Stock Predictor Application

Software Engineering of Web Applications

Authors:

Marc Gamell	[mg900]
Fernando Geraci	[fgeraci]
Varvara Kalokyri	[vk230]
Parishad Karimi	[pk359]
Ali Rostami	[ar943]

Spring 2015

Contents

1 Introduction

Financial markets have always attracted people, since it is thought to be an easy way to earn money without so much effort. Toward this end, people have adopted different kind of approaches in order to predict future trends of stocks based on the knowledge of the past. However, it is really difficult for non-professional investors to gain profit out of it in the long run, since a lot of time and knowledge is required. Professional investors spend their entire careers studying the markets and since this is their job they have the time and the resources to implement the lengthy analysis that is required in order to make an investment.

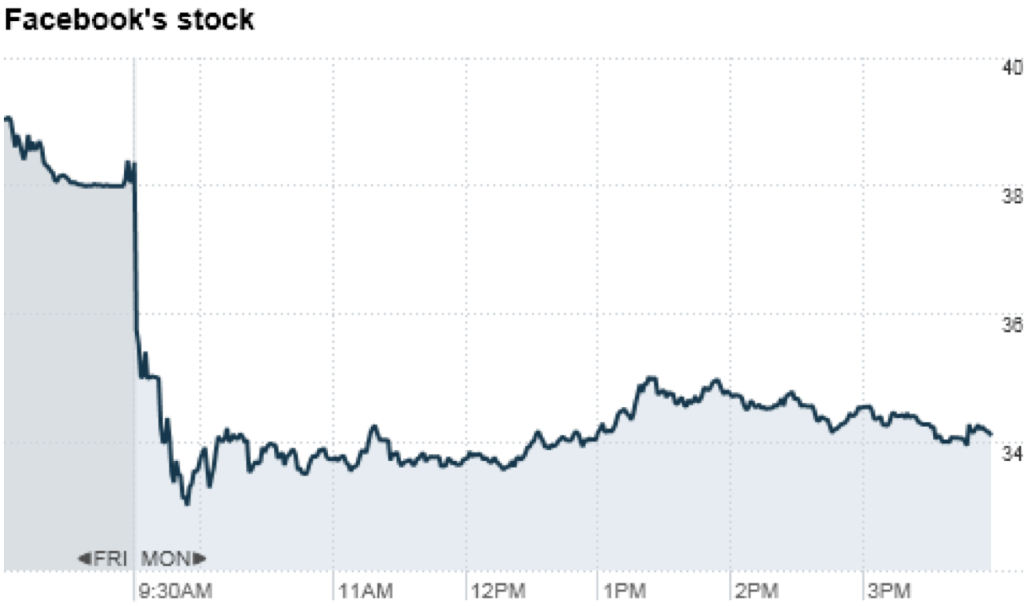
With this realization, our project will attempt to benefit a broad range of investors by providing a web-based application which gives them the opportunity to make their own decision on whether to buy, sell, or hold a stock by providing stock predictions for the stocks of their interest. We will also provide them with the ability to have their own portfolio so as to be able to add their favorite companies as well as check the news for these companies as well.

2 Customer Statement of Requirements

The stock market is one example of a realistic environment in which huge amounts of data are constantly generated. Wikipedia defines the stock market, or equity market, as the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks. Stocks are just shares of a company than are owned by private parties or other companies. Each company has a limited set of stocks, which have a specified price. These prices fluctuate depending on many factors.

Throughout history there have been continuous attempts to try to understand and model the fluctuations in the stock prices of the market.

By definition, trading requires two very basic components: offer and demand. The price of an item is usually determined by the volume of its demand and its availability. A scarce product with high demand will tend to have a higher cost than a more available one with lower demand. This very simple concepts are the core in stock trading. As an example, lets look at Facebook's Initial Public Offering (IPO) which took place on May of 2012. Highly anticipated, the company opted for pricing its share beyond the upper bound of its valuation. Despite of what happened later, it was a success. The company offered around 480 million shares:



http://money.cnn.com/2012/05/21/markets/facebook-stock/

From the above image, we can formulate the following questions:

1. Why was the IPO share value so much higher than the $35/share valued upperbound?
2. What caused the stock price to plummet right after its release to the public?
3. How did people react to such events?

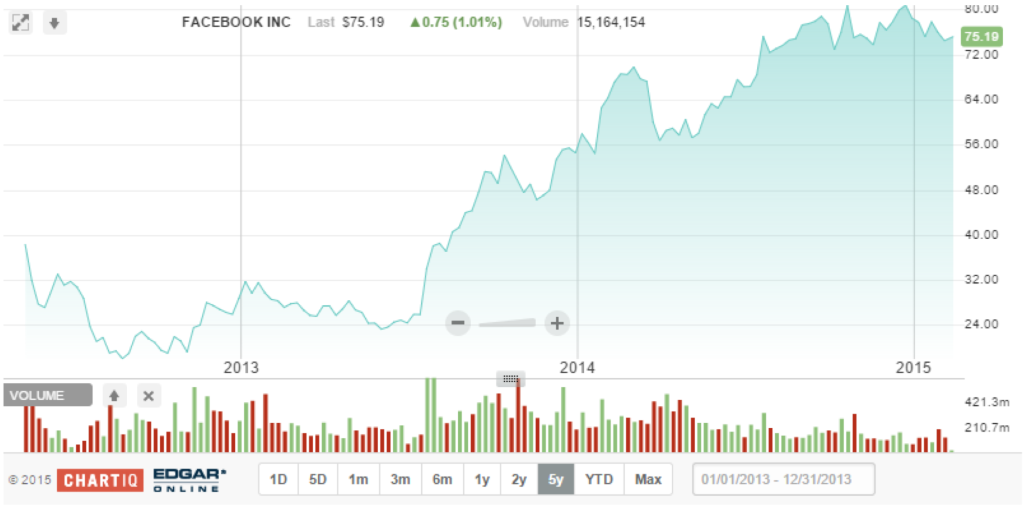
Answering question one simply require from us to go back to the offer/demand concept. Everyone wanted to take a slice of the IPO pie, so the company, taking advantage, valued its IPO share at a higher price knowing all 480 million shares would go to investors hand. Question two plain answer is that the plummeting was caused because investors did not think Facebook was a $100 billion dollar company hence the market regulated itself, which is, despite of expectations there was no offer for a Facebook's IPO share at the actual IPO price.

Question three's answer builds on answers for one and two. After observing the IPO's trend, many people decided to take a loss on their investment and only 30 seconds after the market opened on Friday, approximately 80 million shares (approximately 16% of the total IPO volume) changed hands, hence there was a very generous trading volume, hence, offer. This translated in stabilized share prices. No skyrocketing, nor further plummeting. By the end of the day, Facebook IPO broke a trading volume record of approximately 560 million shares traded on by day one. This volume of offer/demand helped investors to agree on a share price of around $34/share by the close of the trading session.

Going back to this essays hypothesis: is looking at the stock performance on the graph enough to understand what happened on the IPO? Clearly not so much. Experienced traders knew no to get involved in this IPO's initial offer and hold buy orders until the market stabilizes its prize. So for this case, experience won over expectation and many hyped investors ended up selling for a lower price to experienced people who bought at the actual markets one. Now, for the sake of analysis, let's fast-forward until today:

Observe how it wasn't until July of 2013, that Facebook did not start to grow and motivate investment, but again, despite of what the chart is showing here, we must ask why? On July, investors felt very strongly about Facebook's mobile ads platform. This caused, as can be appreciated on lower histogram chart (around July-August 2013) an increase in trading volume, hence all the people who broke-even and wanted out sold their stock to people who believed FB's share would raise due to the company's business model. We can infer one big factor here: how people feel about an investment.

First, we looked how volume creates market's self-regulated trading, and now, we see how investors perception and research creates trends on share



http://www.nasdaq.com/symbol/fb/interactive-chart

prices. Especially on long term investors. People who believe in the model would buy, people who was disappointed by their past investment would sell, volumes raise, people buy, not enough for everyone, share prices go up. Can this be predicted by the previous model? not necessarily, because months before July 2013, the company did poorly in the markets and there were not many news which could stimulate trading. After exposing two major factors on trading, volume of exchange and company's business performance, I can also say that going back to FB day one IPO, it would have not been sufficient to draw conclusion from a highly pronounce downwards triangle or anything alike.

Now that we know that a graph by itself cannot be isolated from concepts such as volume of trading and investors perception, we begin to analyze some additions to the data itself. This soft approaches, are known as FUNDAMENTAL ANALYSIS, which basically, it utilizes business and markets information to understand behaviors and add a degree of certainty to potential future scenarios.

But just like as everything, only fundamental analysis is not enough to understand and try to reduce uncertainty in trading securities, we also need to be familiar with some TECHNICAL ANALYSIS concepts which are commonly used at the time of representing a stock's behavior in the market. This basic concepts will be describe below.

Clearly, people is really creative at the time of analyzing previous data. But going back to the original essay's question: is a graph alone sufficient to predict behavior or at least reduce uncertainty? the answer could be: it depends. The factors which will define its usefulness are:

1. Investment goals
2. Information available
3. Market volatility (can't be controlled by a single force)
4. Trading horizon

These are some among others, and they mean that no simple graph is sufficient if we cannot interpret what is behind it, hence its data's source. Several tools are available from different schools: fundamental and technical analysis. Both are arguably effective and defective. It would be ideal to find a nice blend which could be adapted depending on the context of the trade.

It is always wise to hedge the risk of investment loss by maintaining a diversified investment portfolio. The investor objectives could be classified into short-term-gain and long-term-gain.

In order to make investment decisions and examine the security of investing, two broad methods are used: fundamental analysis and technical analysis. Basically these methods will determine what stock to buy and how much to spend on it based on the future growth trends of stocks. In fundamental analysis the underlying forces that influence the health of economy, industry groups and companies are investigated. The end goal of fundamental analysis is to provide the investor with a value that he/she can compare with security's current price and figure out to under-price or over-price.

On the other hand, technical analysis studies supply and demand in a market to predict the trend (illustrating the direction) on future. Technical analysis is more short term in nature compared to fundamental analysis and the goal of the purchase dictates the method. Technical analysis suits "trades" better and fundamental analysis is used for investments.

3 System Requirements

This section describes the requirements of the system that has to be built.

3.1 Functional Requirements

Identifier	Requirement
REQ1	The system acquires stock prices on a daily basis from Yahoo! Finance and stores the data in a relational database.
REQ2	The system will give the ability to users to search for a stock based either on the stock ticker/symbol or the company name.
REQ3	The user will be able to have his/her own portfolio that will hold his/her own favorite companies.
REQ4	The system will display in a line graph the price trend of a specific stock being search by the user.
REQ5	If the user searches for a stock that doesn't exist in the database, then the system will search real time to find it in the Yahoo! Finance and present it to the user.
REQ6	Each stock will have its own page in the website with all its stock information on the page.
REQ7	The user will be able to register to the system so as to have his own personal space.
REQ8	If the user is logged in then s/he will be able to see the news and tweets for his favorite companies.
REQ9	The system should show the details for a specific stock and also display the predicted price and a suggestion for buying/selling/holding.
REQ10	The stock data stored in the database will be used by the prediction algorithm to calculate the stock's prediction.
REQ11	Understand and implement the prediction model for the closing price of the stock prices.
REQ12	The user will be able to see the current prices for the stock, as well as browse the data based on time (yesterday's data, one month ago data, one year ago data etc.)

Our main requirement is to be able to obtain the historical data for our five initial stocks (REQ-1) from the Yahoo! Finance API. If the user though searches for a stock that doesn't exist then the system will be able to search about it real-time and present it to the user (REQ-5). Generally, the user will be able to search a stock on the website (REQ-2) and the system will return a webpage with all the stock's information, graph of historical data with the prediction as an overlay (REQ-9). The user as well will be able to register (REQ-7) so as to have his personal portfolio (REQ-3) and be able to read the news and tweets about his/her favorite companies that s/he is following (REQ-8).

3.2 Non Functional Requirements

Identifier	Requirement
REQ13	The system shall ensure that all the user credentials will remain secure even if an attack will happen to the db system
REQ14	The system shall be able to run through all prediction models for each and every listed stock within 1 hour.
REQ15	The system shall allow for users to get a prediction for a stock just with one click.
REQ16	The system should insure that any features that do not require a user to be logged in are not hidden to unregistered users.
REQ17	The system should be able to run intact from a smartphone/tablet workspace.

Since the system will be storing user accounts, REQ13 needs to be specified for security purposes. REQ14 ensures that latency is kept to a low level since the system will be a web service where a fraction of a second in loading can make the difference in the user satisfaction. Furthermore, the customer intends for the service to be as easily accessible as possible and thus REQ16 insures visitors to the service are not locked out of key features and REQ15 insures user efforts for the most prominent feature (searching for predictions) is kept to a low level. REQ17 finally insures that the system is able to be used for not only desktop browsers but also smartphones and tablets as well. This requirement is of the least priority because we will be working on the functionality for modern desktop web browsers first and then optimize for mobile if development time allows.

3.3 On Screen Appearance Requirements

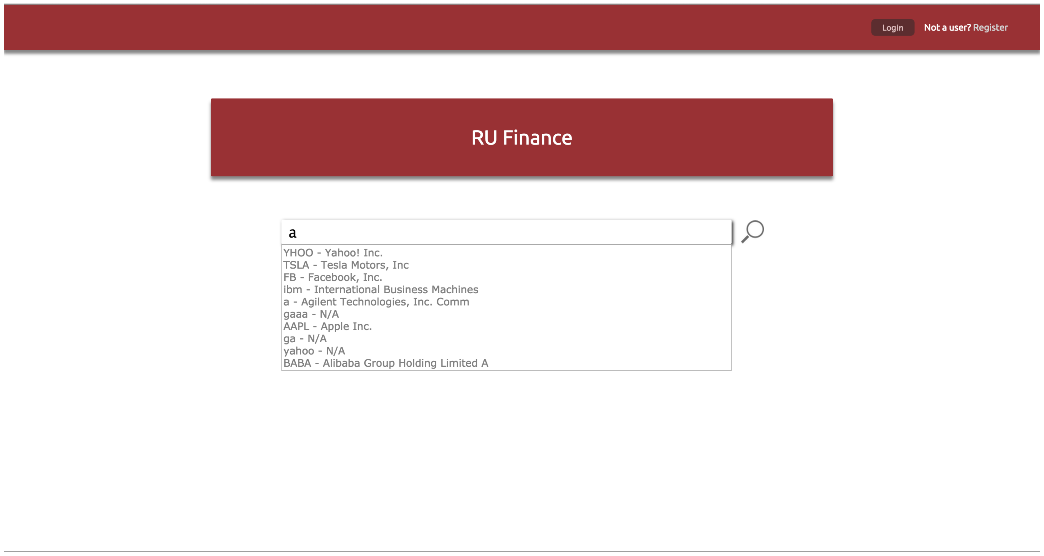
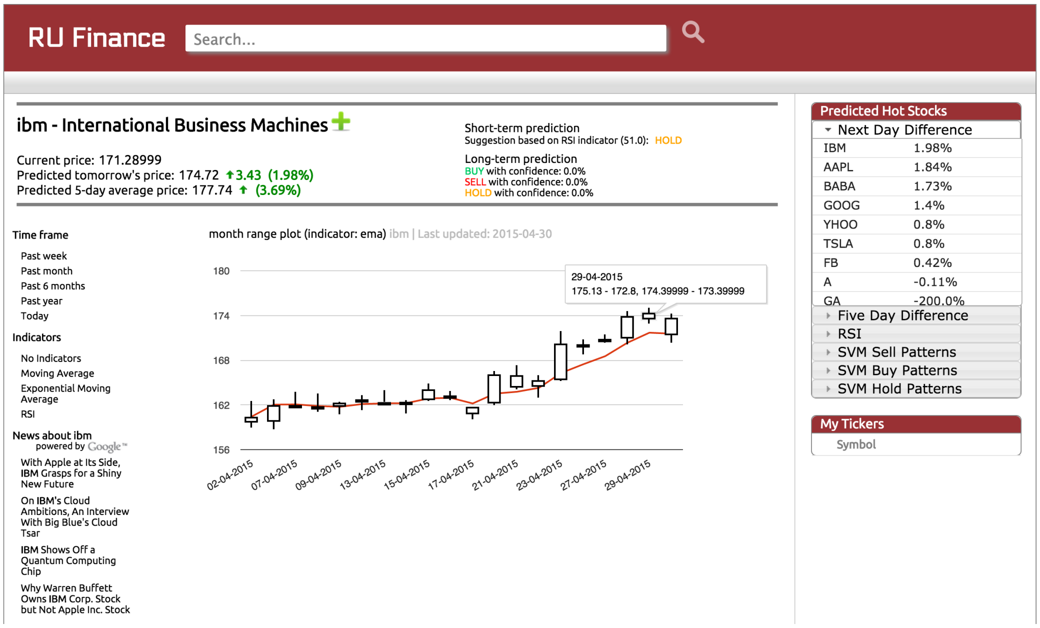


Figure 1: Homepage of our stock prediction webpage

Identifier	Requirement
REQ18	Figure 1 shows the mockup of the first page of the system when a user access it. A search bar is displayed where the user is able to search for a company name or ticker name. This is also where a user is allowed to sign in or register if s/he doesn't have an account.
REQ19	Figure 2 shows the design for the page after the user has searched and selected to see a specific stock. If the user is logged in as well, then s/he will be able to see his favorite companies and see the news concerning these companies. A persistent search bar is displayed at the top while stock data are given, filled with a chart of historical prices, current price, predicted price, confidence value, and predicted decision (buy, sell, hold).



ibm - International Business Machines

Current price: 171.28999
Predicted tomorrow's price: 174.72 ↑3.43 (1.98%)
Predicted 5-day average price: 177.74 ↑ (3.69%)

Short-term prediction
Suggestion based on RSI indicator (51.0): HOLD

Long-term prediction
BUY with confidence: 0.0%
SELL with confidence: 0.0%
HOLD with confidence: 0.0%

month range plot (indicator: ema) ibm | Last updated: 2015-04-30

29-04-2015
175.13 - 172.8, 174.39999 - 173.39999

News about ibm

With Apple at Its Side, IBM Grasps for a Shiny New Future

On IBM's Cloud Ambitions, An Interview With Big Blue's Cloud Tsar

IBM Shows Off a Quantum Computing Chip

Why Warren Buffett Owns IBM Corp. Stock but Not Apple Inc. Stock

IBM	1.98%
AAPL	1.84%
BABA	1.73%
GOOG	1.4%
YHOO	0.8%
TSLA	0.8%
FB	0.42%
A	-0.11%
GA	-200.0%

Figure 2: Results page showing the organization of data

4 Functional Requirements Specification

4.1 Stakeholders

Two different type of stakeholders can be identified:

1. Users: both registered and unregistered users of the system.
2. Administrator: maintains and updates the web system.

4.2 Actors and Goals

Seven actors can be identified:

1. Registered User: a user that has an account in the sytem.

2. Visitor: any unregistered user
3. User: both a registered user and visitor
4. Database: records of stock information (i.e historical prices, prediction results, confidence value, etc), user data (i.e. username, password, tracked stocks, email, etc), and system data (timers, search logs, prediction time logs).
5. Price Provider (i.e. Yahoo! Finance): Provides the current pricing of a stock of interest
6. Grapher: Provide visual charts from raw data.
7. Administrator: a special case user that maintains and updates the web system.

4.3 Use Cases

4.3.1 Casual Description

UC-1: Register - Allow a visitor to fill out the a registration form and become a registered user.

UC-2: Login - Allow user to access his/her account and view information for their favorite stocks.

UC-3: Logout - Allow user to logout from the system.

UC-4: Search - Allow a user or visitor to search for a particular stock through keywords

UC-5: Obtain Prediction - Obtains the input parameters for the prediction algorithm to calculate the prediction. Once the prediction is calculated, a graph will then be generated with both the stock's historical data and prediction.

UC-6: DataAcquisition - In order to obtain a prediction and generate a graph, we will need to retrieve stock and market data from the Yahoo! Finance API and store it into the database. This is done in the background. If the stock is not already in the database, an algorithm will query Yahoo! Finance for the requested data.

UC-7: CalculatePrediction - From the input parameters, the prediction algorithm will calculate the stock's prediction.

UC-8: Add ticker - Allow users to add a certain stock to their "portfolio".

UC-9: Remove ticker - Allow users to remove a certain stock from their portfolio

UC-10: News - Users will have access to current events and news regarding their favorite companies.

4.3.2 Use case Diagram

The use cases provided are illustrated in 3. The use cases are as follows:

- Register - Allow a visitor to fill out the a registration form and become a registered user.
- Login - Allow user to access his/her account and view information for their favorite stocks.
- Logout - Allow user to logout from the system.
- Search - Allow a user or visitor to search for a particular stock through keywords
- Obtain Prediction - Obtains the input parameters for the prediction algorithm to calculate the prediction. Once the prediction is calculated, a graph will then be generated with both the stock's historical data and prediction.
- Data Acquisition - to retrieve stock and market data from the Yahoo! Finance API and store it into the database. algorithm will query Yahoo! Finance for the requested data.

- Calculate Prediction - From the input parameters, the prediction algorithm will calculate the stock's prediction and detected patterns.
- Add ticker - Allow users to add a certain stock to their "portfolio".
- Remove ticker - Allow users to remove a certain stock from their portfolio
- News - Users will have access to current events and news regarding their favorite companies.

5 Prediction

One of the most important parts of the project is the prediction part. The prediction, that it can be considered as the heart of the application, has been made of three separable modules; Short term prediction, long term prediction, and some helpful information that help users to decide which call indicators.

The prediction module has been implemented as an event based module. It means that whenever user makes a query, the prediction module gets the crawled stock data from the MySQL database, and sends the processed information to the web service to update the user's browser.

5.1 Short Term Prediction

Predicting the short-term fluctuations of stock price is a very difficult problem. Historically people have tried to predict stock prices using complicated mathematical models on the real world stock data. In the short run, investors need to decide whether market speculations are optimistic or pessimistic. Short term trading can be useful, but it is also risky to rely on that.

Short-term trading uses many methods and tools to make money; however, users must know how to apply provided tools to achieve success when they are using this strategy. Keeping this in mind, users will be able to make money in both bull and bear markets while keeping their losses at a minimum.

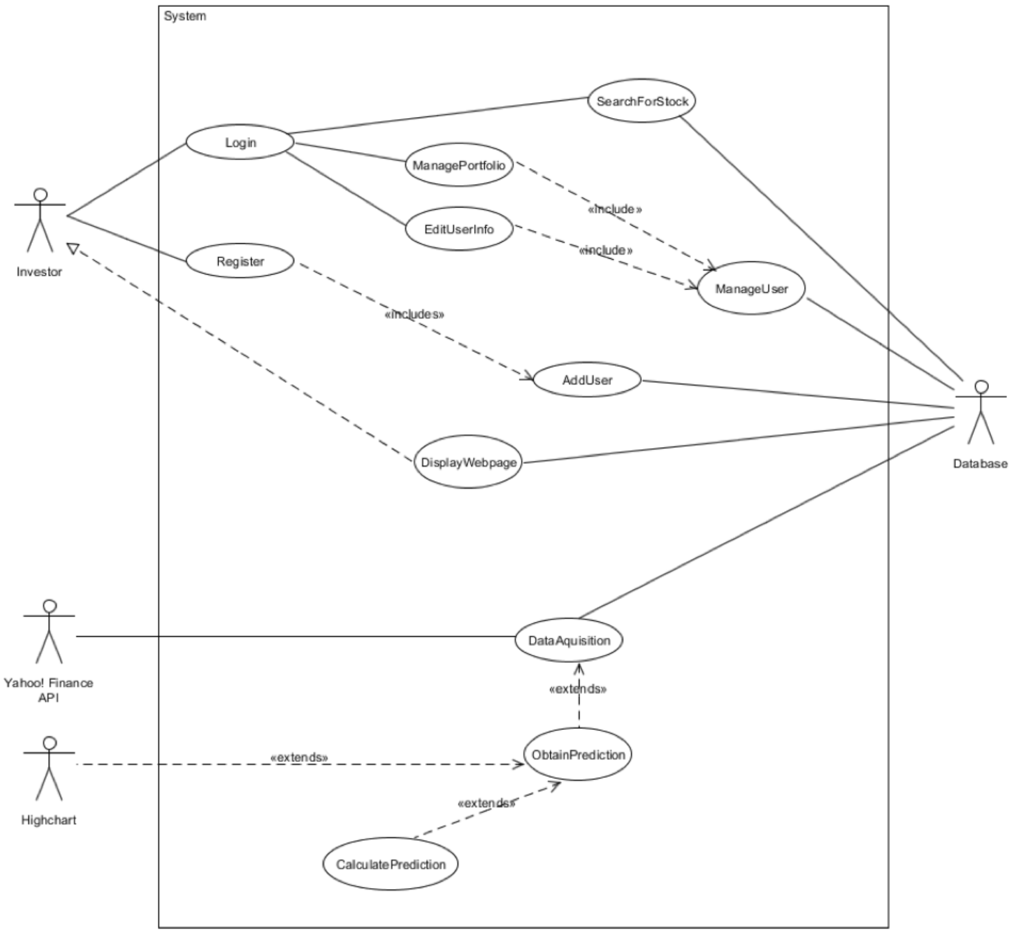


Figure 3: Use case diagram

Even though the stock prediction models provide a good forecast of the short-term behavior, abnormal stock fluctuations increase the uncertainty, therefore the whole investment is in the risk. Short-term trading involves risk, so it is essential to minimize it. Wavelet modulus maxima method is proposed in order to detect the abnormal values of the stock market. The abnormal points detected by wavelet modulus maxima are replaced by the new interpolation points, which will be used as an important index of Kalman Filter algorithm to predict the stock price.

The experimental results show that the proposed method can predict the stock data with higher credibility than Bayesian algorithm. Therefore, the proposed method can reduce the investment risk and plays an important role in the economic development and financial building. Long story short, we want a more accurate prediction, then we should find the abnormal signals (which are the stock prices in this article) and replace them by some less damaging values.

According to modern financial engineering principles, market data is supposed to be a Brownian motion, which means that the daily price changes with a white-noise process. A white noise is a random process in which consecutive values are independent of each other, meaning that a price increase is just as likely as a decrease each day. However, in reality, it is not uncommon for a particular market item to have several consecutive down days, or up days, over a short time span. During such spans the prices are considered to be correlated. The objective is to harness these correlations with a Kalman filter for prediction.

The Kalman filter is a recursive algorithm invented in the 1960's to track a moving target from noisy measurements of its position, and predict its future position. The proposed method consists of the following three steps:

- Detecting the abnormal values using Wavelet Transform
- Replacing the detected values by Interpolation method
- Predict the data by Kalman filter

5.1.1 Discrete Wavelet Transform

As we mentioned, Anomaly detection refers to finding data models that do not conform to expected behavior in the data. They are called outliers in some other contexts. Wavelet transform method is an extension to spectral analysis methods. It can be used for multi-scale analysis in stock market fluctuations, expresses the local features of the signals, and has the self-adaptability and zoom features. It is suitable for non-stationary time series analysis and can estimate the long memory parameters. In this project, the method of discrete wavelet transform is proposed for anomaly detection. The input values are fine-tuned to improve the prediction accuracy based on results of anomaly detection.

Wavelet Transform is acquired through the irregular sampling in the multi-scale wavelet transforms. It can depict the mutation point of signals. The detected point is corresponding to a large change in the original sequence of signal (here the signal is the price sequence). This method adapts to the abnormal detection of stock research just because of the characteristics of the stock signal singularity.

If $W(f(t))$ is the wavelet transform of the function $f(t)$, then if $W'(f(t))$ has one zero at any point of the sequence s, and it satisfies Eq. (1) for all the neighbors of that point, the corresponding point is a local maximum, which we consider it as an outlier.

$$W'(f(t)) \leq W(f(t0)) \tag{1}$$

5.1.2 Interpolation

In Numerical analysis, interpolation is a method of constructing new points within the range of a discrete set of known data points. There are three main methods to do the interpolation:

- Linear interpolation
- Polynomial interpolation

- Spline Interpolation

In this project, we will use the linear interpolation which is using:

$$f(t_0) = \frac{f(t_0 - 1) + f(t_0 + 1)}{2} \tag{2}$$

This will make the graph look smooth and improves the prediction efficiency of the Kalman filter in an uncertain stock market scenario.

5.1.3 Kalman Filter

The Kalman Filter was originally designed to represents information about a moving objects. Moving objects are modeled using a discrete time dynamic system. Here we consider the object to be a tuple of (x, t) where x is the price and t is the sampling time.

Data is supposed to be noisy. Another representation of the prices are x_t. Some states might be missing and other are biased. However it cannot be observed directly, because it is encumbered by hidden (Gaussian) noise w. The system produces visible output vector y that is a simple linear observation (x evolves first-order Markov process), encumbered by noise v. Figure 4 shows the system model.

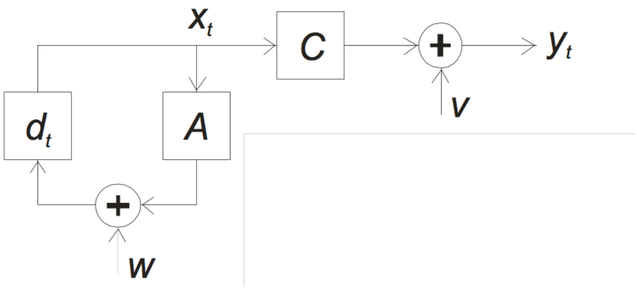


Figure 4: Kalman Filter model

Kalman Filter system dynamic model is determined by Eq. (3).

$$x_t = Ax_{t-1} + Bu_{t-1} + w \tag{3}$$

Where w is as following:

$$w \approx N(0, Q) \tag{4}$$

Likewise, Kalman Filter measurement model is determined by Eq. (5).

$$y_t = Cx_t + v \tag{5}$$

Where v is as following:

$$v \approx N(0, R) \tag{6}$$

In the Figure 4, dt is a delay, A is a state transition matrix; B the optional control matrix with input u, C is the observation or measurement matrix. N is a normal distribution of mean 0, the (covariance) matrices of w and v are Q and R, respectively. Based on Bayesian probability $P(x_t|y_t) = N(\hat{x}_t, P_t)$, as described below. An efficient computational (recursive) tool to estimate the state of a process, in a way that minimizes the mean of error, is the Kalman filter.

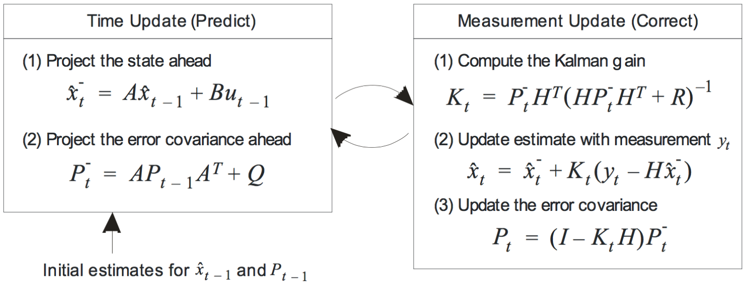

Time Update (Predict)
(1) Project the state ahead
$\hat{x}_t^- = A\hat{x}_{t-1} + Bu_{t-1}$
(2) Project the error covariance ahead
$P_t^- = AP_{t-1}A^T + Q$
Initial estimates for $\hat{x}_{t-1}$ and P_{t-1}
Measurement Update (Correct)
(1) Compute the Kalman g ain
$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1}$
(2) Update estimate with measurement y_t
$\hat{x}_t = \hat{x}_t^- + K_t(y_t - H\hat{x}_t^-)$
(3) Update the error covariance
$P_t = (I - K_t H)P_t^-$

Figure 5: Operation of Kalman Filter

The Kalman filter iteratively applies two stages of computations using feedback control (see Figure 5):

1. Time update computations (prediction). Here $\hat{x}'_t$ is the prior state estimate (predicted) at time t, $\hat{x}_t$ is the posterior (corrected) state estimation at the time ($\hat{x}_t = E[x_k]$), P'_{t-1} is the prior estimate error covariance matrix and P_{t-1} is the posterior error covariance matrix ($P_t = E[(x_t - \hat{x}_t)(x_t - \hat{x}_t)T]$).
2. Measurement update computations (correction). There H is a measurement matrix and K_t is a Kalman gain matrix. The weighting by K minimizes the posterior error covariance, while R approaches zero, the actual measurement y_t is trusted more to the detriment of measurement prediction $H\hat{x}'_t$.

We implemented the Kalman filter in Java based on the Discrete Kalman filter algorithm using the standard JAMA package. Here the stock prices are assumed to be the accelerating target by generating random acceleration increments and updating the velocity and displacement accordingly. We also used the implementation of Bayesian prediction which we had for one of our homework as an comparison to the results given by Kalman filter. The results are very promising, since Kalman shows a closer prediction. Figure 6 depicts a 10 days prediction by Kalman Filter algorithm and Bayesian prediction, as well as the actual values.

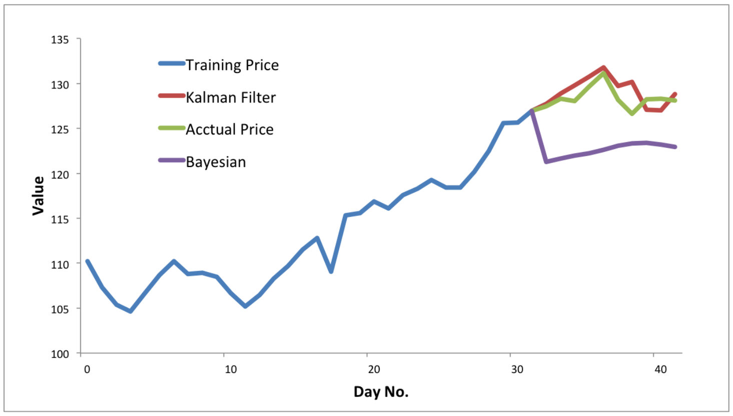


Figure 6: Predicted values via Kalman filter algorithm and Bayesian prediction algorithm

Considering the prediction results from Figure 6, comparison between the

Kalman Filter algorithm and Bayesian is presenter in Figure 7. It is clearly seen that Kalman predicted values are closer to the actual values.

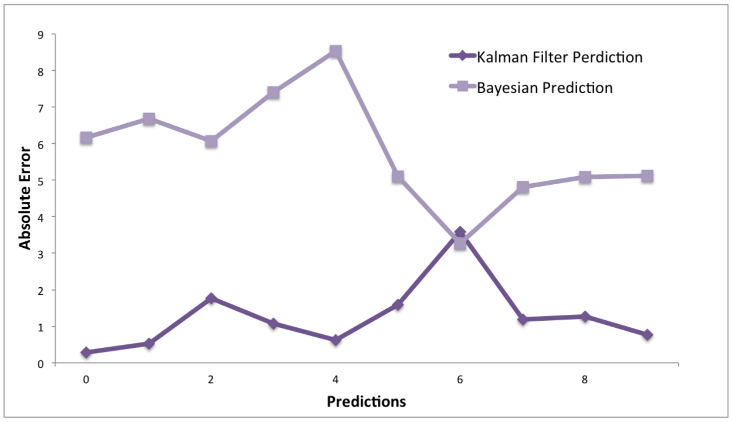


Figure 7: Predicted values via Kalman filter algorithm and Bayesian prediction algorithm

5.2 SVM

SVM (Support Vector Machine) is supervised machine learning algorithm which conducts classification of data into hyper-planes by maximizing the marginal distances between the training data-set points. This method will optimally separate data into multiple categories. The margin is calculated by creating two sets of parallel hyper-planes. The accuracy and performance of the algorithm depend on the distance the hyper-planes will impose on the neighboring data points. As this margin increases, the classifier's error will decrease.

SVM has a couple of features which makes it a reasonable candidate for stock prediction. SVM can be utilized to analyze data and recognize patterns which can be used in classification. This is done by training the SVM to construct a model based on training data, which will later be used to classify the test data.

Traditionally classification problems attempt to address multi-class situations. For example, for a two-class SVM the training data set will be

P=$\{(x_1,y_1)\ldots\ldots (x_N,y_N)\}$, in which y_i can either be -1 or 1, depending on the class xi belongs to, and xi is the value for a specific feature. SVM is capable of creating a non-linear decision boundary, by projecting the data using a non-linear function to a dimension space higher than the original one. A hyper-plane in the higher dimension will specifically be helpful in case of data for which a straight separating line does not exist.

Unlike the more traditional classifications, one-class SVM attempts to classify data by separating the data points from the origin in the feature space. As a result the function returns 1, for the small subset of space, which corresponds to the training data and -1 for all the other points of the space. In this project, we have examined both two-class and one-class SVM classifiers. This will be discussed in detail in the Implementation section.

5.2.1 JAVA Implementation

In order to implement SVM, JAVA library "libsvm" has been used. The reason to choose "libsvm" over other libraries, like MATLAB built-in SVM functions is its better performance in optimization. Any SVM classification consists of two phases, training phase and testing phase. Before explaining the implementation of each of these phases, it should be taken into account that the classification accuracy will be highly impacted, if some data pre-processing methods are not conducted. These methods along with the training and testing phases are described as follows:

5.2.1.1 Data Preprocessing

In order to process the data to meet the requirements of SVM training and testing two step are taken. The first step is feature selection, and the second step is scaling. In the case of stock prediction, the feature set is the daily price of a stock market over a range specified by the user. Since the number of features in the train data and test data should be the same, and having a very large number of features might deteriorate the performance of the SVM, we chose the number of features to be 21. This means that the range of the test data user is interested in, will be downsampled to the number of features.

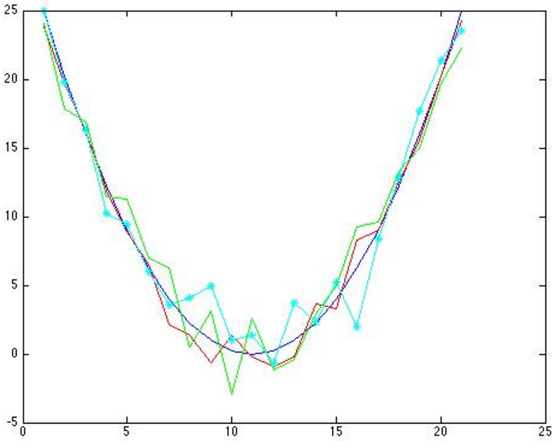


Figure 8: rounding bottom model generated data, along with 3 instances with additive random noise

The second step is to scale the test data with respect to train data. Scaling is advantageous because it avoids the domination of larger values overs smaller ones in classification.

5.2.1.2 Training

In the training phase the SVM trainer requires features of different models of technical patterns discussed above (Cup and Handle, Ascending Triangle, Descending Triangle, Head and Shoulder, Double Tops, Double Bottoms and Round Bottom). The training data is artificially generated. It is fed to the SVM trainer, along with instances of training data with random additive noise. The random additive noise has been chosen in a way that it will not entirely destroy the pattern, but will reflect the instantaneous nature of stock data. In the case of one-class SVM no more training data is required. However, for the case of two-class SVM, we need another class of data with label -1. This data is also artificially generated and fed to the SVM trainer. An example of training data, for the round bottom model is shown in Fig.8.

The linear kernel has been used to train the model and the parameters chosen have been checked both with the built-in method called *svm_ check_ parameter*

Table 1: Cross Validation Accuracy Results, One-class SVM vs. Two-class SVM

SVM	CupHandle	Descending Triangle	HeadShoulder	AsendingTriangle
One-Class	87%	69%	75%	80%
Two-Class	100%	100%	100%	100%

and 3-fold cross validation. Cross validation is a technique use to improve the correctness of the training model. In a 3-fold cross validation, the data is separated in to 3 parts. Each part is tested by training the model with the other two parts, and the percentage of correctly classified data will demonstrate the accuracy of the model. The result of cross-validation to compare one-class SVM with two-class SVM is shown in Table.1. As a result of this comparison, in the rest of the project the two-class SVM is used.

5.2.1.3 Testing

In this phase, the user will select a specific company with the range of days she is interested in. This data will be queried from the SQL database, and after feature extraction it will be tested against different models (like cup and handle, etc) which have already been trained. A probabilistic result will be shown to the user illustrating which model the data is closest to. The sequential flow diagram of this phase is shown in Fig. 9

5.2.2 SVM in practice

In order to test our SVM classifier, we tested it with one year of stock closing price data from several companies. The results for Google are shown in Fig.10 and Fig.11.

The classifier detected head and shoulder pattern for one year stock price of Google. This result is shown in Fig.11

Another example is 150 day YAHOO stock market, which is detected to

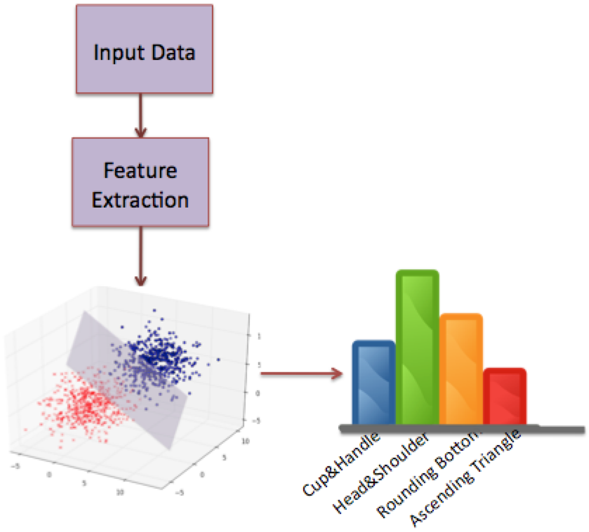


Figure 9: Sequential Flow Diagram of SVM Test Phase

have cup and handle model. It is clear from Fig.12that this model exists.

5.3 Trends and Indicators

Different kinds of activities can be detected and analyzed using the technical indicators. These indicators will provide some estimation of the price behavior. This will be beneficial for predicting the future trends,patterns and ultimately prices.

5.3.1 Moving Average

Moving average is an important indicator in technical analysis to study the stock market by filtering out the noise and smoothing the data. It will enable the user with a mathematical analysis of security's average for a specific period. This will smooth the fluctuations in stock price to provide the user with trends. Among the different types of moving averages, we focused on simple moving average and the exponential moving average. These can be

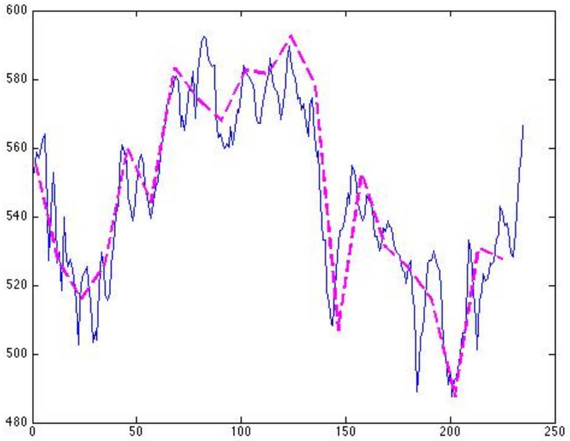


Figure 10: Data Preprocessing of Google's one-year stock data. The blue plot shows daily data, while the purple plot shows the downsampled data, which is the input to SVM classifier.

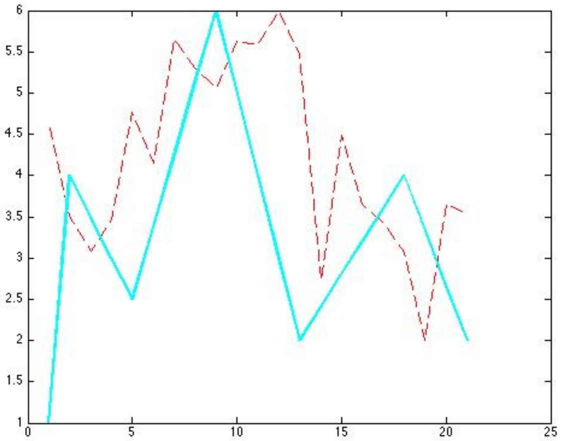


Figure 11: Long Term prediction for Google. The red plot shows the downsampled and scaled test data, The blue plot shows the head and shoulder pattern detected by the SVM classifier.

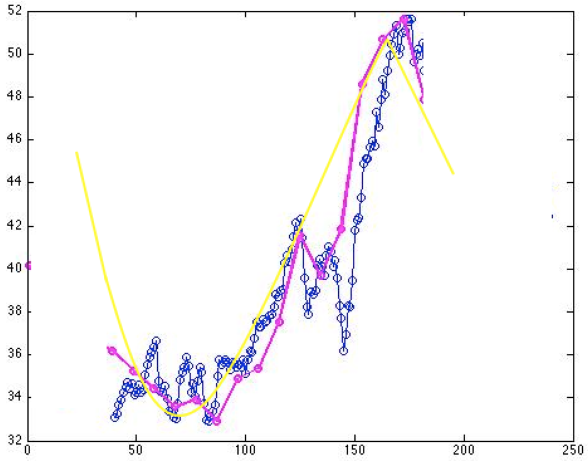


Figure 12: Long Term prediction for YAHOO. The blue plot represents the daily data, the purple plot shows the downsampled data and the yellow plot shows a typical cup and handle figure.

used to determine the direction of the trend and identify potential support and resistance levels. The difference between different types of moving average is the weight assigned to the recent data. Simple moving average assigns equal weight to all the prices within the range specified, while the exponential moving average puts more weight on the most recent data, while not discarding the older data. Since all the moving average methods take old data into account, they are referred to as trend-following or lagging indicators. The important factor in finding a moving average finding one which will be profitable consistently. 39-week (200-day) moving average is popular for this characteristic. The length of a moving average should fit the market cycle one wishes to track.

5.3.1.1 Simple Moving Average

In order to calculate the simple moving average, a range (N) has to be predefined. Then for each day the SMA will be equal to the average of N previous days. The formula below shows the moving average for each day:

Table 2: Algorithm Steps to Find the Exponential Moving Average

Step	Method
1	Calculate the moving average for the first N days
2	Set the first term for exponential moving average equal to the result from step 1.
3	The rest of terms will be found using this equation: $\text{EMA}(i) = \text{Current Price} \times \alpha + \text{EMA}(i-1) \times (1-\alpha)$

$$\frac{\sum_{i=1}^{N} ClosingPrice}{N} \tag{7}$$

Moving average is based on closing prices. As the name suggests it functions as an averaging method that moves through the time scale. Although this method is simple, it is more comprehensible and gives good insight into tracking the prices. The range can be defined based on short, intermediate or long term prediction.

5.3.1.2 Exponential Moving Average

Exponential moving average has less lag compared to simple moving average, because of the more weight it puts on the more recent data. The steps to calculate the exponential moving average for each day is presented in Table.2.

For example a 9% exponential moving average will be calculated by multiplying the current closing price with 9% and then sum it with previous day's EMA multiplied by 91%.

5.3.1.3 Moving Average in Practice

In order to interpret the moving average, we have to compare the moving average with the actual data chart. When the price rises above moving average, it demonstrates the buy signal and when the price falls below the moving average it demonstrates a sell signal.The SMA and EMA for 40-day and 100-day Facebook stock are illustrated in Fig.13 and 15. As can be seen from these figures, the EMA is more sensitive to stock price fluctuations than SMA, and this is more clear by considering the 100-day plot. The buy and sell signal has also been shown in this plot.

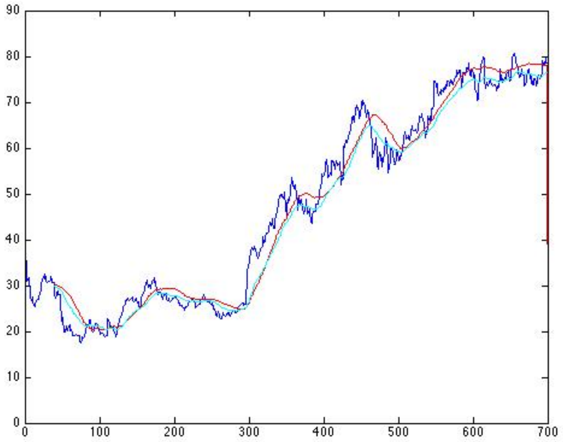


Figure 13: The 40-day Moving Average for Facebook, the blue plot shows the actual daily price, the red plot shows the SMA and the light blue plot shows the EMA.

An example of detecting the support and resistance lines is shown in Fig. **??**. As can be inferred from the figure, it shows that for 40-day Google stock the support line gets broken and it becomes the resistance line around day 160.

5.3.2 Relative Strength Index

The Relative Strength Index (RSI) is a well-versed momentum based oscillator, which is used to measure the speed directional price movement as well as its magnitude change. Essentially RSI graph provides a visual mean to monitor both the current, as well as historical, strength and weakness of a particular market. The strength or weakness is based on closing prices over the duration of a specified trading period creating a reliable metric of

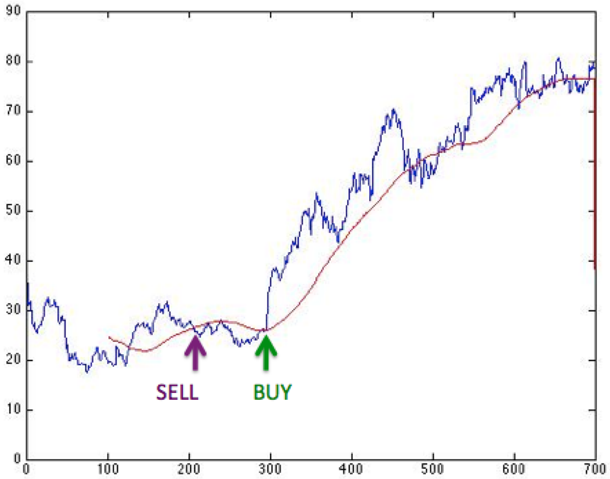


Figure 14: The 100-day Moving Average for Facebook, the blue plot shows the actual daily price, the red plot shows the SMA and the light blue plot shows the EMA.

price and momentum changes. Given the popularity of stock indexes and leveraged financial products, RSI has proven to be a viable indicator of price movements.

To calculate the RSI, we need to have the overall gain over the period that we want to calculate the RSI for. We do this by comparing each two consecutive prices; if it's rose that's the partial gain for those days. If it's fallen, the gain would be zero and the loss would be equal to the amount of difference. After doing the calculation of the overall gain and loss, RSI is given by:

$$RSI = 100 - 100 \times \frac{1}{1 + \frac{A}{B}} \tag{8}$$

Where A is the average gain, and B is the average loss.

How it works? The answer is that we use this indicator to find out if the stock has been oversold or overbought. It is like when the prices rose very rapidly and therefore momentum is high (usually over 70-80), that the underlying financial commodity would have to eventually be considered overbought

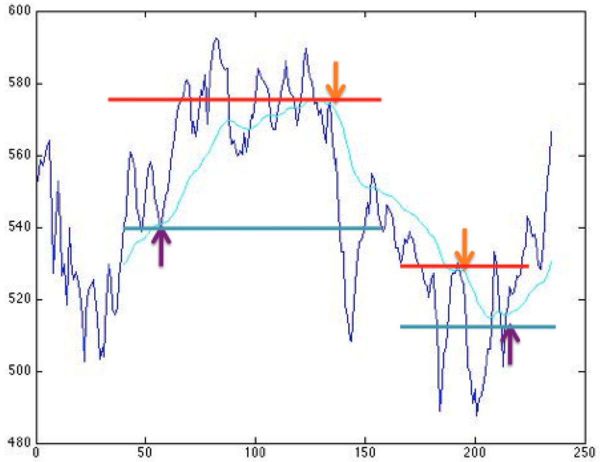


Figure 15: The 40-day Moving Average for Google, the blue plot shows the actual daily price, the red plot shows the SMA, the red and blue horizontal lines show resistance and support lines

and a selling opportunity was possibly at hand. Likewise, when prices drop rapidly and therefore momentum is low (usually below 20-30), the financial commodity would at some point be considered oversold, presenting a possible buying opportunity.

In our project we have a class named RSI that gives a single double variable for an array of prices. Standard implementations use one of the 9, 14, or 25 samples to calculate the RSI for. Figure 16 shows the RSI sequence for apple inc stock price. Each RSI value have been calculated for its previous 25 prices.

6 System Architecture

This section describes the complete system's architecture, identifies its basic components and provides an in depth analysis of the internal functionality. Built as a web application, the system adopts the Spring framework which provides a comprehensive programming and configuration model for modern

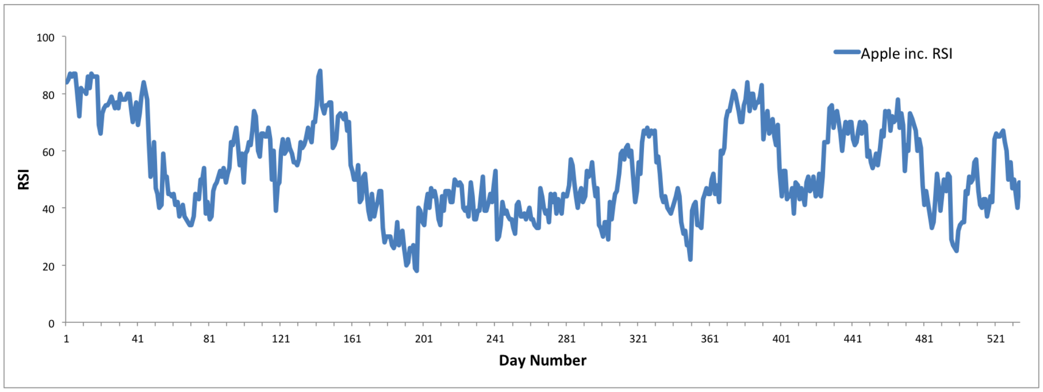


Figure 16: RSI plot for apple inc. stock prices over periods of 25 days

Java-based enterprise applications. Figure 17 displays the overall system architecture.

6.1 MVC Design Pattern

Most screens in a Web application contain controls that allow the users to review application domain data. A user can modify the data and submit the changes. The client logic retrieves the domain data from the server, handles user events, alters other controls on the page in response to the events, and submits the changed domain data back to the server. Including the logic behind these functions in the Web page makes the code complex, difficult to maintain, and hard to test. In addition, it is difficult to share code between Web pages that require the same behavior.

This pushes the need for an architectural design that: Maximizes the code that can be tested with automation. (Web pages containing HTML elements are hard to test.) Code sharing between pages that require the same behavior. Separation of Business logic from User Interface logic to make the code easier to understand and maintain. These requirements drove the creation of the Model-View-Controller (MVC) Design Pattern. MVC introduces the separation of the responsibilities for the visual display and the event handling behavior into different entities named, respectively, the view and the controller.

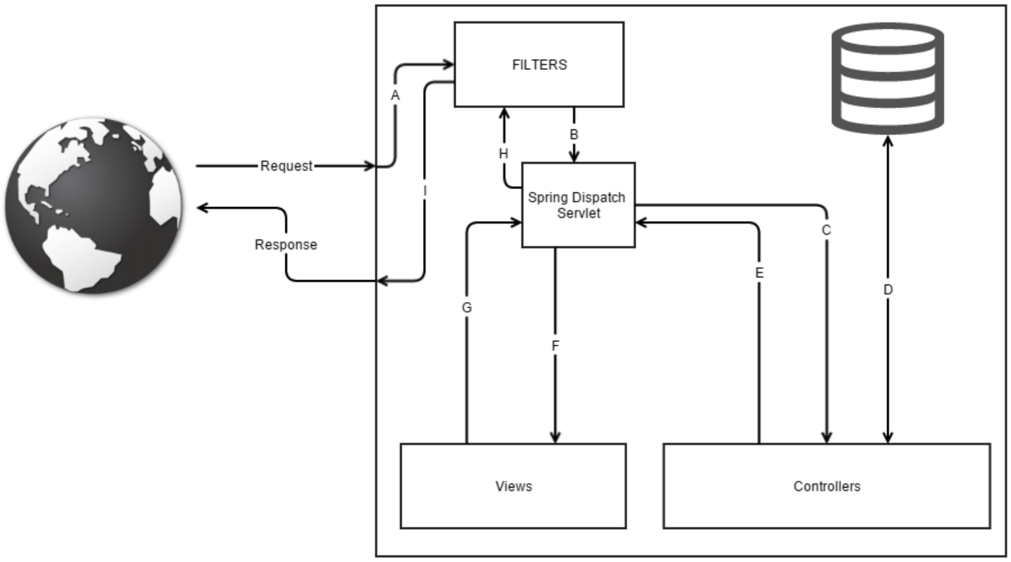


Figure 17: The MVC Spring Model

Figure 17 presents the diagram of this pattern which is composed of the View, the Controller and the Model. The View in this pattern is the component that is responsible for the presentation and interaction with the user. Each action of the user produces an event which is handled by the Controller, who converts the event into an appropriate user action understandable for the model. The Controller notifies the Model of this action, possibly resulting in a change in the Model's state. The Model holds actually all the business objects used by the system and it is in charge of all database related operations.

6.2 Implementation

As this is a regular 3 tier application we have the following three components:

- **Application**: Stockpred App
- **Web Container**: Tomcat
- **Database**: MySQL

The application is mainly programmed in java, and packaged as a WAR (Web Application Archive) file, which is handled by Tomcat. This web container is mainly prepared for dealing with java and all its packaging flavors so Tomcat-Java is a natural combination.

The application is tied up on the other end to its model by an API connector which in our case is the MySQL Connector/J. This provides full access to data operations on the database.

6.3 Operation

Figure 17, shows how a request is handled by the whole application. Firstly a user sends a HTTP GET/POST request to the application. This will be handled by Tomcat, which will resolve the URL and map it to the application's entry point. Once inside the application, a filter will log the request and handle authentication of the session. Then, the filter will forward this request to a central handler called a *dispatch servlet.* This dispatcher will first call a central controller which will execute the business logic for then, based on the data returned by the controller will resolve a view, render it, stuff it into the response's body and send it back to Tomcat. Finally, Tomcat will respond to the original request with the processed packet.

It is important to highlight that the centralized Controller will not exclusively handle all the application's logic but just all the data which corresponds to request and responses. Other activities such as updating and recording stock data will be handled by other independent synchronized entities within the application which have no direct participation in request/responses handling. Specifically, the stock updating is performed by a timered procedure that is invoked every certain amount of time as a separate thread, as described in Section 6.3.5.

More specifically the implementation of the system components are as described in the following subsections.

6.3.1 Model

Till now, the Model of our application consists of the following Java classes:

- **User:** Contains all the information pertaining to a user such as: username, password, firstName, lastName etc.
- **Ticker:** Contains all the information about a ticker which is the ticker name and the ticker symbol.
- **InstData:** Contains all the information concerning an instantaneous stock, such as the stock price, volume, date and time.
- **HistData:** Contains all the information concerning historical stocks, such as open price, close price, volume and date.

6.3.2 View

For this phase of the project it is not required to have a user interface. However, we have implemented the first page of the application that the user visits that contains information about logging in and registering to the application.

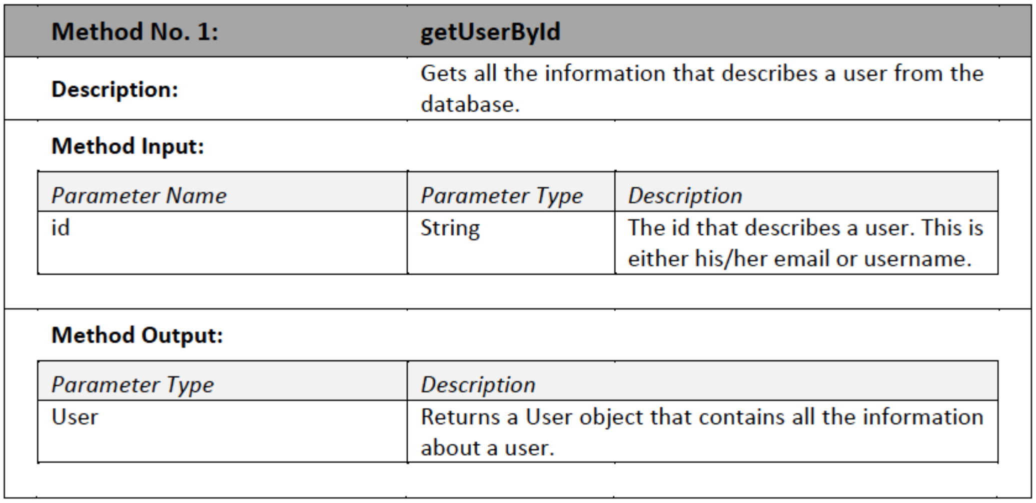
6.3.3 Controller

The controller for our system is implemented in the MainController class. Its main function is to redirect traffic of requests to proper handlers and manage the model for setting up views.

6.3.4 Database Interface

The communication of the system with the database is done via database interface classes. These interface classes are responsible for mapping the business logic of our application to the database. Their functionality is to communicate solely with the database by implementing methods that are responsible for the storage and the fetching of the data. The advantage of these classes is that they separate fully the business logic of the application from the persistent logic of the application. That means that in case in the future the database changes, the only thing that would change is the implementation of the these interfaces classes and nothing more. The rest of the logic of the application remains exactly the same.

The implemented methods are listed in the tables below, providing brief descriptions regarding each method's functionality.

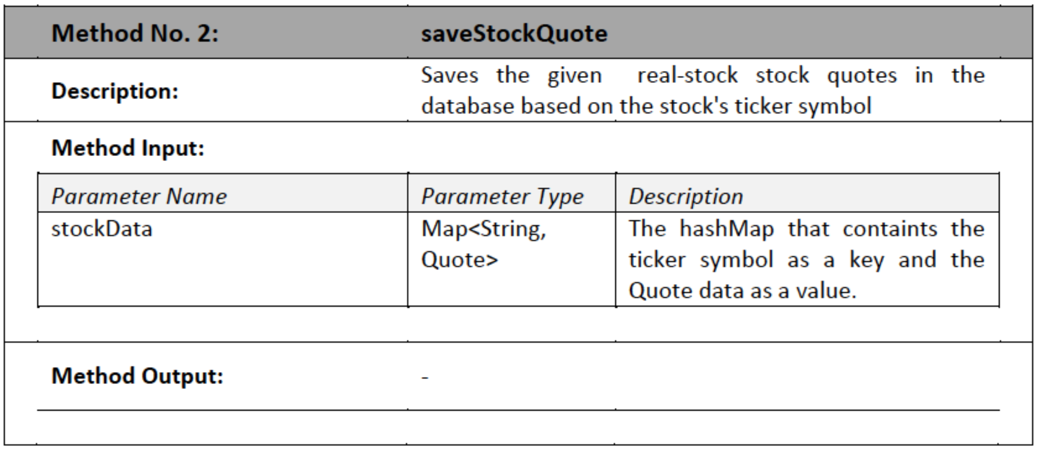


Method No. 1:	**getUserById**
Description:	Gets all the information that describes a user from the database.
Method Input:	

Parameter Name	*Parameter Type*	*Description*
id	String	The id that describes a user. This is either his/her email or username.

Method Output:

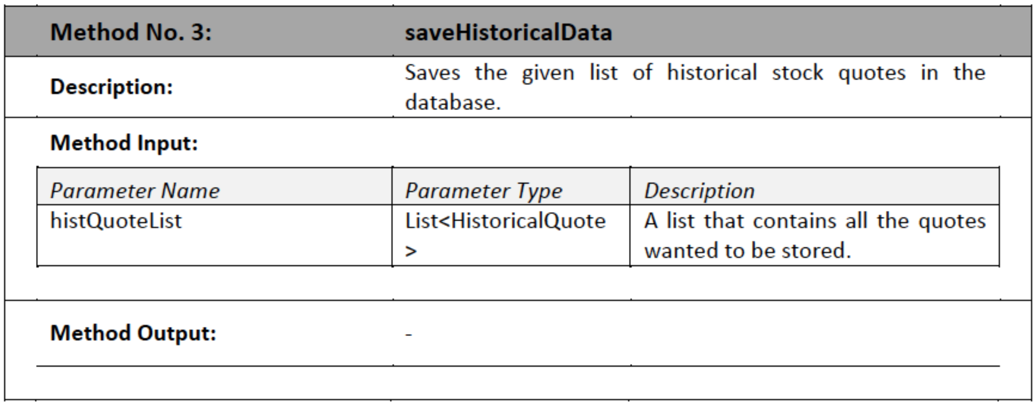
Parameter Type	*Description*
User	Returns a User object that contains all the information about a user.



Method No. 2:	**saveStockQuote**
Description:	Saves the given real-stock stock quotes in the database based on the stock's ticker symbol
Method Input:	

Parameter Name	*Parameter Type*	*Description*
stockData	Map<String, Quote>	The hashMap that containts the ticker symbol as a key and the Quote data as a value.

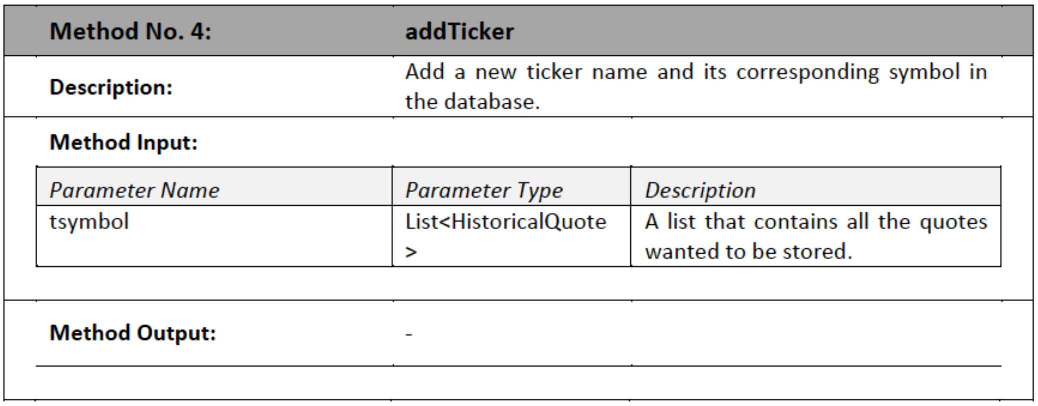
Method Output:	-



Method No. 3:	saveHistoricalData
Description:	Saves the given list of historical stock quotes in the database.
Method Input:	

Parameter Name	*Parameter Type*	*Description*
histQuoteList	List<HistoricalQuote >	A list that contains all the quotes wanted to be stored.

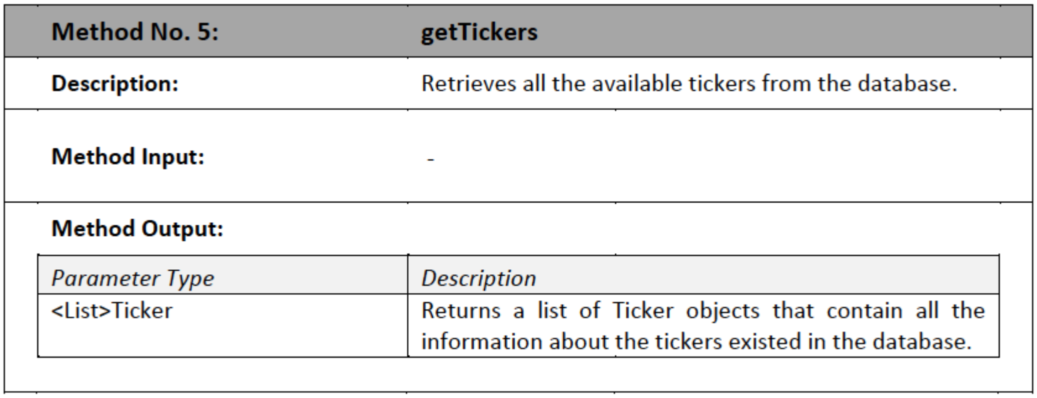
Method Output:	-



Method No. 4:	addTicker
Description:	Add a new ticker name and its corresponding symbol in the database.
Method Input:	

Parameter Name	*Parameter Type*	*Description*
tsymbol	List<HistoricalQuote >	A list that contains all the quotes wanted to be stored.

Method Output:	-



Method No. 5:	getTickers
Description:	Retrieves all the available tickers from the database.
Method Input:	-
Method Output:	

Parameter Type	*Description*
<List>Ticker	Returns a list of Ticker objects that contain all the information about the tickers existed in the database.

6.3.5 Stock Downloader

Updating and recording stock data will be handled by other independent synchronized entities within the application which have no direct participation in request/responses handling.

The stock updating is performed by a timered procedure that is invoked every certain amount of time as a separate thread.

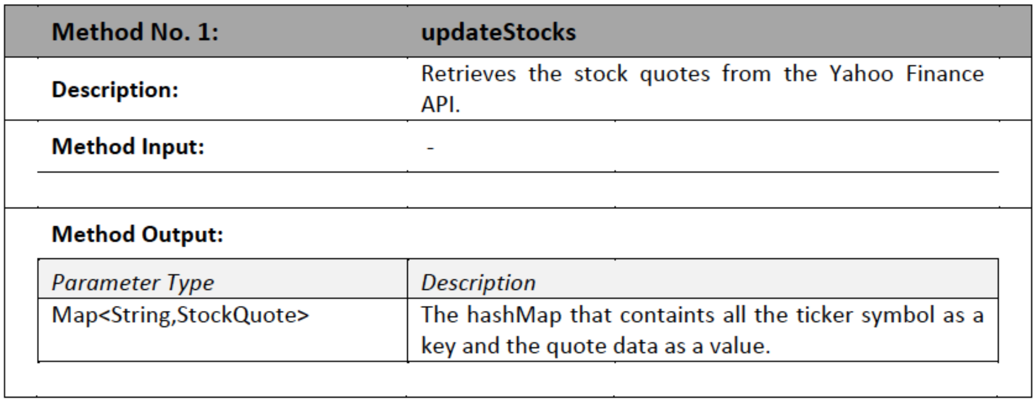
This timered procedure first obtains the list of current tickers from the database and then it downloads the data from Yahoo! Finance website using the Yahoo Finance API, which is an open source library written in Java.

In order to interface with this API we have implemented a class, StockDownloader, which is thread-safe thanks to the use of semaphores. Therefore, the same class can be accessed through the main thread running the Spring container and the timered thread in charge of updating the instantaneous values.

The StockDownloader class contains interfaces to download both instantaneous and historical data, for which it uses the aforementioned Yahoo Finance API.

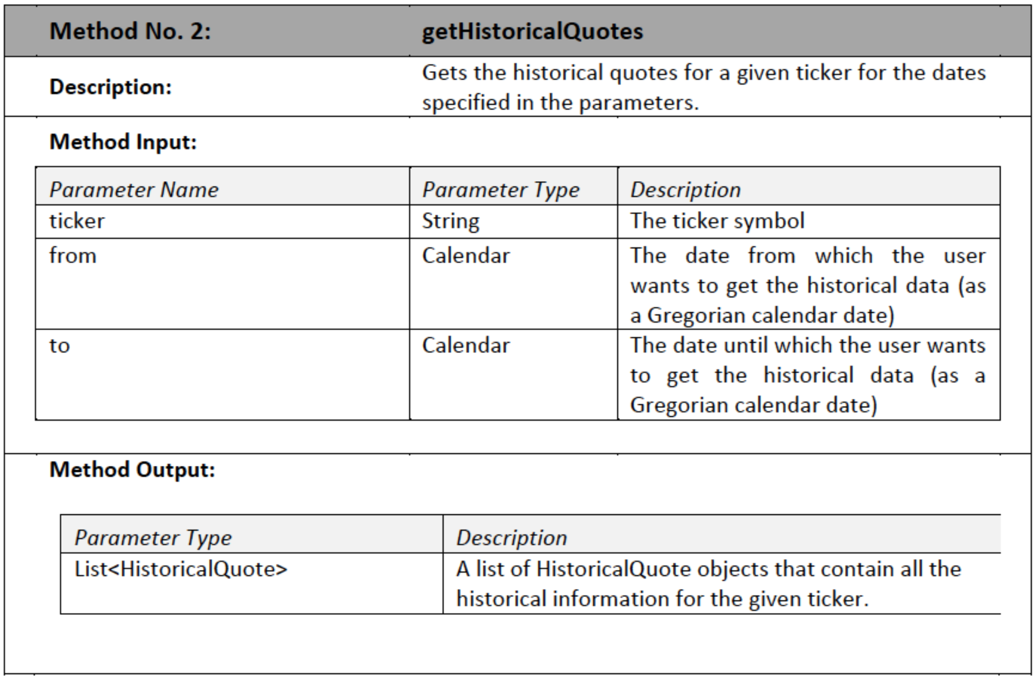
To implement the timered procedure as a separate thread we use a TimerTask, which is a class in the java.utils package. Once it obtains the information from the Yahoo public API, it invokes the model controller, which is the responsible module to store the information in the database.

The method signatures for downloading the instantaneous and historical quotes from the Yahoo! Finance API are given below.



Method No. 1:	**updateStocks**
Description:	Retrieves the stock quotes from the Yahoo Finance API.
Method Input:	-
Method Output:	

Parameter Type	*Description*
Map<String,StockQuote>	The hashMap that containts all the ticker symbol as a key and the quote data as a value.



Method No. 2:	**getHistoricalQuotes**
Description:	Gets the historical quotes for a given ticker for the dates specified in the parameters.

Method Input:

Parameter Name	*Parameter Type*	*Description*
ticker	String	The ticker symbol
from	Calendar	The date from which the user wants to get the historical data (as a Gregorian calendar date)
to	Calendar	The date until which the user wants to get the historical data (as a Gregorian calendar date)

Method Output:

Parameter Type	*Description*
List<HistoricalQuote>	A list of HistoricalQuote objects that contain all the historical information for the given ticker.

6.4 Database Design

In order to store stock information, which we categorize as instantaneous and historical information, MySQL has been used. Yahoo Finance API is a powerful tool to retrieve various pieces of financial data for stocks. The corporations we chose to store their stock data are as following:

- Google Inc.
- Yahoo! Inc.
- Apple Inc.
- Tesla Motors Inc.
- Facebook Inc.

Firstly, it is important to conduct a conceptual design of the database. In order to do this, the entity-relationship model should be created. Since the data being retrieved falls into either categories of instantaneous stock quote, or historical stock quote, two separate ER diagram will be depicted for each. Figure 18 shows these ER diagrams.

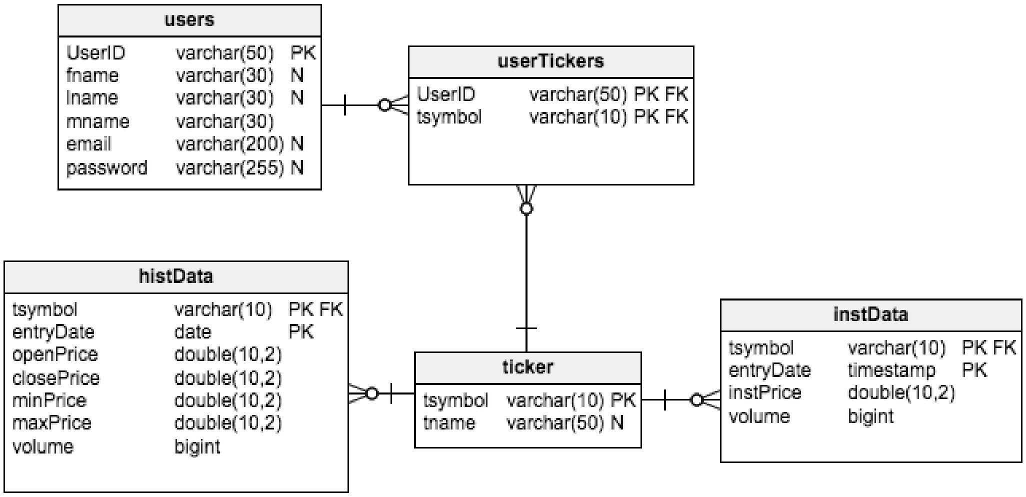


Figure 18: The Database Schema

Now that we have the schema required, appropriate API to retrieve the data will be helpful. The java class *StockQuote* has been used in this step. This class provides the user with plenty of methods to retrieve data. For example in this case, for the instantaneous data the methods to retrieve date and time, price and volume ,and for historical data methods to retrieve opening price, closing price, minimum price, maximum price and volume have been used.

The reason behind having two separate tables for historical data (histData table) and instant data records (instData) about corporations specified in ticker is because the characteristics for a daily record is different than an instant record. As Figure 18 depicts, we do not have time stamp for a historical data because a record from this class belongs to a whole day, not to a specific time. Likewise, there are some characteristics for a historical data record which are meaningless when they come to an instant data record, e.g. opening price, closing price, etc.

Furthermore, it worth to mention that having a separate table for tickers (ticker table) gives the opportunity to add/remove tickers, dynamically.

6.5 UML Diagram

Fig.19 shows the UML diagram for the system. As can be seen the system is composed of several parts, like web service manager, system, controller, machine learning, beans and model. The controller is responsible for connecting and controlling different entities of the application.

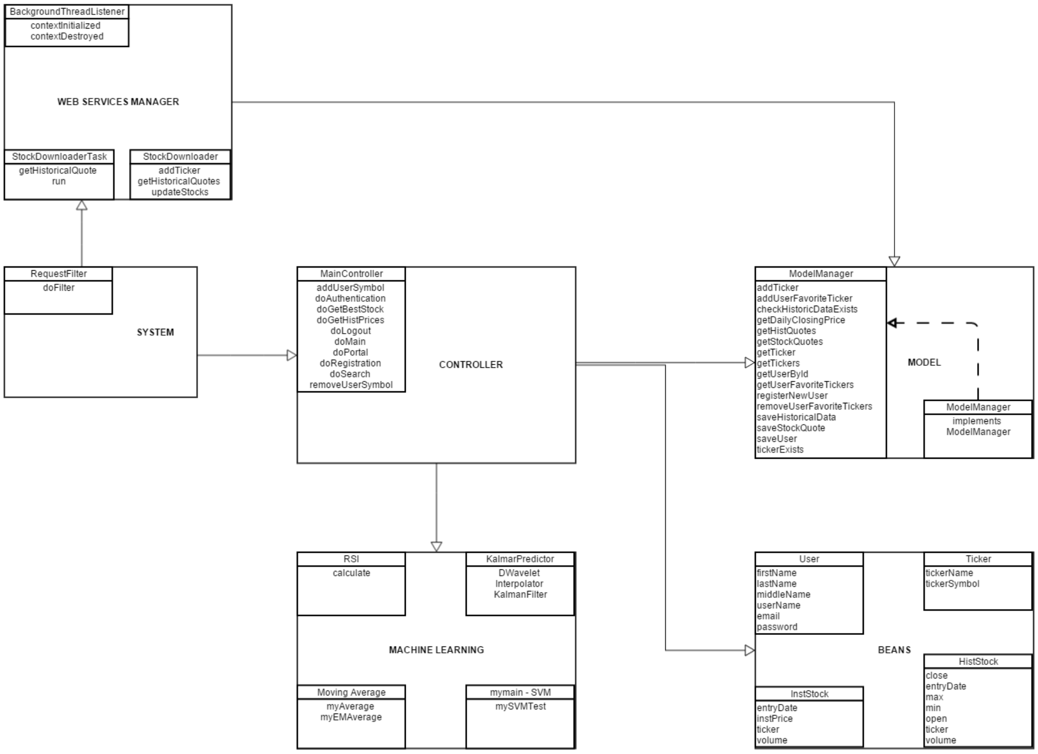


Figure 19: The UML Diagram

7 Additional Features

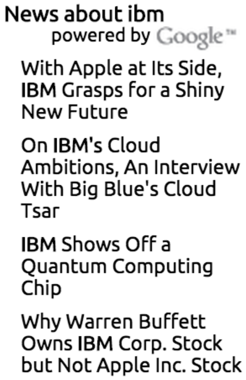
Although the main requirement for the current project is the long term and short term prediction part, a good application can offer more than a machine learning based suggestion system. As we mentioned before, current system model is exposed to noise. This noise (alongside the imperfect model) makes the whole system faulty, and it would be better if the application provides more raw information for users, so their human ability gets involved in reasoning and decision making process. In order to achieve the goal, our application offers some additional features such as related news, personalized results for each registered user, etc.

In this section we show some of the most important additional features that we have.

7.1 Related News

As we mentioned before, the application doesn't rely just on its prediction algorithms to server users. The philosophy is to trust user's sprite and offer him/her any related information that might help him/her to make a better decision. Showing related news could help the user to predict what is coming next. i.e. if the CEO of one company is giving out his personal share of his own company, it might be a bad sign that the aformentioned company is exposed to risk and its stock price might fall shortly.

Although we know that these sort of information makes the model more perfect, but implementing a feature to process the news for the application and maintain appropriate perception of the news is beyond this projects scope. The only way to gain the advantages of related news is to trust to user, show him/her the news and help him/her to monitor stock changes more carefully. Figure 20 is a screenshot from a part of the user interface that provides some related news based on the user interests.



News about ibm
powered by Google™

With Apple at Its Side, IBM Grasps for a Shiny New Future

On IBM's Cloud Ambitions, An Interview With Big Blue's Cloud Tsar

IBM Shows Off a Quantum Computing Chip

Why Warren Buffett Owns IBM Corp. Stock but Not Apple Inc. Stock

Figure 20: The Database Schema

7.2 Personalized Results

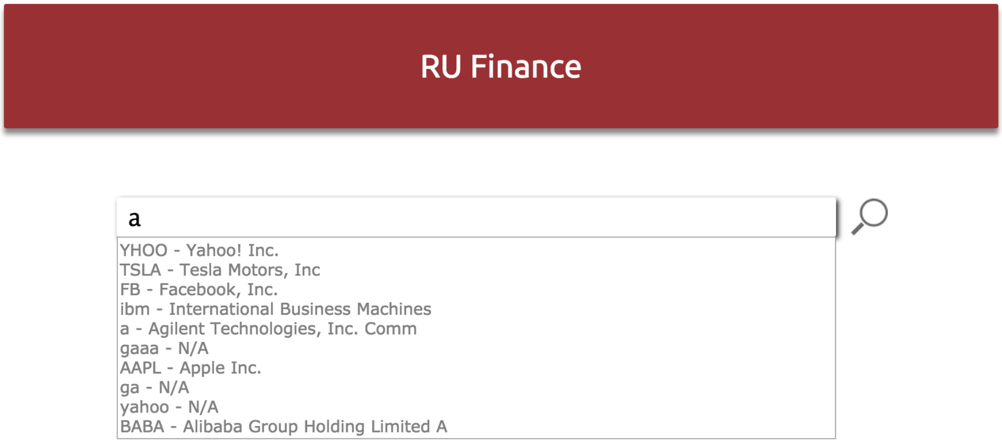
One of the useful additional features that the application has now, is to offer personal profiles to registered users. Once a user a registered, he/she can start to do research and add each company which he/she is interested in to the list of the companies he/she wants to track. This feature is essential to a persistent stock observation, and monitoring all the important events that users are interested in individually.

7.3 Realtime Add Ticker

Another additional feature is the ability to add a ticker which is not already in the database. This makes users able to add their desired company's information even if their target companies are not well-known.The system successfully add the tickers as long as our information source has the company's information (the application is using Yahoo finance as its information source)

7.4 Auto Complete

Since some tickers have different initials than their company name, it's likely that some of the queries fail since the user didn't put the right ticker symbol. With having the auto Users don't have to put the whole name of the ticker, or the company's name. This makes it easy to look for tickers that are already in the database. Figure 21 shows the auto complete feature where the frontend design was still evolving (The final submitted website may look different).



a

YHOO - Yahoo! Inc.
TSLA - Tesla Motors, Inc
FB - Facebook, Inc.
ibm - International Business Machines
a - Agilent Technologies, Inc. Comm
gaaa - N/A
AAPL - Apple Inc.
ga - N/A
yahoo - N/A
BABA - Alibaba Group Holding Limited A

Figure 21: The Database Schema

8 Appendix A: Stock Market Behavior Study

8.1 Introduction

There are two broad categories for analyzing and classifying market analysis techniques in order to take investment decisions: technical analysis and fundamental analysis.

- Technical analysis (or charting) is only interested in the history of price and volume of market movements. It usually implies the use of charts.
- Fundamental analysis is interested in the inner characteristics of each company, such as earnings, dividends, new products and research as well as the economic factors

In this essay we focus on technical analysis techniques.

Technical analysis involve different studies of indicators and patterns that can be represented using graphs. In order to create this graphs the basic needed information are the price and the volume of negotiation. Some sources distinguish between charting analysis and technical analysis even though both areas are known globally as technical analysis. Charting analysis focuses on the study of the patterns that can be seen in a price chart as an indicator of the future trend. Technical analysis, however, try to operate mathematically and statistically with the prices and volumes in order to determine and detect different trend situations.

The focus of this paper is on patterns and indicators used in technical analysis. By statistical analysis of market activity and usage of chart patterns, technical indicates and oscillators or other tools to suggest patterns, technical analysis predicts the future activity. Technical analysis relies on three major principles: 1. The market discounts everything. Unlike fundamental analysis, technical analysis does not take in to account fundamental factors of the company. Instead technical analysts believe that all these factors are priced into the stock. 2. Price movements follow trends. This will let the analysts to predict the future after the establishment of a trend. 3. History tends to repeat itself. Market psychology is thought to result is a repetitive nature for price movement.

8.2 Concepts

8.2.1 Support and resistance

The difference between support and resistance is as follows:

- Support is a level of prices in which the down trend is stopped and the trend goes upward again. The market considers that is a very low price level. Therefore, when the price reaches this value, buy signals are activated.
- On the contrary, resistance is a price level in which the up trend is stopped and the trend goes downwards again. The market considers that this is a very high price level. Therefore, when the price reaches this value, sell signals are activated.

As we can see, the market itself is the one who determines the support points and the resistance points. Support and resistances are detected when we analyze the price history graph where we can observe how there are levels in which the upward price trend is stopped and other levels in which the downward price trend is stopped.

Support and resistances have a specific length in time, because there is a moment in which the price trend changes. If the overall trend goes upwards, the support and resistances will be higher every time, meanwhile if the overall trend goes downwards, the support and resistances will be lower every time.

Support and resistances are stronger if the buy/sell volumes related to those points are higher. Also, if supports and resistances last longer in time, they can be considered stronger.

8.3 Trend lines

Technical analysis assumes that prices trend, and therefore trend Lines are an important tool in technical analysis in order to help identify and confirm trends.

Trend lines connect two or more prices in a chart using a straight line. This line can be extended to the future to help predict how the trend will continue.

Trend lines are drawn using arithmetic-scale or semi-logarithmic scale graphs, and they involve connecting low points or high points. Drawing a line is complex since it involves connecting pivot points, which will determine the slope and position of the line. Furthermore, some sources suggest connecting two points, while others encourage to connect 3 or more points in a straight line.

Two classifications can be seen depending on which characteristic do we focus on. If we ignore the slope of the trend line, we can distinguish between resistance trend line and support trend line. Focusing on the slope of the trend lines, there are two types: up trend lines and down trend lines. The former focuses on periods of time in which the prices keep increasing, while the latter focuses on periods of time in which the prices keep decreasing.

Apparently, trend lines can be applied in any scale of time. However, trends can be classified in three different kinds depending on their length.

- Main trend is usually seen in time ranges greater than a year.
- Secondary trend is usually seen in ranges from three weeks to several months.
- Minor trends are seen in ranges from a single day to several weeks.

8.3.1 Support trend lines

Support trend lines are formed by connecting low points, which are lowest points in a strictly decreasing period of time. Examples can be seen in Figure 22.

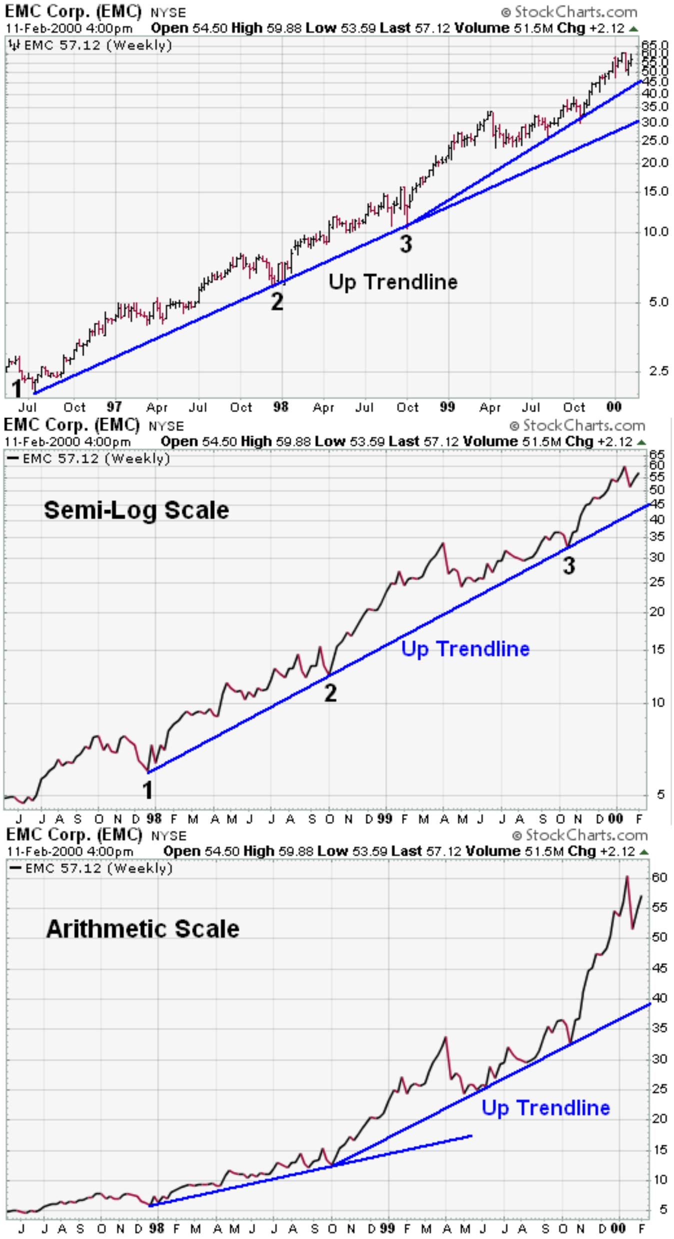


Figure 22: http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:trend_lines

8.3.2 Resistance trend lines

Resistance trend lines are formed by connecting high points, which are highest points in a strictly increasing period of time. Examples can be seen in Figure 23.

8.3.3 Up trend lines

Up trend lines act as a support (see previous section) and has to be constructed by connecting two low points, in which the second low (the most recent one) has to be higher than the first low in order for the slope to be positive, i.e. in order to have an increasing line. It is known as a series of higher highs and higher lows. This is used to determine buy signals.

8.3.4 Down trend lines

Down trend lines act as a resistance (see previous section) and has to be constructed by connecting two high points, in which the second low (the most recent one) has to be lower than the first low in order for the slope to be negative, i.e. in order to have a decreasing line. It is known as a series of lower highs and lower lows. This is used to determine sell signals.

8.4 Trend channels

Trend channels are made by two parallel lines. One is the called the trend line and the other is parallel to the first one. There are three kinds of channels. When the slope of the trend lines is positive, channels are called upward channels. When the slope is negative, channels are called downward channels, and when the slope is 0, channels are called horizontal.

In upward trends, the trend channel is formed by one up trend line and one parallel line to it that moves through the maximums.

In downward trends, the trend channel is formed by one down trend line and one parallel line to it that moves through the minimums.

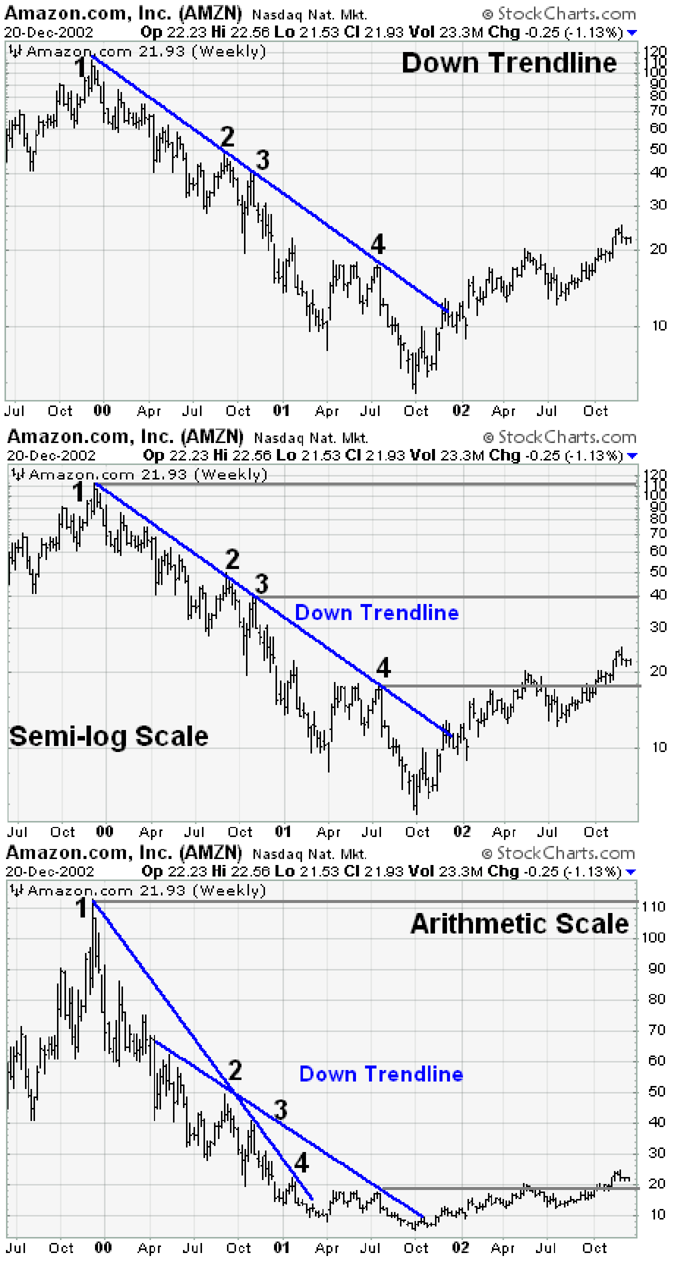


Figure 23: `http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:trend_lines`

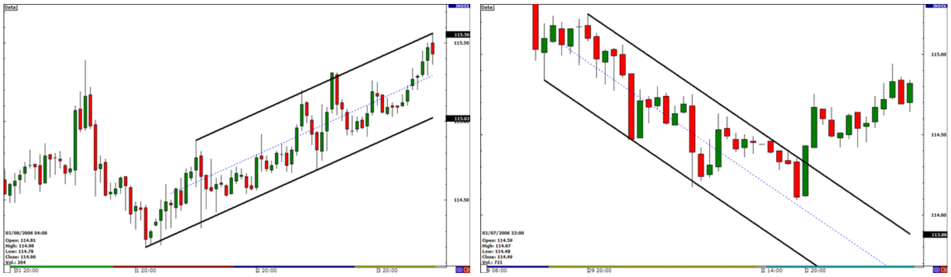


Figure 24: Upward (ascending) channels and downward (descending) channels. `http://www.babypips.com/forexpedia/Trend_Channel`

The wider the channel is, the stronger the trend will be.

Figure 24 is an example of ascending and descending channels.

8.5 Major trend reversals

A major trend reversal is a change of the price trend direction. For example, is the point in which an up trend becomes a down trend, or a down trend becomes an up trend. See above for definitions of up and down trends.

These trend reversal points can be easily seen on a price chart, in which major trend reversals are seen as bottoms or tops.

Figure 25 is a good example of several major trend reversals that occurred during the span of about 4 years.

8.6 Trend reversal patterns

8.6.1 Head-and-shoulders pattern

The head-and-shoulders pattern is a visual configuration which can be usually found in the technical analysis graphs. It is usually found at the end of a trend and implies a trend reversal.

The prices form three ceilings in which the central maximum (the head) is

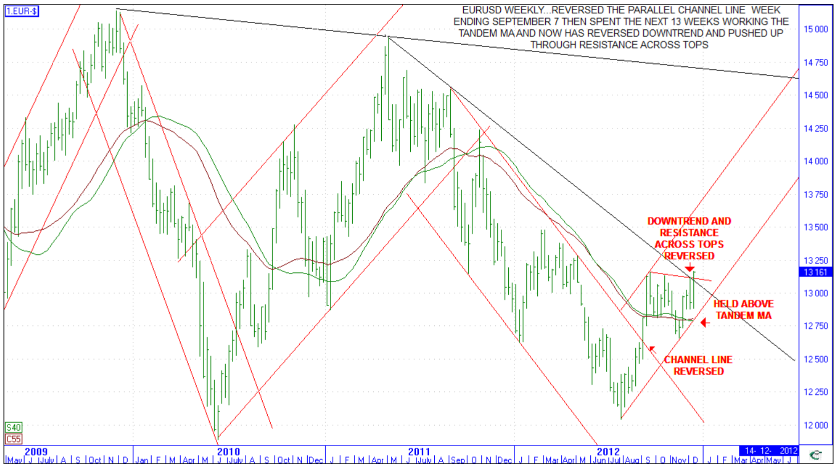


Figure 25: Major trend reversals on several points in time. The figure also shows upward channels and downward channels. `http://investrac.com/blog/eurusd-medium-term-trend-outlook-bullish`

higher than the other two extremes (right and left shoulders). The buy/sell volume tends to be smaller in the head, which indicates a weakness in the trend. The third ceiling, the right shoulder, is smaller than the head, which indicates a weakness in the trend. It is difficult to identify by the investor, because the confirmation is only produced when the right shoulder is well formed, i.e. when the prices already traversed the neck-line, which is the support line.

This figure can be derived in multiple ways, such as inverted head-and-shoulders pattern (a trend reversal from down trend to up trend), or multiple head-and-shoulders pattern.

Figure 26 shows head and shoulders pattern with real market data and in a schematic manner. It also shows the inverted head-and-shoulders pattern.

8.6.2 Double and Triple Top and Bottom

Double and triple tops (see Figure 27) and bottoms are patterns that in general indicate major trend reversal. They are not as predictable as the head-and-shoulders pattern and, therefore, they are more difficult to iden-

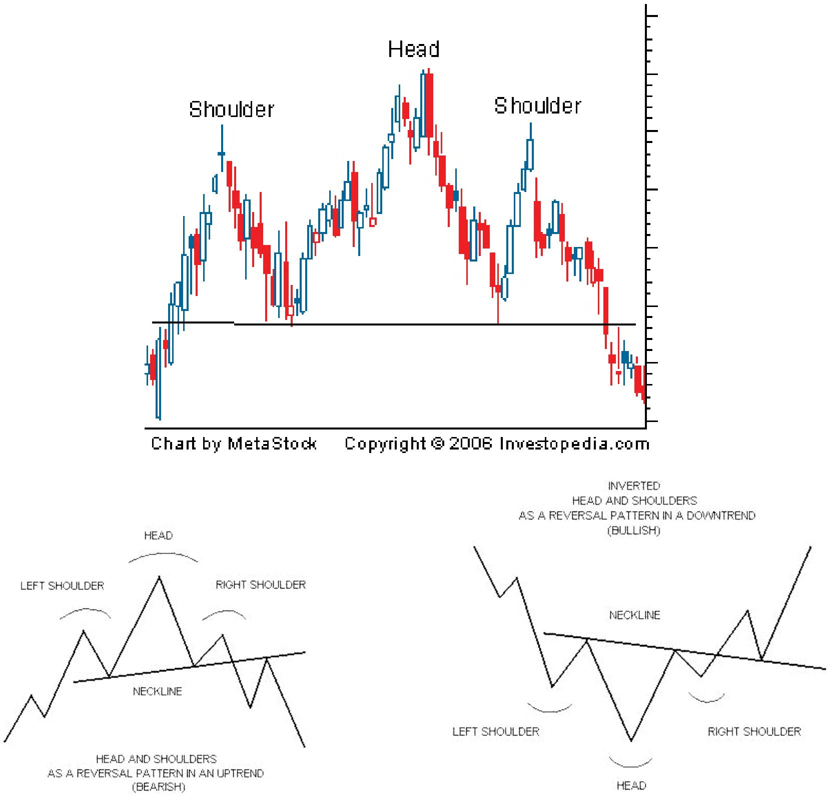


Figure 26: Head and shoulders pattern with real market data and in a schematic manner. Inversed head-and-shoulders pattern. `http://www.investopedia.com/terms/h/head-shoulders.asp`, `http://www.chartpatterns.com/headandshoulders.htm`

tify. They do not frequently appear in the market. However, they are very important because they can indicate the end of a main trend.

There are different stages:

- After an ascending short-range trend, a price is achieved in which the interest to sell gets higher than the interest to buy.
- Low volume price drop.
- Price is recovered with even lower volume. The price reaches the same

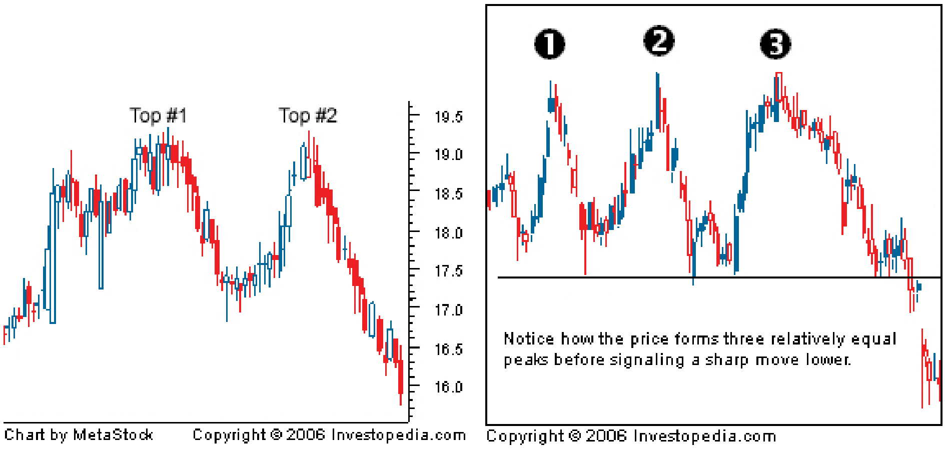


Chart by MetaStock Copyright © 2006 Investopedia.com

Notice how the price forms three relatively equal peaks before signaling a sharp move lower.

Copyright © 2006 Investopedia.com

Figure 27: Double and triple top pattern`http://www.investopedia.com/terms/d/doubletop.asp`, `http://www.investopedia.com/terms/t/tripletop.asp`

level as the preceding point before dropping again with low volume, reaching again the previous minimum price.

- If the price is recovered again with low volume, we identify a triple top.

In order to validate the pattern, it has to occur during a long time interval, approximately from one to two months between each maximum (a little smaller if we are talking about a triple top).

8.6.3 Rounding Top and Bottom

These patterns are not frequently seen in the active markets. These patterns are more usually seen as rounding bottoms than rounding tops. They indicate a slow but gradual price change in a trend.

The rounding bottom (see Figure 28) is generated because the volume descends significantly reaching, in the center of the pattern, a minimal volume during a long period of time of several months. After this, volume can suddenly begin to increase up to the point in which a strong increase that ends the pattern.

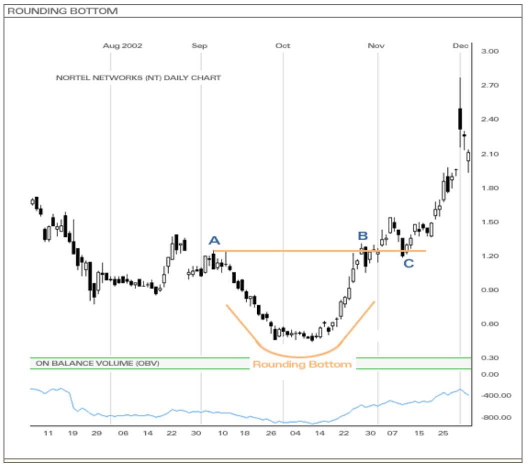


Figure 28: Rounding bottom pattern http://markettechlab.com/blog/rounding-bottoms/

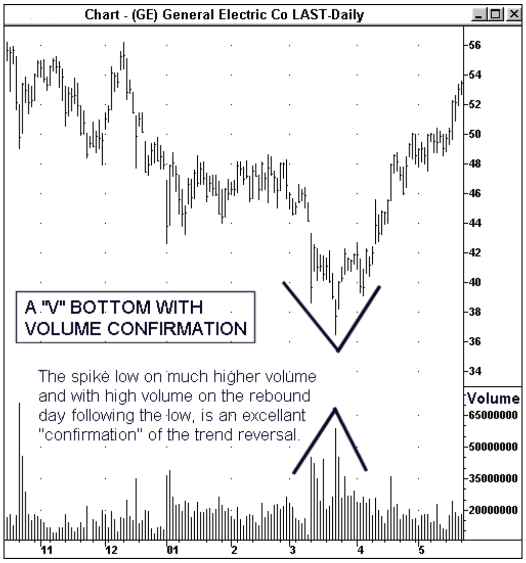
8.6.4 V-tops and V-bottoms

The V pattern is a complex pattern because it represents a sudden trend reversal in the market. An indication that a V pattern can occur, even though it does not apply in general, is the appearance of a large price range within a single day. While this is happening, a large volume usually occurs. The greater the volume and larger the price range, the higher the chances are that a V pattern will occur.

Figure 29 shows an example of V-bottom with volume confirmation.

8.6.5 Saucers and spikes

These patterns are less common compared to the other ones discussed. The spike refers to a sudden change in trend. The word “sudden” is the key to the definition here. What differentiates a spike from a normal reversal is the absence of a transition period. Spikes might happen due to stocks being overbought or oversold. On the other hand, the saucer pattern holds in itself a subtle and gradual change in trend. As the name suggests the pattern resembles a rounding bottom which goes down and maintain a horizontal



A "V" BOTTOM WITH VOLUME CONFIRMATION

The spike low on much higher volume and with high volume on the rebound day following the low, is an excellant "confirmation" of the trend reversal.

Figure 29: V pattern `http://www.optioninvestor.com/page/oin/education/traders/2011/07-07.21-14-39.html`

pattern and the goes up. Examples of spike and saucer pattern can be seen below.

The slow nature of saucer pattern makes it clear as why saucer charts are made on a longer scale of time like weekly or monthly basis for several years.

8.7 Continuation patterns

Patterns that predict a trend continuation are called continuation pattern. In this section we will study some of the most commonly used. There are specific formations/patterns that are used to predict continuation patterns. This patterns are as important as the ones that we already studied, the ones that indicate trend change, because thanks to them we are allowed to stay in advantage situations for longer time. Also, they allow to wait a little more before buying signals, after the down trend has finalized.

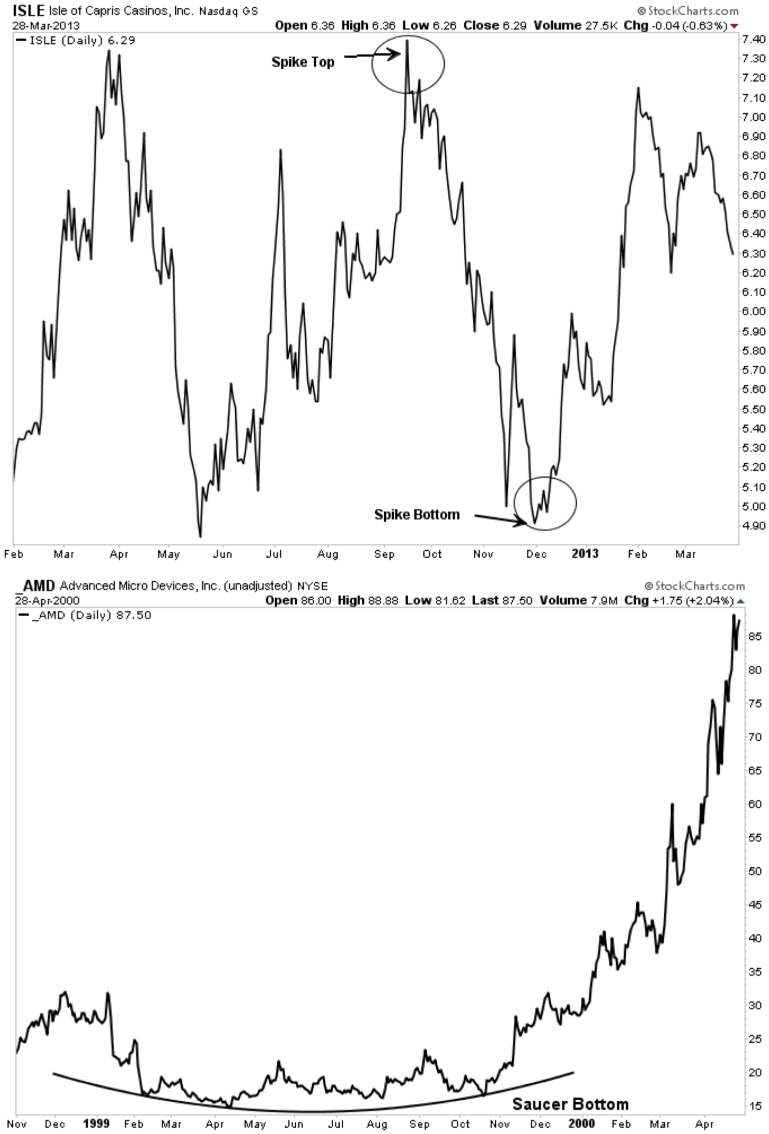


Figure 30: Saucers and spikes `http://stockcharts.com/school/doku.php?id=chart_school:john_murphy_s_chart`

8.7.1 Flags and Pennants

The first patterns that clearly indicate a continuation trend are Flags and Pennants (see Figure 31). These are the most reliable continuation pattern and rarely they will indicate a trend change. Flags are little rectangles formed by the prices, which slope is slightly leading in the contrary direction than the preceding trend.

Pennants are similar figures, except that lines slightly converge, forming figures similar to small wedges, as well slightly leading in the contrary direction than the preceding trend.

Flags and Pennants represent small pauses on active markets. These figures usually appear.

For example, usually the market begins a strong vertical ascending trend due to the abscence of sellers. If the price has gone up a lot, and fast, the most probable situation is that investors want fast earnings. At that point, some interest in selling appears, which slows down the quick ascending trend, but this situation does not last long and sellers are abscent soon.

It is known that flags usually appear more or less in the middle point of an up trend or a down trend. This has led to a rule to calculate the price goal that can be achieved when the main trend is resumed. Measuring the distance between the beginning of the trend and the beginning of the pattern we can estimate the distance that the trend will follow after the flag/pennant pattern is broken.

A flag has to last at most 3 weeks to form, with a minimum of 4 to 5 sessions. With less time, it is very difficult to see the pattern. In down trends, the formation times are usually shorter. In the great majority of situations they only last about a week, which as we just mentioned, is the minimum formation time.

Note that all theoretial concepts about flags can also be applied to pennants. Pennants are presented in situations in which very good benefits can be obtained.

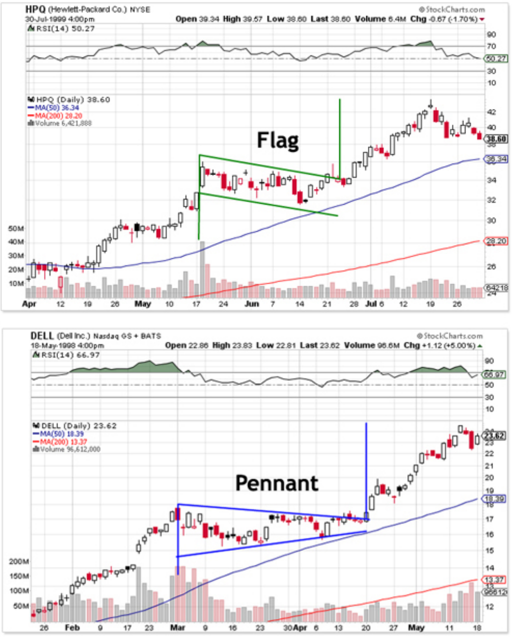


Figure 31: Flag and pennant. http://www.mysmartrend.com/technical-analysis-chart-patterns/flag-and-pennant

8.7.2 Triangles

This patterns are the most difficult to classify. The great majority of times they represent continuation and consolidation patterns, but in some cases they can indicate a trend change.

A set of ascending low and high points indicate an up trend, while a descending low and high points indicate a down trend, as we have seen while talking about trends. However, when descending high points and ascending low points a triangle is being formed. This situation indicates insecurity in the market. A minimum of four points are needed in order to see a triangle: two low points and two high points. The lines that can be drawn connecting the low points and the high points have to converge in a point, which is called the vertex.

Depending on the slope of the two lines that compose the triangle we can classify as a symmetric triangle, ascending triangle, and descending triangle. An special case is the inverted triangle (also known as expanding triangle).

An example of the triangle pattern can be seen in Figure 32.

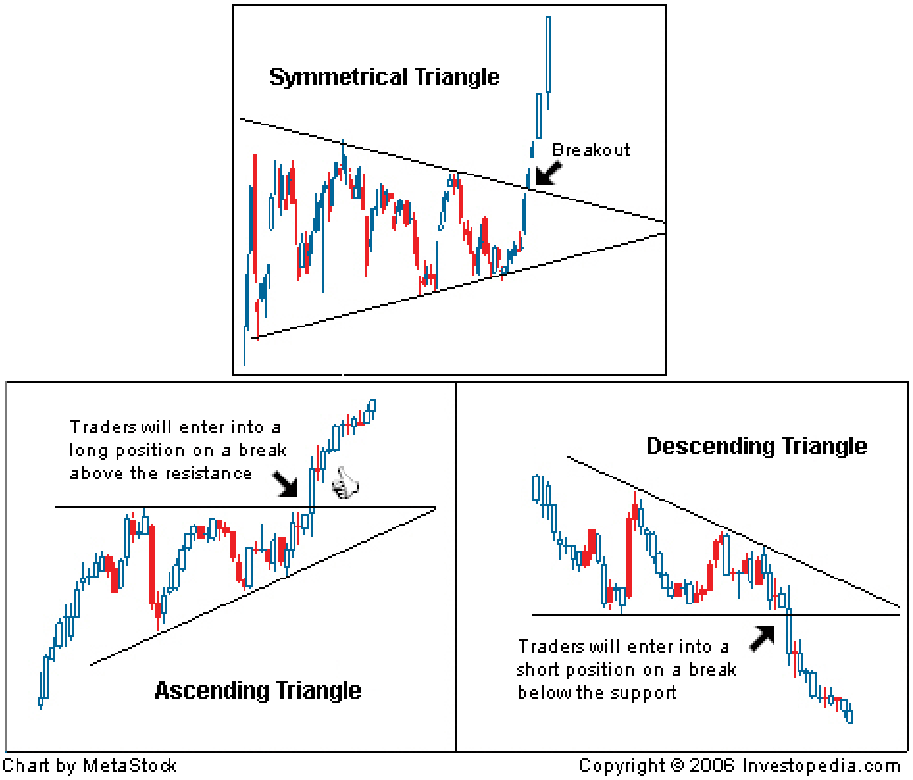


Figure 32: Triangle pattern. Note the three different classifications of triangles. `http://www.investopedia.com/university/technical/techanalysis8.asp`

8.7.3 Rectangle

The rectangle (see Figure 33) is another pattern that pronosticates market unstability. A rectangle appears in the market where there are strong contrary interests. On the one hand, the sellers that want to sell their assets at a determined price. On the other hand, the buyers that want to buy assets at a lower price. The price of the assets fluctuate between both these values until one of the interests loses strength and the pattern break. It is important to observe the volume of the movements that appear inside the rectangle. The duration of the rectangles varies between one and three months.

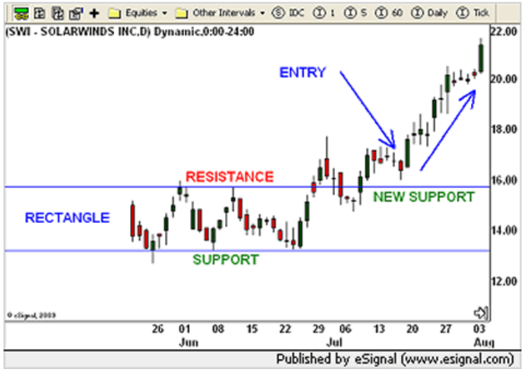


Figure 33: Rectangle pattern. `http://investorplace.com/2009/10/how-to-invest-in-ipos/`

8.7.4 Gap

Gap is an area in a graph in which there are no operations. A gap in an up trend is produced when the price range within a day is higher than the maximum of the previous day' prices. In a down trend, a gap is produced when the price range whithin a day is lower than the minimum price of the previous day' prices.

It is an important pattern due to the consequences that have been observed from the price behavior when a gap occurs. A gap indicate a lack of sell/buy interest in a price range, which can be translated in a low volume during the interval.

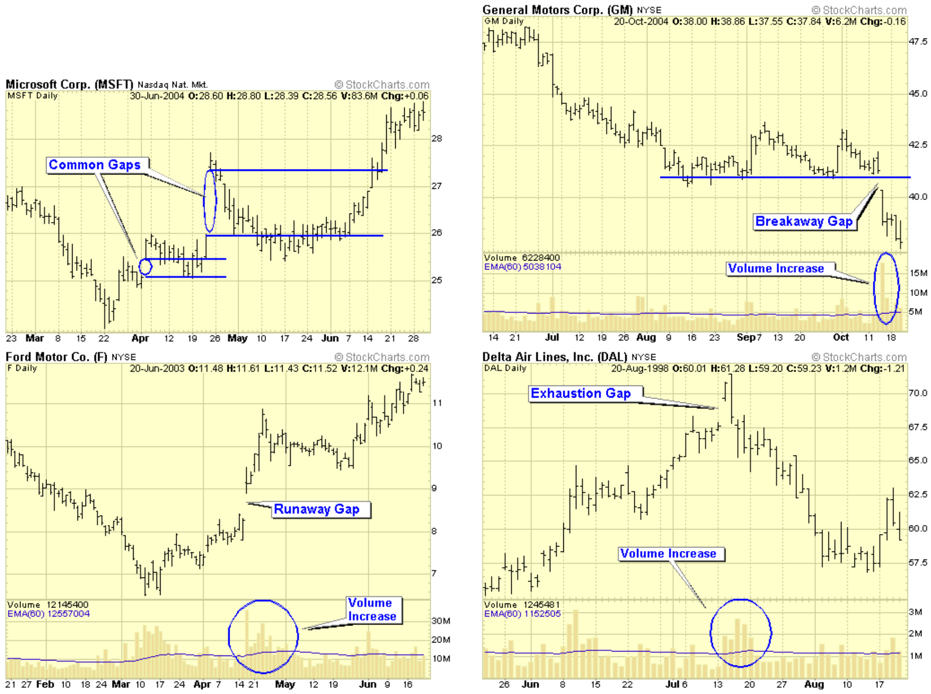


Figure 34: Gap pattern `http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:gaps_and_gap_analysis`

Four gap classes have been observed: the common gap (or area gap), the breakawat gap, the runaway gap, continuation gap and the exhaustion gap. See Figure 34 for an example of these 4 different gap classes.

9 Contribution

Equal.

10 Glossary of Terms

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

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

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

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

Neural network: conceptually based off the central nervous system, it interconnects systems of neurons that can calculate values for inputs by feeding information through the network.

Short term investment: An account in the current assets section of a company's balance sheet. This account contains any investments that a company has made that will expire within one year. For the most part, these accounts contain stocks and bonds that can be liquidated fairly quickly.

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

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

Closing Price: The price a particular stock closes at on a given trading day.

Opening Price: The price a stock starts off at a particular trading day.

Moving Average Prediction Model: A way to predict the future price of stocks based on the assumption of constant underlying mean of given prices

Market Trend: The tendency of a financial market to move in a particular direction over time. Bull market refers to an upward trend, and a bear market refers to a downward trend.

Stock Symbol/Ticker A unique set of symbols that represent a particular company. Ex: GOOG is the stock symbol of Google.