SOFTWARE ENGINNERING OF WEB APPLICATIONS

WEB-BASED STOCK FORECASTERS

Team: Group #25
Rutgers, the State University of New Jersey
Department of Electrical and Computer Engineering

April 29, 2016

Contents

A Individual Contribution Breakdown.

Details:
Introduction and Background Information: Wangzhe Chen
Analysis: Siyuan Zhong, Jianqin Gao, Wangzhe Chen, Kaifei lei
Database Creation: Siyuan Zhong, Jianqin Gao
UML Diagrams: Wangzhe Chen
Use case(s): Kaifei Lei, Wangzhe Chen
Web service interface: Siyuan Zhong, Jianqin Gao
User Interface:Kaifei Lei,Siyuan Zhong, Jianqin Gao
Prediction Strategy:Siyuan Zhong, Jianqin Gao, Wangzhe Chen, Kaifei lei
KDJ indicator: Jianqin Gao, Wangzhe Chen
Volume ratio (VR): Siyuan Zhong, Kaifei Lei
Bayesian Algorithm: Siyuan Zhong, Kaifei Lei
SVM algorithm: Jianqin Gao, Siyuan Zhong
ANN:Wangzhe Chen,Jianqin Gao
Planning: Siyuan Zhong, Jianqin Gao, Wangzhe Chen, Kaifei lei
Report documentation: Kaifei Lei

Totally:
All members contributed equally

B Background

Nowadays, people are more and more eager 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 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 quick and cheap fashion. 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 more easy and 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.

C System Design

C.1 System description

There are three levels of the whole system: interface level, service level and data level.[3] The user input the user id and passcode to login the system (If this is the first time using this Web, the user need to register for an id and passcode). The user input 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 four key option for user to choose: Graphs, Quotes, News and Prediction. 1) Get Quote This service returns the current price for a selected stock from the database. These prices are ticked every 59 seconds from http://finance.yahoo.com 2) Get Graphs These graphs are dynamically generated using software using historical prices 3) Get Prediction This command gets buy-sell recommendations for users using Web services or PHP scripts using Indicator algorithms, as well as predicted price for a particular stock.

If a user wants to see the graph of a particular stock, the information is directly sent to the database, corresponding data is fetched, required actions are performed on it and the required graph is sent back to the top layer that is the interface. If the user wants to get the quotes of a particular stock, he selects that stock and clicks the "quotes" button. In this case the Web Service is called which takes the data from the data base, performs the required actions and send the required data to the user. If the user wants to get the prediction of a stock, the user again selects the stock from the list, clicks the "Prediction" button. There are two ways in which the system works: one is using the Web Services and the other one is using the Indicators which can be invoked by the user. Based on the method selected (Web Services or the Indicators), data is again fetched from the database and the desired results are sent back to the user through the services level. The database is kept up to date with the latest stock data. This is because the PHP data mining script has been scheduled to run every 15 minutes. After every 15 minutes, new data is updated into the database. Google Finance (http://finance.google.com) is being used for fetching the historical data for our system. In the above figure the top cloud represents the data mining application of our project where our PHP script get historical stock data which includes the following: High, Low, Close, Open and volume for a list of companies and stores it in the database (MySQL)[4].

C.2 System block Diagram

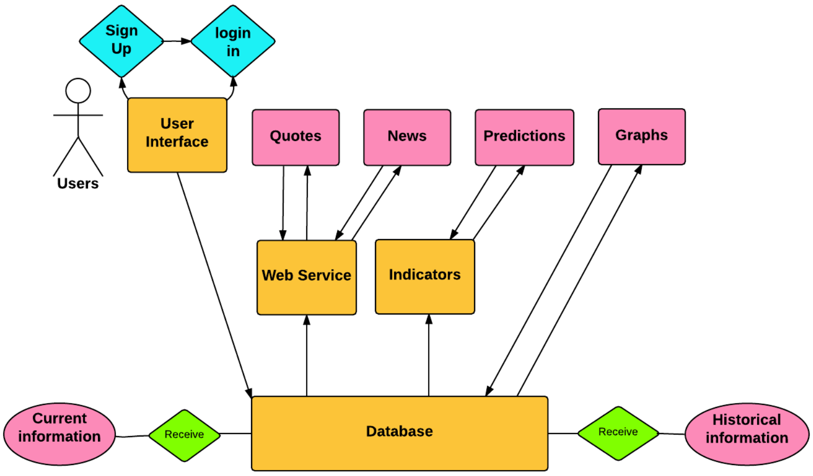


Figure 1: System Design Diagram

D Functional Requirements Specification

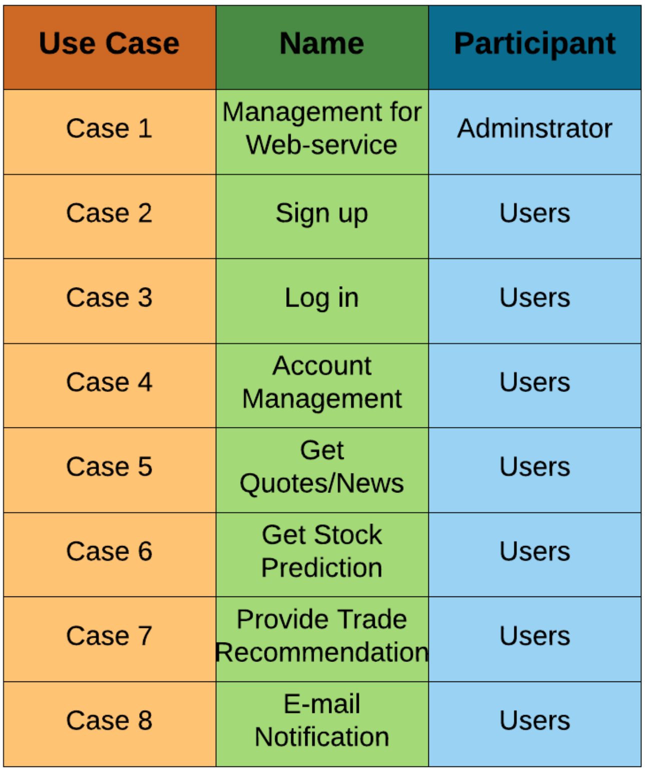
D.1 Stakeholders

Three Stakeholders can be identified:
1. Visitors: any visitors can registered in our system with email, username and password.
2. Users: any user could log in the system and get web services.
3. Administrator: maintains and updates website services.

D.2 Actors and Goals

Our system has both human and non-human actors.
RegisteredUser : a registered user.
Visitor : any unregistered user.
Administrator : The manager that is in charge or keeping the system updated and in working order.
PredictionAlgorithm : 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!FinanaceAPI :The API is where we will pool all stock data from to store into the database to generate the prediction.

D.3 Use Cases



Use Case	Name	Participant
Case 1	Management for Web-service	Adminstrator
Case 2	Sign up	Users
Case 3	Log in	Users
Case 4	Account Management	Users
Case 5	Get Quotes/News	Users
Case 6	Get Stock Prediction	Users
Case 7	Provide Trade Recommendation	Users
Case 8	E-mail Notification	Users

Figure 2: Use cases

UseCase1 : ManagementforWeb – service

For the administrator, they need to analysis whether the website is work normally or not. And also they need to management and maintenance of the Web-service keep the web serving the customers continually. It is a very important part of the whole system. For example, new users need be added to the database and current users will have their information updated in the data base.The diagram is shown as follow.

Adminstrator
Manage the Web service
Keep the Web serving
users continuously

Figure 3: Use Case 1: Management for Web-service Diagram

UseCase2 : Signup

The user of the application wants to create a new account. The user will open the application and be brought to the log in screen. Once at the log in screen the user will then select "Create an Account". The user will then be brought to an account creation page. The user will fill out the required information, such as name, user name, password, email address, job, qualifications, and others. Once completed and submitted, the information will be sent to the database. The database will then check the availability of the user name. If the user name is not available the database will send a notification to the application. The application will then ask the user to choose another user name. If the user name is available, then the database will set up the account, and log the user in. The user will then be brought to the home page.The diagram is shown as follow.

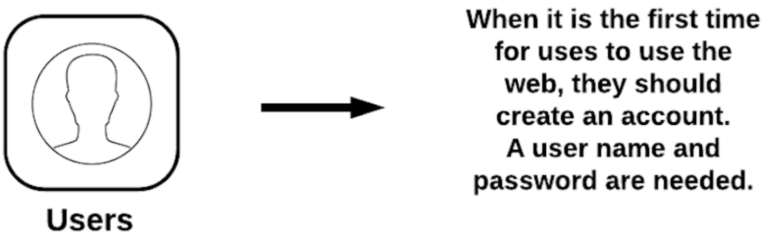

Users
When it is the first time for uses to use the web, they should create an account. A user name and password are needed.

Figure 4: Use Case 2: Sign up

UseCase3 : Login

After registration, the customers already have an account. If the customers input the correct combinations of user name and passcode, then they will login the web service successfully.The diagram is shown as follow.

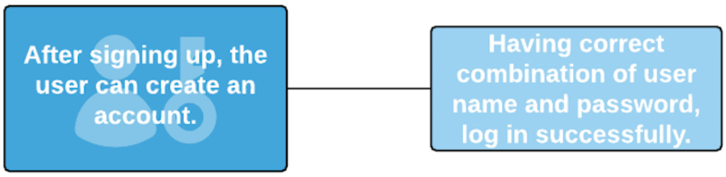

After signing up, the user can create an account.
Having correct combination of user name and password, log in successfully.

Figure 5: Use Case 3: Log in

UseCase4 : AccountManagement

The customers can manage their account. They can select some stock that they are interesting in. They can do "add" and "remove" operation.The diagram is shown as follow.

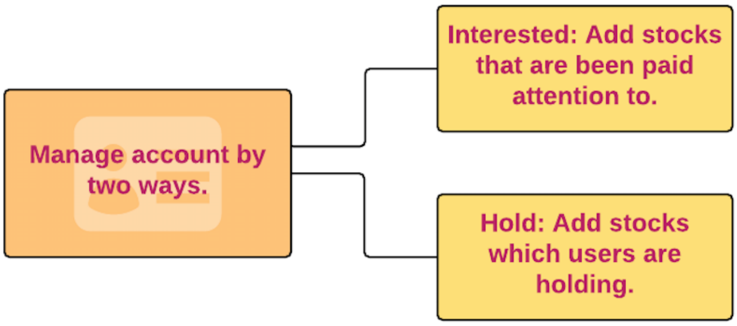

Manage account by two ways.
Interested: Add stocks that are been paid attention to.
Hold: Add stocks which users are holding.

Figure 6: Use Case 4: Account Management

UseCase5 : GetQuotes/Graphs

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.The diagram is shown as follow.

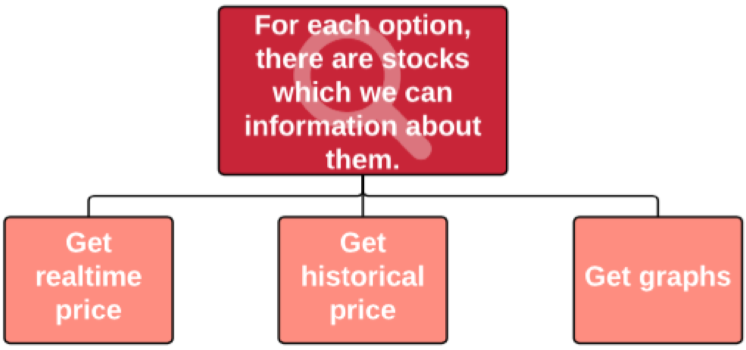

For each option, there are stocks which we can information about them.
Get realtime price
Get historical price
Get graphs

Figure 7: Use Case 5: Get Quotes/Graphs

UseCase6 : GetStockPrediction

The administrator have some patterns and indicators to analyze the stock's information, and then the web will present the several forecast (long-term and short-term) for each stock to customers. For this system, we use five kinds of algorithms to predict the result, the user can compare the result and decide to buy the stock or not. The five algorithms are SVM, Bayesian, Neural Network, VR and KDJ.The diagram is shown as follow.

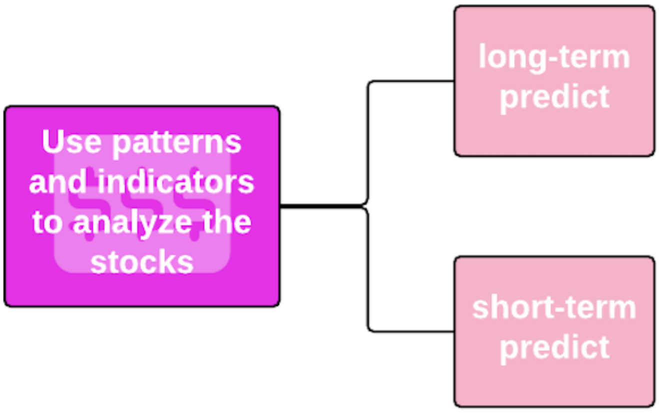
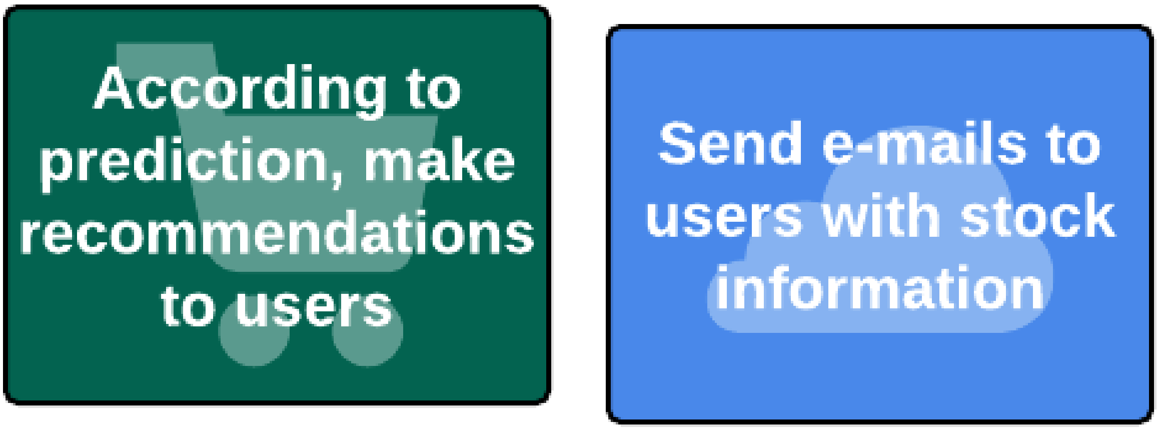


Figure 8: Use Case 6: Get Stock Prediction

UseCase7 : ProvideTradeRecommendation

The web can give the customers recommendation of trade (buy, sell, hold or sit-out), after the web analyze the result of forecast. We will analysis the four results of our prediction, and make a conclusion to users which trade is the most reasonable. We hope the users can reduce the lost and earn more money.The diagram is shown as follow.


According to prediction, make recommendations to users
Send e-mails to users with stock information

(a) Use Case 7: Provide Trade Recommendation

(b) Use Case 8: Send E-mail Notifications

Figure 9

UseCase8 : SendE – mailNotifications

It is our feature, when customers are not online. Our web can send updated information like stoploss price.

E Class Diagram and Interface Specification

E.1 Class Diagram

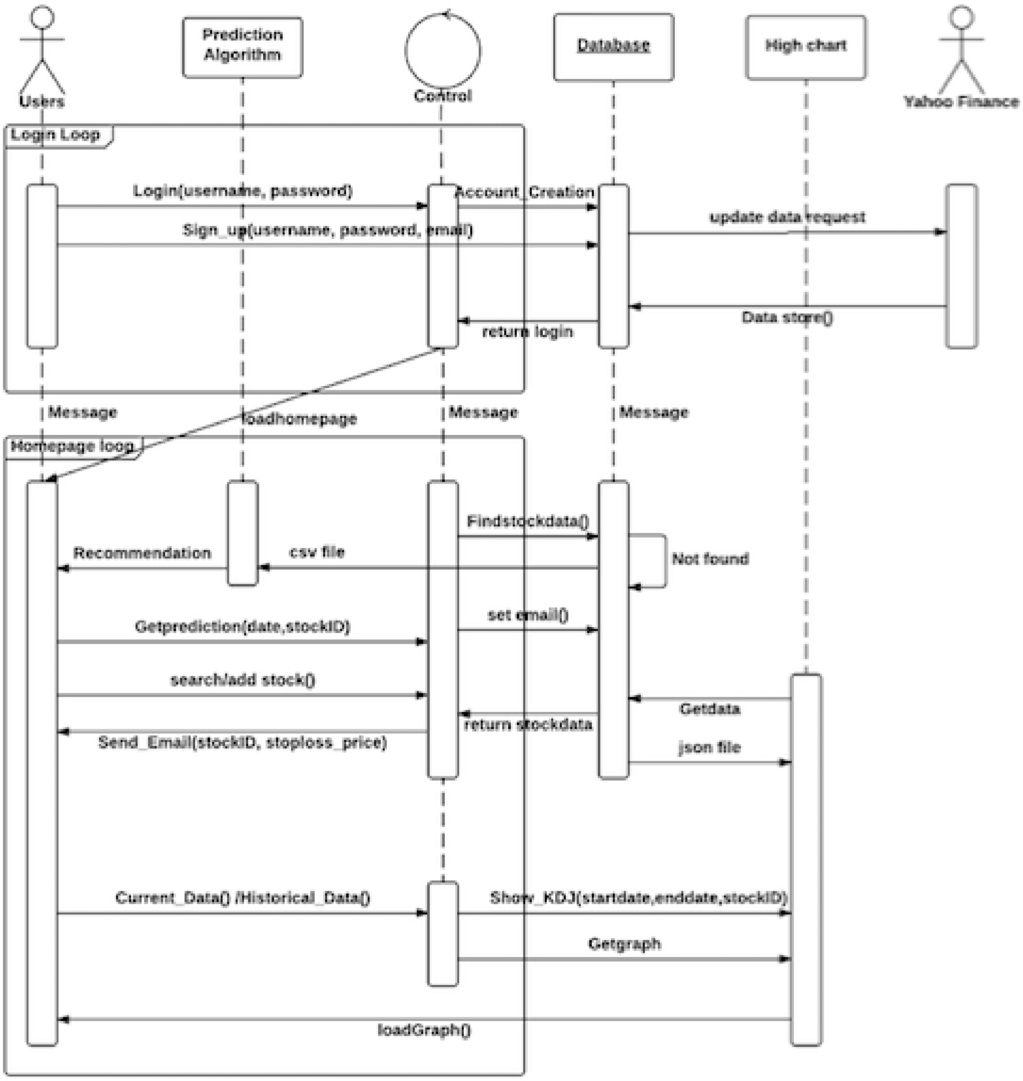


Figure 10: system class diagram

E.2 Data Types and Operation Signatures

Void Login(string username, string password): Let users login our website, enter personal information as well as stock management.
Void Sign_up(string username, string password,string email): Let visitors sign up our website and create a personal account which can store their own stock informations.
Void Add_interested(string StockID, string username): Let users add stock to their Interested list so the website will keep track on them.
Void Add_selected(string stockID, string username,int volume): Let users add stock to their hold list and store stock informations in our system.
Double Get_Prediction_NaiveBayesian(string date,string stockID): Show the prediction results using Naive Bayesian method.
Double Get_Prediction_NeuralNetwork(string date,string stockID): Show the prediction results using Neural Network method.
Double Get_Prediction_SVM(string date,string stockID): Show the prediction results using SVM method.
Void Show_KDJ(string startdate, string enddate, string stockID): Show KDJ graph to indicate the trend of the stock.
Void Show_VR(int volume, string startdate, string enddate): Show VR index to indicate the trend of the stock.
Void Get_recommendation(string stockID): Provide recommendation of trade.
Void Send_Email(string stockID, double stoploss_price): Send email to inform user if the price reach the stop loss price which set by user.
Void DataBase_Connect():Connect our database with the website.
Void Data_Store():Store the data to database which is got from yahoo finance.

Void Account_Creation(string name,string password,string email):Create the user account,including the name,password and the email.
Void Current_Data():Get the real time of the price and the volume of the stock every 40 seconds.
Void Historical_Data():Get the history price information of the stock in the past one year.

F Web service

F.1 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 designed 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.

F.2 Developing Web Services with Axis

F.2.1 Server-side Development with Axis

At the server side (or the Web service side), the steps are as follows:
1. Define Java interface of the Web service (and a class that implements this interface)
2. Generate the WDSL document from the serviceâĂŹs Java interface (Java WSDL)
3. Generate the skeleton Java class (server-side proxy) from the WSDL document(WSDL → Java)
4. Modify the skeleton proxy to interact with the Java class that implements the Java interface (both created in Step 1 above)

F.2.2 Client-side Development with Axis

At the client side (or the service consumer side), the steps are as follows:
1. Generate the stub Java class (server-side SOAP proxy) from the WSDL document
2. Modify the client code to invoke the stub (created in Step 1 above)

G Prediction Strategy

G.1 Bayesian Network

G.1.1 Bayesian Theorem

Bayes' theorem is stated mathematically as the following equation:

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

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), a conditional probability, is the 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

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

G.1.2 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 be able to make predictions for the target variable t given some new value of the input variable x on the basis of a set of $x(x_1,...,x_n)^T$ training data comprising N input values 1 N and their $t(t_1,...,t_n)^T$ 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

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

where, for consistency with the notation in later chapters, we have defined a precision parameter Îš 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 Îś and Îš are fixed and known in advance (in later chapters we shall discuss how such parameters can be inferred from data in a Bayesian setting).

G.1.3 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

$$p(t|x_t, x, t) = \int p(t|x_t, w)p(w|x, t)\mathrm{d}w \tag{4}$$

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, and can be found by normalizing the right-hand side of (1.1). This posterior distribution is a Gaussian and can be evaluated analytically. Similarly, the integration in (1.3) can also be performed analytically with the result that the predictive distribution is given by a Gaussian of the form.[5]

$$p(t|x_t, x, t)N(t|m(x_t), s^2(x_t)) \tag{5}$$

where the mean and variance are given by

$$m(x_t) = \beta\phi(x_t)^T \cdot S \cdot \sum_{n=1}^{N} \phi(x_n)\, t_n \tag{6}$$

$$s^2 = \beta^{-1} + \phi(x_t)^T \cdot S \cdot \phi(x_t) \tag{7}$$

Here the matrix S is given by

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

where I is the unit matrix, and we have defined the vector $\phi(x)$ with elements $\phi_i(x) = x^i$ for i = 0, . . . , 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.

G.1.4 Programming Technologies

We use java under Windows and Eclipse to do Bayesian curve fitting, for the next day stock price prediction. The strategy is below: (1). Read the dataset collected by PHP, these contain at least 10 datasets, each dataset contains at least one year stock price information. (2). Trying to find the last date in that dataset. (3). Using close price in those datasets to do Bayesian curve fitting. Use the last 10 days' close price to predict the close price tomorrow. (4). After the code running up, form a dataset automatically with 1 rows. The row has the information of stock name, the last date data collected and the nest day, and the predicted stock price.

G.2 Support vector machine (SVM)

G.2.1 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 α_i of images of feature vectors x_i 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:

$\sum_i \alpha_i k(x_i, x) = \mathrm{constant}.$ 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.

G.2.2 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 proposed 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 $\frac{1}{2} \| w \|_2$ subject to

$$\begin{cases} y_i - < w, x_i > -b \le \epsilon \\ < w, x_i > +b - y_i \le \epsilon \end{cases} \tag{9}$$

where x_i is a training sample with target value y_i. 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.

G.2.3 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.[6]

G.3 Artificial neural network

G.3.1 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:
1.The interconnection pattern between the different layers of neurons.
2.The learning process for updating the weights of the interconnections
3.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 $f(x) = K\left(\sum_i w_i g_i(x)\right)$, where K (commonly referred to as the activation function[30]) 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 $g = (g_1, g_2, \ldots, g_n)$.

G.3.2 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 $f : X \rightarrow Y$ 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.

G.3.3 Programming Technologies

We totally have 3 layers which means we only set one hidden layer together with input layer and output layer. For input layer, we choose continuous 7 days' prices as 7 inputs, 3 hidden neurons 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 is the 107 days' closing 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 7 days to forecast the next day's price and according to this to predict next 2th day's. We do this repeatedly for 10 times to obtain long term predictions.

G.4 KDJ Indicator

G.4.1 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.[7]

G.4.2 Computation Method

First is to 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:

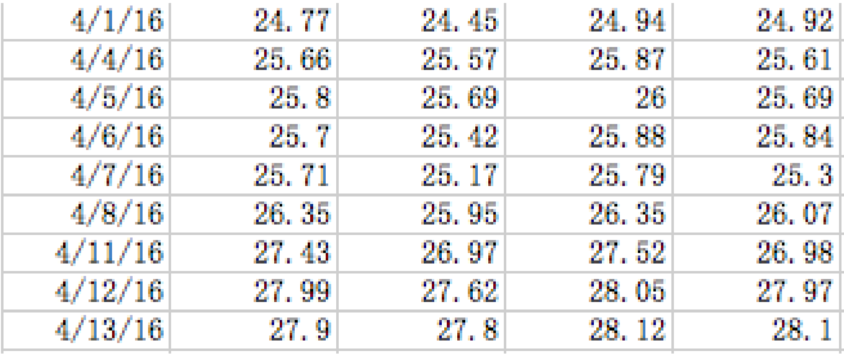
$$N\ day's\ RSV = (C_n - L_n)/(H_n - L_n) * 100 \tag{10}$$

In this formula, Cn is the closing price of the Nth day; Ln is the lowest price during N days; Hn is the highest price during the N days. Then, to calculate the K value and D value:
that day's K = 2/3 * the previous day's K + 1/3 * that day's RSV
that day's D = 2/3 * the previous day's D + 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:

$$9\ day's\ RSV = (C - L_9)/(H_9 - L_9) * 100 \tag{11}$$

In this formula, C is the closing price of the 9th day; L_9 is the lowest price during 9 days; H9 is the highest price during the 9 days.
K = 2/3 * 8th day's K + 1/3 * 9th day's RSV
D = 2/3 * 8th day's D + 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:



4/1/16	24. 77	24. 45	24. 94	24. 92
4/4/16	25. 66	25. 57	25. 87	25. 61
4/5/16	25. 8	25. 69	26	25. 69
4/6/16	25. 7	25. 42	25. 88	25. 84
4/7/16	25. 71	25. 17	25. 79	25. 3
4/8/16	26. 35	25. 95	26. 35	26. 07
4/11/16	27. 43	26. 97	27. 52	26. 98
4/12/16	27. 99	27. 62	28. 05	27. 97
4/13/16	27. 9	27. 8	28. 12	28. 1

Figure 11: KDJ example

The fifth column is the closing prices. Now, we calculate KDJ of date

4/13/2016:

$$\begin{aligned} C &= 28.1 \\ L_9 &= 24.92 \\ H_9 &= 28.1 \\ 9\ day's\ RSV &= (C - L_9)/(H_9 - L_9) * 100 = 100 \end{aligned} \tag{12}$$

Since we lack of the K and D of 4/12/2016, we can set them to 50.
K of 4/13/2016 = 2/3 * K of 4/12/2016 + 1/3 * RSV of 4/13/2016 =66.67
D of 4/13/2016 = 2/3 * D of 4/12/2016 + 1/3 * K of 4/13/2016 =55.56
J of 4/13/2016 = 3 * K of 4/13/2016 - 2 * D of 4/13/2016 =88.89
It is in an increasing trend, the K value is bigger than D value and the K line is increasing to break through the D line, so it is the buying signal.

G.4.3 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.

G.5 Volume ratio (VR)

G.5.1 Definition

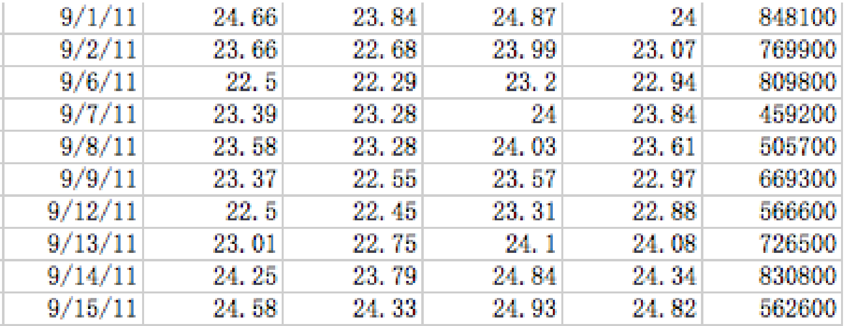
The Up/Down Volume Ratio is a powerful technical tool that identifies stocks that have a high probability of experiencing either a prolonged up or down move. This indicator identifies stocks that are either under accumulation (Bullish) or experiencing distribution (Bearish). The Up/Down Volume Ratio 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.

G.5.2 Calculation method

VR=total volume at rising days of N days/ total volume at declining days of N days
And the application of the law:
(1)VR fell to below 40%, the market is very easy to form the bottom.
(2)The VR value of the most general distribution in around 150%, once over 250%, the market is very easy to produce a bull market.
(3)The VR of more than 450%, should be high-grade crisis consciousness, pay attention to the inversion.

G.5.3 Programming Technologies

We use PHP, SQL and JavaScript to implement the prediction algorithm. Let's take Tesla Inc. as example.



9/1/11	24.66	23.84	24.87	24	848100
9/2/11	23.66	22.68	23.99	23.07	769900
9/6/11	22.5	22.29	23.2	22.94	809800
9/7/11	23.39	23.28	24	23.84	459200
9/8/11	23.58	23.28	24.03	23.61	505700
9/9/11	23.37	22.55	23.57	22.97	669300
9/12/11	22.5	22.45	23.31	22.88	566600
9/13/11	23.01	22.75	24.1	24.08	726500
9/14/11	24.25	23.79	24.84	24.34	830800
9/15/11	24.58	24.33	24.93	24.82	562600

Figure 12: VR example

In the past ten days, if open price is higher than close price, then add the volume to B.

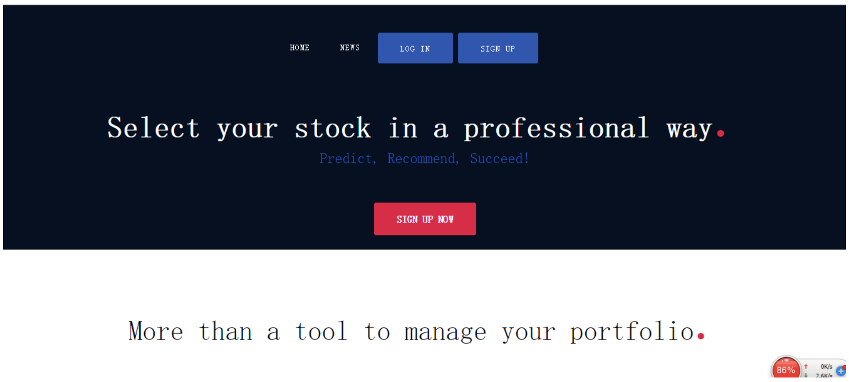
And if the close price is higher than open price, then add the volume to A.

So the VR of the last ten days is VR=A/B=4461200/2287300=1.9504=195.04%

And since 195.04% is in the safe range, so it's safe to say that the stock trend is stable. So the suggestion will be BUY.

H Special Features

H.1 Homepage



Select your stock in a professional way.

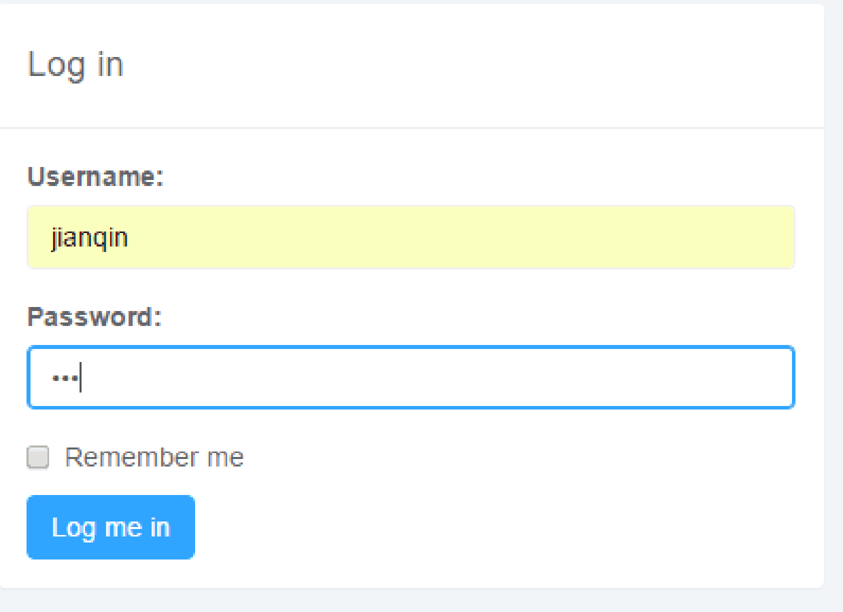
Predict, Recommend, Succeed!

More than a tool to manage your portfolio.

Figure 13: Homepage

The homepage is shown as former. You can click login and sign up button to jump to the certain page.

H.2 Log in



Log in

Username: jianqin

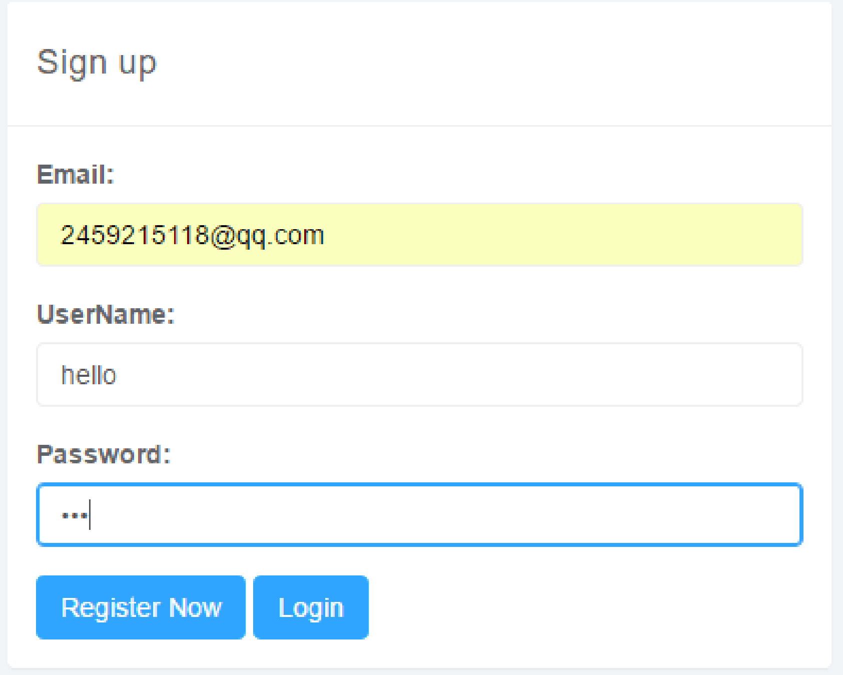
Password: •••

□ Remember me

Figure 14: login

Users input valid username and password to log in the stock system.

H.3 sign up



Sign up

Email:

2459215118@qq.com

UserName:

hello

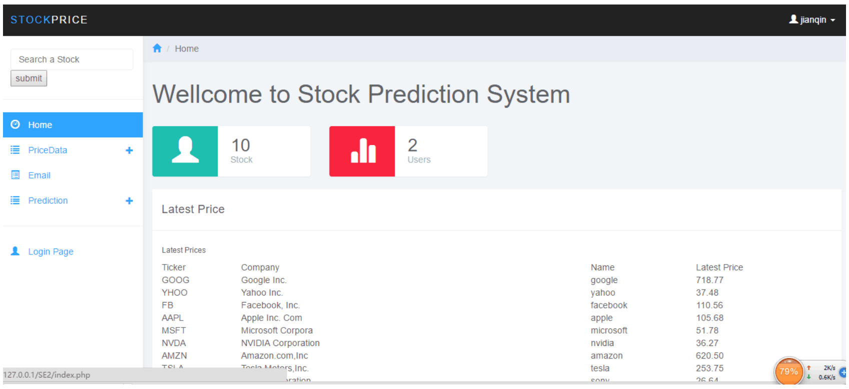
Password:

•••

Figure 15: sign up

Visitors could register in our system.

H.4 index



Wellcome to Stock Prediction System

10 Stock

2 Users

Latest Price

Latest Prices

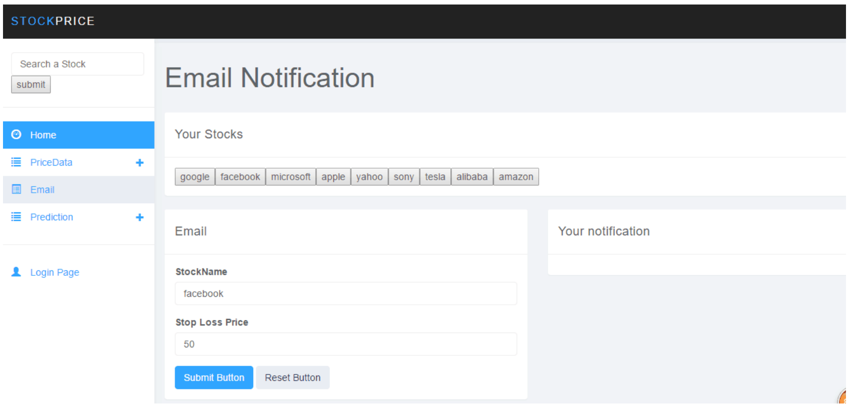
Ticker	Company	Name	Latest Price
GOOG	Google Inc.	google	718.77
YHOO	Yahoo Inc.	yahoo	37.48
FB	Facebook, Inc.	facebook	110.56
AAPL	Apple Inc. Com	apple	105.68
MSFT	Microsoft Corpora	microsoft	51.78
NVDA	NVIDIA Corporation	nvidia	36.27
AMZN	Amazon.com,Inc	amazon	620.50
TSLA	Tesla Motors,Inc.	tesla	253.75
	ration	sony	26.64

127.0.0.1/SE2/index.php

Figure 16: index

After logging in, Users can search any stock that exists in the database, and the website will show the user the description of the stock, the highest price in the last 10 days, the average price and the lowest price in the last one year.Also,we have all the query that project requires.

H.5 Email notification

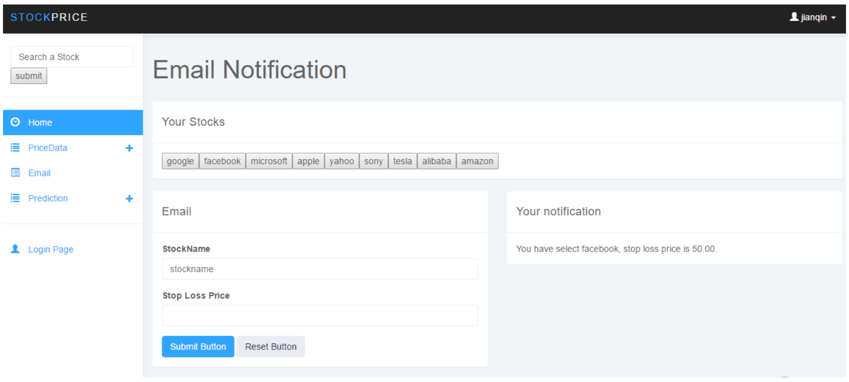


StockName: facebook

Stop Loss Price: 50

Figure 17: email set up

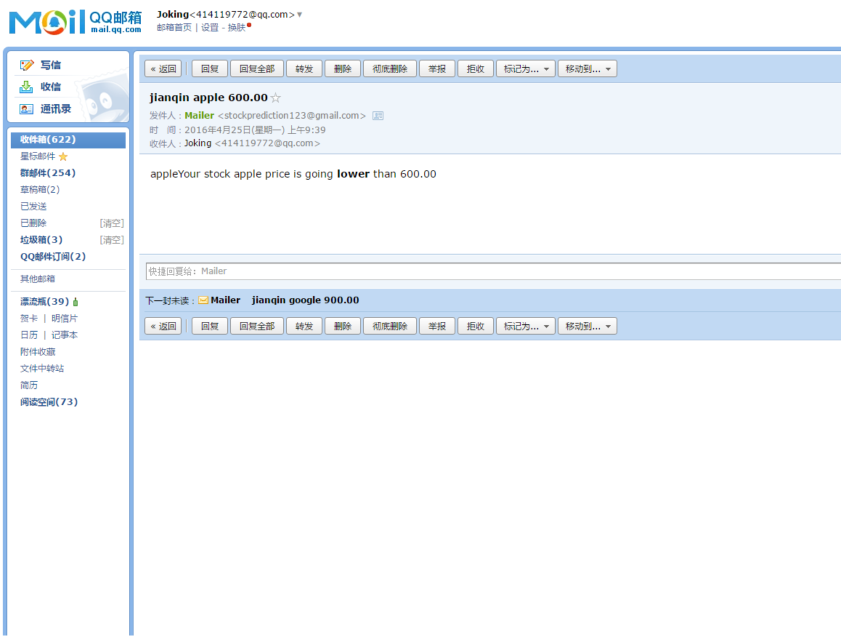
This part is our special features, the system allow users to input stock name which they listen and the stop loss price. After adding listening price and stock name, it will show on web page which is shown as follow,



You have select facebook, stop loss price is 50.00.

Figure 18: after adding

If the price in our database which gets from Yahoo finance is lower than the stop loss price, users will receive an e-mail.



Joking<414119772@qq.com>

jianqin apple 600.00

发件人：Mailer <stockprediction123@gmail.com>

时　间：2016年4月25日(星期一) 上午9:39

收件人：Joking <414119772@qq.com>

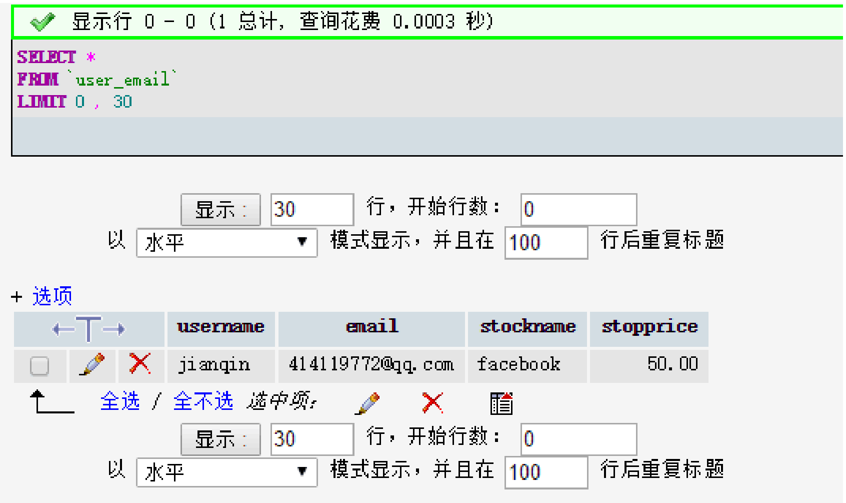
appleYour stock apple price is going **lower** than 600.00

快捷回复给：Mailer

下一封未读：Mailer　jianqin google 900.00

Figure 19: receive email

After sending e-mail, our database will delete the stop loss price that user set. Users could set the next stop loss price.



✓ 显示行 0 - 0 (1 总计, 查询花费 0.0003 秒)

```
SELECT *
FROM `user_email`
LIMIT 0 , 30
```

显示：30 行，开始行数：0

以 水平 模式显示，并且在 100 行后重复标题

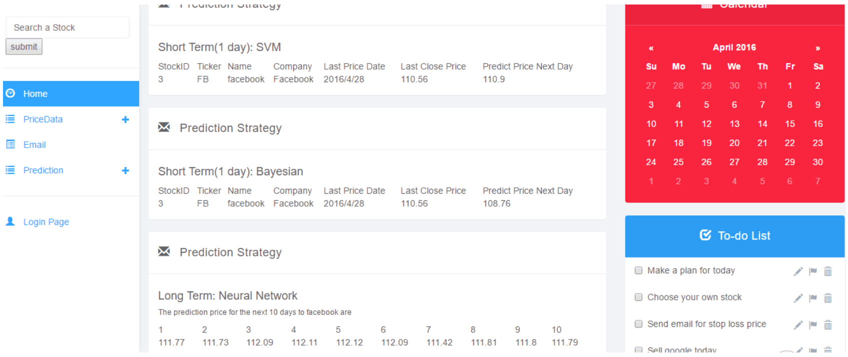
←T→			username	email	stockname	stopprice
□	✎	✗	jianqin	414119772@qq.com	facebook	50.00

显示：30 行，开始行数：0

以 水平 模式显示，并且在 100 行后重复标题

Figure 20: delete price from our database

H.6 prediction strategy



Short Term(1 day): SVM

StockID	Ticker	Name	Company	Last Price Date	Last Close Price	Predict Price Next Day
3	FB	facebook	Facebook	2016/4/28	110.56	110.9

Prediction Strategy

Short Term(1 day): Bayesian

StockID	Ticker	Name	Company	Last Price Date	Last Close Price	Predict Price Next Day
3	FB	facebook	Facebook	2016/4/28	110.56	108.76

Prediction Strategy

Long Term: Neural Network

The prediction price for the next 10 days to facebook are

1	2	3	4	5	6	7	8	9	10
111.77	111.73	112.09	112.11	112.12	112.09	111.42	111.81	111.8	111.79

« April 2016 »

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

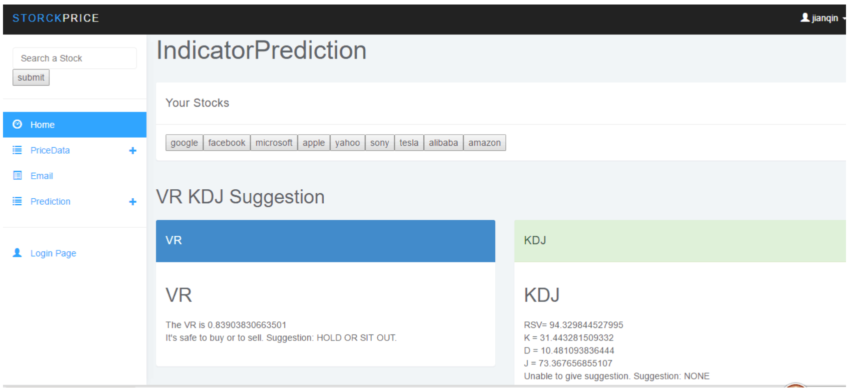
- □ Make a plan for today
- □ Choose your own stock
- □ Send email for stop loss price
- □ Sell google today

Figure 21: prediction strategy

Website will show the user the predicted prices in three different ways. The first and second way is called SVM and Bayesian, respectively, which can predict the stock price for the next stock day.

And the other way is called Artificial neural network, which can predict the stock prices in the next few days. These two ways of prediction can provide the user a better view of how the stock price goes in the next period of time.

H.7 prediction indicators



IndicatorPrediction

Your Stocks

VR KDJ Suggestion

VR

VR

The VR is 0.83903830663501
It's safe to buy or to sell. Suggestion: HOLD OR SIT OUT.

KDJ

KDJ

RSV= 94.329844527995
K = 31.443281509332
D = 10.481093836444
J = 73.367656855107
Unable to give suggestion. Suggestion: NONE

Figure 22: prediction indicators

There will be two indicators, one is called VR, and the other is called KDJ. These two indicators can be really helpful to predict the stock price trend in the future, and based on the existing theoretical statistics, they can give reasonable advices on the operations of stocks.

I Web sources for stock price

We will gather data from Yahoo Finance (http://finance.yahoo.com/) and store the historical price (day by day) for long term prediction. We can also read the real-time price of stocks to do short term prediction. All the price information will store in our databases.

J Acknoledgment

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

References

[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] Pattern Recognition and Machine Learning Bishop 21-32

[6] http://en.wikipedia.org/wiki/Support_vector_machine

[7] http://thompsonshome.blogspot.in/2011/02/kdj-analysis-and-application-of.html