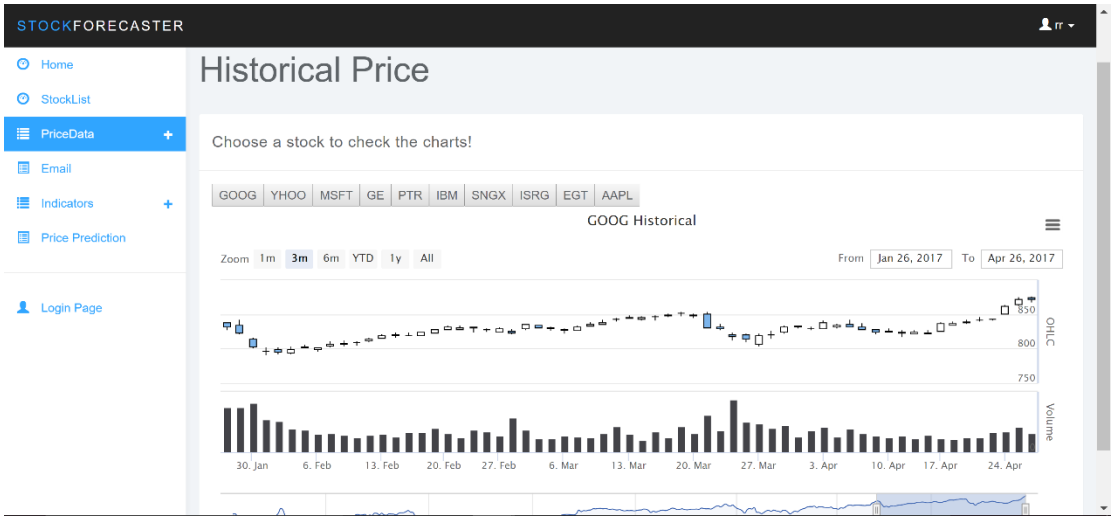
11.2 Home Page

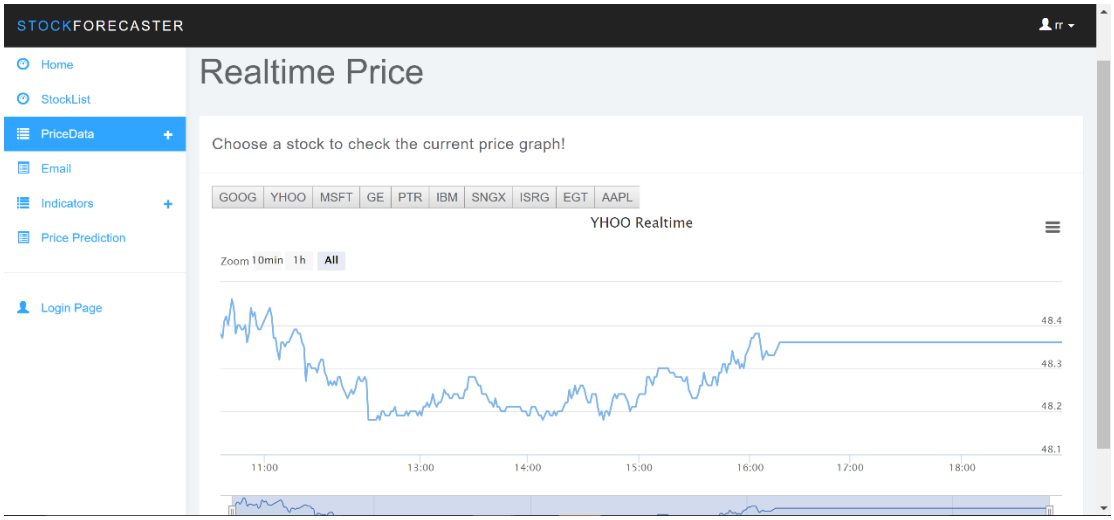


This is our main interface of stock predictor system.

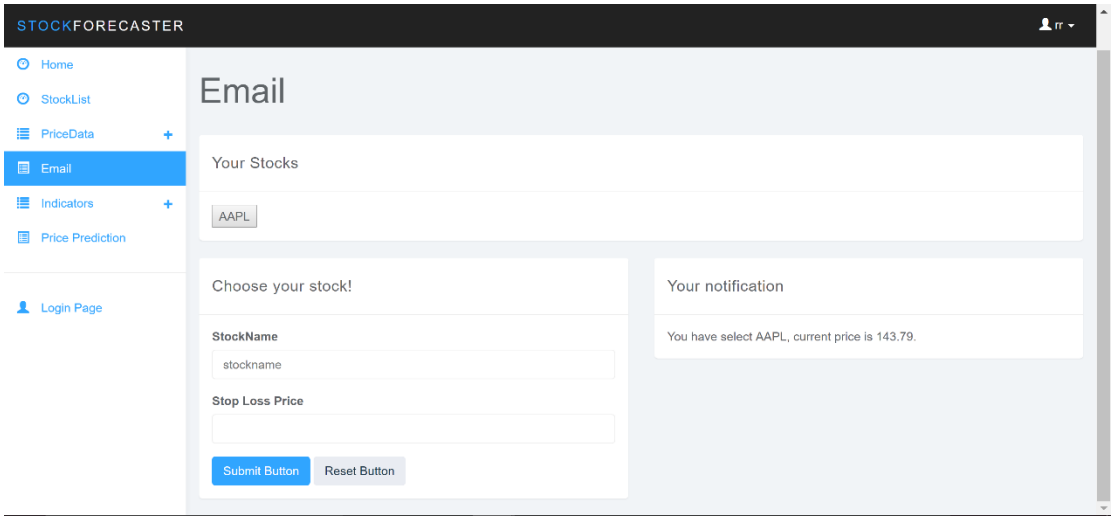
By logging in the website, you can see the main stock price from the main body. At this page, we list the name and latest price of the 10 stocks.

11.3. Main Functions

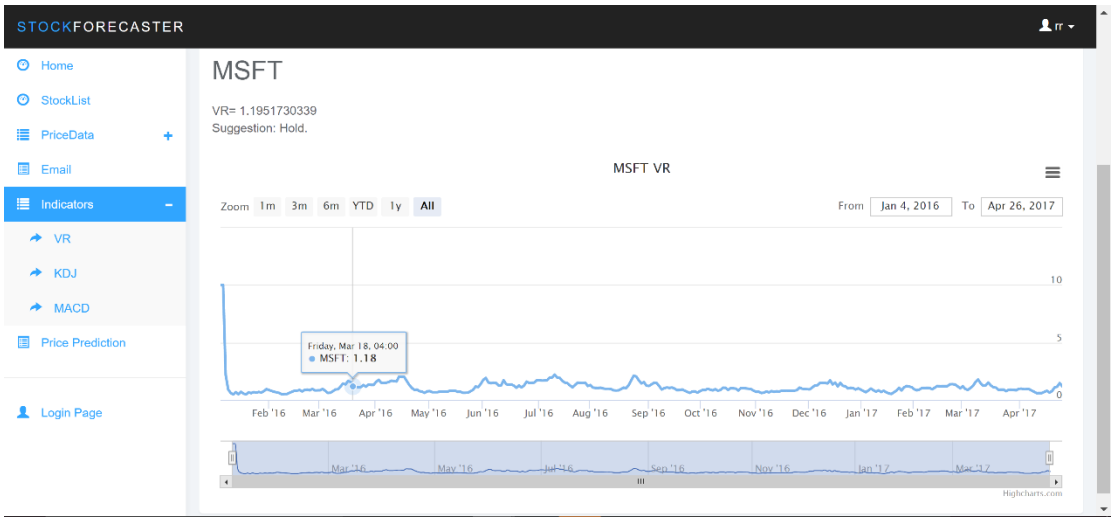


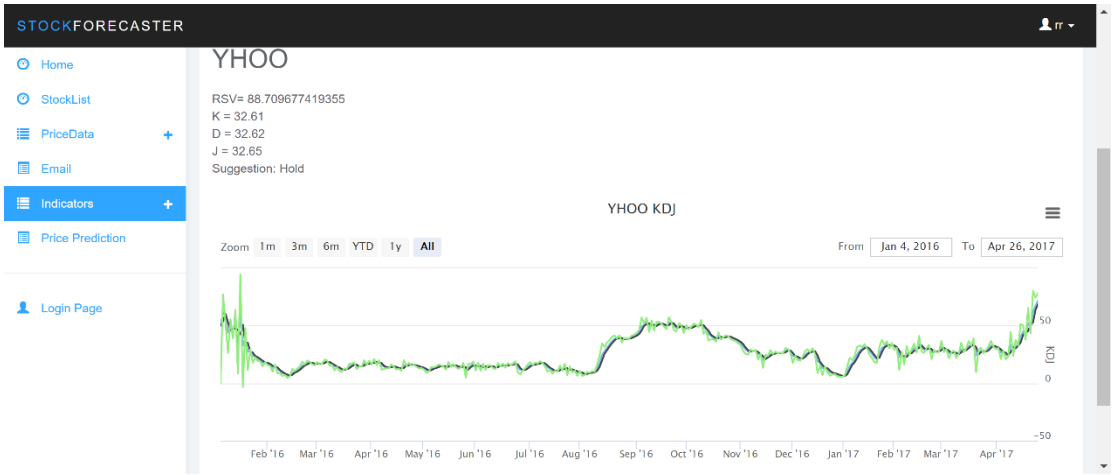


At the priceData button, user can choose a stock by name and choose historical price or real time price. And the website will show the diagram of the chosen stock.

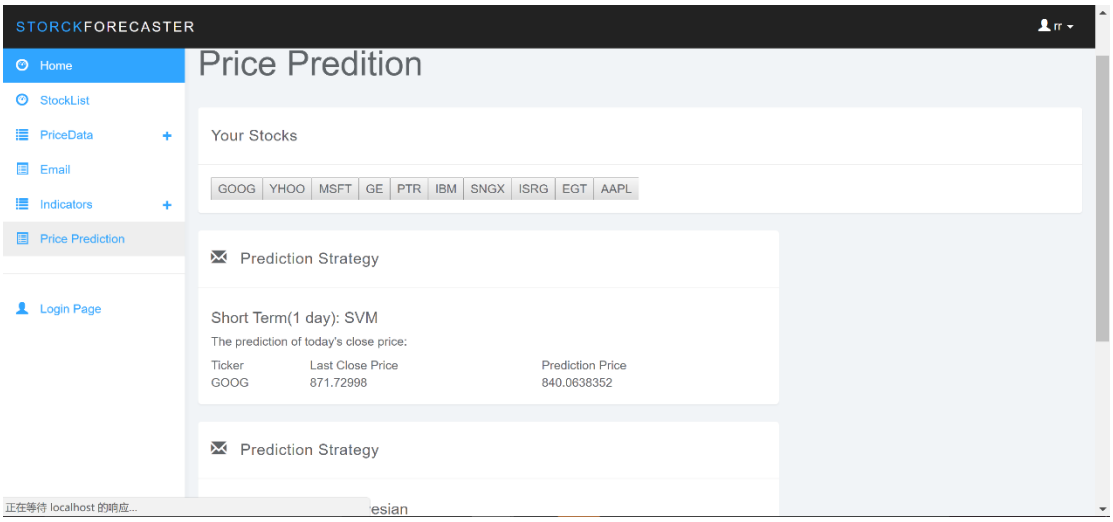


At the E-mail page, user can choose a stock and enter the stop loss price to get E-mail notification.





We implemented 3 indicators to give user advices about the stocks.



We implemented 3 prediction strategies to do short time and long time predictions.

12 FUTURE WORK

In the future, we still plan to do more explorations and optimizing on this project.

- First of all, we will add more stock information to the database. Currently, we just

save 10 stocks and the information of each stock is incomplete. To search the stock, we can only use the symbol. But in the future, we'd better increase more methods pointing to the stock.

- Secondly, we want to add more function to optimize the website. Stocks that users interested in should be collected so that they can be observed better. And also, current query function is just a prototype because the content and range of query have been assigned already. It still need more promotion in the future.
- At last, the login and sign up modules need more work. We find that hashtable could be used to save the user information so that the security of user account will be increased.

13 ACKNOLDGMENT

These guidelines have based out of an outline prepared by Prof. Shiyu ZHOU for ECE 568 web application. We gratefully acknowledge helpful discussions with Prof. Shiyu ZHOU of the ECE department.

14. REFERENCE

[1] http://finance.yahoo.com/

[2] Software Engineering (online version), Chapters 6-8, by Ivan Marsic

[3] Web-Based Application Development, by Ralph F. Grove Publisher: Jones & Bartlett Publishers; 1 edition (April 22, 2009)ISBN-10: 0763759406;

[4] Database Systems: Design, Implementation, and Management

[5] MACD https://en.wikipedia.org/wiki/MACD

[6] ANN https://people.eecs.berkeley.edu/~akar/IITK_website/EE671/report_stock.pdf

[7] SVM http://www.svms.org/finance/HuangNakamoriWang2005.pdf

http://cs229.stanford.edu/proj2012/ShenJiangZhang-StockMarketForecastingusingMachineLearningAlgorithms.pdf

http://research.ijcaonline.org/volume41/number3/pxc3877555.pdf

[8] KDJ https://en.wikipedia.org/wiki/Stochastic_oscillator