A Dissertation Report On “Predicting Stock Price Movements from Daily News Data”

August 19, 2010

As part of our fulfillment of our M.S., I am submitting the dissertation report on the topic “Predicting Stock Price Movements from Daily News Data”. I hope the report prove to be satisfactory.

By: Madhup Tibrewala
M.S. Artificial Intelligence
Regd. No: 091588874
mt155@hw.ac.uk

To: Professor David Wolfe Corne
Professor David Wolfe Corne

Supervised by:
Professor David Wolfe Corne
Acknowledgement

I would like to express my deepest gratitude to Professor David Wolfe Corne for supporting me in my dissertation at Heriot-Watt University. His guidance and inspiration have provided an invaluable experience that will help me in successfully completing my report and therefore in my career.

My sincere appreciation goes to Mr Vinamra Tibrewala (brother) and Mr Virendra Tibrewala (father) for guiding and clearing any doubts during the preparation of dissertation project.
I would like to thank my numerous friends who helped me directly or indirectly throughout this project.

Last but not the least; I would like to thank my parents, brother, sisters and all family members for their unconditional love and support.
Statement of Non-Plagiarism

By this letter I declare that I have written this paper completely by myself, and that I have used no other sources or resources than the ones mentioned.

The sources used have been stated in accordance with the rules and regulations that are applied at the MACS school of the Heriot-Watt University, United Kingdom. I have indicated all quotes and citations that were literally taken from publications, or that were in close accordance with the meaning of those publications, as such.

In case of proof that the essay, paper or thesis has not been constructed in accordance with this declaration, the MACS school considers the paper as negligence or as a deliberate act that has been aimed at making correct judgment of the candidate's expertise, insights and skills impossible.

In case of plagiarism the examiner has the right to exclude me from any further participation in the particular assignment, and also to exclude me from further participation in the MSc program at the MACS school of the Heriot-Watt University.

Name: Madhup Tibrewala

Place: Edinburgh, United Kingdom

Date: Thursday, 19 August 2010

Signature:
Abstract

The dissertation contributes to the techniques in “Predicting Stock Price Movements from Daily News Data” using Sentiment Analysis. Stock market prediction has always been a fascinating task to researchers, and news can have huge impacts on markets. Announcements about corporate profits, result of the football world cup, management alteration swings the prices of companies shares thus affecting the market behavior. This paper presents various methods in predicting the behavior of S&P500 stock market using sentiment analysis.

The RSS news feeds provided by Yahoo finance are used as the basis for accessing running sentiments in the market. On the news feeds articles, different experiments have been performed using TF-IDF encoding scheme. String handling, Part of speech tagging and WordNet features of Python language are been extensively used to grasp the exact meaning of words in news articles. Using Weka, several machine learning algorithms are been implemented on TF-IDF encoded articles and results are analyzed, compared and discussed.

The paper starts with the introduction of stock market, briefly describing different software’s, packages and techniques used. Then, it summarizes and comments on the previous work done to carry out Sentiment Analysis of various stock markets. In the next chapter, I have explained different experiments performed, along with the data preprocessing required by each of them. The next four chapters displays, analyzes, discusses and compares the results of the experiments. This paper ends with the conclusion of my understanding, limitations of my research and recommendation on the further activity.
Table of Contents

1. Chapter 1 : Introduction
   1.1. Need to discover trading rules
   1.2. A word of mouth about Sentiment Analysis
   1.3. Assumptions
   1.4. News Feeds (RSS)
   1.5. Term Frequency: Inverse Document Frequency
      1.5.1. Stemming
      1.5.2. Stop Words
   1.6. Python
   1.7. Natural Language Toolkit (NLTK)
   1.8. NLTK and Python: Combined together
   1.9. Data Mining and Machine Learning
   1.10. Machine Learning algorithms
      1.10.1.1. K – Nearest Neighbor Algorithm
      1.10.1.2. Naive Bayes Algorithm
   1.11. Weka
   1.12. Part of Speech
   1.13. WordNet

2. Chapter 2: Literature Review

3. Chapter 3: Research Methodologies
   3.1. Data Preprocessing
   3.2. Experimental Design
      3.2.1. Experiment: Only TF-IDF Experiments
         3.2.1.1. One article as a document
         3.2.1.2. All day articles as one document
      3.2.2. Experiment: TF-IDF with POS tagging
         3.2.2.1. One article as a document with POS tagging
         3.2.2.2. All day articles as one document with POS tagging
         3.2.2.3. One article as a document with only verbs
         3.2.2.4. All day articles as a document with only verbs
      3.2.3. Experiment: TF-IDF augmented with WordNet
         3.2.3.1. One article as a document with WordNet
         3.2.3.2. All day articles as one document with WordNet
4. Chapter 4: Experiment: Only TF-IDF
   4.1. One article as a document
       - Nearest Neighbor Approach
       - Naive Bayes Approach
   4.2. One day article as a document
       - Nearest Neighbor Approach
       - Naive Bayes Approach

5. Chapter 5: Experiment: TF-IDF with POS tagging
   5.1. One article as a document with POS tagging
   5.2. All day articles as one document with POS tagging
   5.3. One article as a document with only verbs
   5.4. All day articles as a document with only verbs

6. Chapter 6: Experiment: TF-IDF augmented with WordNet
   6.1. One article as a document with WordNet
   6.2. All day articles as one document with WordNet

7. Chapter 7: Comparison: All the experiments
   7.1. One article as a document
   7.2. All day article as one document

8. Chapter 8: Conclusion and Limitations
   8.1. Conclusions
   8.2. Limitations

9. Chapter 9: Future Work and Recommendations

10. References

11. Tables and Figures

12. Appendices
Chapter 1: Introduction

The chapter starts with the introduction of stock market and the need to discover rules in it. Then, it gives a basic idea about Sentiment Analysis. Later, it outlines the assumptions made to carry out the dissertation. In the last sections, it briefly describes the different algorithms implemented, languages and software’s used to carry out the analysis.

A stock is an official certificate stating you own a small fraction of the company/corporation. A Company issues a stock to grow. To use the money to hire people, expand business etc. Combination of different companies engaging in buying and selling of stocks on a common index is termed as Stock Market Index. The market value of a stock is determined in 2 ways. One is, by analyzing cash flow, sales and profit of the company. This valuation is purely based on historic ratios, statistics and measurable. Another way is presence of Supply and Demand in the market. This valuation is difficult to predict, and it often drives the short-term stock market trends. Stock index acts as an alternative of the economic conditions of the entire market. They involve the total market capitalization of the companies weighted by their effect on the index. The larger stocks would make more of a difference to the index as compared to a smaller company. A general formula to calculate the index is:

\[
\text{Index level} = \sum \left( \text{Price of stock} \times \text{Number of shares} \times \text{Free float factor} \right) / \text{Index Divisor}.
\]

Free float Adjustment factor represents the proportion of shares that is free floated as a percentage of issued shares and then its rounded up to the nearest multiple of 5% for calculation purposes. Dividing the portfolio market value by a factor, usually called the Index divisor, does the scaling.

1.1. Need to Discover Rules:

There are many reasons which justify the need to discover rules in the stock market. Investors fail in making profit. These are the few reasons which led to their failure.

- Poor Understanding of the market data.
- Insufficient knowledge about the system.
- Inadequate Patience
- Unrealistic Expectations
- Lack of Discipline
- Natural Calamity
- Low liquidity, you can't find a buyer of a stock
• Hyping of stock price in low liquidity shares to manipulate stock prices.
• Inability to cut losses
• Publicity by companies

Some of these reasons can be outplayed by scientific analysis of the data. Technical Analysis (TA) and Sentiment Analysis (SA) would be a promising approach for it. Due to the limited availability of the time and massive nature of stock market, I have focused my dissertation on Sentiment Analysis.

1.2. A word of mouth about Sentiment Analysis

Sentiment Indicators are defined to quantify the levels of optimism and pessimism present in the market. News articles are a good source about market sentiment but they are merely a bag of words. A human mind can easily understand and extract knowledge from it. Based on extracted knowledge and own interpretation, he/she invest in market and gains profit or book loss. By the advent of new technologies, stock market news keeps on updating by the second. For a human, it is a tough job to interpret and calculate combined knowledge from them. It is an easy job for a machine provided it is connected to a well-defined program which interprets those bags of words. Understanding and interpretation those bag of words is still a much difficult task for a machine. However, if we can extract important words which quantify the semantics/meaning of article, we might be able to predict the actual meaning of the article. Again, it is a tricky task for a machine to interpret categorical data rather than numerical data. Data mining and machine learning algorithms works well on numerical data. If we use correct conversion techniques to change extracted words in meaningful numerical data, we can train our program to learn and extract knowledge from it. TF-IDF (described later) encoding does the same. It extracts important words from a document and assigns a numerical weight to it based on their importance. “If an article used similar words with similar importance with some other document, it might have the same effect on market as by that document”. The above statement forms the basis of my dissertation.

1.3. Assumptions:

In order to carry out my dissertation, I have made some assumptions. They are as listed below:

• All the news articles carry an equal effect on the market. This is certainly not true. News of big companies or sectors has greater impact on the stock market. Therefore,
in SA of stock market, news about big companies or sectors should be given more
weightage compared to other news. I have marked this point as a recommendation and
would like to work it out in future if asked to do so.

- I have calculated the effect of the news on the next day trading. Although, news
impacts the market the next moment, it is published. Morning news impacts the
market throughout the day, but evening news (or just before market closes) or later
impacts the market the next day.
- I have assumed that only news articles maintain the sentiment of market. I haven’t
considered other factors like future prediction, suggestions by analysts etc.
- I have assumed that all the news in the market is authenticated and true. This might
not the true, because people use to spread rumors and are not authenticated by the
company sources.

1.4. News Feeds (RSS)

RSS is a format for syndicate news and the content of news-like sites of websites on to our
website. News feeds are also RSS feeds which collects news stories from a website and
displays it on to your website. News feeds comes in XML format and thus, are easily
readable from programs.

There exists news which directly or indirectly affects the stock market. For instance, if a
listed company obtains a new contract or a deal, it will automatically boost up the current
price of its share which in turn has a direct positive impact on the stock market. Again, when
France was in the football world cup 2010, CAC (France stock market) was going at a steady
pace. But, later, when France was out of the world cup, the CAC crashed. Since, the football
world cup hasn’t got any direct effect on the stock market, but it virtually drives the sentiment
of the people. Based on those sentiments, CAC crashed. This is an example of indirect effect
of news on the stock market.

There are many news feeds available for the S&P500 stock market. To name a few, Yahoo
finance, Google finance, money.cnn, Bloomberg etc. They all provide market news feeds but
differ in their sources. I have used yahoo finance (http://finance.yahoo.com/news/category-
stocks/) news feeds. Reason being, it provides daily updates of news feeds of U.S. markets
and it compresses news stories from different sources viz. cnnmoney.com, Reuters, Market
Watch etc. This is good because, you can find news feeds from different sources at one place
only. Also, it nullifies the effect of sentiments of an individual who writes the article for particular news.

1.5. Term frequency Inverse Document Frequency (TFIDF)

TF-IDF is a weighting model which estimates the importance of the terms in a document with reference to the terms in entire set of documents. It doesn’t concern about the order of words; rather it retains information based on frequency of occurrence of words in the documents. Let me describe it in some detail. It consists of 2 terms, Term frequency and Inverse Document Frequency. Given some term $i$, and a document $j$, the term count $n_{ij}$ is the number of times that term $i$ occurs in document $j$. Given a collection of $k$ terms and a set $D$ of documents, the term frequency $tf_{ij}$ is defined as:

$$tf_{ij} = \frac{n_{ij}}{\sum_{k=1}^{k} n_{kj}}$$

The Inverse Term Frequency of term $i$ is defined as Log of the number of documents in the master collection divided by the number of those documents that contain the term.

$$idf_i = \log\left(\frac{|D|}{|\{d_j : d_j \in D\}|}\right)$$

So, the TFIDF encoding of $i$ vector for document $j$ is

$$tf_{ij} \cdot idf_i$$

The TFIDF encoding, can possibly, extract the important terms in a document. They also provide the importance of the term based on their number of occurrences in a document. The news articles are not more than a bunch of texts. So, if we can extract important terms from the news articles with their strength, we can have an idea about the sentiment attached with that document.

“If a particular news article has similar set of words and relative importance with some other document, then it will probably affect the market the same way as done by that document”. The above statement forms the basis of my dissertation.
1.5.1.  **Stop Words**

There are many words in English language viz. and, what, where etc. which gives no information about the content of the document. So, we should ignore calculating the TFIDF encoding of these words. Including these words would result an increase in computational time and depreciation of the accuracy and thus gives a negative impact on our results.

3.5.2.  **Stemming**

The process of removing affixes from the main part of a word is termed as Stemming. For example, given the word "produced", its lemma is "produce", however the stem is "produc": Stemming is necessary because it reduce the size of text thus making faster information retrieval. In TFIDF encoding, there exists words which exists in different forms. Through Stemming, we can reduce the words to their basic form and thus, can reduce the number of terms in text. Since, we are calculating the TFIDF encoding of a different forms of words only once; this would results in faster interpretation of text and an increase in accuracy.

1.6.  **Python**

Python is a object-oriented, high-level programming and interpreted language that is used in developing software’s for a wide variety of application domains viz. Internet, software and network development, education and scientific computing. Python has simple readable syntax, high introspection capabilities and it supports modules/packages extensibility capabilities which make it easy and powerful language to develop diverse applications. It also has a good string handling capability. Moreover, python extendible libraries are freely available for all platforms and are freely distributed which makes Python platform independent and portable.

Python is an excellent tool for scanning and manipulating textual data. Python also has several string-handling features that do not exist in other high-level languages. For instance, In Python, strings are “immutable sequences”. Programmer can refer to elements or sub-string directly in a string but they can’t be modified as in the case of tuples. It consists of functions viz. ‘, join’ and ‘.split’ which allow programmers to modify string values by changing string to tuples and vice versa.
1.7. Natural Language Toolkit (NLTK)

According to Mertz David⁴, “The Natural Language Toolkit is a Python library for analyzing and otherwise processing collections of textual data, particularly in terms of the concepts and techniques developed in academic linguistics. Some of these techniques overlap with what goes by the name "text processing"--or perhaps to lexing/parsing in computer science--but other capabilities for syntactic and even semantic analysis are specialized to the more subtle texts and grammars of natural languages.” In other words, NLTK (Natural Language Toolkit) is a suite of packages and libraries for research and development in natural language processing and text analytics, with distributions for Windows, Mac OSX and Linux⁵. NLTK consists of several built-in packages which are very useful in parsing natural language. It supports different types of functions and packages which supports automatic processing of natural language.

NLTK provides different algorithm to perform the same task. For example, NTLK offers three stemming algorithm; namely “Porter”, “Lancaster” and “regexp” stemming algorithm. Each of these stemming algorithms has advantages and disadvantages of their own. These algorithms differ in their complexity, speed and accuracy.

1.8. Python and NLTK: Combined together

Together, Python and NLTK provide the basis for sentiment analysis. They have several packages which, when used sensibly, can help in understanding the sentiments of the news. NLTK comes with many pre-processed corpora to different degrees based on the different experiments. User can try or experiment with different features of the language on these corpuses. In order to extract meaning from sentences, NLTK first tokenize the sentence and tag the words. It offers certain techniques (each having their advantages and disadvantages) that could be applied in order to generate or extract knowledge from it. The group of words could be parsed in grammatical elements to identify their part of speech. Or, the relation between two or more tokenized words is been calculated based on the WordNet library of NLTK. It also gives you the ability to generate statistics, inference results and graphs based on different features applied at various levels on textual data.

You can always integrate the vast string handling and other functional capabilities of python with the above structure of NLTK. For example, you can change the corpora with your own corpora and perform various analyses on it.
1.9. Data Mining and Machine Learning

Huge amount of data is available from many sources like business, finance etc. They might be a potential valuable resource. Raw data is useless until we can extract knowledge from it. Raw data has certain recorded facts or they have certain patterns lying under it. Data mining is an approach to detect patterns and regularities in the data. In other words, they are used to extract information from the raw data. These information might be implicit, previously unknown or potential useful. Machine Learning is the techniques to explore structural dependencies among the data. They can be used to predict outcome in a new situation. According to Tom Mitchell, “Machine Learning is the study of computer algorithms that improve automatically through experience.”

Definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Stock market is all about data. They run by the news and sentiments flowing in the market. These news articles are nothing other than raw data. Data mining and machine learning algorithms provides a way to extract the structural dependency of news articles. By using derived structural dependency, we can predict the behavior of the market when new news comes in.

1.10. Machine learning algorithms

The purpose of machine learning algorithms is to use observations (experiences, data, patterns) to improve a performance element, which determines how the agent reacts when it is given particular inputs. By receiving feedback on the performance, the learning algorithms adapt or adjust the performance element to enhance its capabilities. These algorithms, when learned or trained, can prove vital in solving many problems and exploring tasks. They can be used in extrapolate patterns in data, or in predicting action in an unknown environment.

For the experiments, I have used few machine learning algorithms. These algorithms are described below:
1.10.1. K-Nearest Neighbor (K-NN) algorithm

This is the simplest type of greedy algorithm for machine learning. In this algorithm, for the test case, the K nearest neighbors is identified and is allocated the same class field as majority of its neighbors. The difference between two instances is calculated by the Euclidean distance. For the dissertation, I have taken the value of K as 1 and 5. I have chosen this algorithm because TF-IDF (described below) encoding marks the words with their strength and if an article has approximately same strength of words, it would probably affect the market in the same way, thus having same class field.

1.10.2. Naive Bayes Algorithm

It is a technique based on Bayesian theorem. According to this algorithm, features are independent of the other features. That is, features have no relation among themselves. The classifier assumes that all features independently contribute to the predictability of the class. News articles are nothing but a collection of words. So, for this experiment, I assumed that all words contribute equally to the predictability of the class. There is no inter-dependability among the words and each word carries equal sentiment about the meaning of the sentence. For this reason, I have used the classifier.

1.11. Weka

Weka (Waikato Environment for Knowledge Analysis) is a software package written in java to implement various machine learning algorithms (or to implement own machine learning schemes) for data mining tasks. It encompasses different machine learning algorithms together to have a comparative study on the datasets. The software package can directly be used on the datasets, or, called from your other supporting languages like Java, Perl or Python. The Java interface available to all algorithms enables embedding them in any user’s program. It offers tools to perform data pre-processing, classification, regression, clustering, association rules, and visualization.

Weka comes with a command line interface (CMI) and a graphical user interface (GUI). CMI offers some functionality which is not accessible via GUI and also, consumes less memory. For simplicity purpose, I have used the graphics user interface (GUI). Weka consists of 4 GUI modules: Explorer, Experimenter, Knowledge Flow and Simple Command Line Interface.  

Page 14 of 56
Main use of Explorer module is for preprocessing (applying filters), classifying based on machine learning algorithms, clustering of data, selecting important attributes which help in classifying the data and visualizing the output. There are numerous classifiers available in Weka. Each classifier has a short description and referencing of parameters associated with it. Some of the classifiers are tabulated in table 1.

Explorer also has select attribute feature. Some machine learning algorithms works slower and weaker when there are large number of attributes. With the help of attribute selection, Weka can choose the combination of attributes, based on training data, which classifies the classes of dataset considerably. The last feature of explorer is Visualization. It graph the class attribute with the other entire attributes. By this way, we can see how well or bad each attribute can separate the data. Other approach to achieve similar functionality as provided by Explorer feature of Weka is by Knowledge flow. Here, all the explorer features like clustering, classification are dragged on the canvas and are connected in a graph. The additional use of Knowledge flow is designing and execution of configurations for streamed data processing.

To perform multiple experiments, both Explorer and Experimentation are not suitable. Machine learning datasets need to be evaluated on different datasets and parameter settings. The Experimenter can automate this process and allows analyzing the results. The experimenter allows us to choose the datasets and also split data in training and test dataset.

I used Weka because it allows me to quickly try out different machine learning algorithms and ability to integrate with my source code.

1.12. Part of Speech (POS)

Words in English language have different part of speech depending on their usage. Part of speech doesn’t explain the meaning of the word; rather it depicts that how the word is used. It expresses for instance, a word is used as a noun, a verb or as adverb in a sentence. Verbs have more control over the semantics of the sentence compared to other classes of POS. POS is the classifications of words according to their relations to each other and to the things they represent. It gives a correct idea about the context of the word used in a sentence. For example, people use normally positives words in negative sentences, but the word is preceded by "not" (or some other negative word), such as "not bad". Simple machine learning algorithm assumes every word is independent, it cannot learn that "not great" is a negative. So, to identify the correct meaning of the sentences, POS tagging plays an important part.
The NLTK provides different types of POS taggers. Based on your requirements, you can choose any of them. The table 2 gives an example of 2 POS taggers supported by NLTK. I have used a simple POS tagger provided by python which tag every single word in the document.

3.13. WordNet

WordNet is a large lexical database of English (also supports other languages)\(^8\). It contains all sorts of interesting relationship between words. It can categorize words into hierarchies or find the parts of an object. It also provides information about how closely the words are related to each other. Remember, news articles are only a bag of words. Different news articles use altered set of words. If we can find the correlated strength between them, it would help in increasing our accuracy.

The paper proceeds as follows. In Chapter 2 “Literature Review”, I have summarized and commented on the previous work done in the sentiment analysis of stock market. In Chapter 3 “Research Methodology”, I discussed about the research methodology including data selection and data processing processes. In the subsequent 4 chapters, I have analyzed and discussed results by experimenting with preprocessed data. Thereafter, in the next chapter (Chapter 7), I have compared the results of the algorithm with each other. I conclude the paper in Chapter 8 with future work and recommendation in Chapter 9.
Chapter 2: Literature Review

The purpose of this chapter is to summarize and comments on the previous work done in the selected topic. For the literature review chapter, I have used Research Methods and Project Planning (F21RP_2009-2010) deliverable as the basis, but updated it greatly in the direction of progress of the project. For instance, a section in RM deliverable titled “Technical Analysis (TA)” is removed because I have focused more on “Sentiment Analysis”. However, I have commented on the limitations of TA which directly or indirectly motivated me to explore the world of Sentiment Analysis (SA). In the next sections, I have discussed the limitations of TA and thereafter, I will discuss the previous work done in predicting stock market behavior using Sentiment Analysis.

In simple terms, TA practices past price movements with other trading strategies like volume, summarized in price charts, to forecast the trading motion of the market\(^1\). Incorporation of High Performance Computing and computing methods viz. Machine Learning, Genetic Algorithms etc in algorithms\(^2\) has added honey to the sugar. Technical analysts examines the current demand and supply in the trading market and based on them, they identify support and resistance levels\(^3\). From this, they ascertain the probable future trading range and direction of price movement for a particular share or the whole share market index. Technical analysts also believe that stock prices move in trends, and that price patterns repeat themselves. Numerous researchers have attempted to construct an accurate model for the stock market. Artificial Neural Networks\(^4\), Genetic Algorithm \(^5\) and Fuzzy System \(^6\), have proved to be an asset to them. However, these methods have their own limitations. For instance, Back-propagation neural network\(^7\) suffers from the risk of over-fitting and large number of parameters. More importantly, they have neglected other source of information such as mass media that will greatly affect the behavior of investors. So, it would be a good idea to carry out the research in the lagging part of the researchers. For this reason, I have focused myself in carrying out sentiment analysis rather than technical analysis.

**Sentiment Analysis (SA)** is a completely different way to predict the behavior of stock market. Sentiment Analysis is defined to quantify the levels of optimism and pessimism present in the market\(^8\). SA judges the general feeling of the investors towards the market. However, if investors become too active in one extreme, then the market will behave opposite to their expectations (Contrary opinion theory)\(^9\).

We know that stock index is collection of different stocks distributed over many sectors. Also, it is rightly said that “Stock market drives the economic health of the country”\(^10\).
Economic health of a country is reflected in the financial news of the country. On taking it other way round, financial news can be used in predicting the behavior of the stock market. This statement, according to me, forms the core of sentiment analysis of the market.

At first, Mishkin\textsuperscript{14} (1978) proclaimed the dependence of consumer sentiment on the US economy during 1954 Q1 – 1976 Q4 based on household balance sheet. He finds that, rising financial assets leads to optimism of household’s income. But, Mishkin only used household balance sheet, but there exist loads of more other indicators which are more dominant and useful than household balance sheet. However, Mishkin study proved to a bench mark for research in this area. It opens the gate for further study in this particular topic. Later, American Association of Individual Investors (AAII) \textsuperscript{15} in 1987 introduces a survey to catch the opinion of investors. The results of the survey then suggest the expected trend (Bearish, Bullish or neutral) of the stock market. They have considered only a short range of people (around 500) so; this is only individual sentiment indicator. I believe, they should have considered the global sentiment indicator by interviewing a large number of people. Mishkin and AAII studies and research proved the presence of sentiments available in the market. However, they were not able to capture and use them.

Illing and Aaron (2005)\textsuperscript{16} shows there are many risk appetite indexes capturing market sentiment in some recent period. For example\textsuperscript{16}, these indexes are able to identify the 1998 Russian crisis as a period of low risk appetite and signal a high degree of risk appetite during the bull market in the US in 2003. Although, the indexes used were volatile and, as a result, sometimes give multiple signals in a given period. This is not desirable as this fails to establish any link between the level of risk appetite and swings in asset prices.

Classical finance theory\textsuperscript{28} neglects the role of the sentiment indicators by claiming that investors are supposed to be rational. That is, rational factors don’t have any effect on the market. On the other hand, (Cai and Zheng, 2004)\textsuperscript{28} proves that if Institutional investors account for more than 50% of trading in the US \textsuperscript{29}, their sentiments should have a considerable effect on the market. Cai and Zheng\textsuperscript{28} have focused only on the impact of institutional trading on stock returns. Later, Bohl and Brzeszczynski (2006)\textsuperscript{30} studied the influence of institutional ownership on volatilities. They found that institutional investors stabilize exchanges, as volatility drops after institutional investors become principal shareholders. That is, they analyze the effect of institutional trading and ownership. However, for their experiments, they have considered small investors as noise traders. This should not be the case. Small investors, although, they are massive in numbers, carries a large amount of
sentiments. The combination of sentiments of these small investors plays a major part in calculating the overall sentiment of the market.

Despite of large number of small investors, researchers have tried to capture the sentiments of these investors. The results proved to be very ambiguous. Brown and Cliff (2001) 32 found long-run predictability of returns for a period of about 2–3 years but they failed in forecasting short term returns. Kenneth and Statman (2000) 33 detected a significantly negative correlation in short terms returns of S&P 500 and people sentiments. However, they failed in forecasting long term investment returns.

Baker and Wurgler (2006, 2007)17, 18 research on stock market price response on response to earnings announcements. They found that the stock market reacts to the global news which was confirmed by Mian and Sankaraguruswamy19. Mitchell and Mulherin20 studied the influence of public information reported by Dow Jones and concluded that a direct relation does exists between released news articles and stock market activities. News contains valuable information of a company and can change the mind set of an investor over a stock. This change in mind-set results in movement of stock prices/index. Qui and Welch21 finds that changes in consumer confidence can explain the excess returns on small stocks. However, they failed to find the same thing with large stocks. DeLong, Shleifer, Summers, and Waldmann (1990) confirmed the presence of noise in the stock market. According to them, sentiment can influence security pricing in two necessary conditions: [1] the assets are held predominantly by sentiment (noise) traders, and [2] transaction costs are high enough to prevent systematic arbitrage by arbitrageur22. Mittermayer 23 proposed a trading system to predict stock price trends immediately after the release of a news article through text mining techniques. They found the system significantly outperforms a random trader. However, only news that is directly related to the stock is included in their study while NoSinger’s study suggests that both the firm specific and the general economic news affect trading behaviors24.

response to good news increases with sentiment and the negative stock price response to bad news decreases with sentiment. Mian and Sankaraguruswamy also finds strong influence of sentiment with respect to stock prices response to news for small firms, young firms, volatile firms, and growth and value firms.

To make the investment decisions with timely information, the listed entities on the Australian stock exchange are required to fully inform the investors. The entities include negotiations of purchase, director appointment/resignation and divestitures of businesses. Most of this information is in news articles, so news feeds become a good source of information in assisting the traders. Mitchell and Mulherin proved direct relation in Dow Jones between news articles and stock market behavior. News article acts as an easy, quick and updated source of company related movements. NofSinger showed that in some cases, investors tend to buy after positive news which results in buying pressure and push the price higher; and sell after negative news which results in a drop in price. Mittermayer developed a trading system to predict stock price trends immediately after the release of a news article through text mining techniques. The system has easily outplayed a random trader. But, he included only the articles which are directly related to the stock. However, there are general economic news also effects the behavior of the market. There is no doubt that news articles effects the sentiments of market, therefore, have an impact on stock market behavior, not much research is been done in predicting the market behavior using this methodology.

Several researchers have documented various exogenous factors that capture mood (and therefore investor sentiment) which are correlated with stock returns. Previous researchers have utilized a variety of variables as mood indicators, such as sunshine (Saunders, 1993; Hirshleifer and Shumway, 2003); sleep patterns, daylight (Kamstra, Kramer, and Levi, 2000) temperature (Cao and Wei, 2005); and international soccer results (Edmans, Garcia, and Norli, 2007). Edmans et al. (2007) argued that the chosen mood indicator must satisfy three criteria before the researcher can rationalize its link with stock returns. “First, the selected variable must drive the mood in a substantial and unambiguous manner so that its effect is vigorous enough to be reflected in asset prices. Second, the variable must affect the mood of a large proportion of the population, so it is likely to influence investors. Third, the effect must be correlated across the majority of individuals within a country.” Terrorist activity has also has a substantial effect on the market. For instance, on September 11, 2001, WTC terrorist attacks in New York City, the MSCI World Index lost 1.98% of its value (U.S. were closed) A second round loss of 2.57% was recorded on September 17 when the U.S. market re-opened. Similarly, on March 11, 2004, the day of the Madrid attacks, the MSCI fell
1.72%. All of these news articles come in the day to day financial news. So, the news articles give an almost clear idea about the sentiments of the people.

Gidofalvi, 2001\(^{46}\) uses bag of words approach to represent the news articles. Once news articles have been collected they must be represented. The method was good but it has certain drawbacks. It includes noise from seldom-used terms and has scalability problems. Tolle & Chen, 2000\(^{47}\) reduced scalability problem by considering only the noun phrases. McDonald, Chen, et al., 2005 extends noun phrases by selecting proper nouns defined in well-defined categories to allow better generalization of previously unseen terms. (Schumaker & Chen, 2006) compared these representation techniques and found that proper Noun representation was more effective in symbolizing news articles. Joachims, (1998) eliminates noise from lesser used terms as well as reducing the number of features represented by introducing a threshold of term frequency. Joachims, (1998) also changes the raw terms into binary format (1 if in article otherwise 0) to run the machine learning algorithms.

Lavrenko et al, 2000 tested the effects of training on the entire universe of news articles and training based on articles for a specific company. He found that training a system on Stock-Specific news articles led to more accurate predictions of price direction. Lavrenko hasn’t considered the grouping of companies in sectors. Peramunetilleke\(^{48}\) experiment of weighting TF x CDF approach in forecasting U.S. dollar performs the best among TF-IDF, conventional numeric time, neural net approaches or random guessing.

Peter\(^{49}\) study of stock message boards reveals important results. He found that stories on message boards are not just noise. Rather, they link to company information environment. A dramatic increase in postings about your company could be a sign of investor discontent and signal the need to fill the information gap. So, in other words, he strengthens the bond between people sentiments and stock market. Vivek\(^{50}\) further proves his point about people sentiments in message boards by applying different machine learning algorithms and analyzing the results.

Corso, Gulli, and Romani (2005)\(^{53}\) confirms that more recent news stories are more important than the old stories. Their experiment on based approach produced far better results compared to the same weightage approach. They also authorizes about the clustering of various news articles. However, with this approach they fail in predicting the long term investments returns. Macskassy and Provost (2001)\(^{54}\) states that if there is a news article about a particular company then there will be a significant change in the stock price of that company. Unluckily, due to massive nature of stock market, it is very difficult to classify news on to the
specific companies. Also, the news changes at a moment, so again, it was getting very difficult to handle these large quantities of textual data.

Sanders et al\textsuperscript{25} proves that investors which only depend on SI may be misguided. This is because; SA only indicates marginal ability to predict future prices. Wright (2009) claims that “for many businesses, online opinion has turned into a kind of virtual currency that can make or break a product in the marketplace”.

From the literature, I realized that using TF-IDF encoding, not much work is been done in extracting sentiments from news articles to predict the behavior of stock market. So, I would like to focus my work in performing SA of stock market through TF-IDF encoding of the news articles.
Chapter 3: Research Methodologies

I have divided the chapter 3 (Research Methodologies) of my dissertation in two sections, “Data Preprocessing” and “Experimental Design”. Data preprocessing includes cleaning, transforming and reducing data so as to prepare them for the experiments. Experimental design speaks about the building and execution of different experiments performed with preprocessed data. In the next sections, I would like to discuss each of the experiment in detail.

3.1. Data Preprocessing

The first challenge in predicting stock market using SA is to process enormous amounts continuously updating and previously documented information. There are countless sources for such documented information. To perform sentiment analysis of S&P500, I have used RSS news feed provided by Yahoo Finance. Yahoo Finance is a compilation of 45 different news sources, therefore, reducing the time and effort to search for relevant news feeds. The source of the feeds can be found at the link, “http://feeds.finance.yahoo.com/rss/2.0/category-stocks?region=US&lang=en-US”.

Initially, using PHP (Hypertext Preprocessor), I have downloaded the news articles (from 8 July till 8 August) and stored them in MySQL database. Since, the RSS feeds displays only the news for the current day, therefore, to store the old news, I have altered the “query string” parameter of RSS feeds. I have also used the CURL feature of PHP language which enables it to run at designated time of each day provided system is connected to internet and the local host server is switched ON. This is done so as to keep the system updated. However, news affects the market the next movement, it is published. In order to reduce system complexity, firm time limit of dissertation and save system processing time, I have defined it at 11:55 PM of each day.

After storing them in MySQL, the HTML and unwanted tags have been removed using urllib2 package of python. This is done because; they provide no information about the document. For the same reason, special characters, punctuation marks and brackets are been removed from the documents. Again, for the same reason, the stop words are been removed.

The relevance of a document is accomplished by extracting important keywords/terms from the document (indexing). The effectiveness of any such term-based approach is directly
related to the accuracy with which a set of terms represents the content of a document, as well as how well it contrasts a given document with respect to other documents.

Machine Learning algorithm works better on numerical data than categorical data. So, a technique was needed which converts those bag of words in numerical values and also maintains the effectiveness of terms for the document. TF-IDF encoding does the same thing. It encodes the importance of terms in a given document with reference to terms of all the documents. So, this acts as a good approach to extract important terms from a document and assigns them a numerical weight. I have done the same thing. I have performed different experiments using TF-IDF encoding. In the later sections of the chapter, I have discussed each of the experiments in detail.

I have assumed that news affects the market the next day it is published. Therefore, I have taken a nominal class field and it is defined as the difference of the opening price of next day and closing price of the current day. If the difference is positive then class field value is “U” and if difference is negative, then I have marked the class field value as “D”. Please note that, the accuracy of the data is been tested on a new set of news articles taken from a different day and thereafter, the algorithms are been performed either by randomizing or by preserving the order of instances. The test data is being created by taking the articles of day news, calculating its TF-IDF encoding with reference to other documents.

3.2. Experimental Design

Different experiments performed in the dissertation require processing of data at various levels. These levels are based on the use of different techniques implemented to find the overall sentiment of the documents. The experiments performed along with the required data processing are described below:

3.2.1. Experiment: Only TF-IDF encoding

In “Only TF-IDF encoding” experiments, I have performed experiments only with TF-IDF encoding scheme. That is, the TF-IDF weighing of every term is important (except stop words). Each term contributes to the sentiments of the text according to its TF-IDF encoding weight-age.
I have performed 2 small sub-experiments using only TF-IDF encoding. They are described as follows:

3.2.1.1. One article as a document:

For this experiment, I have assumed that each news article has an effect on the market behavior. That is, the sentiment associated with each article affects the market in one way or the other. However, each article has its own sentiment which can be positive or negative. I haven’t taken this factor in account since it is much difficult to calculate because of huge amount of documented textual information and limitations in Natural Language Processing.

In this experiment, I have taken each news article as a separate document. Now, for each document, the stop words are being removed and remaining terms are being stemmed. The TF-IDF encoding of all the articles are calculated. The class field is nominal and is calculated in 2 steps. In the first step, for each article, I have calculated the difference of next day opening price and closing price for current day. In the second step, if the difference is positive, I have marked the class field as “U”, or else “D”.

I have run this experiment twice. The first one is the same as defined above. For the second run, I have added an extra attribute named “difference” which holds the difference of next day opening price and closing price for current day. This is done to add extra information about the effect of the news in the stock market.

I have implemented 2 machine learning algorithms on to the TF-IDF encoded data. K-Nearest Neighbor and Naive Bayes algorithm. I can directly run the K-Nearest Neighbor algorithm on to the encoded dataset. However, in order to run Naive Bayes algorithm, I need to preprocess the data. Preprocessing here refers to discretization of data, since this is the prime requirement to run Naive Bayes algorithm. The results and analysis for the sub-experiments are discussed in section 4.1.

3.2.1.1. One day article as a document:

For this experiment, I have assumed only one news article can’t predict the behavior of the market. Rather, it is the combined effect of all the news that affects the behavior of the market. It might be possible that many articles have a negative sentiment and others have a positive sentiment. But the overall sentiment of the all the articles, taken together, have a positive sentiment. So, the market behaves accordingly.
In this experiment, I have compressed the article of an each day in a separate single document. Now, for each document, the stop words are being removed and remaining terms are being stemmed. The TF-IDF encoding of the entire documents are calculated.

Again, I have run this experiment twice. The first one is the same as defined above. For the second run, I have added an extra attribute which holds the difference of next day opening price and closing price for current day. This is been done to incorporate the effect of the news in the stock market. For each run, I have implement 1-NN, 5NN and Naive Bayes algorithms. The results and analysis for the experiment are discussed in section 4.2.

3.2.2. Experiment: TF-IDF with POS tagging

All the words in the above experiment are given equal weight-age in predicting the class field. However, certain words like verbs in a sentence should be given more importance. These are the words which tell more about the semantic of the document. Keeping this point in mind, I have extracted verbs from the sentences.

I have used the POS tagger package given by NLTK Python to extract the verbs. This package when implemented with Python assigns part of speech such as noun, verb, or adjective to the words of the sentence. Thereafter, I have extracted verb from the tokenized word. In order to separate them from other words I have appended “_verb” to the end of the word.

Removing stop words and punctuation symbols initially might alter the part of speech of the sentence. Therefore, firstly, I have extracted verbs from the document and then, removed the stop words and special symbols. Certain stop words of English language; like “is”; are also verbs and stop words. These words or verbs like “is” provide no information about meaning of the language, therefore, are of no use. After appending “_verb”, it is difficult to remove them. To overcome the problem, I have to use the regular expression package of NLTK-Python. After removing all the stop words, I have performed different machine learning algorithms by giving verbs twice as much importance compared to other words in the same experiment. Again, the class field is nominal and is calculated in 2 steps. In the first step, for each article, I have calculated the difference of next day opening price and closing price for current day. In the second step, if the difference is positive I have marked the class field as “U”, or else “D”. In the next sections, I will describe each of the experiments performed in detail.
3.2.2.1 One article as a document with POS Tagging

In this sub-experiment, I have implemented POS tagger package of NLTK python on to each article separately. Thereafter, I have removed stop words, special characters and punctuation symbols from the documents using regular expressions. Then, I have calculated the TF-IDF encoding of all the documents.

In this experiment, I have given twice as much weightage to verbs as compared to other words of document. That is, if a particular feature matches and if it is a verb (appended with _verb), then it amounts twice compared to other non-verbs terms in calculating the neighborhood. In order to implement different machine learning algorithm, I have written my own code. I have performed different machine learning algorithm on it, the results of which are discussed in section 5.1.

3.2.2.2 All day articles as a single document with POS tagging

In this sub-experiment, I have merged all the documents of each day in one single separate document. Thereafter, stop words, special characters and punctuation symbols are removed. Then, the TF-IDF encoding of all documents are calculated and stored.

Again, similar to above experiment (Experiment: 3.2.2.1), verbs are given twice weightage compared to other important extracted terms in calculating the nearest neighborhood. Again, I have used my own code to implement different machine learning algorithms and the results are analyzed and compared in section 5.2.

After performing and analyzing the above sub-experiments, I felt, if verbs are so important for a document, it would be a good idea to perform the TF-IDF encoding only on these verbs. Therefore, by using Regular Expression of Python, I extracted only the verbs from the document. Again, I have performed different machine learning algorithm which are discussed below. The class field is taken to the same as in above experiments.

3.2.2.3 One article as a document with only verb

In this experiment, I have extracted verbs from individual articles using regular expression of python. Again, stop words are been removed using regular expression. Later, punctuation and special symbols are being removed. Since, removing them in the initial stage might alter the part the speech. Then, I have calculated the TF-IDF encoding of the articles. On having at the look at it, I find, there were lot more of duplicate values. Since, these duplicate instances might depreciate the true accuracy of the program. I just deleted them and performed the
experiments with unique values. Thereafter, I have performed different machine learning algorithms on it whose results are been discussed and analyzed in section 5.3.

3.2.2.4 All day articles as document with only verb

For this experiment, firstly, I have compressed all the articles of news for each day in one article. I have extracted verbs from individual articles using regular expression of python. Later, stop words, punctuation and special symbols are being removed. Again, there were many duplicate instances and I have deleted them. I have performed various machine learning algorithms on the data. The dataset has 65 instances and 29 attributes. The results of which are analyzed in section 5.4.

3.3. Experiment: TF-IDF augmented with WordNet

Articles are merely a bag of words. These words have their own senses and meaning. However, there are several words in English language which have the same meaning. They just differ in their usage and strength. So, I thought, it would be a good idea to compress all those words in one word and later, calculate the TF-IDF encoding of them.

In order to extract the words with same meaning, I have used WordNet package of NLTK Python. In this experiment, first of all, I have gone through the contents of the document with one word at a time. Then for each word, I search for the words which have similar meaning to that if the given word. Then, I have changed the similar words to that of current word. I continued the above step till I reduced dataset to an extent where there are no or quite less words with same meaning. However, it should be noted that all the similar words can’t be extracted due to limitations of WordNet module and ambiguity of English language. After wrapping the similar words, I have calculating the TF-IDF encoding of the words. I have performed two sub-experiments with WordNet feature. The sub-experiments are described below:

3.3.1. One article as a document with WordNet

In this experiment, I have extracted similar terms of one article at a time. In the next step, I have calculated the TF-IDF encoding of each of the article. For this experiment, I have assumed that it is frequency of similar words that governs the actual meaning of the article. The results of this experiment are analyzed in section 6.1.
3.3.2. All day articles as one document with WordNet

For this experiment, First of all, I have compressed all the articles of a day in one document. Then, using WordNet, I have sorted out similar words of compressed articles and perform the TF-IDF encoding of each document with reference to other documents. The results of this experiment are displayed and analyzed in section 6.2.

In the next chapter, I have discussed and analyzed the result of the experiments performed only with TF-IDF encoding.
Chapter 4: Experiment: Only TF-IDF

In this chapter, as stated in section 3.2.1., I have analyzed, discussed and compared the results of experiments performed using only TF-IDF encoding scheme. The results of the sub-experiments are discussed below.

4.1. One article as a document:

I have applied two different machine learning approach in classifying numeric news feeds of articles only with TF-IDF encoding scheme. The result of the two approaches is discussed below:

- **Nearest Neighbor Approach**

Table 3 tabulates the results of 1-NN and 5-NN machine learning algorithms on correctly classified instances (in %) over 4 runs of randomized test and training sets. The table 3 also depicts the accuracy of algorithm over test set. The accuracy is based on 90% training data and 10 % test data. The total numbers of attributes are 1350 excluding the class field. It can be clearly seen from the table that they have classified the data almost equally. However, 1-NN has a slightly better accuracy than 5-NN algorithm. It might be because; due to large number of attributes and dominance of class “U” data. Also, the accuracy achieved by both the algorithms is not so good. On comparing the results on test set, we can that the accuracy further decreased considerably. This might be due to the addition of new terms in the news articles. In order to further investigate the results, I have added an extra parameter to the dataset. As stated in section 3.2.1.1, I have added the field “difference” as an extra attribute.

After adding the field and normalizing it, I have again run the 1 NN algorithm on it. The results of which are displayed in the figure 1.

Figure 1 illustrates the results of 1-NN algorithm after addition of new attribute “difference”. It results in a slight increase of accuracy. This might be because; we have added the effect of news on the market. Also, from the confusion matrices, it can be seen that the system fails to predict correctly the instances of class D. That is, maximum instances of class “D” are predicted to be of class “C”. For the experiment, the figure also depicts the different parameters like mean squared error, kappa statistic, precision rate, recall rate etc.
• Naive Bayes Approach

Figure 2 graphs the results of Naive Bayes algorithm on TF-IDF encoding scheme. The algorithm is run over new day test data. It results in a good increase of 10% in accuracy compared to 1 NN algorithm. Also, from the confusion matrices, we can see that it has a significant increase in predicting the class “D” as the correct class. But here, we find a good true positive rate and an average false positive rate which is not good.

4.2. One day article as a document:

As stated earlier, in this experiment, I have compressed all the articles of a day and taken them as a single document. After doing the data preprocessing, I have implemented several machine learning algorithms on it. The results and analysis of the algorithms are discussed here:

• Nearest Neighbor Approach

Table 4 compares the results of 5 NN and 1 NN machine learning algorithms over 4 runs on training test randomized dataset and 1 run over test set for TF-IDF encoding of all articles of one day taken as a single document. The total number of attributes in the experiment comes out to be 1887. Again, in this case, both the algorithms have an almost equal accuracy. For the experiment, the number of instances of class U is one and a half times larger than instances of class D. From the table, we can infer, over randomized versions of data that the accuracy for 1 NN varies largely from 50.4% to 72.5%. This might be due to the fact that we are choosing only one nearest neighbors which are been randomly placed in test and training sets. So, accumulation of one class in a particular set might result in increase or decrease of accuracy. For the test set, the accuracy decreased marginally, but it is better than the experiment discussed in section 4.1. Might be; because; we have considered the combined effect of all the news in this case.

• Naive Bayes Approach

Since, the numbers of instances are less (30), I have run the algorithm with a test set and training set of 25% and 75% respectively. It gives an accuracy of 75%.

Table 5 tabulates the confusion matrices as the result of the algorithm. From the confusion matrices, we can see that it hasn’t classified any of the instances as of class D. Of the total instances, 63% of the instances were from Class C and 37% are from class D. All the
instances are predicted to be of class C. Therefore, the algorithm fails in classifying the data accurately. It might be possible due to less number of training and test instances.

On comparing the results of both the above sub-experiments, we see that both of them have predicted the class almost equally. This might, be, because we fail in extracting important terms or haven’t given proper weight to the terms. Since, verbs carry more importance than other words. So, in the next experiment, I have given twice much importance to verbs as to other terms.

In the next chapter, I have discussed and analyzed the results of different POS tagging experiments.
Chapter 5: Experiment: TF-IDF with POS tagging

In this chapter, as stated in section 3.2.2., I have analyzed, discussed and compared the results of experiments performed using TF-IDF encoding scheme and POS tagging. The results of the sub-experiments are discussed below.

5.1. One article as a document with POS tagging

In the experiments, I have given twice weight-age to verbs compared to other words of the article. The results of the machine learning algorithms are discussed here:

- **Nearest Neighbor Approach**

Table 6 compares the results of 5-NN and 1-NN machine learning algorithms over 4 runs on training test randomized dataset and 1 run over test set for TF-IDF encoding of all articles with POS tagging of one day taken as a single document. We can see clearly a good increase in both, the randomized version and test set version, of correctly classified instances accuracy. Again, in this case, for each algorithm the randomly training and test accuracies differ much. This is again due to concentration of a class in training or test datasets. So, after performing this experiment, I can say that my assumption about verbs was correct and they play an important part in determining meaning of the articles. Thus, they can help in accessing sentiments of stock market.

- **Naive Bayes Approach**

Table 7 tabulates the results of Naive Bayes algorithm on test data by giving twice weightage to verbs. The algorithm performed much better compared to all the Naïve Bayes experiments so far. It performed better in classifying both the datasets. Surprisingly, this time, it has classified more instances of class “U” to class “D” compared to class “D” to class “U”. Altogether, it has classified instances better than all the approaches followed so far.

5.2. All day articles as one document with POS tagging

After compressing all the articles of a day in a single document, I have calculated the TF-IDF encoding of the document. Thereafter, I have run machine learning on the TF-IDF encoded datasets. The results of which are discussed below:
• **Nearest Neighbor Approach**

Table 8 depicts the comparison of 1-NN and 5-NN algorithms over randomized cases and test set. As expected, the accuracy increased in the case. This is because; the overall sentiments of all news articles are taken into account. However, for a change, in this case the accuracy of 1-NN is greater than 5-NN. I don’t know the exact reason. Might be, because; number of verbs in article and twice weightage given to each of them towards the accuracy.

• **Naive Bayes Approach**

Table 9 depicts the results of algorithm in confusion matrices. With any doubt, the algorithm does well in predicting the class of instances. But, still, it has classifies many instances of the right class to the other class. Therefore, still there is a chance of improvement which arise the need to perform other experiments.

5.3. **One article as a document with only verbs**

While preparing the dataset, I found that there were many duplicates in the dataset. These duplicates might hinder the accuracy of machine learning algorithms. So, I have removed them and performed he experiment only with unique values. The results and analysis of the algorithms applied on this dataset are discussed below:

• **Nearest Neighbor approach**

The results of this algorithm are shown in table 10. The table depicts the comparison of 1-NN and 5-NN correctly classified instances over test set for 4 runs and with test set. From table 5, we can deduce that 5–NN algorithm has performed much better compared to 1-NN algorithm. This might be, because after removing duplicate values, the system succeeds in classifying correctly the classes of unpredicted instances. From the table, for both the algorithms, we can see that there is range of accuracies. This is probably due to the randomly chosen test cases.

• **Naive Bayes approach**

The results after applying Naive Bayes classifier are depicted in figure 3. However, the algorithm has given 61% accuracy, but it failed to classify the instances. From the confusion matrices, it has classified all the instances to class “U”. We can say the algorithm failed completely in order to predict classes.
5.4. All day articles as one document with only verbs

The dataset contains very less, only 30 instances. This is because I have used the news articles of only one month. Due to less number of test and training instances the algorithms might not perform accurately. For this reason, I have run this algorithm using cross-validation technique. The results of machine learning algorithms are discussed here:

- **Nearest Neighbor Approach**

Table 11 depicts the results of 1-NN and 5-NN over 4 runs and on the test set. Again, similar results over randomized training and test data, 5-NN has outperformed 1-NN algorithm by a big margin of 10%. The algorithm has also worked better on the test set.

- **Naïve Bayes Approach**

Table 12 depicts the results of Naïve based algorithm on to the cross validated data. The algorithm performed much better this time. Most of the instances are predicted to be correct. Although, we have performed our experiments with cross – validation. It would be a good idea to perform it using a new set of data.
Chapter 6: Experiment: TF-IDF augmented with WordNet

In this chapter, as stated in section 3.2.3., I have analyzed, discussed and compared the results of experiments performed using TF-IDF encoding scheme and WordNet. The results of the sub-experiments are discussed below.

6.1. One article as a document with WordNet

The results of the experiment are discussed here.

- **Nearest Neighbor Approach**

  Table 13 tabulates the results of 1-NN and 5-NN algorithm over 4 runs on randomized data and a run over test data. The algorithms have outperformed all the results so far. The algorithm did very well in classifying data on randomly built on training and test dataset. However, for the test data, strangely, 1-NN has done much better. This might be; because; by compressing similar words in one, the classifier succeeds in finding the almost exact match for the news test dataset.

- **Naïve Bayes Approach**

  Table 14 displays the results of Naïve Bayes algorithm in confusion matrices. Again, the algorithm performed much better and time, and for a change, it correctly classified much instances of class “D” to class “D”. Might be; due to the compression of similar words, it succeeds in finding the correct match to the test set data.

6.2. All day articles as one document with WordNet

The results of the experiments are discussed below:

- **Nearest Neighbor Approach**

  Table 15 tabulates the results of 1-NN and 5-NN algorithm over 4 runs on cross validation data and a run over test data. The algorithms give varied results. It performed good on randomized data but slightly less on test data. Interestingly, 1-NN algorithm is better it performed slightly less than 5-NN on test data. This is probably, again for the same reason. Due to the compression of similar words, it succeeds in finding the correct match to the test set data.

- **Naïve Bayes Approach**
Table 16 displays the results of Naïve Bayes algorithm in confusion matrices. Interestingly, this time the algorithm failed in classifying more of class “U” instances correctly. Probably, the WordNet failed in compressing similar words or due to ambiguity of natural language.
Chapter 7: Comparison: All the experiments

In this chapter, I have compared all the results of my experiments based on the preprocessing of the dataset. The next section compares the each of the result in detail:

7.1. One article as a document

In this section, I have compared all the “One articles as a document” experiment cases with each other. Figure 4 and Table 15 depicts the results of the experiments on the bar chart and table respectively. From the chart, we can see that, as expected, the entire algorithms in all the experiments lowers their accuracy levels in test data compared to the randomized versions of the dataset. Again, Experiments with WordNet and 5-NN has performed best among all in randomized cases but there is a big drop in its accuracy when compared with the test data. WordNet with 1-NN algorithm has given an almost stable result over test and randomized versions. This might be because; the compression of similar terms in a single term results in a close match with the test data. As expected, only TF-IDF encoding (both 1-NN and 5-Nn algorithm) hasn’t performed better. This is because, English is an ambiguous language. There are many ways to write the same thing with different words. That is also the reason the WordNet performed good because we have compressed all the similar words in one, therefore an increase in the accuracy.

7.2. All day article as a document

In this section, I have compressed the results of “All day article as a single document” experiments performed. The results are been tabulated in figure 5 and table 16. In this case, the POS tagging, where only verbs are been taken, has performed better. WordNet with 1-NN algorithm has just performed slightly weaker than the POS tagging with only verb. Again, the only TF-IDF failed in this case.
Chapter 8: Conclusion and limitations

This chapter concludes my research and points out the limitations which has affected the accuracies of the algorithms.

8.1 Conclusions

“Daily news data carries sentiments and they can be extracted and utilized to predict stock price movements”. The statement has been proved by carrying out different experiments with TF-IDF encoding and features of Python NLTK. The experiments has given varied results depending on preprocessing of dataset, language or string handling features used or the type of machine learning algorithm.

5-NN algorithm has performed better than 1-NN and Naïve Bayes algorithm. Moreover, in most cases, ‘One article as one document’ version of dataset tested over test data has performed better result with 5-NN algorithm. However, this is not the case with other algorithms. The confusion matrices results of Naïve Bayes algorithm has given mixed results. It failed with ‘Only TF-IDF’ experiments but performed very well with “WordNet” experiments. 1-NN has performed worthy over randomized versions of dataset. But, it gives inconsistent results over test data.

Of all the experiments performed in carrying out the dissertation, experiments with WordNet have topped the chart. Might be because; we have compressed words with similar meaning into a single word. POS tagging experiments has also done well and this proves that words are very much important to determine the meaning/ sentiment flowing in the news articles.

To conclude, I would say, with a couple of more experiments, the accuracy of algorithms can be increase and I can rely on them to invest my money in market to make profit.

8.2. Limitations

Certain limitations which depreciate the accuracy of the results are listed below:

- News articles are written by individuals, they have their own sentiments which might affect the calculations.
- The ambiguity of natural language also affects the accuracy of the results.
- The result can easily be diverted if a person manipulates the news feed or the content of the news.
- The NLTK have some limitations.
• News articles come throughout the day in the market. If the news comes in the morning, then its effect can be clearly seen in the market rather than news which came just 5 minutes before the closing.
Chapter 9: Future work and Recommendations:

- Some news has a greater impact on financial markets than others. So, a proper weight-age should be given to that news. It could be, by creating a Roulette wheel selection of share prices.

- News feeds are produced by a human, so his/her sentiment also comes into play. To avoid this issue, news feeds from different sources should be taken into consideration.

- Certain shares have more weight-age in the stock market price. The sentiment should be calculated based on that.

- Certain shares are grouped in sectors. News articles related to certain sectors have more weight-age towards calculating the stock market index.

- All the news articles are given equal weightage. However, most recent articles are more important than older articles. So, they should be given more weightage.

- In the part of speech tagging, proper noun, verbs affect should be given varied weightage.
11. References

6. Corne David, Professor, *Heriot Watt University.*
7. Murphy John. Technical references of financial markets
11. Wiering Macro, Department of Artificial Intelligence *University of Groningen*
    techniques. *Proc. of the 37th Hawaii International Conference on System Sciences*,
    Hawaii.
    Banking & Finance*. 25, 1339-1366.
    Contrary Opinion: A Test Using Sentiment Indices In Futures Markets. *Journal of
    Agribusiness*. 21 (1), 1428-1460.
26. Anonymous, Standards and Poor’s, Available at:
    http://www2.standardandpoors.com/portal/site/sp/en/la/page.topic/ratings_au/2,1,1,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.html.
27. Murphy John. Technical references of financial markets
    prices? Evidence from an emerging market. *Journal of International Financial
    *Working Paper, Kenan-Flagler Business School, University of North Carolina*.
    paper, *University of Pennsylvania. Available at SSRN: http://ssrn.com/abstract=831487*
    Market Response to Corporate News G. *The NUS Business School National
    University of Singapore*. 


# 11. Tables and Figures

## Index of Tables

<table>
<thead>
<tr>
<th>S. No</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Example of classifiers provided by Weka.</td>
</tr>
<tr>
<td>2.</td>
<td>POS taggers by NLTK.</td>
</tr>
<tr>
<td>3.</td>
<td>Comparing 1 NN and 5 NN algorithms for one article as a document.</td>
</tr>
<tr>
<td>4.</td>
<td>1 NN algorithm after adding the difference of stock index prices.</td>
</tr>
<tr>
<td>5.</td>
<td>Confusion Matrices: Naïve Bayes algorithm all day article as a document (Only TF-IDF).</td>
</tr>
<tr>
<td>6.</td>
<td>Comparison of 1NN and 5NN on one article as a document with POS tagging.</td>
</tr>
<tr>
<td>8.</td>
<td>Comparison of 1-NN and 5-NN of all day article as a document with POS tagging.</td>
</tr>
<tr>
<td>9.</td>
<td>Confusion Matrices: Naïve Bayes algorithm all day article as one document (With POS tagging)</td>
</tr>
<tr>
<td>10.</td>
<td>Comparing 1 NN and 5 NN algorithms for only verbs</td>
</tr>
<tr>
<td>11.</td>
<td>Comparing 1 NN and 5 NN algorithms for all day article as a document (with only verbs)</td>
</tr>
<tr>
<td>12.</td>
<td>Naïve Bayes algorithm all day article as one document only with verbs</td>
</tr>
<tr>
<td>13.</td>
<td>Comparing 1 NN and 5 NN algorithms for one article as a document with WordNet.</td>
</tr>
<tr>
<td>15.</td>
<td>Comparing 1 NN and 5 NN algorithms for all articles of a day as a single document with WordNet.</td>
</tr>
<tr>
<td>16.</td>
<td>Confusion Matrices: Naïve Bayes algorithm all day article as one document with WordNet.</td>
</tr>
<tr>
<td>17.</td>
<td>Comparing results of “One article as a document” experiments for all algorithms</td>
</tr>
<tr>
<td>18.</td>
<td>Comparing results of “All articles of a day as a single document” for all algorithms</td>
</tr>
</tbody>
</table>

## Index of Figures
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1 NN algorithm after adding the difference of stock index prices.</td>
</tr>
<tr>
<td>2.</td>
<td>Naive Bayes Approach on Only TF-IDF encoding scheme.</td>
</tr>
<tr>
<td>3.</td>
<td>Naive Bayes approach on only verbs</td>
</tr>
<tr>
<td>4.</td>
<td>Comparing results of “One article as a document” experiments for all algorithms</td>
</tr>
<tr>
<td>5.</td>
<td>Bar chart comparing results of “All articles of a day as single document” experiments for all algorithms.</td>
</tr>
</tbody>
</table>
**12. Appendix:**

**Tables**

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>Standard probabilistic Naive Bayes classifier.</td>
</tr>
<tr>
<td>J48</td>
<td>C4.5 decision tree learner.</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>Back propagation neural network.</td>
</tr>
<tr>
<td>SVM Reg.</td>
<td>Support vector machine for regression.</td>
</tr>
<tr>
<td>IB-k</td>
<td>K-Nearest Neighbor classifier.</td>
</tr>
</tbody>
</table>

**Table 1:** Example of classifiers provided by Weka

<table>
<thead>
<tr>
<th>NgramTagger</th>
<th>A tagger that chooses a token's tag based on its word string and on the preceding n word's tags.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrigramTagger</td>
<td>A tagger that chooses a token's tag based its word string and on the preceding two words' tags</td>
</tr>
</tbody>
</table>

**Table 2:** POS taggers by NLTK

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy (% Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>307</td>
<td>34</td>
</tr>
<tr>
<td>5 NN</td>
<td>(90%)</td>
<td>(10%)</td>
</tr>
</tbody>
</table>

**Table 3:** Comparing 1 NN and 5 NN algorithms for one article as a document (Only TF-IDF)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy (% Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>27(90%)</td>
<td>3(10%)</td>
</tr>
<tr>
<td>5 NN</td>
<td></td>
<td>66.6</td>
</tr>
</tbody>
</table>
Table 4: Comparing 1 NN and 5 NN algorithms for all day article as a document (Only TF-IDF)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy (% Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>300(88%)</td>
<td>41(12%)</td>
</tr>
<tr>
<td>5 NN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Confusion Matrices: Naive Bayes algorithm all day article as a document (Only TF-IDF)

Table 6: Comparison of 1NN and 5NN on one article as a document with POS tagging

Table 7: Confusion Matrices: Naive Bayes algorithm: One article as a document (With POS tagging)
Table 8: Comparison of 1-NN and 5-NN of all day article as a document with POS tagging

<table>
<thead>
<tr>
<th>Confusion Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
</tr>
<tr>
<td>u</td>
</tr>
<tr>
<td>d</td>
</tr>
</tbody>
</table>

Table 9: Confusion Matrices: Naive Bayes algorithm all day article as one document
(With POS tagging)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy (% Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>46</td>
<td>18</td>
</tr>
<tr>
<td>5 NN</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Comparing 1 NN and 5 NN algorithms of one article as a document for only verbs

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy (% Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>5 NN</td>
<td>66.1</td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Comparing 1 NN and 5 NN algorithms for all day article as a document (with only verbs)

<table>
<thead>
<tr>
<th>Confusion Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
</tr>
<tr>
<td>u</td>
</tr>
<tr>
<td>d</td>
</tr>
</tbody>
</table>

Table 12: Confusion Matrices: Naive Bayes algorithm all day article as one document only with verbs
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy ( in % Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>307</td>
<td>34</td>
</tr>
<tr>
<td>5 NN</td>
<td>68.7</td>
<td>69.7</td>
</tr>
</tbody>
</table>

Table 13: Comparing 1 NN and 5 NN algorithms for one article as a document with WordNet

<table>
<thead>
<tr>
<th>Confusion Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>u</td>
</tr>
<tr>
<td>d</td>
</tr>
</tbody>
</table>

Table 14: Confusion Matrices: Naive Bayes algorithm one article as one document with WordNet

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Instances (in %)</th>
<th>Accuracy ( in % Correct Classified)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
<td>Test set</td>
</tr>
<tr>
<td>1 NN</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>5 NN</td>
<td>68.7</td>
<td>71.1</td>
</tr>
</tbody>
</table>

Table 15: Comparing 1 NN and 5 NN algorithms for all day articles as one document with WordNet

<table>
<thead>
<tr>
<th>Confusion Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>u</td>
</tr>
<tr>
<td>d</td>
</tr>
</tbody>
</table>

Table 16: Confusion Matrices: Naive Bayes algorithm all day article as one document with WordNet
### Table 17: Comparing results of “One article as a document” experiments for all algorithms

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Algorithm</th>
<th>Accuracy (in %)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Train-test set (Avg)</td>
<td>Test set</td>
</tr>
<tr>
<td>Only TF-IDF</td>
<td>1-NN</td>
<td>62.6</td>
<td>58.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>62.4</td>
<td>56.1</td>
<td></td>
</tr>
<tr>
<td>TF-IDF + POS (twice weight to verbs)</td>
<td>1-NN</td>
<td>68.45</td>
<td>65.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>71.55</td>
<td>67.68</td>
<td></td>
</tr>
<tr>
<td>TF-IDF + only verbs</td>
<td>1-NN</td>
<td>70.7</td>
<td>66.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>68.4</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>TF-IDF + WordNet</td>
<td>1-NN</td>
<td>70.6</td>
<td>67.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>72.4</td>
<td>61.4</td>
<td></td>
</tr>
</tbody>
</table>

- All day articles as one document

### Table 18: Comparing results of “All articles of a day as single document” experiments for all algorithms

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Algorithm</th>
<th>Accuracy (in %)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Train-test set (Avg.)</td>
<td>Test set</td>
</tr>
<tr>
<td>Only TF-IDF</td>
<td>1-NN</td>
<td>61.75</td>
<td>60.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>65.75</td>
<td>61.68</td>
<td></td>
</tr>
<tr>
<td>TF-IDF + POS (twice weight to verbs)</td>
<td>1-NN</td>
<td>67.43</td>
<td>67.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>69.1</td>
<td>64.3</td>
<td></td>
</tr>
<tr>
<td>TF-IDF + only verbs</td>
<td>1-NN</td>
<td>62.2</td>
<td>66.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>75.1</td>
<td>69.75</td>
<td></td>
</tr>
<tr>
<td>TF-IDF + WordNet</td>
<td>1-NN</td>
<td>74.1</td>
<td>66.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-NN</td>
<td>69</td>
<td>68.4</td>
<td></td>
</tr>
</tbody>
</table>
Figures

Figure 1: 1 NN algorithm after adding the difference of stock index prices

Figure 2: Naive Bayes Approach on Only TF-IDF encoding scheme
Figure 3: Naive Bayes approach on only verbs

Figure 4: Bar chart comparing results of “One article as a document” experiments for all algorithms
Figure 5: Bar chart comparing results of “All articles of a day as single document” experiments for all algorithms