Predicting Stock Price Volatility

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Computer Science & Engineering

under the Supervision of

Ms. Sanjana Singh

By

Prashant Singhal

111224

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled "Predicting Stock Price Volatility using Machine Learning", submitted by Prashant Singhal in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Supervisor's Name**

Designation

Acknowledgement

There are many people who are associated with this project directly or indirectly whose help and timely suggestions are highly appreciable for completion of this project. First of all, I would like to express my sincere gratitude to my supervisor Ms Sanjana Singh, for her super vision, encouragement, and support which has been instrumental for the success of this project. It was an invaluable experience for me to be one of her students. Because of her, I have gained a careful research attitude.

I would like to thank Prof. Dr. SP Ghrera, Head, Department of Computer Science Engineering for his kind support and constant encouragements, valuable discussions which is highly commendable.

Lastly, I would also like to thank my parents for their love and affection and especially their courage which inspired me and made me to believe in myself.

Date: Prashant Singhal

111224

Table of Content

List of Figures

List of Tables

Abstract

Stock market is an important and active part of nowadays financial markets. Stock time series volatility analysis is regarded as one of the most challenging time series forecasting due to the hard-to-predict volatility observed in worldwide stock markets. Stock market state is dynamic and invisible but it will be influenced by some visible stock market information. Current models for predicting volatility do not incorporate information flow and are solely based on historical volatilities. The future stock volatility is better predicted by our method than the conventional models. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a difficult task. The key factor for each investor is to earn maximum profits on their investments. Numerous techniques used to predict stocks in which fundamental and technical analysis are one among them. In this project, prediction algorithms and functions are used to predict future share prices and their performance is compared. The results from analysis shows that M5P algorithm offers the ability to predict the stock prices more accurately than the other existing algorithms.

CHAPTER 1

1. INTRODUCTION

Volatility, defined as the variation of return around some expected value, is a commonly used estimate of risk in financial assets. The expected future volatility is therefore a key parameter for portfolio selection, risk management, and pricing-equity-related derivative instruments. Risk anticipation also has an important implication for policy makers such as central banks and financial regulators.

Stock Market research encapsulates two elemental trading philosophies; Fundamental and Technical approaches. In Fundamental analysis, Stock Market price movements are believed to derive from a security's relative data. Fundamentalists use numeric information such as earnings, ratios, and management effectiveness to determine future forecasts. In Technical analysis, it is believed that market timing is key. Technicians utilize charts and modeling techniques to identify trends in price and volume. These later individuals rely on historical data in order to predict future outcomes.

When predicting the future prices of Stock Market securities, there are several theories available. The first is Efficient Market Hypothesis (EMH). In EMH, it is assumed that the price of a security reflects all of the information available and that everyone has some degree of access to the information. From the tenets of EMH, it is believed that the market reacts instantaneously to any given news and that it is impossible to consistently outperform the market.

A different perspective on prediction comes from Random Walk Theory. In this theory, Stock Market prediction is believed to be impossible where prices are determined randomly and outperforming the market is infeasible. Random Walk Theory has similar theoretical underpinnings to Semi-Strong EMH where all public information is assumed to be available to everyone. However, Random Walk Theory declares that even with such information, future prediction is ineffective.

The whole idea of this project is to do predictive analysis of stock price changes. The predictive analysis will be accompanied with the help of Collection of data[News Article], Bag-of-words approach, Data-Clustering Algorithm, Semantic Analysis on the data, Training Data, Regression Analysis, Result Evaluation.

1.1 NATIONAL STOCK EXCHANGE OF INDIA (NSE)

NSE was incorporated in November 1992, and received recognition as a stock exchange under the Securities Contracts (Regulation) Act, 1956 in April 1993. Since its inception in 1992, NSE of India has been at the vanguard of change in the Indian securities market. This period has seen remarkable changes in markets, from how capital is raised and traded, to how transactions are cleared and settled. The market has grown in scope and scale in a way that could not have been imagined at that time. Average daily trading volumes have jumped from Rs. 17 crore in 1994-95 when NSE started its Cash Market segment to Rs.16,959 crore in 2009-10. Similarly, market capitalization of listed companies went up from Rs.363,350 crore at the end of March 1995 to Rs.36,834,930 crore at end March 2011. Indian equity markets are today among the most deep and vibrant markets in the world. NSE offers a wide range of products for multiple markets, including equity shares, Exchange Traded Funds (ETF), Mutual Funds, Debt instruments, Index futures and options, Stock futures and options, Currency futures and Interest rate futures. Our Exchange has more than 1,400 companies listed in the Capital Market and more than 92% of these companies are actively traded. The debt market has 4,140 securities available for trading. Index futures and options trade on four different indices and on 223 stocks in stock futures and options as on 31st March, 2010. Currency futures contracts are traded in four currency pairs. Interest Rate Futures (IRF) contracts based on 10 year 7% Notional GOI Bond is also available for trading. The role of trading members at NSE is to the extent of providing only trading services to the investors; the Exchange involves trading members in the process of consultation and participation in vital inputs towards decision making.

Stock Market

A stock market index is a method of measuring a stock market as a whole. The most important type of market index is the broad-market index, consisting of the large, liquid stocks of the country. In most countries, a single major index dominates benchmarking, index funds, index derivatives and research applications. In addition, more specialized indices often find interesting applications. In India, we have seen situations where a dedicated industry fund uses an industry index as a benchmark. In India, where clear categories of ownership groups exist, it becomes interesting to examine the performance of classes of companies sorted by ownership group.

Stock Classification

Stocks are often classified based on the type of company it is, the company's value, or in some cases the level of return that is expected from the company. Below is a list of classifications which are generally known to us Growth Stocks, Value Stocks, Large Cap Stocks, Mid Cap Stocks, and Small Cap Stocks. Stocks are usually classified according to their characteristics. Some are classified according to their growth potential in the long run and the others as per their current valuations. Similarly, stocks can also be classified according to their market capitalization. S&P CNX NIFTY has NIFTY (50), Junior NIFTY (50), CNX IT (20), Bank NIFTY (12), NIFTY Midcap50, CNX Realty (10) and CNX Infra (25). The sectoral distribution of NSE are Financial services or banks, Energy, Information Technology, Metals, Automobile, FMCG, Construction, Media & Entertainment, Pharma, Industrial Manufacturing, Cement, Fertilizers & Pesticides, Textiles, Power and Telecom [11].Two ways of analyzing stock prices namely fundamental analysis and technical analysis are described in the next section.

Fundamental Analysis

Fundamental analysis involves analysis of a company's performance and profitability to determine its share price. By studying the overall economic conditions, the company's competition, and other factors, it is possible to determine expected returns and the

intrinsic value of shares [14]. This type of analysis assumes that a share's current (and future) price depends on its intrinsic value and anticipated return on investment. As new information is released pertaining to the company's status, the expected return on the company's shares will change, which affects the stock price. So the advantages of fundamental analysis are its ability to predict changes before they show up on the charts. Growth prospects are related to the current economic environment.

Technical Analysis

Technical analysis is a method of evaluating securities by analyzing the statistics generated by market activity, such as past prices and volume. To predict the future movement and identify patterns of a stock, technical analysts use tools and charts ignoring the fundamental value. Few analysts rely on chart patterns and while others use technical indicators like moving average (MA), relative strength index (RSI), on balance volume (OBV) and moving average convergence-divergence (MACD) as their benchmark. In any case, technical analyst's exclusive use of historical price and volume data is what separates them from their fundamental counterparts. Unlike fundamental analysts, technical analysts don't care whether a stock is undervalued - the only thing that matters is a security's past trading data and what information this data can provide about where the security might move in the future. Technical analysis really just studies supply and demand in a market in an attempt to determine what direction, or trend, will continue in the future.

Technical analysis studies the historical data relevant to price and volume movements of the stock by using charts as a primary tool to forecast possible price movements. According to early research, future and past stock prices were deemed as irrelevant. As a result, it was believed that using past data to predict the future stock price was impossible, and that it would only have abnormal profits. However, recent findings have proven that there was, indeed, a relationship between the past and future return rates. Furthermore, arguments have been made that by using past return rates, future return rates could also be forecasted. There are various kinds of technical indicators used in futures market as well. There are 26 technical indicators which can be primarily used to

analyze the stock prices. Based upon the analysis, stock trend either up or down can be predicted by the investor.

1.2 Machine Learning

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. Machine learning is the modern science of finding patterns and making predictions from data based on work in multivariate statistics, data mining, pattern recognition, and advanced/predictive analytics.

Machine learning methods are particularly effective in situations where deep and predictive insights need to be uncovered from data sets that are large, diverse and fast changing — Big Data. Across these types of data, machine learning easily outperforms traditional methods on accuracy, scale, and speed. For example, when detecting fraud in the millisecond it takes to swipe a credit card, machine learning rules not only on information associated with the transaction, such as value and location, but also by leveraging historical and social network data for accurate evaluation of potential fraud.

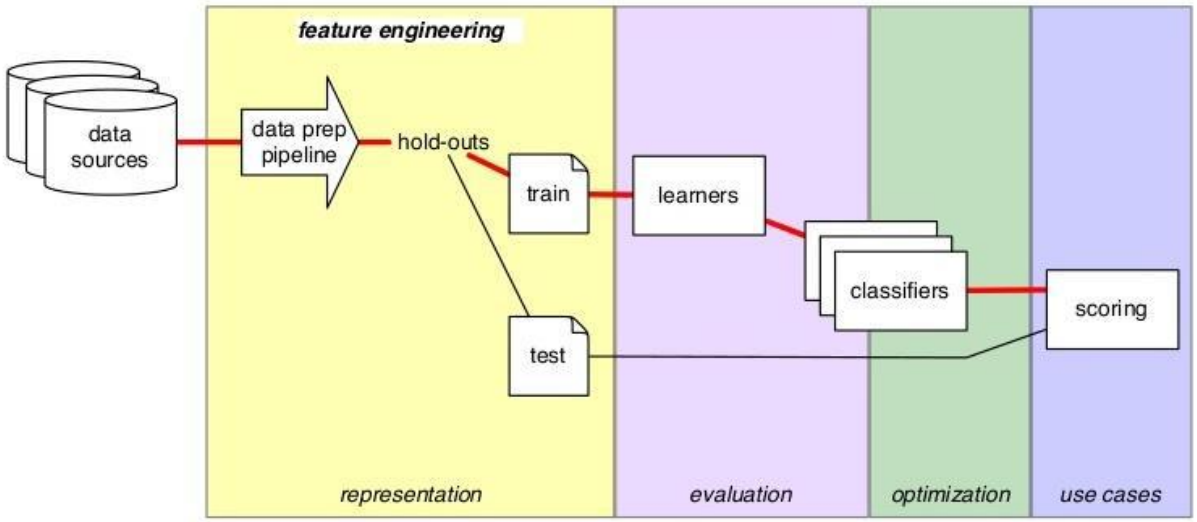


Fig. 1 Machine Learning Process

Different Techniques

1.2.1 Supervised Learning

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of *training examples*. In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

Supervised Learning Techniques:

- **Decision tree learning**

  It uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

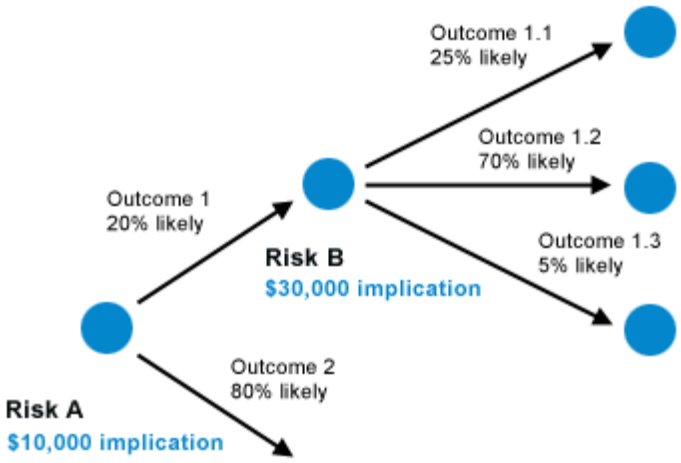


Fig. 2: Decision Tree to Predict Different outcomes and their Implications

- **Support vector machines**

  In machine learning, support vector machines (SVMs) is a supervised learning model with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

- **Artificial neural networks (ANNs)**

  In machine learning, **artificial neural networks (ANNs)** are a family of statistical learning algorithms inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected

"neurons" which can compute values from inputs, and are capable of machine learning as well as pattern recognition thanks to their adaptive nature.

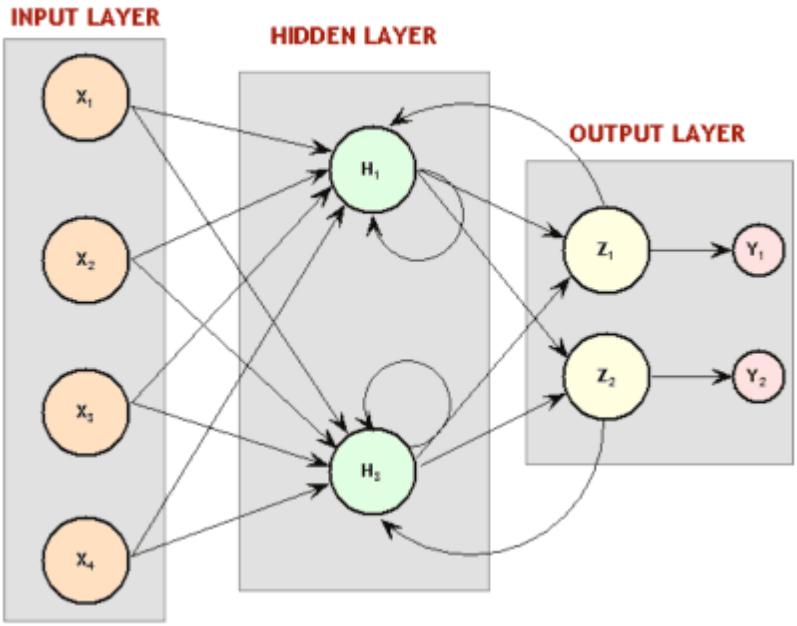


Fig.3: Schematic Representation of ANN

1.2.2 Unsupervised Learning

- **Cluster Analysis**

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as

a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results.

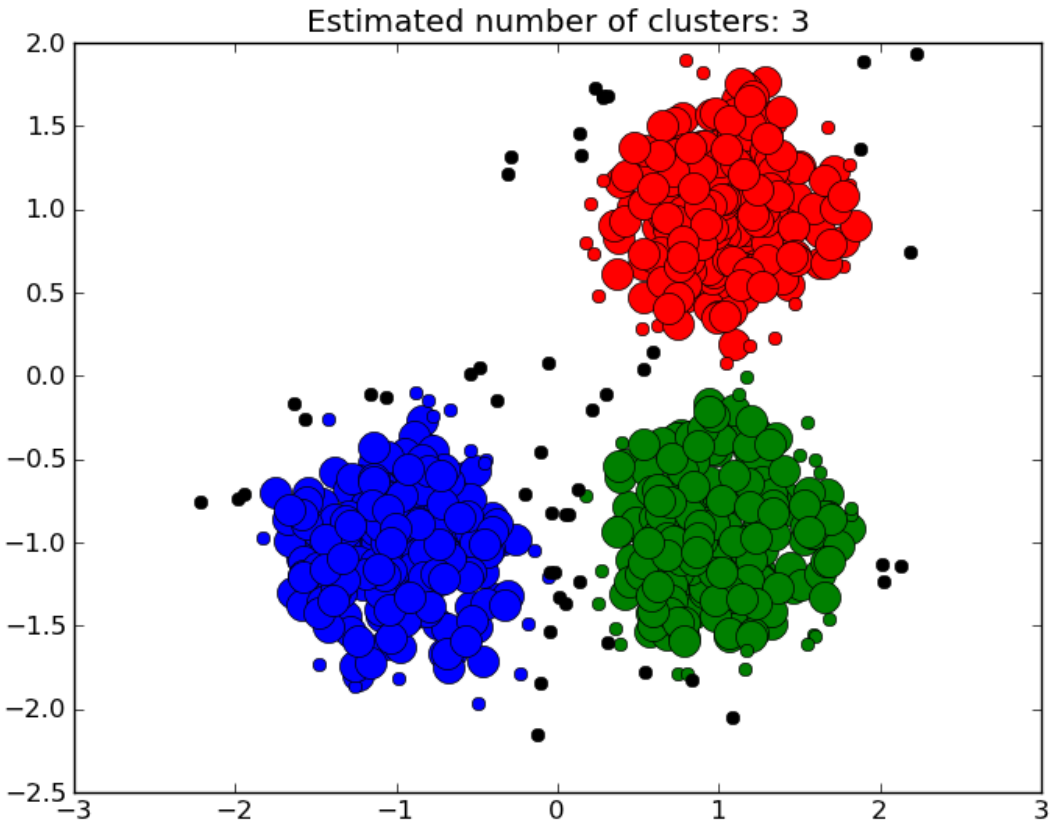


Fig.4: An Example of Clustering

- **Hidden Markov Model**

A **hidden Markov model** (**HMM**) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (*hidden*) states. An HMM can be presented as the simplest dynamic Bayesian network. The mathematics behind the HMM was developed by L. E. Baum and coworkers. It is closely related to an earlier work on the optimal nonlinear filtering problem by Ruslan L. Stratonovich, who was the first to describe the forward-backward procedure. In simpler Markov

models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a *hidden* Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a 'hidden' Markov model even if these parameters are known exactly. Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.

- **Singular Value Decomposition**

  In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix, with many useful applications in signal processing and statistics. Formally, the singular value decomposition of an $m \times n$ real or complex matrix M is a factorization of the form $M = U\Sigma V^*$, where U is an $m \times m$ real or complex unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V^* (the conjugate transpose of V, or simply the transpose of V if V is real) is an $n \times n$ real or complex unitary matrix. The diagonal entries $\Sigma_{i,i}$ of Σ are known as the singular values of M. The m columns of U and the n columns of V are called the left-singular vectors and right-singular vectors of M, respectively.

1.3 Data Mining

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Although data mining is a relatively new term, the technology is not. Companies have used powerful computers to sift through volumes of supermarket scanner data and analyze market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost.

What can data mining do?

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

For example, Blockbuster Entertainment mines its video rental history database to recommend rentals to individual customers. American Express can suggest products to its cardholders based on analysis of their monthly expenditures.

Data Mining Techniques

1.3.1 Classification

Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels. For example, we can build a classification model to categorize bank loan applications as either safe or risky. Such analysis can help provide us with a better understanding of the data at large. Many classification methods have been proposed by researchers in machine learning, pattern recognition, and statistics. Most algorithms are memory resident, typically assuming a small data size. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large amounts of disk-resident data. Classification has numerous applications, including fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.

Following are the examples of cases where the data analysis task is Classification −

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

1.3.2 Clustering

Clustering is the process of making a group of abstract objects into classes of similar objects. Clustering algorithms find groups of items that are similar. It divides a data set so that records with similar content are in the same group, and groups are as different as possible from each other.

Applications of Cluster Analysis

- Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.
- In the field of biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.

1.3.3 Regression

Regression is a data mining function that predicts a number. Profit, sales, mortgage rates, house values, square footage, temperature, or distance could all be predicted using regression techniques. For example, a regression model could be used to predict the value of a house based on location, number of rooms, lot size, and other factors.

Regression analysis seeks to determine the values of parameters for a function that cause the function to best fit a set of data observations that you provide. The following equation expresses these relationships in symbols. It shows that regression is the process of estimating the value of a continuous target (*y*) as a function (*F*) of one or more predictors ($\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$), a set of parameters (θ_1 , θ_2 , ..., θ_n), and a measure of error (*e*).

$$y = F(x,\theta) + e$$

The predictors can be understood as independent variables and the target as a dependent variable. The error, also called the **residual**, is the difference between the expected and predicted value of the dependent variable. The regression parameters are also known as **regression coefficients.**

1.3.4 Association

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository.

Association rules are created by analyzing data for frequent if/then patterns and using the criteria *support* and *confidence* to identify the most important relationships. *Support* is an indication of how frequently the items appear in the database. *Confidence* indicates the number of times the if/then statements have been found to be true.

In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, catalog design and store layout.

Programmers use association rules to build programs capable of machine learning. Machine learning is a type of artificial intelligence (AI) that seeks to build programs with the ability to become more efficient without being explicitly programmed.

CHAPTER 2

Related work through NLP

Natural language processing (NLP) is the ability of a computer program to understand human speech as it is spoken. NLP is a component of artificial intelligence (AI).The development of NLP applications is challenging because computers traditionally require humans to “speak” to them in a programming language that is precise, unambiguous and highly structured or, perhaps through a limited number of clearly-enunciated voice commands. Human speech, however, is not always precise -- it is often ambiguous and the linguistic structure can depend on many complex variables, including slang, regional dialects and social context.

Current approaches to NLP are based on machine learning, a type of artificial intelligence that examines and uses patterns in data to improve a program's own understanding. Most of the research being done on natural language processing revolves around search, especially enterprise search.

2.1 Major Tasks in NLP

- **Machine translation**

  Automatically translate text from one human language to another. This is one of the most difficult problems, and is a member of a class of problems colloquially termed "AI-complete", i.e. requiring all of the different types of knowledge that humans possess (grammar, semantics, facts about the real world, etc.) in order to solve properly.

- **Named entity recognition (NER)**

  Given a stream of text, determine which items in the text map to proper names, such as people or places, and what the type of each such name is (e.g. person, location, organization). Note that, although capitalization can aid in recognizing named entities in languages such as English, this information cannot aid in determining the type of named entity, and in any case is often inaccurate or insufficient. For example, the first word of a sentence is also capitalized, and named entities often span several words, only some of

which are capitalized. Furthermore, many other languages in non-Western scripts (e.g. Chinese or Arabic) do not have any capitalization at all, and even languages with capitalization may not consistently use it to distinguish names. For example, German capitalizes all nouns, regardless of whether they refer to names, and French and Spanish do not capitalize names that serve as adjectives.

- **Parsing**

  Determine the parse tree (grammatical analysis) of a given sentence. The grammar for natural languages is ambiguous and typical sentences have multiple possible analyses. In fact, perhaps surprisingly, for a typical sentence there may be thousands of potential parses (most of which will seem completely nonsensical to a human).

- **Sentiment analysis**

  Extract subjective information usually from a set of documents, often using online reviews to determine "polarity" about specific objects. It is especially useful for identifying trends of public opinion in the social media, for the purpose of marketing.

- **Information extraction (IE)**

  This is concerned in general with the extraction of semantic information from text. This covers tasks such as named entity recognition, Co-reference resolution, relationship extraction, etc.

2.2 Semantic and Non-semantic Models to predict stock price volatility

The semantic content of language conveys a wealth of information that typically is immediately understood by people due to its meaningful nature. At the same time, this information is often ignored in scientific studies due to lack of methods to quantify the semantic content. However, more recently, methods that allow quantification of meanings of words have been emerging. These methods utilize the empirical fact that text tends to keep to a certain semantic theme so that words within a certain context (i.e., sentence, paragraph, or document) are more likely to have a more similar meaning than those within other contexts. To be able to quantify this semantic content, it is necessary to have access to huge collections of text data, typically in the order of 100 MB or larger, where appropriate statistical methods are required for identifying the semantic representation.

Semantic spaces were early on proposed as a theory or model for how children acquire an understanding of the meanings of words. Semantic spaces have been used in a number of fields—assessing the quality of essays measuring context coherence; measuring values of social groups ; studying how object relations of mother, father, and self are influenced by long-term psychotherapy; studying semantic linguistic maturity in children and teenagers; disambiguating different meanings of *holy* in blogs etc. Here, we show how semantic space can be applied for studying and predicting stock price volatilities.

- **Latent Dirichlet Allocation**

Latent Dirichlet Allocation (LDA) assigns a discrete latent model to words and let each document maintain a random variable, indicating its probabilities of belonging to each topic LDA has mainly been used to model text corpora, where the notion of exchangeability corresponds to the "bag-of-words" assumption that is commonly employed in such models.LDA models each document as a mixture over topics, where each vector of mixture proportions is assumed to have been drawn from a Dirichlet distribution. A topic in this model is defined to be a discrete distribution over words from some finite lexicon.

For example, an LDA model might have topics that can be classified as CAT_related and DOG_related. A topic has probabilities of generating various words, such as *milk*, *meow*, and *kitten*, which can be classified and interpreted by the viewer as "CAT_related". Naturally, the word *cat* itself will have high probability given this topic. The DOG_related topic likewise has probabilities of generating each word: *puppy*, *bark*, and *bone* might have high probability. Words without special relevance, such as *the*, will have roughly even probability between classes (or can be placed into a separate category). A topic is not strongly defined, neither semantically. It is identified on the basis of supervised labeling and (manual) pruning on the basis of their likelihood of co-occurrence. A lexical word may occur in several topics with a different probability, however, with a different typical set of neighboring words in each topic.

- **Latent Semantic Analysis**

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text . The underlying idea is that the aggregate of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other. The adequacy of LSA's reflection of human knowledge has been established in a variety of ways. For example, its scores overlap those of humans on standard vocabulary and subject matter tests; it mimics human word sorting and category judgments; it simulates word–word and passage–word lexical priming data; and, it accurately estimates passage coherence, learnability of passages by individual students, and the quality and quantity of knowledge contained in an essay.

- ## Latent Semantic Indexing

Latent semantic indexing (**LSI**) is an indexing and retrieval method that uses a mathematical technique called singular value decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts. LSI is also an application of correspondence analysis, a multivariate statistical technique developed by Jean-Paul Benzécri in the early 1970s, to a contingency table built from word counts in documents called Latent Semantic Indexing because of its ability to correlate semantically related terms that are latent in a collection of text, it was first applied to text at Bellcore in the late 1980s.

- ## Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Indexing is a novel approach to automated document indexing which is based on a statistical latent class model for factor analysis of count data. Fitted from a training corpus of text documents by a generalization of the Expectation Maximization algorithm, the utilized model is able to deal with domain specific synonyms well as with polysemous words. In contrast to standard Latent Semantic Indexing (LSI) by Singular Value Decomposition, the probabilistic variant has a solid statistical foundation and defines a proper generative data model. Retrieval experiments on a number of test collections indicate substantial performance gains over direct term matching methods as well as over LSI. In particular, the combination of models with different dimensionalities has proven to be advantageous.

2.3Stock Price Prediction based on Textual Analysis of Financial News Articles

Nowadays, a huge amount of valuable information related to the financial market is available on the web. A majority of this information comes from Financial News Articles, Company Reports and Expert Recommendations (Blogs from valid sources can also act as a source of information) Most of this data is in a textual format as opposed to a numerical format which makes it hard to use. Thus the problem domain can now be viewed as one that involves Mining of Text Documents and Time Series Analysis concurrently.

One method which is generally used involves defining the news impact on a particular stock: Positive, Negative, and Neutral.

A news is considered to have a positive impact (or negative impact) if the stock price rises (or drops) significantly for a period, after the news story has been broadcasted. If the stock price does not change dramatically after the news is released, then the news story is regarded as neutral.Another method which we study in this paper relates to detecting and determining atterns in the news articles which correspond directly to a rise or fall in the stock price. The general architecture is as follows:

A crawler continuously crawls news articles and indexes them for a particular stock portfolio. The learning environment requests the news since the last T minutes from the indexer. The learning environment consists of several base learners which look for specific information in the text document (i.e. patterns like “profits rise” inside a just released news article, or “share prices will go down” on the blog of a veteran Wall Street Trader/Speculator etc.). A Bag-Of-Words consisting of Positive Prediction Terms and Negative Prediction Terms and Phrases is used by the learning environment. Each time a word/phrase from the Positive Prediction Term occurs in a particular news article, a PostiveVote is assigned to the article.

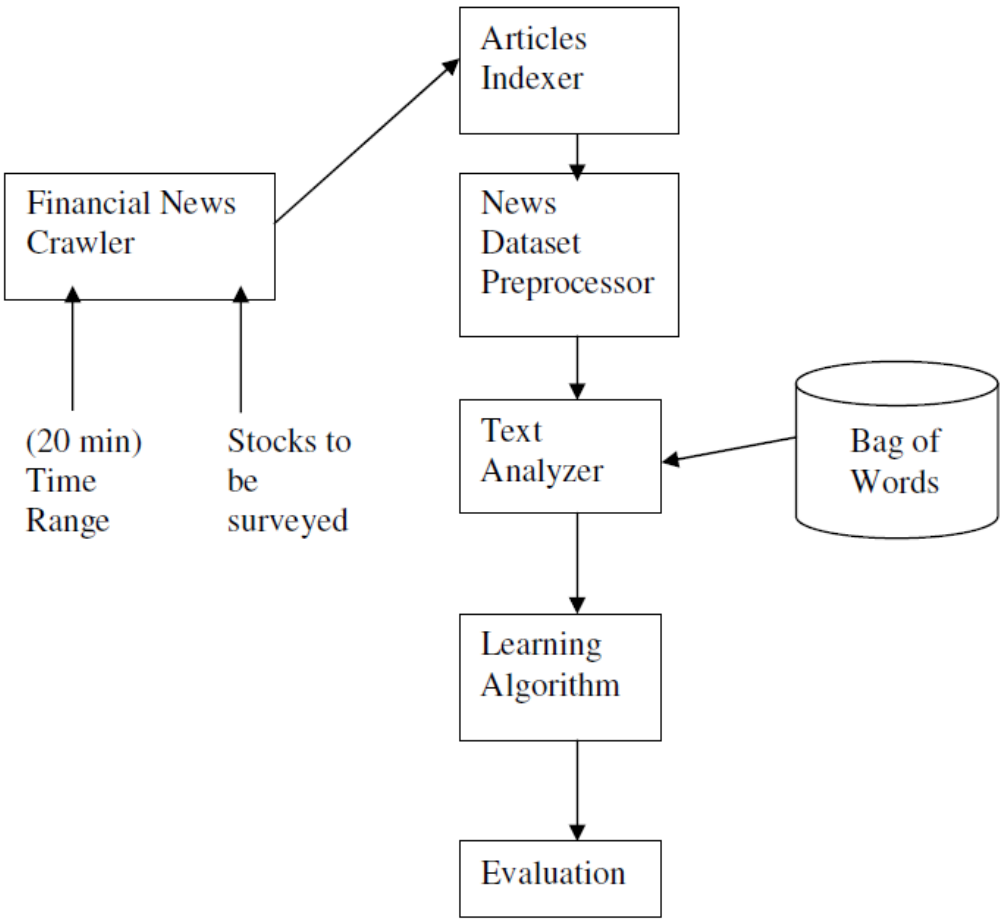


Fig:2.1 General Architecture of Semantic based system

CHAPTER 3

Literature Survey

3.1 Research Papers

Title of Paper	Textual Analysis of Stock Market Prediction Using Financial News Articles
Author	Robert P Schumaker, Hsinchun Chen
Year	2006
Summary	This paper examined the role of financial news articles on three different textual representations; Bag of Words, Noun Phrases, and Named Entities and their ability to predict stock prices twenty minutes after an article release. Using a Support Vector Machine (SVM) derivative, they showed that our model had a statistically significant impact on predicting future stock prices compared to linear regression
Publishing Details	Most (2006) Volume: Acapulco,, Publisher: Citeseer, Pages: 1-20
Weblink	http://ailab.arizona.edu/intranet/papers/Textual%20Analysis%20of%20Stock%20Market.pdf

Title of Paper	An Efficient Approach to Forecast Indian Stock Market Price and their Performance Analysis
Author	K.K.Sureshkumar, Dr.N.M.Elango
Year	2011
Publishing Details	International Journal of Computer Applications (0975 – 8887) Volume 34– No.5, November 2011
Summary	This paper examined and applied different neural classifier functions by using the Weka tool. By using correlation coefficient compared various prediction functions, and found that Isotonic regression function offer the ability to predict the stock price of NSE more accurately than the other functions such as Gaussian processes, least mean square, linear regression, multilayer perceptron, pace regression, simple linear regression and SMO regression.
Weblink	http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.2508&rep=rep1&type=pdf

Title of Paper	Time Series Forecasting Of Nifty Stock Market Using Weka
Author	Raj Kumar, Anil Balara
Year	2014
Publishing Details	JRPS International Journal for Research Publication & Seminar Vol 05 Issue 02 March -July 2014
Summary	In this research, they examined and applied different forecasting techniques by using the Weka tool. Also, compared various prediction functions, and found that SMO regression function offer the ability to predict the stock price of NSE more accurately than the other functions such as Gaussian processes, linear regression, multilayer perceptron. This analysis can be used to reduce the error percentage in predicting the future stock prices
Weblink	http://jrps.in/uploads/March%202014/research_paper_raj.pdf

3.2 Literature Summary

For prediction of Stock Prices generally, there are two analytical models. They are fundamental analysis and technical analysis. Fundamental analysis includes economic analysis, industry analysis and company analysis. Technical analysis is a method for predicting future price based on the past market data. Prediction is made by exploiting implications hidden in past trading activities and by analyzing patterns and trends shown in price and volume charts.

Different Algorithms can be used for studying the complex relationships between the input and output variables in the system.

For the task of predicting the stock price of a company from today to tomorrow by using the past values of the company stock. Waikato Environment for Knowledge Analysis version 3.6.3 (Weka 3.6.3) tool can be applied to preprocess the stock previous close price, open price, high price and low price data.

There are numerous methods to measure the performance of systems. In which, Root mean squared error and relative absolute error are very common in literature. In order to evaluate the net performance of the stock value four different indicators can be calculated. The indicators are mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE) and root relative squared error (RRSE). These indicators are used to evaluate the error rate of the stock price.

CHAPTER 4

Predicting Stock Price Volatility in this project

Recently, a lot of interesting work has been done in the area of applying Machine Learning Algorithms for analyzing price patterns and predicting stock prices and index changes. Most stock traders nowadays depend on Intelligent Trading Systems which help them in predicting prices based on various situations and conditions, thereby helping them in making instantaneous investment decisions.

Stock Prices are considered to be very dynamic and susceptible to quick changes because of the underlying nature of the financial domain and in part because of the mix of known parameters (Previous Days Closing Price, P/E Ratio etc.)

An intelligent trader would predict the stock price and buy a stock before the price rises, or sell it before its value declines. Though it is very hard to replace the expertise that an experienced trader has gained, an accurate prediction algorithm can directly result into high profits for investment firms, indicating a direct relationship between the accuracy of the prediction algorithm and the profit made from using the algorithm.

4.1 The dataset

News media is perhaps the best source of daily information about stocks. This source of information also has the advantage of being highly accessible and having a specific time mark associated, namely the date of publication.

For this experiment, the historical data was downloaded from the yahoo finance section. In particular, the stock prices of two companies were studied, namely Google Inc. (GOOG) and Yahoo Inc. (YHOO)

The dataset available has the following attributes:

Date Open High Low Close Volume Adj. Close

Definition of these attributes

Opening Price:

The price at which a security first trades upon the opening of an exchange on a given trading day. A security's opening price is an important marker for that day's trading activity, especially for those interested in measuring short-term results, such as day traders. Additionally, securities, which experience very large intra-day gains and losses, will have those swings measured relative to their opening price for the day.

Quite commonly, a security's opening price will not be identical to its closing price. This is due to after-hours trading and to changes in investor valuations or expectations of the security occurring outside of trading hours. Nasdaq uses an approach called the "opening cross," to decide on a price level that would serve as the best opening price, given the orders that have accumulated overnight.

Closing Price:

The final price at which a security is traded on a given trading day. The closing price represents the most up-to-date valuation of a security until trading commences again on the next trading day.

Most financial instruments are traded after hours (although with markedly smaller volume and liquidity levels), so the closing price of a security may not match its after-hours price. Still, closing prices provide a useful marker for investors to assess changes in stock prices over time - the closing price of one day can be compared to the previous closing price in order to measure market sentiment for a given security over a trading day.

High:

A security's intraday high trading price. Today's high is the highest price at which a stock traded during the course of the day. Today's high is typically higher than the closing or opening price. More often than not this is higher than the closing price. When you look at a stock quote, you can find today's high by looking at the second number listed next to "Range." One way that day traders and technical analysts use today's high, along with today's low, is to help them identify gaps or sudden jumps up or down in a stock's price with no trading in between those two prices. For example, if today's low is $25 and the previous day's high is $20, there is gap. The identification of a

gap, along with other market signals such as changes in trading volume and overall bullish or bearish sentiment, helps market analysts generate buy and sell signals for particular stocks.

Low:

A security's intraday low trading price. Today's low is the lowest price at which a stock trades over the course of a trading day. Today's low is typically lower than the opening or closing price.

When you look at a stock quote, you can find today's low by looking at the first number listed next to "Range." Today's low and today's high are important to day traders and technical analysts, who seek to earn profits from a security's short-term price movements and identify and track trends. One way that day traders use today's low along with today's high is to identify gaps, or sudden jumps up or down in a stock's price with no trading in between. Gaps are used in technical analysis to identify directional movement, average true range/price volatility, candlestick patterns and more. Traders then analyze these patterns to determine profitable entry and exit points.

Adjusted Closing Price

A stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open. The adjusted closing price is often used when examining historical returns or performing a detailed analysis on historical returns.

The adjusted closing price is a useful tool when examining historical returns because it gives analysts an accurate representation of the firm's equity value beyond the simple market price. It accounts for all corporate actions such as stock splits, dividends/distributions and rights offerings.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Volume	Adj Close
2	13-03-15	124.4	125.4	122.58	123.59	51675300	123.59
3	12-03-15	122.31	124.9	121.63	124.45	48145700	124.45
4	11-03-15	124.75	124.77	122.11	122.24	68582700	122.24
5	10-03-15	126.41	127.22	123.8	124.51	68240700	124.51
6	09-03-15	127.96	129.57	125.06	127.14	88347500	127.14
7	06-03-15	128.4	129.37	126.26	126.6	72225800	126.6
8	05-03-15	128.58	128.75	125.76	126.41	56004600	126.41
9	04-03-15	129.1	129.56	128.32	128.54	30977700	128.54
10	03-03-15	128.96	129.52	128.09	129.36	37816300	129.36
11	02-03-15	129.25	130.28	128.3	129.09	48096700	129.09
12	27-02-15	130	130.57	128.24	128.46	62014800	128.46
13	26-02-15	128.79	130.87	126.61	130.42	91287500	130.42
14	25-02-15	131.56	131.6	128.15	128.79	74711700	128.79
15	24-02-15	132.94	133.6	131.17	132.17	69228100	132.17
16	23-02-15	130.02	133	129.66	133	70974100	133
17	20-02-15	128.62	129.5	128.05	129.5	48948400	129.5
18	19-02-15	128.48	129.03	128.33	128.45	37362400	128.45
19	18-02-15	127.63	128.78	127.45	128.72	44891700	128.72
20	17-02-15	127.49	128.88	126.92	127.83	63152400	127.83
21	13-02-15	127.28	127.28	125.65	127.08	54272200	127.08
22	12-02-15	126.06	127.48	125.57	126.46	74474500	126.46
23	11-02-15	122.77	124.92	122.5	124.88	73561800	124.88
24	10-02-15	120.17	122.15	120.16	122.02	62008500	122.02
25	09-02-15	118.55	119.84	118.43	119.72	38889800	119.72
26	06-02-15	120.02	120.25	118.45	118.93	43372000	118.93

Fig 4.1 Sample of raw data

News outlets can be differentiated from wire services in several different ways. One of the main differences is that news outlets are centers that publish available financial information at specific time intervals. Examples include Bloomberg, Business Wire, CNN Financial News, Dow Jones, Financial Times, Forbes, Reuters, and the Wall Street Journal. In contrast, news wire services publish available financial information as soon as it is publicly released or discovered. News wire examples include PRNewsWire, which has free and subscription levels for real-time financial news access, and Yahoo Finance, which is a compilation of 45 news wire services including the Associated Press and PRNewsWire. Besides their relevant and timely release of financial news articles, news

wire articles are also easy to automatically gather and are an excellent source for computer-based algorithms. Stock Quotations are an important source of financial information. Quotes can be divided into various increments of time from minutes to days, however, one minute increments provide sufficient granularity for machine learning.

4.2 Data Pre-Processing

Intuitively, based on the EMH, the price of the stock yesterday is going to have the most impact on the price of the stock today. Thus as we go along the time-line, data-points which are nearer to today's price point are going to have a greater impact on today's price. For a time-series analysis we can take the Date as the X-Axis with integer values attached to each date, such that the most recent Date Tag in the dataset gets the highest value and the oldest Date Tag gets the lowest value.

We add one more attribute to the above attributes, this attribute will act as our label for predicting the movements of the stock price. This attribute will be called "Indicator" and will be dependent on the other available attributes. For our experiments we use the EMA (Exponential Moving Average) as the indicator function.

Indicator Functions

We now take a brief look at the attributes and indicators that are normally used in the technical analysis of stock prices:

Indicators can be any of the following:

Moving Average (MA):

The average of the past n values till today.

Exponential Moving Average (EMA):

Gives more weightage to the most recent values while not discarding the older observation entirely.

Rate of Change (ROC):

The ratio of the current price to the price n quotes earlier. n is generally 5 to 10 days.

Chaikin Money Flow Indicator:

Chaikin's money flow is based on Chaikin's accumulation/distribution. Accumulation/ distribution in turn, is based on the premise that if the stock closes above its midpoint [(high+low)/2] for the day, then there was accumulation that day, and if it closes below its midpoint, then there was distribution that day. Chaikin's money flow is calculated by summing the values of accumulation/distribution for 13 periods and then dividing by the 13-period sum of the volume.

$$\text{CMI} = [\frac{sum(AD,n)}{sum(vol,n)}]$$

$$\text{AD} = \text{vol}[\frac{(CL-OP)}{(HI-LO)}]$$

Relative Strength Index (RSI):

Measures the relative size of recent upward trends against the size of downward trends within the specified time interval (usually 9 – 14 days).

For this Project, the EMA was considered as the primary indicator because of its ability to handle an almost infinite amount of past data, a trait that is very valuable in time series prediction (It is worth noting that the application of other indicators might result in better prediction accuracies for the stocks under consideration).

EMA (t) = EMA (t-1) + alpha * (Price (t) - EMA (t-1))

Where, alpha = 2/ (N+1), Thus, for N=9, alpha = 0.20

In theory, the Stock Prediction Problem can be considered as evaluating a function F at time T based on the previous values of F at times t-1,t-2,t-n while assigning corresponding weight function w at each point to F.

*F (t) = w1*F (t-1) + w2*F (t-2) + … + w*F (t-n)*

	A	B
1	day	ema
2	1055	377.8674
3	1054	378.3499
4	1053	378.8321
5	1052	379.32
6	1051	379.8048
7	1050	380.2861
8	1049	380.7698
9	1048	381.2552
10	1047	381.7379
11	1046	382.2205
12	1045	382.7049
13	1044	383.192
14	1043	383.6767
15	1042	384.1659
16	1041	384.6501
17	1040	385.134
18	1039	385.6261
19	1038	386.1216
20	1037	386.618
21	1036	387.1176

Fig 4.2 Sample of data after pre-processing

4.3 Use of WEKA

One we have learned a model, it can be used to classify new unseen data. These notes describe the process of doing some both graphically and from the command line. First, the file with cases to predict needs to have the same structure that the file used to learn the model. The difference is that the value of the class attribute is "?" for all instances.

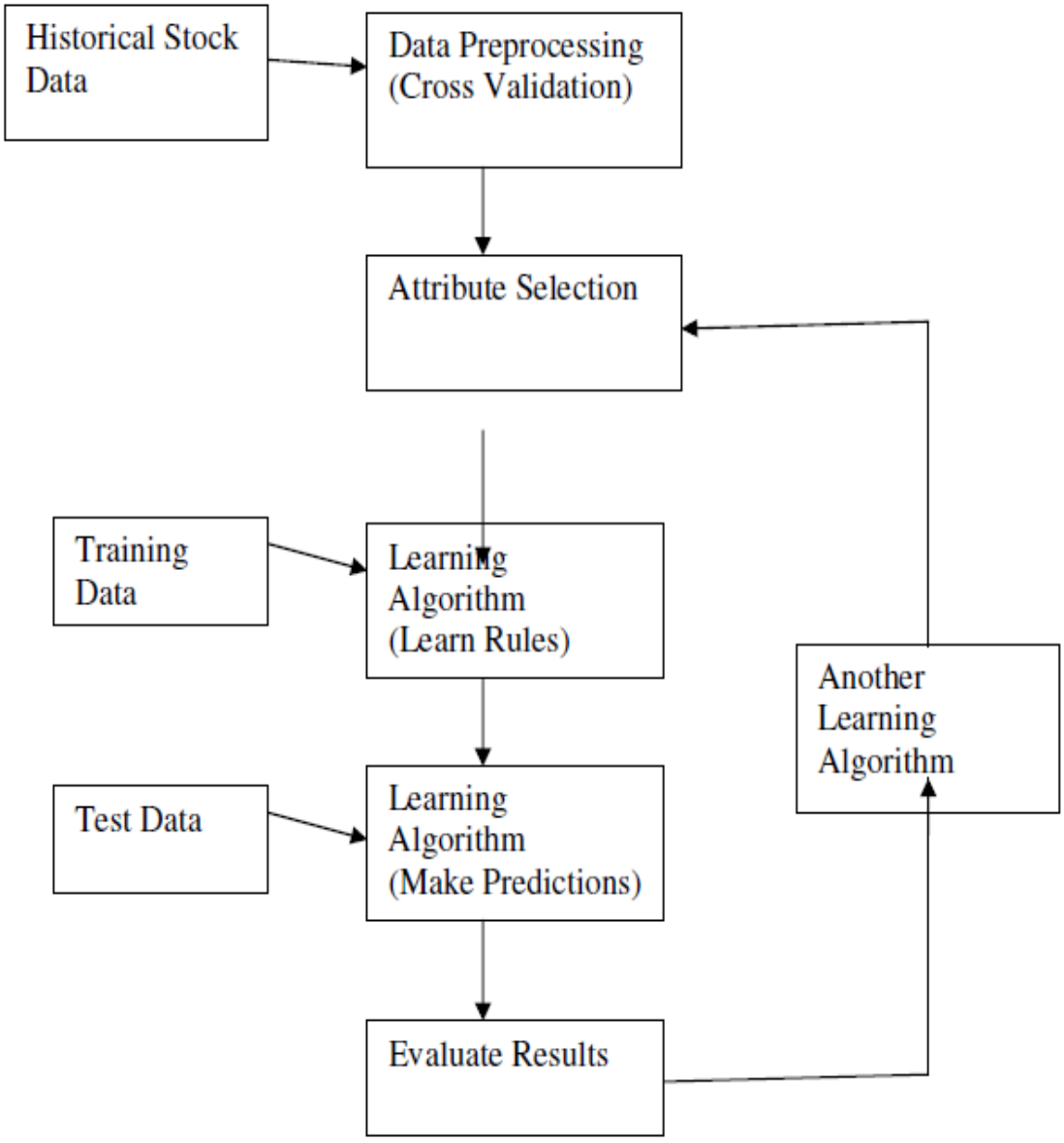


Fig. 4.1 Flow-Chart

Fig 4.2 Activity Diagram

4.4 Classifying Algorithms

Classification consists of predicting a certain outcome based on a given input. In order to predict the outcome, the algorithm processes a training set containing a set of attributes and the respective outcome, usually called goal or prediction attribute. The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Next the algorithm is given a data set not seen before, called prediction set, which contains the same set of attributes, except for the prediction attribute – not yet known. The algorithm analyses the input and produces a prediction. The prediction accuracy defines how "good" the algorithm is. For example, in a medical database the training set would have relevant patient information recorded previously, where the prediction attribute is whether or not the patient had a heart problem. Table 1 below illustrates the training and prediction sets of such database.

Training set

Age	Heart rate	Blood pressure	Heart problem
65	78	150/70	Yes
37	83	112/76	No
71	67	108/65	No

Prediction set

Age	Heart rate	Blood pressure	Heart problem
43	98	147/89	?
65	58	106/63	?
84	77	150/65	?

TABLE 1 – TRAINING AND PREDICTION SETS FOR MEDICAL DATABASE

Among several types of knowledge representation present in the literature, classification normally uses prediction rules to express knowledge. Prediction rules are expressed in the form of IF-THEN rules, where the antecedent (IF part) consists of a conjunction of conditions and the rule consequent (THEN part) predicts a certain predictions attribute value for an item that satisfies the antecedent. Using the example above, a rule predicting the first row in the training set may be represented as following: IF (Age=65 AND Heart rate>70) OR (Age>60 AND Blood pressure>140/70) THEN Heart problem=yes In most cases the prediction rule is immensely larger than the example above. Conjunction has a

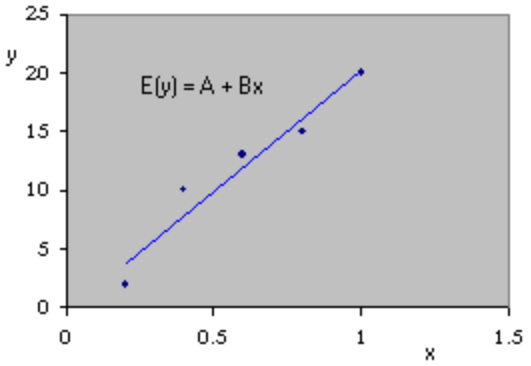
nice property for classification; each condition separated by OR's defines smaller rules that captures relations between attributes. Satisfying any of these smaller rules means that the consequent is the prediction. Each smaller rule is formed with AND's which facilitates narrowing down relations between attributes. How well predictions are done is measured in percentage of predictions hit against the total number of predictions. A decent rule ought to have a hit rate greater than the occurrence of the prediction attribute. In other words, if the algorithm is trying to predict rain in Seattle and it rains 80% of the time, the algorithm could easily have a hit rate of 80% by just predicting rain all the time. Therefore, 80% is the base prediction rate that any algorithm should achieve in this case. The optimal solution is a rule with 100% prediction hit rate, which is very hard, when not impossible, to achieve. Therefore, except for some very specific problems, classification by definition can only be solved by approximation algorithms.

4.4.1 Simple Linear Regression (SLR)

Simple linear regression is a method that enables you to determine the relationship between a continuous process output (Y) and one factor (X). The relationship is typically expressed in terms of a mathematical equation such as $Y = b + mX$.

Suppose we believe that the value of y tends to increase or decrease in a linear manner as x increases. Then we could select a model relating y to x by drawing a line which is well fitted to a given data set. Such a deterministic model – one that does not allow for errors of prediction –might be adequate if all of the data points fell on the fitted line. However, you can see that this idealistic situation will not occur for the data of Table 11.1 and 11.2. No matter how you draw a line through the points in Figure 11.2 and Figure 11.3, at least some of points will deviate substantially from the fitted line.

The solution to the proceeding problem is to construct a probabilistic model relating y to x- one that knowledge the random variation of the data points about a line. One type of probabilistic model, a simple linear regression model, makes assumption that the mean value of y for a given value of x graphs as straight line and that points deviate about this line of means by a random amount equal to e, i.e.



y = A + B x + e,

where A and B are unknown parameters of the deterministic (nonrandom) portion of the model.

If we suppose that the points deviate above or below the line of means and with expected value E(e) = 0 then the mean value of y is

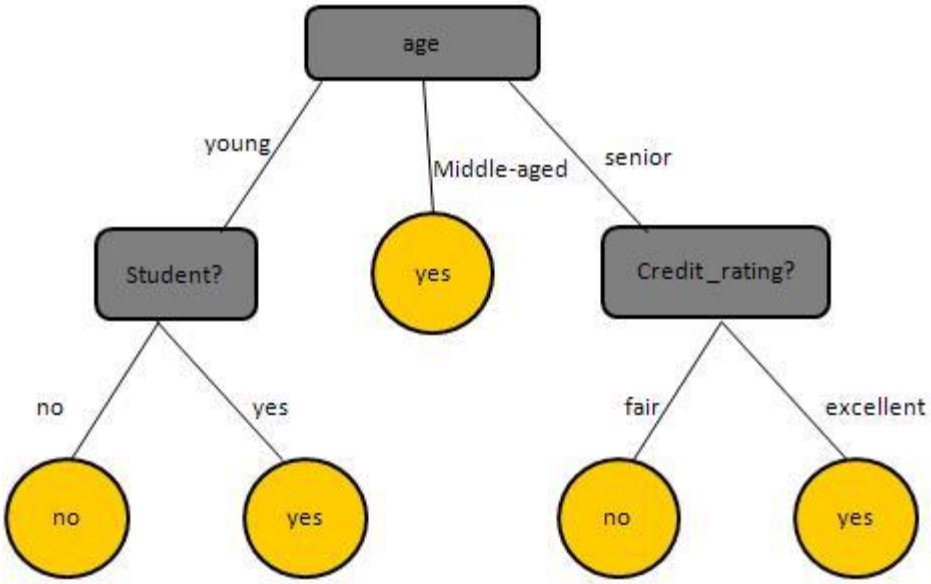
y = A + B x.

Therefore, the mean value of y for a given value of x, represented by the symbol E(y) graphs as straight line with y-intercept A and slope B.

4.4.2 Decision Table

The decision tree is a structure that includes root node, branch and leaf node. Each internal node denotes a test on attribute, each branch denotes the outcome of test and each leaf node holds the class label. The topmost node in the tree is the root node.

The following decision tree is for concept buy_computer, that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents the test on the attribute. Each leaf node represents a class.



4.4.3 Decision Stump

Decision stumps are basically decision trees with a single layer. As opposed to a tree which has multiple layers, a stump basically stops after the first split. Decision stumps are usually used in population segmentation for large data. Occasionally, they are also used to help make simple yes/no decision model for smaller data with little data. Decision stumps are generally easier to build as compared to decision tree. At the same time, SAS® coding for decision stumps are more manageable compared to CART and CHAID. The reason being that the decision stumps is just one single run of the tree algorithm and thus does not need to prepare data for the subsequent splits. At the same time, there is no need to specify the data for the subsequent split which make renaming of output simpler to manage.

4.4.4 M5P

M5P [WAN97] is a reconstruction of Quinlan's M5 algorithm [QUI92] for inducing trees of regression models. M5P combines a conventional decision tree with the possibility of linear regression functions at the nodes. First, a decision-tree induction algorithm is used to build a tree, but instead of maximizing the information gain at each inner node, a splitting criterion is used that minimizes the intra-subset variation in the class values down each branch. The splitting procedure in M5P stops if the class values of all instances that reach a node vary very slightly, or only a few instances remain. Second, the tree is pruned back from each leaf. When pruning an inner node is turned into a leaf with a regression plane. Third, to avoid sharp discontinuities between the subtrees a smoothing procedure is applied that combines the leaf model prediction with each node along the path back to the root, smoothing it at each of these nodes by combining it with the value predicted by the linear model for that node. Techniques devised by Breiman et al. [BRE84] for their CART system are adapted in order to deal with enumerated attributes and missing values. All enumerated attributes are turned into binary variables so that all splits in M5P are binary. As to missing values, M5P uses a technique called "surrogate splitting" that finds another attribute to split on in place of the original one and uses it instead. During training, M5P uses as surrogate attribute the class value in the belief that this is the attribute most likely to be correlated with the one used for splitting. When the splitting procedure ends all missing values are replaced by the average values of the corresponding attributes of the training examples reaching the leaves. During testing an unknown attribute value is replaced by the average value of that attribute for all training instances that reach the node, with the effect of choosing always the most populous sub-node. M5P generates models that are compact and relatively comprehensible.

4.5 Evaluation

We use a number of measures to evaluate the predictive power of our suggested approach relative to the rival models. In all these measures, we compare the volatility prediction of a specific model with the realized volatility. Realized volatility for week *t* is defined as the standard deviation of the daily return in that week.

Correlation Coefficient

Pearson's correlation coefficient between two variables is defined as the covariance of the two variables divided by the product of their standard deviations. The form of the definition involves a "product moment", that is, the mean (the first moment about the origin) of the product of the mean-adjusted random variables; hence the modifier *product-moment* in the name.

The formula for ρ is:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

where:

- cov is the covariance
- σ_X is the standard deviation of X

The formula for ρ can be expressed in terms of mean and expectation. Since

- $\operatorname{cov}(X,Y) = E[(X - \mu_X)(Y - \mu_Y)]$

Mean Absolute Error

In statistics, the **mean absolute error (MAE)** is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i| .$$

As the name suggests, the mean absolute error is an average of the absolute errors $|e_i| = |f_i - y_i|$, where f_i is the prediction and y_i the true value. Note that alternative formulations may include relative frequencies as weight factors.

The mean absolute error is a common measure of forecast error in time series analysis.

Root Mean squared Error

The RMS value of a set of values (or a continuous-time waveform) is the square root of the arithmetic mean of the squares of the values, or the square of the function that defines the continuous waveform.

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_n^2\right)}.$$

Relative Absolute Error

The **relative absolute error** is very similar to the relative squared error in the sense that it is also relative to a simple predictor, which is just the average of the actual values. In this case, though, the error is just the total absolute error instead of the total squared error. Thus, the relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor.

Mathematically, the **relative absolute error** E_i of an individual program *i* is evaluated by the equation:

$$\frac{|\,p_1 - a_1\,| + \ldots + |\,p_n - a_n\,|}{|\,\bar{a} - a_1\,| + \ldots + |\,\bar{a} - a_n\,|}$$

Root Relative Squared Error

The **root relative squared error** is relative to what it would have been if a simple predictor had been used. More specifically, this simple predictor is just the average of the actual values. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the simple predictor. By taking the square root of the relative squared error one reduces the error to the same dimensions as the quantity being predicted.

Mathematically, the **root relative squared error** E_i of an individual program i is evaluated by the equation:

$$\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(\overline{a} - a_1)^2 + \ldots + (\overline{a} - a_n)^2}$$

4.6 Analysis and Results

4.6.1 Analysis on Raw Data

- **Simple Linear Regression**

```
Time taken to build model: 0 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    123.25     306.715    183.465
     2    124.24     306.912    182.672
     3    123.38     307.109    183.729
     4    126.69     307.306    180.616
     5    127.21     307.503    180.293
     6    125.9      307.699    181.799
     7    127.5      307.896    180.396
     8    128.47     308.093    179.623
     9    127.04     308.29     181.25
    10    124.95     308.487    183.537

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  0.6196
Mean absolute error                    181.738
Root mean squared error                181.7438
Relative absolute error                 63.4125 %
Root relative squared error             63.4133 %
Total Number of Instances               10
```

- **Decision Stump**

```
Time taken to build model: 0 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    183.25     106.616    -76.634
     2    184.24     106.616    -77.624
     3    183.38     106.616    -76.764
     4    186.69     106.616    -80.074
     5    187.21     106.616    -80.594
     6    185.9      106.616    -79.284
     7    187.5      106.616    -80.884
     8    188.47     106.616    -81.854
     9    187.04     106.616    -80.424
    10    184.95     106.616    -78.334

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  0
Mean absolute error                 79.247
Root mean squared error             79.2659
Relative absolute error             34.9727 %
Root relative squared error         34.98   %
Total Number of Instances           10
```

- **Decision Table**

```
=== Predictions on test set ===

 inst#     actual  predicted      error
     1    183.25     114.374    -68.876
     2    184.24     114.374    -69.866
     3    183.38     114.374    -69.006
     4    186.69     114.374    -72.316
     5    187.21     114.374    -72.836
     6    185.9      114.374    -71.526
     7    187.5      114.374    -73.126
     8    188.47     114.374    -74.096
     9    187.04     114.374    -72.666
    10    184.95     114.374    -70.576

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  0
Mean absolute error                 71.489
Root mean squared error             71.5099
Relative absolute error             31.549  %
Root relative squared error         31.5573 %
Total Number of Instances           10
```

- **M5P**

```
Time taken to build model: 0.27 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    183.25     131.172    -52.078
     2    184.24     131.101    -53.139
     3    183.38     131.03     -52.35
     4    186.69     130.959    -55.731
     5    187.21     130.889    -56.321
     6    185.9      130.818    -55.082
     7    187.5      130.747    -56.753
     8    188.47     130.676    -57.794
     9    187.04     130.605    -56.435
    10    184.95     130.534    -54.416

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  -0.6196
Mean absolute error                      55.0098
Root mean squared error                  55.0413
Relative absolute error                  24.2766 %
Root relative squared error              24.2897 %
Total Number of Instances                10
```

4.6.2Analysis on Pre-Processed Data

- **Simple Linear Regression**

```
Time taken to build model: 0 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    373.629    512.341    138.711
     2    374.1      512.218    138.118
     3    374.57     512.096    137.526
     4    375.043    511.973    136.93
     5    375.511    511.851    136.34
     6    375.98     511.729    135.749
     7    376.452    511.606    135.154
     8    376.923    511.484    134.561
     9    377.393    511.362    133.969
    10    377.867    511.239    133.372

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                 -1
Mean absolute error                136.043
Root mean squared error            136.0537
Relative absolute error            191.962  %
Root relative squared error        191.9421 %
Total Number of Instances           10
```

- **Decision Stump**

```
Time taken to build model: 0 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    373.629    483.514    109.885
     2    374.1      483.514    109.414
     3    374.57     483.514    108.944
     4    375.043    483.514    108.471
     5    375.511    483.514    108.003
     6    375.98     483.514    107.534
     7    376.452    483.514    107.062
     8    376.923    483.514    106.592
     9    377.393    483.514    106.121
    10    377.867    483.514    105.647

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  0
Mean absolute error                107.7674
Root mean squared error            107.7759
Relative absolute error            152.064  %
Root relative squared error        152.0483 %
Total Number of Instances               10
```

- **Decision Table**

```
Time taken to build model: 0.01 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    373.629    406.548     32.919
     2    374.1      406.548     32.448
     3    374.57     406.548     31.978
     4    375.043    406.548     31.505
     5    375.511    406.548     31.037
     6    375.98     406.548     30.568
     7    376.452    406.548     30.096
     8    376.923    406.548     29.625
     9    377.393    406.548     29.155
    10    377.867    406.548     28.681

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  0
Mean absolute error                 30.8013
Root mean squared error             30.8309
Relative absolute error             43.4618 %
Root relative squared error         43.4957 %
Total Number of Instances           10
```

- **M5P**

```
Time taken to build model: 0.13 seconds

=== Predictions on test set ===

 inst#     actual  predicted      error
     1    373.629    381.248      7.619
     2    374.1      381.735      7.635
     3    374.57     382.222      7.651
     4    375.043    382.708      7.665
     5    375.511    383.195      7.684
     6    375.98     383.682      7.702
     7    376.452    384.169      7.716
     8    376.923    384.655      7.733
     9    377.393    385.142      7.749
    10    377.867    385.629      7.762

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient                  1
Mean absolute error                      7.6916
Root mean squared error                  7.6918
Relative absolute error                 10.8532 %
Root relative squared error             10.8514 %
Total Number of Instances               10
```

DataSet	Algorithm Used	Error	Test Instances
Raw Data	Simple Linear Regression	183%	10
	Decision Stump	80%	10
	Decision Table	70%	10
	M5P	52%	10
Pre-Processed Data	Simple Linear Regression	138%	10
	Decision Stump	109%	10
	Decision Table	32%	10
	M5P	7.6%	10

Table 1. Comparison Analysis of Various Algorithms

CHAPTER 5

Conclusion

The prices in the financial market reflect the inflow of information via media. However, this information is difficult to quantify and statistically measure. We develop an automatic method for measuring and tracking the effect of company information in media on the company's stock price volatility. The suggested method for volatility forecasting is based on the semantic content of information in media text. Using a larger dataset would help offset any market biases that are associated with using a compressed period of time, such as the effects of cyclic stocks, earnings reports, mergers and other unexpected surprises.

The results of the project shows a comparison between the various data-mining algorithms i.e. M5P, Decision Stump, Decision Tree, Simple Linear Regression with the raw data and pre-processed data.

Out of all experiments carried out M5P shows the best results with the pre-processed data set.

CHAPTER 6

Future Scope

6.1Money Management

The money management strategy is responsible for determining the amount of money to invest on a given trade signal. A simple approach to this problem would be to always invest a fixed fraction of available investment capital, a fixed sum, or a fixed number of shares on every trade signal.

6.2Portfolio Simulations

Portfolio simulations are executed by simulating buy and sell transactions based trade signals generated by some prediction model and an amount of initial investment capital. The amount of stock to buy or sell is either defined by a fixed order total or determined by the money management strategy.

CHAPTER 7

References

[1] Stock Quotes Information Available, [online]

http://www.finance.yahoo.com, (Last Accessed: May 5 ,2015)

[2] Data Mining Softaware in JAVA

http://www.cs.waikato.ac.nz/~ml/weka/, (Last Accessed: Nov,2014)

[3] Raj Kumar, Anil Balara(2014). Time Series Forecasting Of Nifty Stock Market Using Weka:JRPS International Journal for Research Publication & Seminar Vol 05 Issue 02 March -July 2014

[4] K.K.Sureshkumar, Dr.N.M.Elango(2011). An Efficient Approach to Forecast Indian Stock Market Price and their Performance Analysis:International Journal of Computer Applications (0975 – 8887) Volume 34– No.5, November 2011

[5] Robert P Schumaker, Hsinchun Chen(2006). Textual Analysis of Stock Market Prediction Using Financial News Articles:Most (2006)
Volume: Acapulco, Publisher: Citeseer, Pages: 1-20

Appendix

Net Beans

NetBeans is a software development platform written in Java. NetBeans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The NetBeans Team actively support the product and seek feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactorings, version control (supporting CVS, Subversion, Git, Mercurial and Clearcase).

Modularity: All the functions of the IDE are provided by modules. Each module provides a well defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

License: From July 2006 through 2007, NetBeans IDE was licensed under Sun's Common Development and Distribution License (CDDL), a license based on the Mozilla Public License (MPL). In October 2007, Sun announced that NetBeans

would henceforth be offered under a dual license of the CDDL and the GPL version 2 licenses, with the GPL linking exception for GNU Classpath.

Integrated Modules

These modules are part of the NetBeans IDE.

NetBeans Profiler

The NetBeans Profiler is a tool for the monitoring of Java applications: It helps developers find memory leaks and optimize speed. Formerly downloaded separately, it is integrated into the core IDE since version 6.0.

The Profiler is based on a Sun Laboratories research project that was named JFluid. That research uncovered specific techniques that can be used to lower the overhead of profiling a Java application. One of those techniques is dynamic bytecode instrumentation, which is particularly useful for profiling large Java applications. Using dynamic bytecode instrumentation and additional algorithms, the NetBeans Profiler is able to obtain runtime information on applications that are too large or complex for other profilers. NetBeans also support Profiling Points that let you profile precise points of execution and measure execution time.

NetBeans JavaScript editor

The NetBeans JavaScript editor provides extended support for JavaScript, Ajax, and CSS.

JavaScript editor features comprise syntax highlighting, refactoring, code completion for native objects and functions, generation of JavaScript class skeletons, generation of Ajax callbacks from a template; and automatic browser compatibility checks.

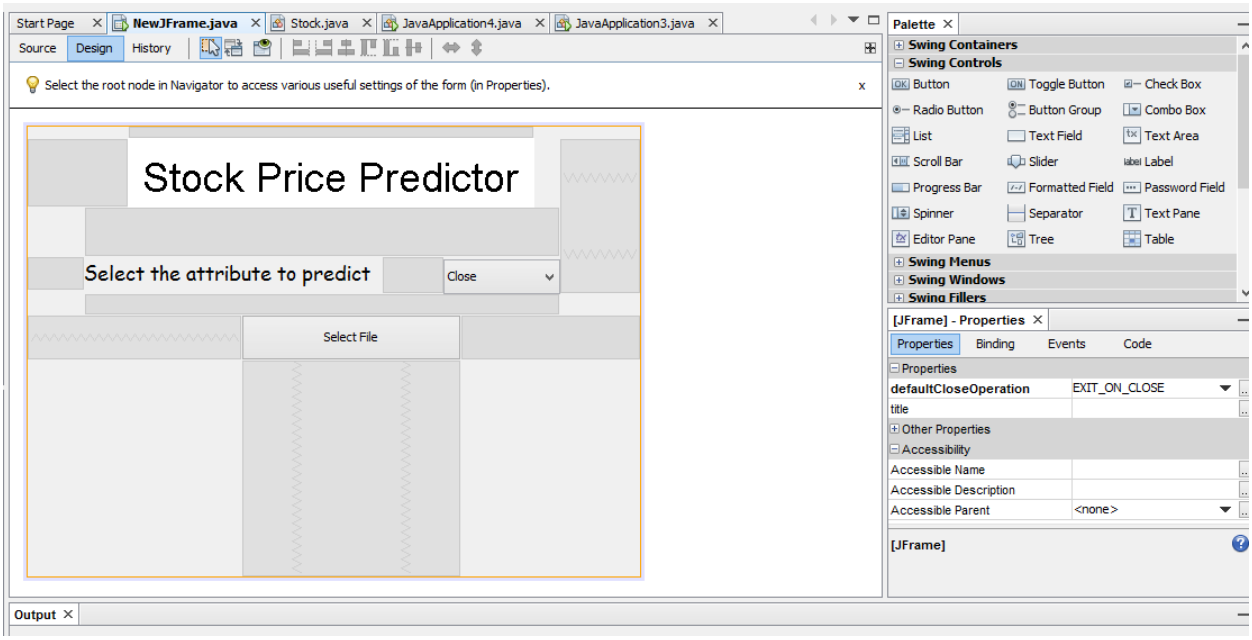
CSS editor features comprise code completion for styles names, quick navigation through the navigator panel, displaying the CSS rule declaration in a List View and file structure in a Tree View, sorting the outline view by name, type or declaration order (List & Tree), creating rule declarations (Tree only), refactoring a part of a rule name (Tree only).

The NetBeans 7.4 and later uses the new [Nashorn] JavaScript engine developed by Oracle.

GUI design tool

Formerly known as *project Matisse*, the GUI design-tool enables developers to prototype and design Swing GUIs by dragging and positioning GUI components.

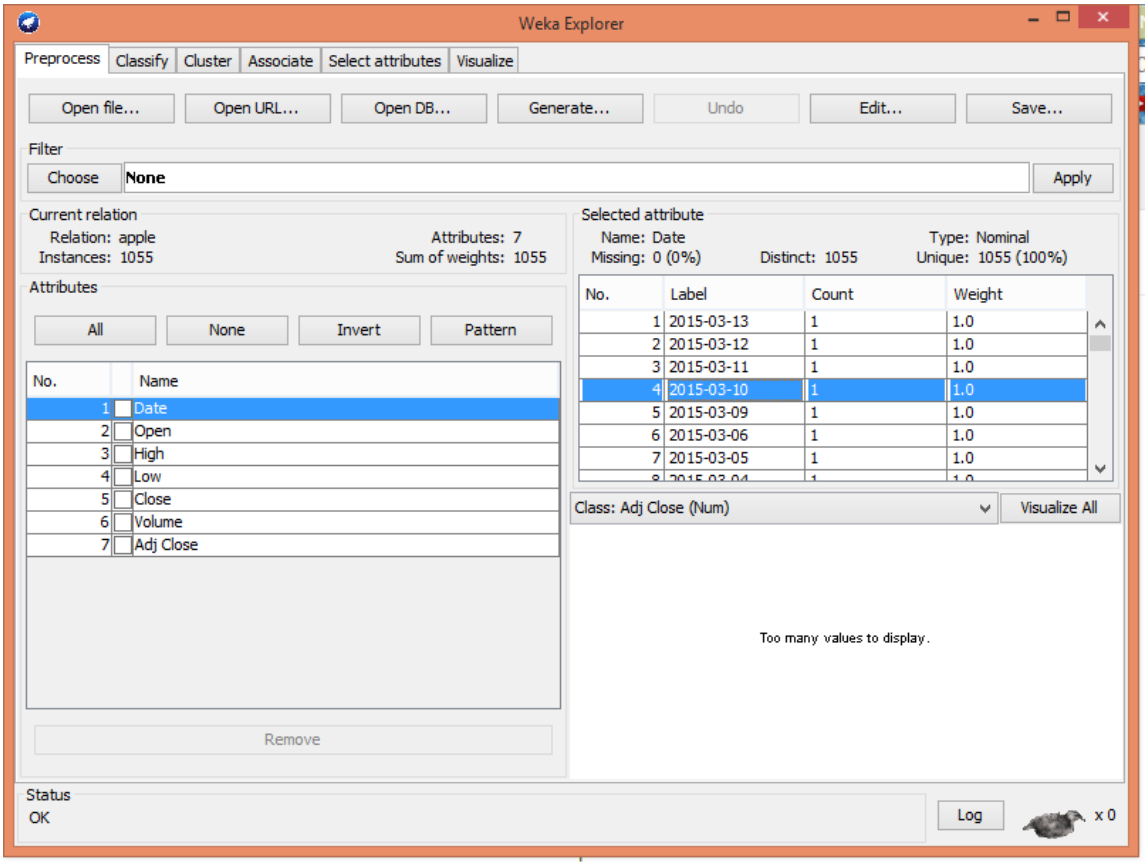
The GUI builder has built-in support for JSR 295 (Beans Binding technology), but the support for JSR 296 (Swing Application Framework) was removed in 7.1.



WEKA

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka is free software available under the GNU General Public License.

Weka is a workbench that contains a collection of visualization tools and algorithms fordata analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. The original non-Java version of Weka was a TCL/TK front-end to (mostly third-party) modeling algorithms implemented in other programming languages, plus data preprocessing utilities in C, and a Makefile-based system for running machine learning experiments. This original version was primarily designed as a tool for analyzing data from agricultural domains, but the more recent fully Java-based version (Weka 3), for which development started in 1997, is now used in many different application areas, in particular for educational purposes and research.



Advantages of Weka include:

- free availability under the GNU General Public License
- portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform
- a comprehensive collection of data preprocessing and modeling techniques
- ease of use due to its graphical user interfaces.

Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification,regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining, but there is separate software for converting a collection of linked database tables into a single table that is suitable for processing using Weka. Another important area that is currently not covered by the algorithms included in the Weka distribution is sequence modeling.