S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

2019

Project Title	Stock Price Fluctuation Prediction Based on Text Analysis of Stock Market News
Student Name	H.U.J.S. Ariyarathne
Registration No. & Index No.	2014/MCS/001 14440016
Supervisor's Name	Dr. Ajantha Athukorala

For Office Use ONLY

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : H.U.J.S. Ariyarathne
Registration Number : 2014/MCS/001
Index Number : 14440016

_____

Signature: Date:

This is to certify that this thesis is based on the work of
Mr./~~Ms~~. H.U.J.S. Ariyarathne
under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by :
Supervisor Name : Dr. Ajantha Athukorala

_____

Signature: Date:

Stock Price Fluctuation Prediction Based on Text Analysis of Stock Market News

A dissertation submitted for the Degree of Master of Computer Science

H.U.J.S. Ariyarathne
University of Colombo School of Computing
2019

Abstract

This thesis is presented with a prediction of stock market price change using news and announcements. Stock Market News provides invaluable information to brokers and investors to take their crucial decisions on investing in stock market. Most of modern stock market data dissemination software systems and tools provides various types if indicators to facilitate these decisions. It includes real-time indicators and historical graphs of stock prices. Adding more indicators to this list is an added advantage to any of those systems due to the impact on stock market activities. There are several researches done on same subject by several people. However, most of those investigations were limited to a specific region in the world (US stock markets), specific resources (Yahoo finance, Google trends etc) and specific prediction parameters (price trend positive or negative).

The aim of this project is to build a data model and prediction system which facilitate another dimension of indicators. It would provide information on the stock price trend for few days ahead, based on current stock market news information. This research includes enhanced prediction methodology to existing methods by providing additional information such as the number of days that the trend will be retained.

Since this is a text analysis-based project, few classification algorithms have been used. Weka tool has been used to feature extraction process and classification. Results were obtained as a comparison between each algorithm used. Best training data set was identified using K-Fold Cross Validation techniques applied to all algorithms. The best results were obtained using Random Forest algorithm with a significant accuracy level.

Acknowledgements

I would like to use this opportunity to express my gratitude to everyone who supported me throughout this course and this project. First, my deep gratitude and heartfelt thanks to Dr. Ajantha Athukorala for his guidance, monitoring and constant encouragement throughout the project as my supervisor and Dr. Kasun Gunawardena for providing necessary instructions and infrastructure for the project as the coordinator. My sincere thanks go to Mr. Ruwan Aluthgedara and Mr. Gayanath Senanayake from DirectFN Pvt Ltd who provided me necessary data and sample simulation programs related to my project. Also, the help given by all my seniors in Mitra Innovation is highly appreciated allowing me to work on project without any obstacles. Furthermore, my special thanks go to my colleagues Lakmal and Sampath, who helped me in numerous ways through the course and this project. Finally, I thank almighty, my parents and wife for their blessings, endless support and encouragement in my studies and work, without which this course or project would not be possible.

Table of Contents

List of Figures 4

List of Tables 6

Chapter 1: Introduction 8
1.1 Motivation 8
1.2 Aims and Objectives 8
1.3 Scope 9
1.4 Limitations 10
1.5 Structure of the Thesis 10

Chapter 2: Background 12
2.1 Stock Market Basics 12
2.1.1 Stocks 12
2.1.2 Stock Market 12
2.1.3 Stock Market Announcements 13
2.1.4 Stock Market News 14
2.1.5 Fundamental Information 14
2.1.6 Technical Information 14
2.1.7 Efficient Market Hypothesis (EMH) 15
2.2 Sources of Stock Market News 15
2.2.1 Saudi (Tadawul) Stock Exchange 16
2.2.2 Reuters 16
2.2.3 Yahoo Finance 17
2.2.4 Bloomberg 17
2.3 Stock Prices and News 17
2.3.1 Open Price 18
2.3.2 Close Price 19
2.3.3 Correlation between news/announcements and stock prices 19
2.4 Previous Researches 20

Chapter 3: Analysis and Design 24
3.1 Data Pre-Processing 25
3.2 Weighting News Items 26
3.2.1 Weighting Algorithm 30
3.3 Feature Extraction and Data Model 32
3.3.1 Weighing news items 32
3.3.2 Feature Extraction 33
3.3.3 Evaluating different algorithms 33

Chapter 4: Implementation 35

4.1 Persistent Storage 35
4.2 Data Pre-Processing 36
4.2.1 HTML Parsing 37
4.2.2 Stop word removing 37
4.2.3 Stemming 37
4.3 Feature Extraction 39
4.4 Training and Prediction 40
4.4.1 Load data to Weka 40
4.4.2 Attribute selection within classification 41
4.4.3 Selection of Feature Selection method 41
4.4.4 Selection of Test Dataset 44
4.4.5 Prediction with selected attributes 45

Chapter 5: Evaluation and Results 47
5.1 Class distribution 47
5.2 Feature Selection 47
5.3 Initial Classification 48
5.4 Classification with further preprocessing 49
5.5 Further Classification with configuration changes 50
5.5.1 Naive Bayes 50
5.5.2 SMO 51
5.5.3 Bagging 52
5.5.4 J48 52
5.6 Summarized Results and Evaluation 53
Top Performers 53
Performance By Algorithm 54
Performance By Data Set 55
Performance By Algorithm with 10-Fold Cross Validation 56
Performance By Algorithm with 66% Split 57
Performance By Algorithm with 80% Split 58
5.7 Evaluation of Final Data Model 59

Chapter 6: Conclusion and Future Work 61
6.1 Conclusion 61
6.2 Future Work 62

References 63

Appendices 65
Appendix A – Tools and Software 65
Weka 65
Jsoup 65
Exude 65
PtnPlanet 65

OpenNLP 65
Java-ML 66
Appendix B – Code Repository 66
Appendix C – More Classification Results 66
Performance of Naive Bayes algorithm 66
Performance of Naive Bayes Multinomial algorithm 67
Performance of SMO (SVM algorithms) 68
Performance of Bagging 69
Performance of J48 70
Performance of Random Forest 71

List of Figures

Figure 1 : News entry in Saudi Stock Exchange website 16
Figure 2 : News entry in Reuters website 16
Figure 3 : News entry in Yahoo Finance website 17
Figure 4 : News entry in Bloomberg website 17
Figure 5 : The effect of news on stock prices 19
Figure 6 : High level project overview 24
Figure 7 : Detailed design of the project 25
Figure 8 : Categorization of news with weight +1 27
Figure 9 : Categorization of news with weight +2 27
Figure 10: Categorization of news with weight +3 28
Figure 11: Categorization of news with weight -1 28
Figure 12: Categorization of news with weight -2 29
Figure 13: Categorization of news with weight -3 29
Figure 14 : Pseudo code of the weight calculation algorithm 31
Figure 15: Database design 36
Figure 16 : Weka compatible arff file with news data 38
Figure 17 : arff file after applying StringToWordVector filter 39
Figure 18 : Initial data loading to Weka 40
Figure 19 : Attribute selection options in Weka 41
Figure 20 : Weka evaluator and search method options 42
Figure 21 : BestFirst search method configurations 42
Figure 22 : ClassifierAttributeEval search method configurations 43
Figure 23 : Classifier selection option in ClassifierAttributeEval 43
Figure 24 : Weka test data set options 44
Figure 25 : Weka experimenter configuration with multiple algorithms 45
Figure 26 : Weka experimenter test output 46
Figure 27 : Top performance by algorithm 54
Figure 28 : Top performance by dataset 55
Figure 29 : Top performance by algorithm with 10-Fold cross validation 56
Figure 30 : Top performance by algorithm with 66% split 57
Figure 31 : Top performance by algorithm with 80% split 58

Figure 32 : Sample arff file format with empty class value 59
Figure 33 : Sample results with predicted class value 60
Figure 34 : Performance of Naive Bayes algorithm 67
Figure 35 : Performance of Naive Bayes Multinomial algorithm 68
Figure 36 : Performance of SMO algorithm 69
Figure 37 : Performance of Bagging algorithm 70
Figure 38 : Performance of J48 algorithm 71
Figure 39: Performance of Random Forest algorithm 72

List of Tables

Table 1: Sample data structure which contains news words, weight and news ID 32
Table 2: Class breakdown 47
Table 3: Attribute selection results 48
Table 4: Initial classification results 48
Table 5: Detailed classification results 50
Table 6: Naive Bayes additional configuration results 51
Table 7: SMO additional configuration results 51
Table 8: Bagging additional configuration results 52
Table 9: J48 additional configuration results 53
Table 10: Top performance by accuracy 54
Table 11: Top Performance by algorithm 54
Table 12: Top performance by dataset 55
Table 13: Top performance by algorithm with 10-Fold cross validation 56
Table 14: Top performance by algorithm with 66% split 57
Table 15: Top performance by algorithm with 80% split 58
Table 16: Evaluation results for the best data model 60
Table 17 : Performance of Naive Bayes Algorithm 66
Table 18 : Performance of Naive Bayes Multinomial algorithm 67
Table 19 : Performance of SMO algorithm 68
Table 20 : Performance of Bagging algorithm 69
Table 21 : Performance of J48 algorithm 70
Table 22 : Performance of Random Forest algorithm 71

List of Abbreviations

EMH	Efficient Market Hypothesis
ER	Entity Relationship
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IPO	Initial Public Offering
ML	Machine Learning
NLP	Natural Language Processing
RSS	Really Simple Syndication
SEC	Security Exchange Commission
SQL	Structured Query Language
SMO	Sequential minimal optimization
SVM	Support Vector Machine
SVR	Support Vector Regressor
TF-IDF	Term Frequency – Inverse Document Frequency
URL	Unified Resource Locator
US	United States
USD	United States Dollar

Chapter 1: Introduction

Stock market trading is one of main financial activity for a country which decides the economy growth of the country. There are millions of users use real-time stock market information through various data vendors and brokerages to analyses and execute trading in stock exchanges. Trading experts have the knowledge on key price indicators which will affect day today stock prices. They use several factors to analyses and predict market behaviour based on their experience and detailed analysis of past data. Such factors include Daily market data (trade price, volume, turnover), Historical market data, Historical graphs, Financial indicators provided by stock exchanges, Market announcements and Market news.

1.1 Motivation

There were several researches done in the past within stock market data analysis domain to find the relationship between stock market news and stock price changes. Some of those resulted in good findings which can be used in actual stock market data dissemination systems. However, there will be several improvements can be done in terms of types of factors which can be predicted using stock market news. This research focuses on these additional factors and possibility of integrating with an actual stock market data dissemination system.

1.2 Aims and Objectives

This project has a main objective of implementing a forecasting technique to effectively predict market price fluctuations using real-time news information. The analysis will be based on feature extraction and other machine learning techniques with a data set extracted from actual past data.

Following are the main objectives of the project;

1. Analyse past actual data and provide a model to predict future market prices in real-time
2. Analyse existing algorithms and provide an improved algorithm with higher accuracy to achieve more accurate result and predict more factors other than stock price.
3. Compare analysed data with already existing research outcomes and provide enhancements in algorithms to achieve better analysis.
4. Adopt the implementation to an actual stock market data dissemination system with simulated data as real-time data feed.

The implementation of the solution will contain data model which is created based on properly trained past data. Real-time news information will be fed to the data model to predict price fluctuations for one-week time ahead.

1.3 Scope

There are hundreds of stock market data resources in the world which provides both price information and content data (news, announcements etc). This project will use actual data from top 5 stock exchanges in Middle East Region. Both historical data and news data will be taken into consideration during building the data model, training and prediction. This information is directly taken from an actual stock market dissemination system which operates in above geographical region. Therefore, it guarantees the accuracy and the consistency of data with its original form without any adjustments.

Following stock exchange data will be used in this project.

- Tadawul Stock Exchange
- Dubai Financial Market
- Abu Dhabi Stock Market
- Muscat Securities Market
- Doha Securities Market

Analysis will be done on existing algorithms, classification techniques and their improvements. Existing algorithm will be improved to get more accurate prediction.

Comparison will be done between predicted prices and actual price. Data model will be integrated with real-time market data dissemination system to predict prices and it will be demonstrated using simulated market data feed without any graphical interface (as a proof of concept only). Dataset will be taken from same system, which will contain both news and related historical price data for respective dates to do the prediction and training. Prediction will give positive/negative or neutral direction of price movement based on news content. It will not be a quantitative figure on same.

Once the data model is created, it can be used to predict price fluctuation in real-time. However, incorporating the model with an actual real-time data prediction is out of scope of this project.

1.4 Limitations

This project will provide a valuable input to existing stock market data dissemination systems which will provide predictability of price trend. However, there are limitations in integrating with existing systems due to various reasons like technology differences, limited scope of stock exchanges used in the project etc. Therefore, the project will be demonstrated using analysis and conclusions rather than implementing the prediction logic within an actual system.

Also, there are limitations in finding actual news data disseminated from stock exchanges (not manipulated by third party providers). Therefore, scope is limited to only five stock exchanges.

1.5 Structure of the Thesis

In the following chapters, relevant literatures were discussed and then a system analysis, design and implementation is introduced. Afterwards the prototype of the system is evaluated and discussed.

Chapter 2 gives a background information, discusses and analyse on related work carried out in these areas. In Chapter 3 discusses about the design of the system. There, the architecture,

overall design, data model and prediction. Chapter 4 discusses all the details about the implementation of the system's design discussed under Chapter 3. The evaluation of the system and its results are discussed in Chapter 5 which includes evaluations of different types of predictions and algorithms. As the Chapter 6 is the final chapter discusses the conclusion and future work suggestion about this research area.

Chapter 2: Background

2.1 Stock Market Basics

2.1.1 Stocks

Stock trading is a common term used in world economics related to money flow among different organizations, people and countries. Basic element of this trading is a “stock” which everyone involved with this process own. A stock is a type of security that signifies ownership in a corporation and represents a claim on part of the corporation's assets and earnings. [18]. People can buy stocks which are published by the company and, they can sell their purchased stocks in an appropriate time which they can get a profit from selling their stocks.

There are two main types of stocks available. One type is common stocks which owners have privilege to vote in company’s meetings and to receive dividends. Other type is preferred stocks which owners do not have right to vote like in common stocks holders, but they will be having higher claim on assets and earnings than common stockholders.

Owner of stocks in a company claims that the person is a shareholder of the company. In other words, shareholder has the ownership of the company with respect to the percentage of shares he/she owns. For example, if a person has 100 stocks out of 1000 total stocks published by the company, that person has ownership of 10% of the company assets and earnings.

2.1.2 Stock Market

Stock Market is a place where all shareholders and companies list their stocks for buying and selling which is governed by Security Exchange Commission (SEC) of each country [19]. It consists of multiple stock exchanges/markets. Shareholders may be individuals as well as investment companies/brokers.

There are two types of markets; Primary Market is the place where initial listing of company shares which goes for public (Initial Public Offering - IPO) is done. Institutional investors buy

most of these shares through investment banks. All other trading happens within the secondary market including day today buying and selling of stocks. This includes trading from both institutional and individual investors.

2.1.3 Stock Market Announcements

Stock trading has become a major investment opportunity for not only investment companies but also for individuals who are searching for more investment options. The number of investors and companies listed in stock exchanges are increasing day by day. While trading is becoming active, the requirement of information flow from exchange to its clients has become a equally major requirement. Most of exchanges use electronic trading engines which consist of software systems with outstanding rate of transactions. Same software systems have capabilities of dispatching information regarding the trading.

One of most important stock trading related information is stock market announcements. Generally, an announcement is a piece of article which consists of a latest update of any stock listed in the stock exchange. Typical example of a stock market announcement which is issued at trading suspended for temporarily for a company in the stock exchange. This will be extremely crucial information for investors to take their decisions and therefore it should be received by them within near zero time.

Other usage of announcement is to analyse past trading information to predict investment options for trading experts. They usually become interested on corporate actions and stock dividends related announcements for stock which occurred during past few years.

Next type of announcement is Market Announcements which does not related to any stock but contains information related to the whole stock market. Most of cases, the information included in market announcements are related to all companies listed in the stock market as well as all investors involved in trading.

2.1.4 Stock Market News

The main difference between announcement and news is that rather than having information limited to stock exchange, news contains information related to the country's economy and its influence with the company. This information is more related to the financial decisions of the government of the country which affects stock exchange and related activities. This information is provided either by stock exchanges itself or independent news agencies. Some of those worldwide news agencies include Dow Jones, Reuters, Bloomberg etc. People use this news information to take their important decisions on stock market investments.

There are two types of information that can be predicted using past data of stock market system.

2.1.5 Fundamental Information

News and announcements are called Fundamental Information. Predicting technical information using fundamental information is a complex process which involves lot of learning and prediction algorithms.

Announcements are mainly used by stock exchanges to provide valuable information during market trading hours. That information contains suspending/resuming trading for stocks, listing/delisting of stocks, applying splits for prices etc.

The content of these news will have direct effect in stock price fluctuations and other market activities. But sometimes, with the busy working schedule, investors and traders will not focus on these news items and their contents in details. It leads to missing some valuable information about future trend of stocks which will ultimately result in big losses in terms of trading revenue.

2.1.6 Technical Information

All factors other than news and announcements are numerical values are called Technical Information. Predicting technical information are solely based on mathematical calculations.

2.1.7 Efficient Market Hypothesis (EMH)

EMH denotes stock market prices are essentially unpredictable. This project focuses on a solution for the above problem which it provides real-time indicators for trading users about future price fluctuations based on the content of stock market news items. In actual scenario, it will take time (sometimes few days) to take the change effective to the price fluctuation, but based on this methodology, it can be predicted at the time of news/announcement published.

There are many researches done on the same subject. Most of factors considered for these researches are not valid today due to drastic changes in technology improvements and competition for investing. Most of stockbrokers and investment firms have developed and published many tools to analyse past data and predict future values. But most of those are proof of concepts and not implemented as a real integration to any actual stock market dissemination system.

2.2 Sources of Stock Market News

There are several sources of stock market news available for investors and customers. Most of stock exchanges operating all around the world provide their own news feeds through different types of transport mediums such as text data feeds, RSS feeds, Web Services, Email notifications etc. On the other hand, there are several other words recognised news providers such as Bloomberg [1], Dow Jones [2], Yahoo Finance [3] etc. who act as intermediate between users and stock exchanges. They take news information from stock exchanges and provide to customers in their own methodologies in terms of contents, categorizations, subscriptions and latency.

For example, Dow Jones news has two types of news feeds. One is real-time feed which provides news information at the same time it was published from sources. Users need to pay higher value for this type of news subscription. Other type is delayed news data, which is less expensive than previous, but it will not transmit in real-time. It will be delayed by 5 mins as default. Also, there are different schemes that users can select based on delay time and the amount charged.

Following figures show how different types of stock market news vendors publish their news articles in web sites which relates to Saudi Arabian Economy.

2.2.1 Saudi (Tadawul) Stock Exchange

Figure 1 shows a news item from Saudi Stock Exchange official web site.

Figure 1: News entry in Saudi Stock Exchange website

2.2.2 Reuters

Figure 2 is a news item from Reuters web site.

Figure 2: News entry in Reuters website

2.2.3 Yahoo Finance

Figure 3 shows Yahoo Finance web site extract.

Figure 3: News entry in Yahoo Finance website

2.2.4 Bloomberg

Figure 4 is a news item from Bloomberg web site.

Figure 4: News entry in Bloomberg website

2.3 Stock Prices and News

Stock exchange has vast number of different information which are used in daily trading activities. This information is valuable in different ways on analysis. Some of this information is required to predict stock prices within next few days, weeks or months even. There are lot of analytical algorithms built on top of basic calculations from stock market data. Other information is related to create statistics for history of stock markets. These are extremely useful in investing in stock markets and those are used by experienced investors and brokers of stock exchanges.

For this research, we mainly focus on following fields in stocks which are disseminated from daily stock market data feeds.

- News and Announcements
    - Date
    - Headline
    - Body
    - Exchange
    - Symbol
- Historical Prices
    - Date
    - Open Price
    - Close Price

Most of above feels related to news and announcements are self-descriptive. Those are used in general vocabulary rather than in stock market specific. However, Open Price and Closed Price fields related to historical prices are specific to stock market trading. It would be required to have more knowledge on those fields before going into the detailed analysis.

2.3.1 Open Price

Open price of a stock symbol specifies the price which the first trade of the day happened. For example, if the first trade of the day for the symbol ABC occurred at the price of USD 30, the Open Price of that symbol for the day is 30. This value will never be changed during the rest of the trading day because there will not be any other first trade for that symbol for the day. However, some stock exchanges have different rules to define these prices and those may be adjusted at the end of the trading day based on these rules. In such scenario, Open Price will not be the exact price which the first trade occurred for the day.

There are some standards in stock exchange regarding prices. One of that behaviour is that most of the time, first trade of the day occurs not at the price of the last trade on previous trading day for that symbol. Theoretically, it should be same unless there is not an overnight change in financials of the company. But most of the time, it differs because of some rules and adjustments of prices by stock exchanges before the trade begins on next day.

2.3.2 Close Price

In general, Close Price means the price of the last trade occurred for the day of the stock symbol. But most of stock exchanges adjust this price based on several theorems and algorithms to have the trading continues in subsequent days. This price adjustments normally happens few minutes after the stock market is closed for the day. Therefore, the adjusted prices are updated in data feeds to ensure all investors and dealers are aware of prices changes.

2.3.3 Correlation between news/announcements and stock prices

Having extracted all required information, it is important to identify relationships among this information. It helps to establish a good data model which will ultimately provide accurate results in prediction.

Stock open and close prices can be used to calculate the price trend for the stock. However, close price is more effective to get an understanding on factors that applied during the trading day. For example, if there is a positive news published during the trading session, it will affect the close price of that stock, rather than the open price of the next trading day. Figure 5 shows this price change example based on positive and negative news items.

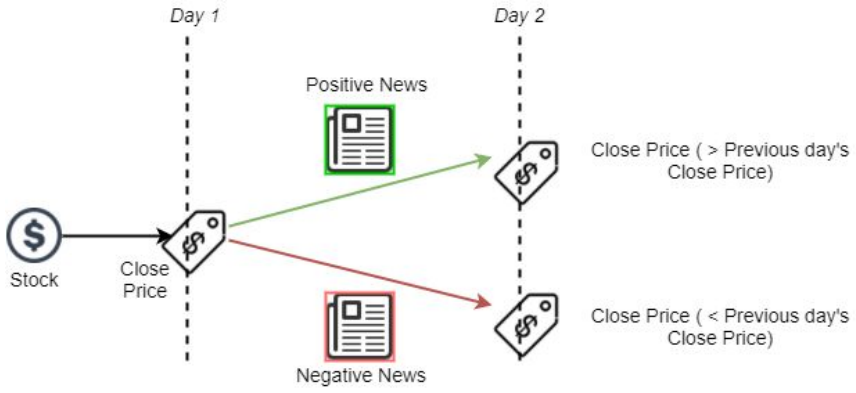

Figure 5: The effect of news on stock prices

2.4 Previous Researches

Jerry Chen and the team conducted similar research [4] which aimed to validate whether Efficient Market Hypothesis (EMH) is still valid with latest stock market behaviours. It took 40 news items from 30 small companies which were extracted from Google News. Scoring of words was done using "*Sentementr*". Based on the score of individual words, it gave a score for the whole article. Subsequently, related price information was considered before 3 days and after 7 days of the article published date to create the data model for prediction. Bid and ask prices were taken into the consideration as price information. Research could not refuse EMH using the scope of the project. It requires more broader scope to finalize the validity of EMH.

There were few drawbacks identified with this research. Bid and ask prices are used for the data model preparation. But those price fields do not directly reflect trading prices. Also, it had very small data set to predict price fluctuations. Results can be biased with some factors when this type of small dataset is used. Apart from that, research has been conducted using Google News only. It is also a disadvantage to get accurate data. Finally, there was no data cleansing done before applying to the training and analysing. These issues will be addressed with my research based on same topic.

In another research, Steven Hesten and Nitish Sinha [5] examine the stock returns predictability using Reuters news and sophisticated neural network. Results showed that news over one day have high predictability for 1 to 2 days. When the aggregated news over a week, it produced dramatic improvement of predictability. Positive news affected only for a week while negative news was affecting for a quarter. This research based on stock returns (turnover) values of stocks, not the trading prices. My research will be based on stock price (increase or decrease) which will reduce above errors.

The data set was tagged with different topic codes which explains the relevance of the news article. STX, RES and MRG are most commonly used codes. STX indicates addition and deletion of stocks. RES indicates all corporate financial results while MRG indicates mergers and acquisitions. Other codes are reserved for economic news.

In the implementation, sentiment engine first categorizes words into its root words (e.g. gone, went, goes categorized as go). Once tone is identified it goes through a three-layer back propagation neural network. The results show that positive sentiment affects stock returns for only two or three weeks while negative sentiments effect for a whole quarter.

This research is more based on predicting long term stock returns by accumulating news for a period of one week. In my research, it will be short term prediction using real-time news items without waiting for accumulated news. Also, short term price fluctuations are more effective on day today trading activities rather than long term influences.

Selene Yue Xu has done another research on stock price prediction [6] using information from Yahoo Finance and Google Trend. In this paper, Selene aimed to combine conventional time series analysis techniques with information from google trend and yahoo finance web sites to predict weekly changes in stock prices.

Both fundamental and technical data was collected from internet. Fundamental data was in the form of news articles and analyst opinions. Technical data consisted of stock prices. Each news item was given a rating of either +1 or -1 based on influence on stock price. In this research, Selene has used only the data from "Apple Inc." stock. Also, the data available in google and yahoo web sites are not comprehensive and therefore the calculations may not be 100% accurate. It can be overcome by using actual stock market data and it should not depend on 3rd party news website for critical price data. Third party news web sites may have modified content according to their requirement of space, bandwidth etc. Therefore, it is essential to use actual news and announcements disseminated through market data feed from stock exchange.

Another research has done by Kalyani Joshi and the team [7]. It has used 3 phase models where in phase 1, it for text processing and set polarity to each news item. In phase 2, it builds the data model using above analysis. In final phase, it analyses the relationship using several graphs. Text processing included removing common words which do not affect process and update words in more generic terms (e.g. developed, development, developing can be considered as "develop").

To detect sentiments, Bag of Ward technique is used. It contains positive and negative words which will be used to evaluate a news article. For classifier, SVM, random forest and Naive Bayes classifiers are used. Evaluation is done for all three algorithms to find the best approach.
While doing the analysis, it showed that SVM classifier performs well for unknown data. Random forest also went well.

Kibum Kim and the team has done another research [8] on the same subject and have suggested a stock price prediction system based on opinion mining and mechanical learning. It has used past one-year data for seven companies. It uses news and twitter as the data input which is analysed by vocabulary analyser and sentiment analysis. Finally, the stock price predictor provides most accurate predicted prices.

Drawback of this analysis is that it has used very limited data set. Also, it has used twitter data to do the analysis, but these data may not contain actual news texts and therefore there is a question regarding the accuracy of the outcome.

Robert P. Schumaker and Hsinchun Chen has done another good research [9] on stock market price prediction using techniques like Bag of Words, Noun phrases and Named entities for textual analysis. These techniques identify patterns in news information and store it in a database. It will be added with stock quote details from stock markets and finally, evaluation matrices are generated using machine learning algorithms like SVR. This data model can be used to predict stock market prices based on different input details.

This is one of the best researches conducted. But still it has not used to calculate prices in real-time with actual data from a stock exchange. The behaviour and performance of the prediction algorithm matters when it is required to use this type of system in actual environment where millions of transactions happen within a second. Therefore, there will be many advantages of having a real-time calculation-based prediction system which provides excellent support for investors to make their decisions.

There were dozens of researches done on stock market data prediction. Some were planned to predict price values while some of those were done to analyse patterns of stock market price

changes. There are very few researches done with target of applying real-time actual market news and announcements. Most of researches have analysed algorithms to find a best possible approach and compared values generated by those algorithms with actual stock market history data.

Also, all previous researches have used public web sites such as google, yahoo, Reuters and many others to gather news and price data. Drawback of this is some of these websites will not publish identical news item which is originally published by the stock exchange. But the investors who are directly dealing with the stock exchange are not using any of these public web sites to see news. They use data terminals given by stock exchange itself of brokerage firms. Therefore, they are taking decision based on actual raw news and announcements received from exchange.

Further, those researches were done based US stock market data and it is based on mostly active few stocks (like Apple, Google etc.). Therefore, is it not 100% independent because most these symbols will have positive news items in most of the time.

To overcome these issues, this research is totally based on actual raw data from stock exchange. To avoid any bias on any specific type of stock, it is considered most active symbols from many exchanges for last 5 years. All the price information was also taken from actual stock market data for these exchanges. By using these information, it is expected to get effective output and unbiased evaluation which also can be applied to real-time data from stock exchanges.

Chapter 3: Analysis and Design

This chapter contains the methodology used in the project to experiment data set using different algorithms and classifiers to predict most accurate information.

High level architecture of the project for building data model, prediction and integration with actual system will be as follows. This includes how it can be integrated with an actual stock market system, which does not include within the scope of the project. Project scope limited to build the best data model and provide as an input to a such system.

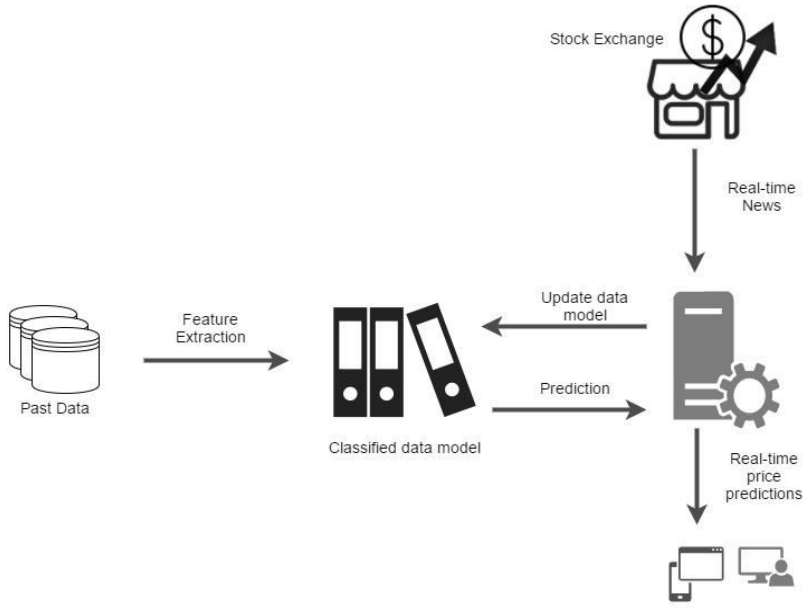

Figure 6: High level project overview

Based on Figure 6, which shows the high level overview of the project, Figure 7 illustrates detailed methodology used in the project to train and predict price information.

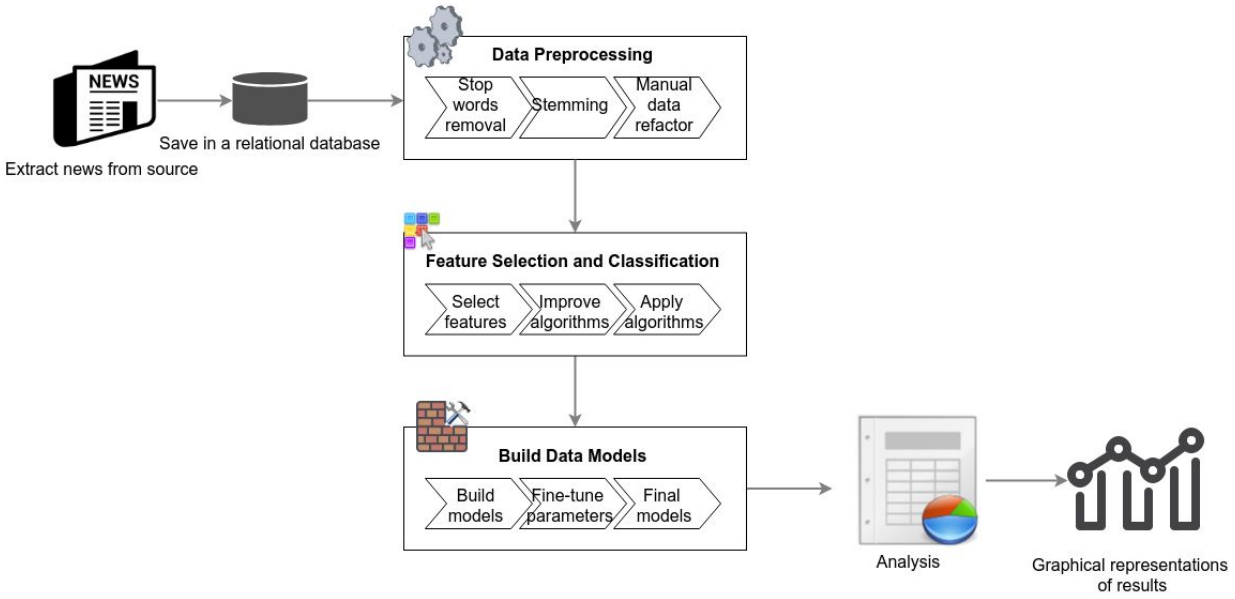

Figure 7: Detailed design of the project

3.1 Data Pre-Processing

Data set is the most important factor in training and prediction-based research project. This project uses actual stock market news items from stock exchanges as mentioned in previous chapter. Since the data contains more specific information related to finance data, it is required to execute lot of data cleaning activities before taking into actual training and prediction.

Data pre-processing includes following set of activities.

1. Removing stop words
2. Stemming
3. Removing unwanted characters
4. Removing short words

Stop word removal has been done using a defined set of words which was taken from www.ranks.ln website [10]. (Refer Appendices) It is a well-known set of words used for most stop word removing activities in researches.

Stemming is the process of generalizing different forms of similar words. For example, after stemming, words like “participate”, “participation”, “participating” will be converges to a single word “particip”. This will allow the prediction does not depend on any specific form of word, but for the meaning of the word.

Data received from stock exchanges may contain some HTML Formatting characters which enables client applications and interfaces to display in a pre-formatted manner. Therefore, it is required to remove all these formatting metadata from news information to extract the actual content related to financial news. Also, it is not correct to have symbols like parentheses, commas, full stops and dollar signs etc. in the data used for prediction. These types of characters also have been removed at this stage. This has been done using simple Java programming.

Words which have shorter length like one or two characters are not considered or required in text classification. Most of these will not have a proper meaning, but those are used as joining words or adjectives in English language. Some of these may have removed using stop word removal process. However, there was an extra process to remove all short words from the data to make sure classification gives proper results.

3.2 Weighting News Items

Weighting news items is done based on historical data analysis. Historical data contains close prices for each stock in stock exchanges. Weight is given for each news item since how many days that stock price has increased or decreased from the date of news published. It is assumed that, if the price is continuously increased/decreased for 3 days (maximum of 3 days has been considered here), after the news item has been published, should have a higher weightage than a news item which caused the price to increase/decrease only for 1 or 2 days. This basis is used for weighting all news items and therefore, possible weight values for any item may vary from -3 to +3 excluding 0. The date of the news item is known as a Hotspot in this project.

Following set of diagrams show how Hot Spots are detected based on historical price changes and how weight is assigned based in price movement during subsequence dates after the Hot Spot.

Weight of the news item would be +1, if the stock price has a negative trend compared to previous day and shows a positive trend only for next day after the news item has been published. Figure 8 shows this transition of the price.

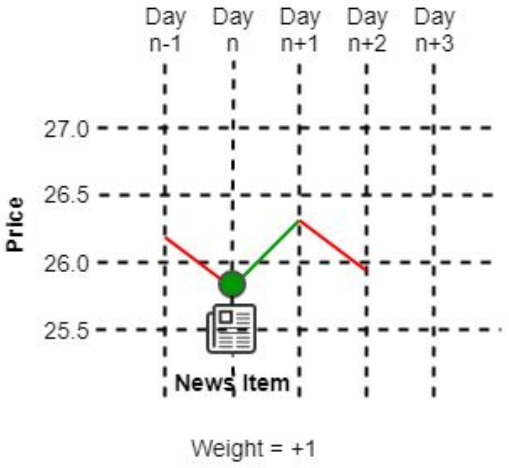

Figure 8: Categorization of news with weight +1

Weight of the news item would be +2, if the stock price has a negative trend compared to previous day and shows a positive trend only for next two days after the news item has been published. Figure 9 shows the variation of price based on this weight.

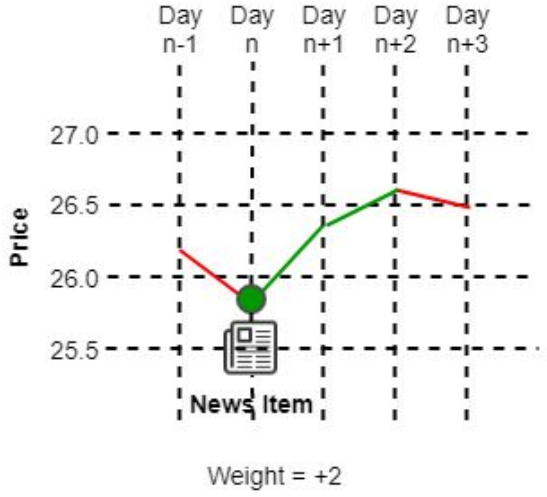

Figure 9: Categorization of news with weight +2

Weight of the news item would be +3, if the stock price has a negative trend compared to previous day and shows a positive trend only for next three days after the news item has been published. This weight calculation logic is shown in Figure 10.

Figure 10: Categorization of news with weight +3

Weight of the news item would be -1, if the stock price has a positive trend compared to previous day and shows a negative trend only for next day after the news item has been published. Figure 11 shows the wight calculation example for -1.

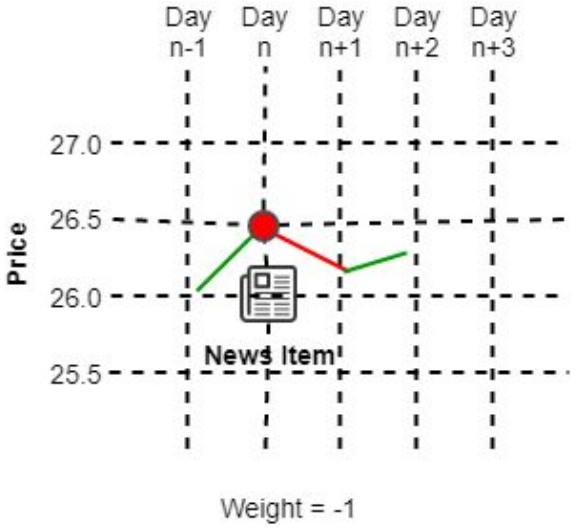

Figure 11: Categorization of news with weight -1

Weight of the news item would be -2, if the stock price has a positive trend compared to previous day and shows a negative trend only for two days after the news item has been published. Figure 12 contains the scenario for weight being -2.

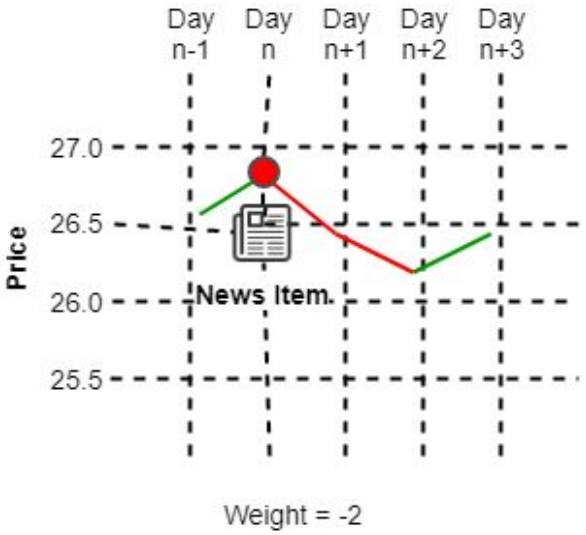

Figure 12: Categorization of news with weight -2

Weight of the news item would be -3, if the stock price has a positive trend compared to previous day and shows a negative trend only for three days after the news item has been published. Figure 13 shows the logic for weight -3.

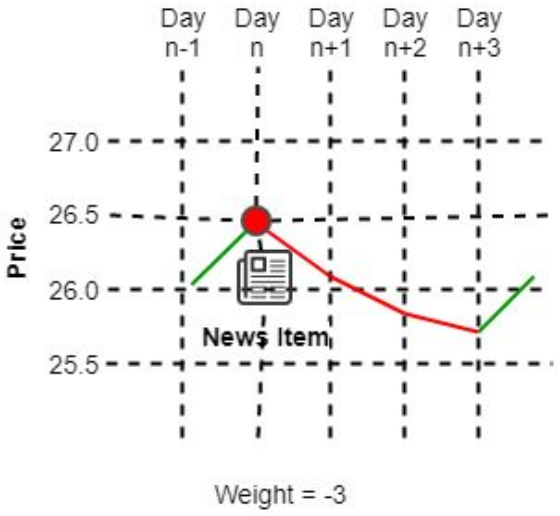

Figure 13: Categorization of news with weight -3

Most of past researches done regarding this subject, do not contain this type of fine grained technique to measure weight of news items. It just checks for price increase or decrease and apply +1 or -1 values. However, in this research, an improvement has been done to categorize news into more granular level and having a range of weight values rather than having two values. With old researches, it can predict only whether price will go higher or lower than today's price. Using the above categorization, it will provide more useful information out from the prediction which includes, not only price going up or down, but also how many days the price increase or decrease will remain same. This would be one of the best indicator that stock market investors and brokers searching for.

3.2.1 Weighting Algorithm

Pseudo code in Figure 14 illustrates the algorithm used to generate above weights for each news article.

It first iterate through all stock symbols in the data set. Each symbol may contain last 5 years data based on its activeness. While iterating these symbol set, algorithm pick the dates when stock price has changed positively or negatively. Those are called Hot Spots.

Then the algorithm takes all hot spots calculated and go through set of calculations within the hotspot. It includes trend direction calculation and price change value calculation. Based on these calculations, each news item for those hotspots dates, will be tagged.

Once each news item has been weighted based on above criteria, those data is saved for the use of classification. However, before classification, it is required to apply feature extraction techniques to select best set of features related to the data model.

```
for each symbol {
        findHotspotDates ();
        for each hotSpotDay(n) {
                previousTrend = day(n)Price – day(n-1) Price
                nextTrend = getNextTrendAndWeight(day(n)).nextTrend
                if (previousTrend is not equal to nextTrendAndWeight.nextTrend ) {
                        Set news item's trend as nextTrendAndWeight.nextTrend
                        Set news item's weight as nextTrendAndWeight.weight
                }
        }
}
findHotspotDates () {
        for each date of news or announcement {
                check next day's close
                if today's close not equal to next day's close {
                        addToHotSpotDays (today);
                }
        }
}
getNextTrendAndWeight(day(n)) {
        priceChangeDay(n+1) = day(n+1)Price – day(n)Price
        priceChangeDay(n+2) = day(n+2)Price – day(n+1)Price
        priceChangeDay(n+3) = day(n+3)Price – day(n+2)Price

        if (priceChangeDay(n+1) > 0) {
                nextTrend = 1
                weight = 1
                if (priceChangeDay(n+2) > 0) {
                   nextTrend = 1
                   weight = 2
                   if (priceChangeDay(n+3) > 0) {
                      nextTrend = 1
                      weight = 3
                   }
                }
             } else if (priceChangeDay(n+1) < 0) {
                nextTrend = -1
                weight = 1
                if (priceChangeDay(n+2) < 0) {
                   nextTrend = -1
                   weight = 2
                   if (priceChangeDay(n+3) < 0) {
                      nextTrend = -1
                      weight = 3
                   }
                }
             }
        return nextTrend, weight
}
```

Figure 14: Pseudo code of the weight calculation algorithm

3.3 Feature Extraction and Data Model

Feature extraction and data model creation with classification is done with various techniques and algorithms to ensure correct prediction methodology is selected with highest accuracy.

3.3.1 Weighing news items

Once the weighing is done based on above approach, refactored data set contains set of words related to each news item and a weight number associated with it. This will allow to create a master words list which represent all distinct words that contained in each of these news items. When feature extraction and classification techniques are applied, it would be required to have Words Master and presence of each word within each news article.

News ID	Weight	Words List
17340771	-1	[adx,fall,lowest,level,bank,sector,loss]
19671069	2	[abnic,profit]
16499782	2	[abu,dhabi,aviat,generat,mln,profit,fy]
17193933	-1	[abu,dhabi,aviat,board,meet,earli,novemb]
17340771	3	[adx,fall,lowest,level,bank,sector,loss]
17387578	-1	[three,stock,benefit,adx,chief,decis]
19289273	-1	[abu,dhabi,aviat,foray,estat,sector]
11754952	-1	[adcb,board,review,financi,juli,th]
15500490	-3	[adcb,board,discuss,agenda,item]
15700435	1	[client,base,rise,adcb,islam,bank,unit]
16178832	-3	[adcb,buy,back,mln,share,reach,regul,maximum,mubash,trade]
16208032	3	[adcb,commit,sustain,martin,scott]
16225106	1	[adcb,profit,growth,boost,lower,provis,expens,mubash,trade]
16225366	1	[global,remain,optimist,adcb,perform]
16225372	1	[nbk,capit,maintain,hold,adcb]
16306596	-3	[ci,affirm,adcb,rate]

Table 1: Sample data structure which contains news words, weight and news ID.

As observed in words list above, some of words have been modified from its original version due to initial data pre-processing techniques. For example, words like “expens”, “provis”,” regul” and “estat” are modified versions of their original words contained in actual news items from stock exchange.

3.3.2 Feature Extraction

Once the master words list is created, feature extraction techniques can be used to identify best suitable words as features of the data set. This will provide most optimum result by using most affecting set of words in prediction as well.

There are several ways of extracting features from a defined data set. In this project, following techniques have been used and tested for the optimum technique which gives best prediction results.

Weka is used as the tool for feature extraction. Since the data contains textual information, it is required to convert it to numerical representation in order to apply for most of algorithms. Therefore, as the first step, original data set was converted to numerical representation using 'StringToWordVector" filter which can be found under unsupervised attribute filter.

Feature selection was done with different combinations of search methods and evaluators. In order to limit the scope, only single evaluator is used in each search method available in Weka, and also a single algorithm is used (Naive Bayes) as the algorithm in search method whenever applicable. Next chapter contains details of different algorithms, search methods and evaluator used for classification and attribute selection within this project.

3.3.3 Evaluating different algorithms

As mentioned in previous section, this project used different search methods and evaluators to choose the best set of features for the data model. Therefore, for different combinations of evaluator, search methods, different feature sets were generated. Following options used as combinations to build the classification models.

Algorithms:

1. Naive Bayes
2. Naive Bayes Multinomial
3. SMO
4. Bagging
5. J48

6. RandomForest

Search Methods:

1. Ranker
2. BestFirst
3. GreedyStepwise

Evaluator:

1. ClassifierAttributeEval(CAE)
2. CfsSubsetEval(CLSE)
3. GainRatioAttributeEval(GRA)
4. InfoGainAttributeEval(ING)
5. PrincipalComponents(PCA)
6. SymmetricalUncertAttributeEval(SYM)
7. WrapperSubsetEval(WSE)

Experiment Type:

1. 10-Fold cross validation
2. 66% split
3. 80% split

Some of these search methods can be used with specific set of evaluators. It was considered when executing the classification and therefore every search method was not used with every evaluator.

All the results and feature lists generated from weka were captured and saved for future use. It does not need to re-run the the feature selection again in order to apply the classification in real-world scenario.

Chapter 4: Implementation

There are several sections in this project required new implementations other than using standard tools and libraries. This chapter describes the technologies and implementation methods used in each stage of the project.

4.1 Persistent Storage

News and announcement information taken from stock exchanges are stored in MySQL database for further processing. This information is kept in its original format without applying any filters or corrections to use in different classification projects or improvements in future. Following is the database structure for the whole database used in the project.

Figure 18 shows the ER diagram for the basic tables in the database. These data are not accessed frequently because all classification and storing data will be done in-memory/file system, Also, it does not have any relationship each other. Therefore, no foreign keys are defined.

News and Announcements tables contain raw data taken from stock exchanges. There are the text data of articles without having any formatting characters and HTML tags. However, these data is not cleaned and ready to use in classifications. It will need techniques like stop word removal, stemming etc. to make the data experiment ready.

History table contains historical information related to all stocks listed in selected stock exchanges. These data are available for last 5 years as same as for News and Announcements. The information stored in this table contains main fields like Date, Open Price and Close price.

Hotspots table was used to store hotspot information retrieved during Weighting News Items process. It contains the dates which stock price has been changed the trend direction. Also, it includes news and announcement availability for the date and the weight.

Classifier table was used to store data related to word count and weight. These data were used in different types of classification techniques. For example, to select features from the data set, average weight of words was used without having a tool like WEKA to do the task. It was just an experiment to check any improvements in the implementation. At the end, this table was not used because the feature extraction was done using WEKA tool.

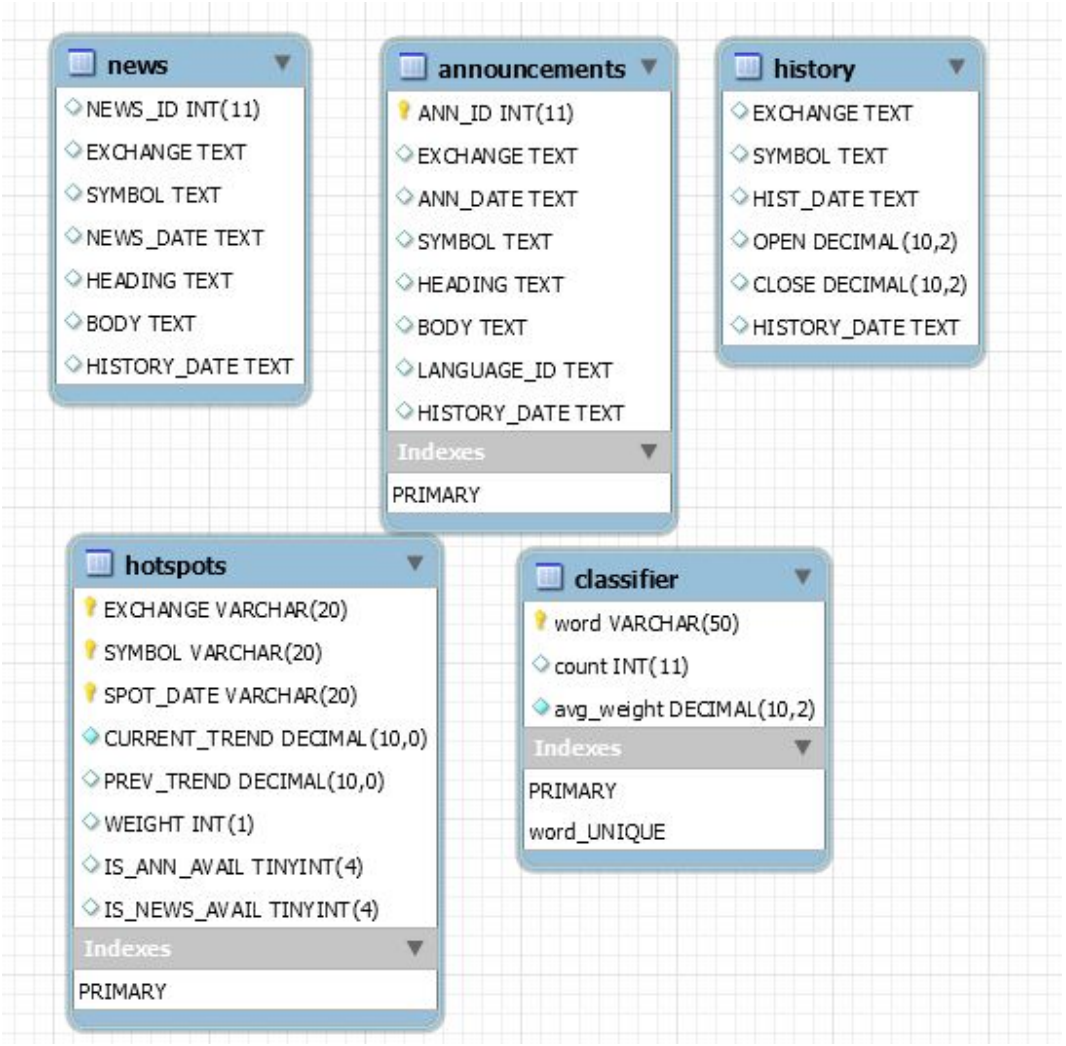

Figure 15 : Database design

4.2 Data Pre-Processing

There were different types of data processing required to execute at the initial data preparation stage due to the unwanted data included in original data set obtained. For example, original news articles taken from stock exchanges contained HTML characters, formatting characters, stop words etc. Following libraries and resources were used to clean-up and remove all

unwanted data from the data set before applying into feature extraction and classification algorithms.

There are many different libraries and programs available to execute this type of text cleaning processed. This project uses Java based applications and libraries which are available to use free of charge.

4.2.1 HTML Parsing

Jsoup[12] library was used to remove HTML tags and unwanted HTML characters in the original news and announcements data. This was resulted in row data set with only printable and processable characters for the data analysis.

4.2.2 Stop word removing

Exude [13] library was used to remove stop words from news and announcements raw data. This library takes a String as the input parameter and provides a String Array containing the list of refactored words (removing any stop words). Advantage of this library is that, it returns an Array by default. Therefore, it was easy to proceed with further processing in the classification. Otherwise, it must be converted to an Array from the project to get the list of words.

4.2.3 Stemming

OpenNLP [14] was used to stem words in news and announcement articles. Snowball Stemmer is the class used to do the process. It accepts a word as the input and output the stemmed word.

Detailed list of applications and libraries used in the project can be found in Appendix B.
Even though these methods used to clean data extensively, it was observed that, classification was not successful due to some specific data contained in news items. These data including;

1. Company Names
2. Stock Market Names
3. Financial terms

Therefore, at initial classification, most of successful prediction percentages were obtained around 30% - 40%. In order to get more accurate results. it was required to remove above common set of words explicitly. The method followed to remove those words was including those words in stop words list. Then the stop word list contains words originally obtained from the web site and words added manually after running through first stage of classification and feature selection.

After these data cleansing methods applied, refined data set was used to create .arff file which is required by Weka in a pre-defined format.

A first few lines of the .arff file generated can be found in Figure 16 below.

```
@relation train

@attribute Document string
@attribute class-name {one-up,two-up,three-up,one-down,two-down,three-down}

@data

"adx fall lowest level bank sector loss ",one-down
"abnic profit ",two-up
"abu dhabi aviat make mln raises",one-up
"abu dhabi aviat board propos reward",two-up
"abu dhabi aviat board propos ",two-up
"abu dhabi aviat gener mln profit acquiring",two-up
"abu dhabi aviat board meet earli novemb negative",one-down
"adx fall lowest level bank sector loss ",three-up
"three stock benefit from adx chief decis ",one-down
```

Figure 16 : Weka compatible arff file with news data

It has two attributes “Document” and “Class-name”. Document specifies a list of words in a particular news item which was retained after data cleansing process. It will not give any meaning by looking at it, because it has gone through several data refactoring process from its original form. Class name attribute provides the actual classification class of each news item which calculated at initial data processing stages using a Java program.

This .arff file was created from a Java program using a pre-defined format of .arff files accepted by Weka tool. However, in order to continue with classification process, these textual data should be converted to a form of work vector and into a numerical form.

In order to convert the data into a numerical vector, special filter was used in Weka. It was called “StringToWordVector” filter which can be found under unsupervised attribute section of filters. Once this filter has been applied to the original data set, it looks like in Figure 17 below.

```
@relation
'train-weka.filters.unsupervised.attribute.StringToWordVector-R1-W100000-prune-rate-1.0-C-T-I-N
0-L-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.WordsFrom
File                                                                    -stopwords
/home/jagatha/Documents/weka/final/stoplist.txt-M1-tokenizerweka.core.tokenizers.WordTokenizer
-delimiters                \"                \\r                \\t.,;:\\\'\\\"()?!/
-_><&#@$\\\%|\\\\^*\"-dictionary/home/jagatha/Documents/weka/final/dictionary'

@attribute class-name {one-up,two-up,three-up,one-down,two-down,three-down}
@attribute acquir numeric
@attribute acquiring numeric……..

@data
{0 one-down,32 2.758717,50 2.615926}
{0 two-up,60 1.321333}
{52 3.191046,62 2.987426}
{0 two-up}
{0 two-up}.....
```

Figure 17 : arff file after applying StringToWordVector filter

Now the textual data has been converted into a numerical vector format which will be used in classification in next stage of the process.

4.3 Feature Extraction

Feature extraction was done through Weka tool. Here, it is required to provide data set in arff format as explained in previous section. This is the format that Weka tool accepts for classification and feature extraction process. Once converted, this file was set as the input to Weka.

There are several options in Weka for feature selection. It can select Evaluation Method as well as a Search Method when executing feature selection for each data set. Few combinations of these options were used in this project for feature selection and separate data sets were created for each of those combinations. Those data sets were used for classifications. More details can be found in next section which includes Training and Prediction as well as feature selection.

4.4 Training and Prediction

The main objective of this project is to create data model which predicts stock market price trend using news data. At this stage, all background work has been completed and training and prediction needs to be done.

4.4.1 Load data to Weka

Once the arff file is created by applying Weka filter "StringToWordVector", it was loaded to Weka as in Figure 18. It shows the attributes, number of classes and classes distribution within the dataset. As discussed with the supervisor, there were some changes done in order to keep the class distribution uniformly throughout the dataset. It was done by including more data with classes which has less frequency and getting those into a level as other class frequencies.

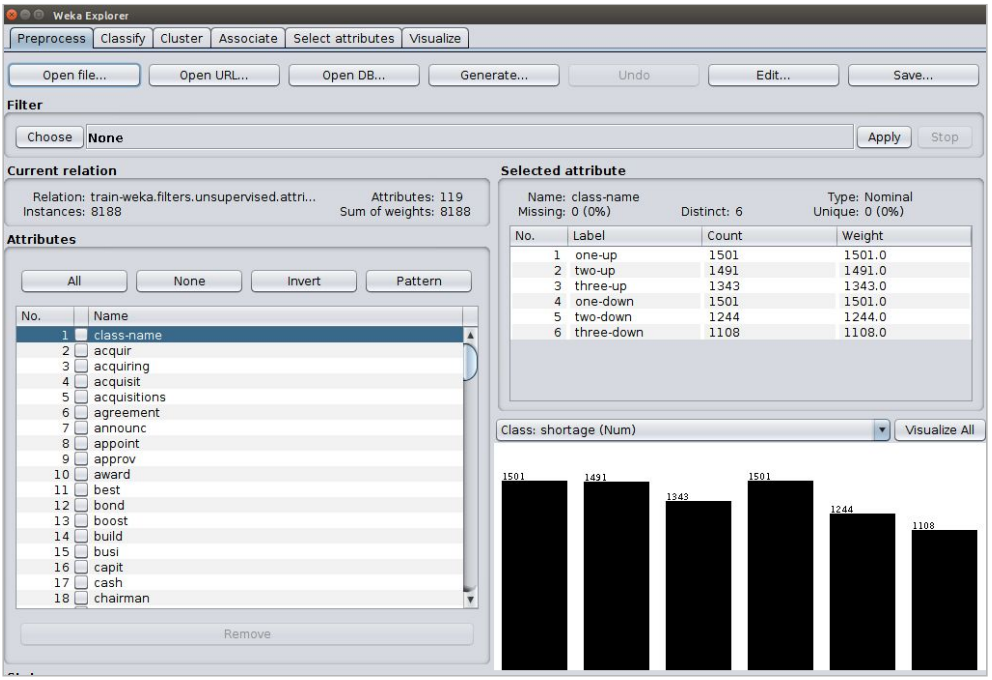

Figure 18: Initial data loading to Weka

When the data is loaded, the next step was to apply different types of feature selection and classification algorithms to the data set and obtain data models for each combination.

4.4.2 Attribute selection within classification

As mentioned in previous section, attribute selection and classification was done using Weka's different search methods and evaluators. It can be selected in Weka explorer as in Figure 19, in "Select Attributes" tab;

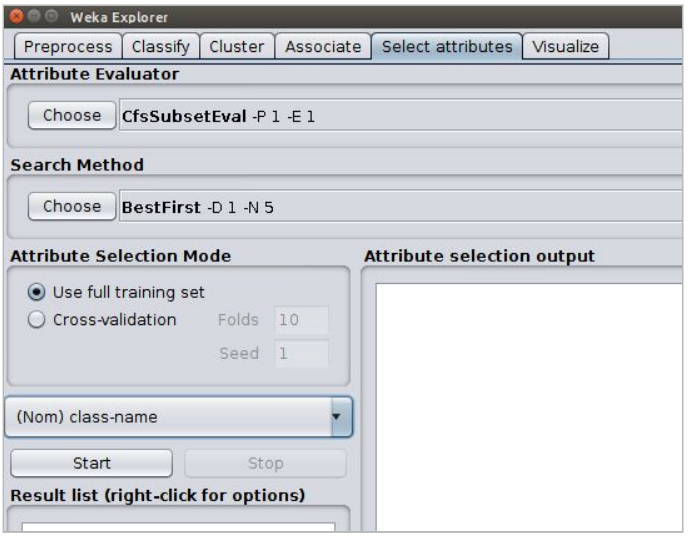

Figure 19: Attribute selection options in Weka

Following options are available in "AttributeSelectedClassifier" in Weka which can be configured for the requirement of the classification.

4.4.3 Selection of Feature Selection method

Selector and Evaluator are main options in feature selection. Within the selector, it has an option to select an algorithm for the selector as well. For the simplicity and consistency, within this project, selector algorithm has been selected as same as classification algorithm. Figure 20 shows the main configuration window of the feature selection tab in Weka.

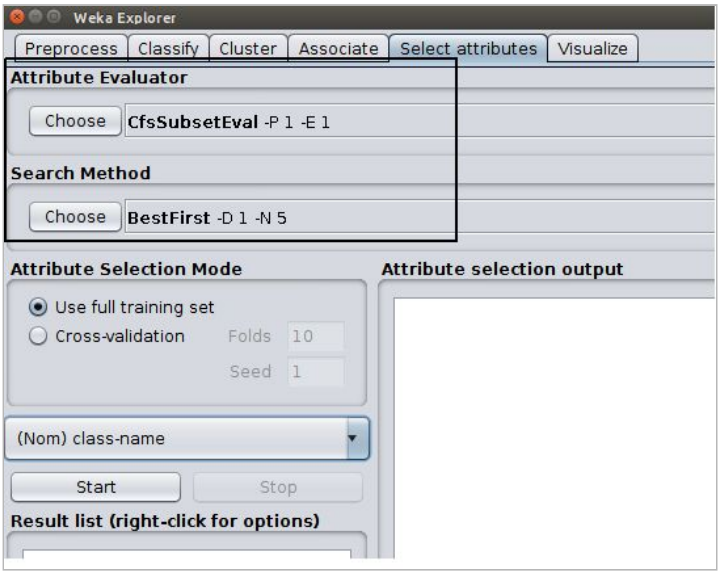

Figure 20: Weka evaluator and search method options

Each of these search methods and evaluators have their own configurations. For the simplicity and due to the limited scope, default configurations of Weka attribute selection is used within this project. Figure 21 and Figure 22 contain the sample configurations exist in "ClassifierAttributeEval" and "Best First" options.

Best First - Search Method configurations

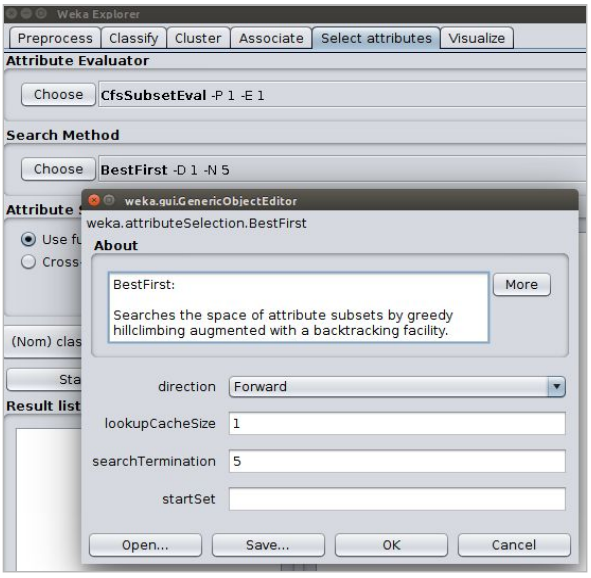

Figure 21 : BestFirst search method configurations

ClassifierAttributeEval - Evaluator configurations

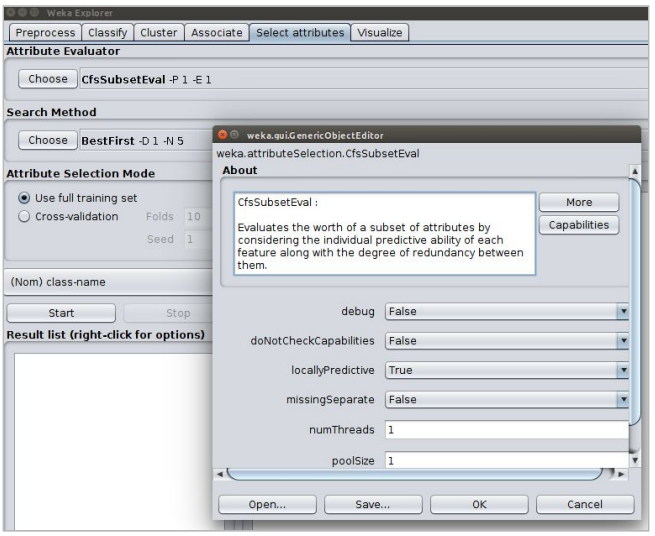

Figure 22 : ClassifierAttributeEval search method configurations

Selection of Evaluator Algorithm

Some of evaluators available in Weka supports configuration of a classification algorithm to be used in feature selection. It can be configured as in Figure 23 in "ClassifierAttributeEval" evaluator.

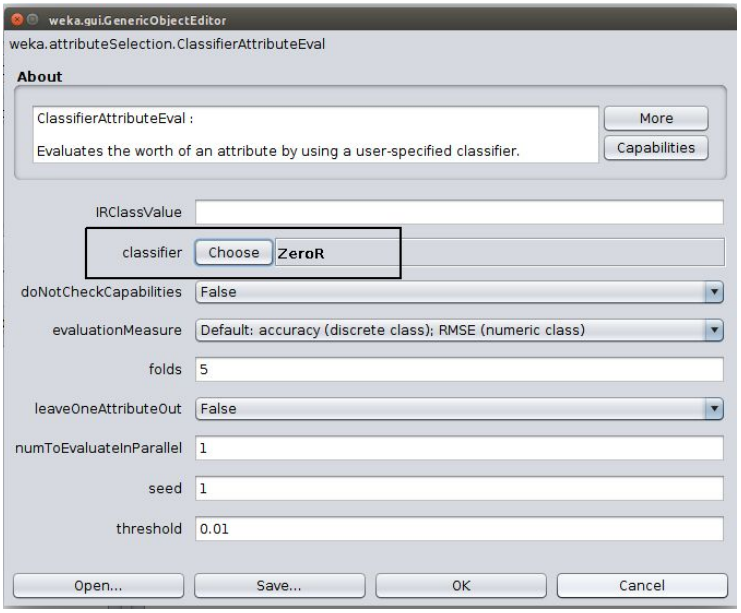

Figure 23 : Classifier selection option in ClassifierAttributeEval

or the simplicity and the limited scope of the project, only Naive Bayes algorithm is used in all available such configurations in this project.

4.4.4 Selection of Test Dataset

These are several training methods as shown in Figure 24, which are available in Weka. They are;

- **Use training set** - Use the data set provided for the training itself. It would be the data set provided in pre processing as the first input.
- **Supplied test set -** Use a separate data set as an input to the Weka for testing. It is different from the data set given originally for preprocessing.
- **Cross validation** - Use folds in the given training set as the test set. These folds are just partitions of the existing training set. Different partition is selected as the test set in each iteration and the classification is done several iterations matching with number of folds. For example, if it is selected as 10-fold cross validation, then the training data set will be partitioned into 10 partitions and classification is done 10 time. At each iteration, different partition is used as the test set and rest of 9 partitions used for training.
- **Percentage Split** - In this method, whole training set provided is partitioned by the given percentage. Higher portion is used as the training set and lower portion is used as the test set. For example, if the percentage split is done with 66%, then 66% of data will be used as the training set and rest of 33% will be used as the test set.

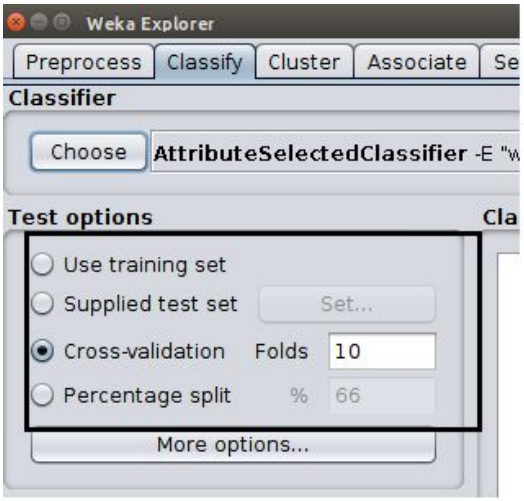

Figure 24 : Weka test data set options

Within this project, first set of classifications were done with both Cross validation and Percentage Split (66% and 80%) options. Results of these tests can be found in next chapter.

4.4.5 Prediction with selected attributes

Once the feature selection is done, there will be a set of features taken from different combinations of search methods and evaluators as described in previous sections. Each of these features were fed to Weka Experimenter and performed a test using different algorithms. Sample Weka configuration window with 6 algorithms for experiment, can be found as in Figure 25;

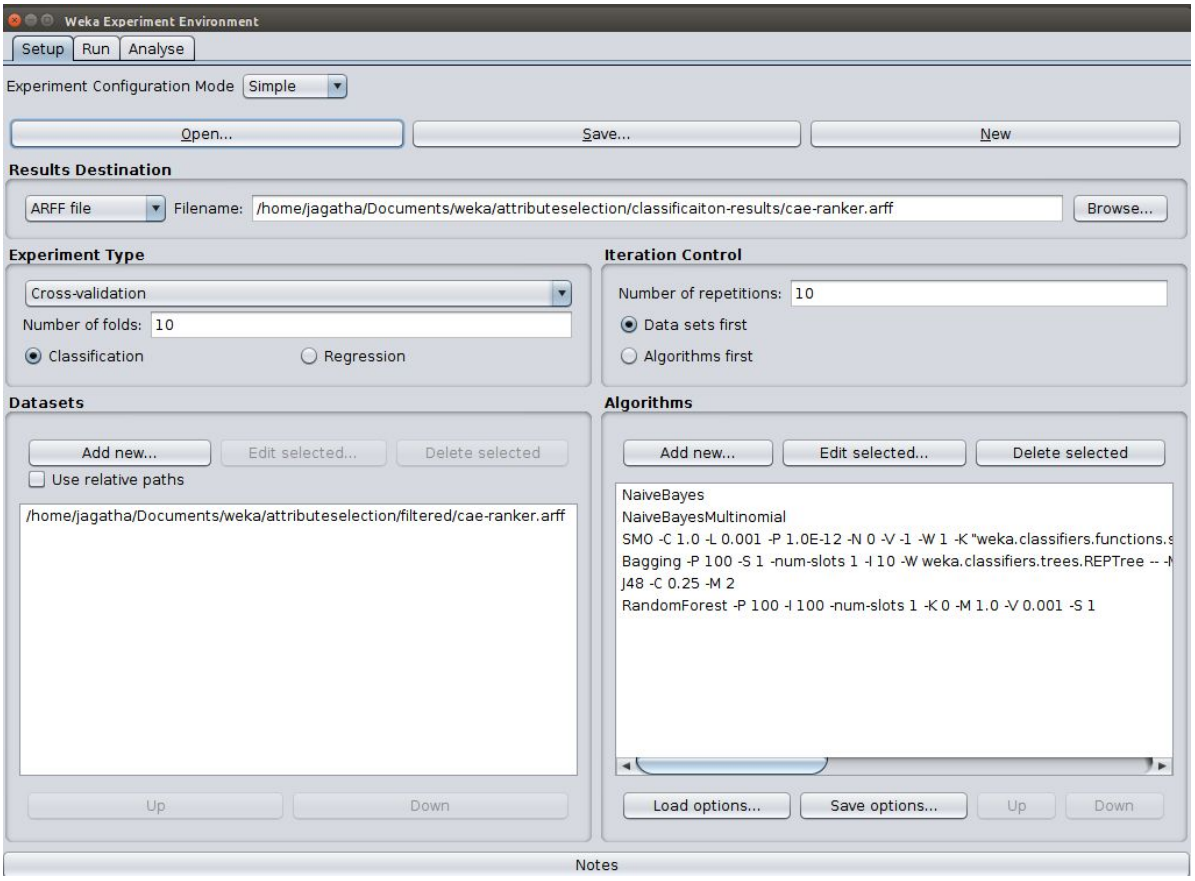

Figure 25 : Weka experimenter configuration with multiple algorithms

Once the experiment is executed and analysis is done, results are shown as in Figure 26 in Weka experimenter which provides comprehensive details of the classification against each algorithm.

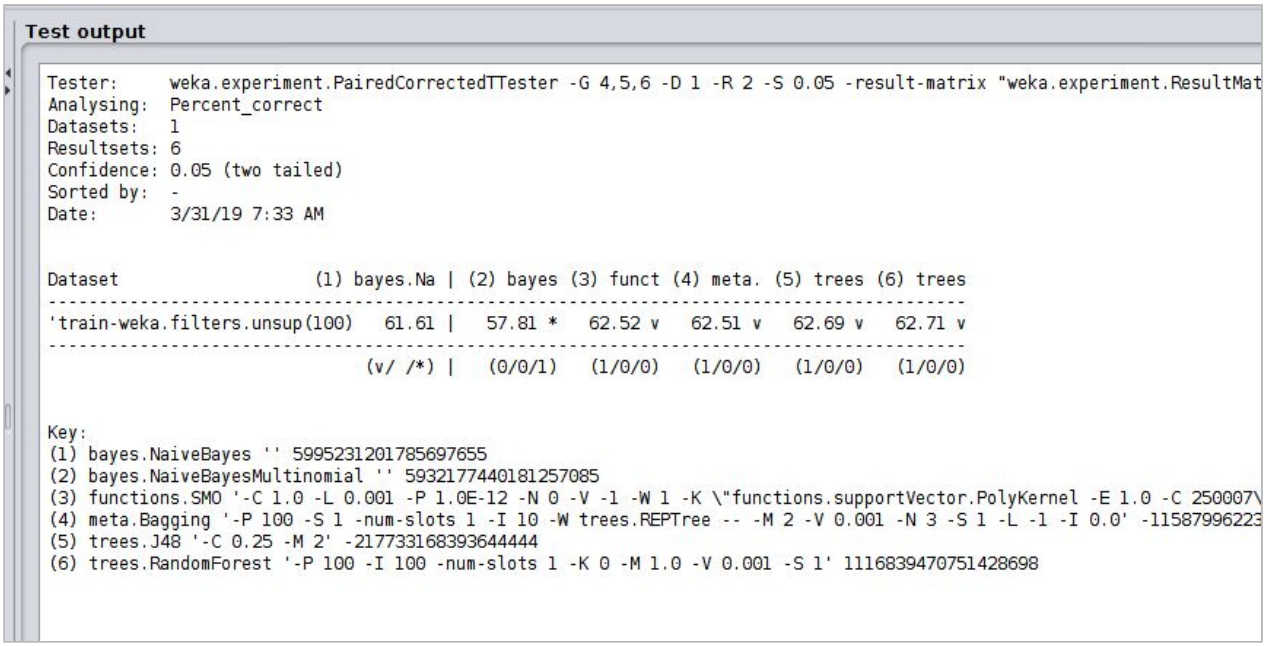

Figure 26 : Weka experimenter test output

Here, tabular structure displays the information on the accuracy of each algorithm with the provided data set and features. Key section shows algorithms used in the experiment with relevant configurations of each.

After executing experiments for all combinations of feature selection methods against six different algorithms mentioned above, results were captured as .arff files for future references.

Chapter 5: Evaluation and Results

Experiments were done with different algorithms as mentioned in previous chapter. Results were obtained and evaluated based on maximum accuracy of the prediction. There were few stages which the results were taken and analyzed. Based on the result of each stage, minor changes were done to the approach and new methods were taken to get maximum benefits of the prediction.

5.1 Class distribution

There were six classes used in the dataset of this project. Distribution of each class within the dataset can be found in below table

Class	Number of Entries within the dataset
One up	1501
Two up	1491
Three up	1343
One down	1501
Two down	1244
Three down	1208

Table 2: Class breakdown

5.2 Feature Selection

As mentioned in the previous chapter, feature selection is done based on selected combinations of Evaluators and Search Methods. Based on the selection done, Table 3 contains results which were obtained in terms of number of features (in this case, feature is a unique word in the whole data set) selected as best effective features for classifications.

Some of methods did not give effective number of attributes when it is selected with the threshold 0.01. In that case, feature selection was done with the threshold value of 0.05 for those scenarios to obtain effective set of attributes. Also some of feature selection methods

gave very less and very large number of attributes as the selected set. Those scenarios were ignored assuming that it will not give correct results in classification.

Evaluator	Search method	Threshold	Initial Attribute Count	Selected Attribute Count
CfsSubsetEval	BestFirst	0.01	119	40
ClassifierAttributeEval	Ranker	0.01	119	36
ClassifierSubsetEval	Greedy Stepwise	0.01	119	53
CorrelationAttributeEval	Ranker	0.05	119	42
GainRatioAttributeEval	Ranker	0.05	119	48
InfoGainAttributeEval	Ranker	0.01	119	43
OneRAttributeEval	Ranker	0.01	119	118
PrincipalComponents	Ranker	0.01	119	107
ReliefFAttributeEval	Ranker	0.01	119	3
SymmetricalUncertAttributeEval	Ranker	0.01	119	42
WrapperSubsetEval	BestFirst	0.01	119	69

Table 3: Attribute selection results

5.3 Initial Classification

Initially, the data set taken into the project was pre-processed using a standard list of stop lists taken from www.ranks.ln website [10]. However, when the standard words list was used, the results from the prediction was not good and it was around 24% - 31% accuracy. Table 4 illustrates summary of accuracy results for each algorithm,

Algorithm	NaiveBayes	RandomForest	SMO	Bagging	J48	NB-Multinomial
Accuracy %	25.14	30.42	28.02	29.14	26.79	24

Table 4: Initial classification results

When it was inspected the pre-processed data, it was observed that, most of words are common in all news items regardless of the classified class. Those words and terms include;

- Stock exchange names - abu dhabi stock market, kuwait stock exchange, dubai financial market, adx, dfm, kse
- Financial terms - bank, currency, market, money
- Country names - dubai, kuwait

Apart from above, the pre-processed data set contains a lot of words which has less frequency in many news items. Those have a frequency like 1 to 10 in while data set of 10000 records.

Steps taken to remove above anomalies in the data set. First, the words were selected based on standard names like above and removed those from the dataset. Second step was to remove words which has no impact on the result, i.e. which has very less frequency compared to number of records in the data set. Those words also removed from the data set. This can be achieved by applying TF-IDF (Term Frequency - Inverse Document Frequency) techniques available in Weka. However, due to the time limitation, it was not done within the scope of this project and can be done as a next step in the project.

Removing of above words were not done manually. Instead, those words also included in the stop words list take originally and given to Weka as a file of stop words, to be applied at the preprocessing and feature selection stage. Then Weka applied these and removed these words from the data while do the classification.

5.4 Classification with further preprocessing

With this change applied, the accuracy of the prediction was increased from 30% to 60% which gave acceptable outcome from the classification process. Following results in Table 5 were taken after applying the above change, and therefore it has given a result around 60% accuracy. Complete list of stop words used in this process can be found in the github repository mentioned in the appendix, which includes the words removed after initial analysis as mentioned above.

Data sets are abbreviated for the simplicity of the presentation as follows;

cfs-bf - CfsSubsetEval + BestFirst
clae-ranker - ClassifierAttributeEval + Ranker
cse-gs - ClassifierSubsetEval + Greedy Stepwise
cae-ranker - CorrelationAttributeEval + Ranker
igae-ranker - InfoGainAttributeEval + Ranker
suae-ranker - SymmetricalUncertAttributeEval + Ranker
grae-ranker - GainRatioAttributeEval + Ranker
wse-gs - WrapperSubsetEval + Greedy Stepwise

Data Set	Naive Bayes			Naive Bayes Multinomial			SMO			Bagging			J48			Random Forest		
	10 fold	66%	80%	10 fold	66%	80%	10 fold	66%	80%	10 fold	66%	80%	10 fold	66%	80%	10 fold	66%	80%
cfs-bf	59.29	59.4	59.51	56.62	57.05	57.06	59.81	59.86	60.04	59.76	59.99	60.1	59.83	60.13	60.13	59.99	60.15	60.27
clae-ranker	56.56	56.73	56.84	53.62	53.95	53.89	57.46	57.44	57.74	57.55	57.62	57.96	57.75	57.86	58.03	57.65	57.86	57.96
cse-gs	61.72	61.64	62.25	54.76	54.91	54.95	61.96	61.9	62.16	62.75	62.69	62.96	62.7	62.74	62.92	62.67	62.54	62.72
cae-ranker	61.61	61.56	61.27	57.81	57.67	57.18	62.52	62.36	62.27	62.51	62.37	62.29	62.69	62.49	62.31	62.71	62.49	62.38
igae-ranker	61.8	61.73	61.43	59.41	59.32	58.75	62.64	62.4	62.34	62.81	62.48	62.52	63.18	62.85	62.67	63.15	62.81	62.78
suae-ranker	61.59	61.75	61.98	59.07	59.39	59.44	62.57	62.62	62.97	62.57	62.67	62.99	62.69	62.89	63.16	62.72	62.91	63.16
grae-ranker	62.06	62.15	62.36	59.7	60.16	60.15	63.01	62.81	63.06	63.02	63.07	63.17	63.31	63.54	63.6	63.47	63.44	63.73
wse-bf	61.6	61.55	61.43	60.27	59.93	59.47	62.11	61.85	61.81	62.35	62.16	62.15	62.61	62.3	62.43	62.51	62.2	62.28
Min Accuracy	56.56	56.73	56.84	53.62	53.95	53.89	57.46	57.44	57.74	57.55	57.62	57.96	57.75	57.86	58.03	57.65	57.86	57.96
Max Accuracy	62.06	62.15	62.36	60.27	60.16	60.15	63.01	62.81	63.06	63.02	63.07	63.17	63.31	63.54	63.6	63.47	63.44	63.73
Standard Deviation	1.91	1.85	1.86	2.45	2.35	2.27	1.91	1.85	1.8	1.96	1.87	1.82	1.98	1.9	1.88	2	1.87	1.89

Table 5: Detailed classification results

5.5 Further Classification with configuration changes

5.5.1 Naive Bayes

Naive Bayes algorithm was re-configured to have different configurations from default configuration as follows.

- useKernalEstimator = true
- useSupervisedDiscresion=true

It was observed that, it always gives better results when default parameters have been changed, as shown in Table 6 below.

Data Set	Default Values	useKernalEstimator = true		useSupervisedDiscresion=true	
		Accuracy %	Progress	Accuracy %	Progress
cfs-bf	59.29	59.91	▲	59.67	▲
clae-ranker	56.56	57.65	▲	57.67	▲
cse-gs	61.72	62.75	▲	62.84	▲
cae-ranker	61.61	62.99	▲	63.02	▲
igae-ranker	61.8	63.2	▲	63.23	▲

suae-ranker	61.59	63.07	▲	63.11	▲
grae-ranker	62.06	63.49	▲	63.29	▲
wse-bf	61.6	62.78	▲	61.94	▲
Min Accuracy	**56.56**	**57.65**	▲	**57.67**	▲
Max Accuracy	**62.06**	**63.49**	▲	**63.29**	▲
Average Accuracy	**60.77875**	**61.98**	▲	**61.84625**	▲
Standard Deviation	**1.91**	**2.08**	▲	**2.08**	▲

Table 6: Naive Bayes additional configuration results

5.5.2 SMO

SMO algorithm was re-configured to have different configurations from default configuration as in Table 7 below.

- normalizedPolyKernel

Data Set	**Default Values**	**normalizedPolyKernel**	
		Accuracy %	**Progress**
cae-ranker	62.52	62.86	▲
cfs-bf	59.81	59.93	▲
clae-ranker	57.46	57.81	▲
cse-gs	61.96	62.35	▲
grae-ranker	63.01	63.39	▲
igae-ranker	62.64	63.31	▲+
suae-ranker	62.57	62.89	▲
wse-bf	62.11	62.32	▲
Min Accuracy	57.46	57.81	▲
Max Accuracy	63.01	63.39	▲
Average Accuracy	61.51	61.8575	▲
Standard Deviation	1.91	1.97	▲

Table 7: SMO additional configuration results

It was observed that accuracy was increased with "*normalizedPolyKernel*" option enabled.

5.5.3 Bagging

Bagging algorithm was re-configured to have different configurations from default configuration as in Table 8 below.

- seed=7
- seed=7 & Hoeffding Tree as classifier

Maximum and average accuracy were reduced with above change of configurations.

Data Set	seed=7			seed=7 & Hoeffding Tree	
	Default Values	Accuracy %	Progress	Accuracy %	Progress
cae-ranker	62.51	62.49	▼	62.11	▼
cfs-bf	59.76	59.78	▲	56.65	▼
clae-ranker	57.55	57.5	▼	55.44	▼
cse-gs	62.75	62.72	▼	60.44	▼
grae-ranker	63.02	62.99	▼	60.01	▼
igae-ranker	62.81	62.75	▼	57.77	▼
suae-ranker	62.57	62.54	▼	57.51	▼
wse-bf	62.35	62.34	▼	59.28	▼
Min Accuracy	**57.55**	**57.5**	▼	**55.44**	▼
Max Accuracy	**63.02**	**62.99**	▼	**62.11**	▼
Average Accuracy	**61.665**	**61.63875**	▼	**58.65125**	▼
Standard Deviation	**1.96**	**1.96**	▼	**2.2**	▼

Table 8: Bagging additional configuration results

5.5.4 J48

J48 algorithm was re-configured to have different configurations from default configuration as in Table 9.

- subtreeRaising=false
- reducedErrorPruning = true

Maximum accuracy given was increased with both options. Average accuracy was increased only with "*subtreeRaising*" configuration change.

Data Set	Default Values	subtreeRaising=false		reducedErrorPruning = true	
		Accuracy %	Progress	Accuracy %	Progress
cae-ranker	62.69	62.8	▲	62.82	▲
cfs-bf	59.83	59.88	▲	59.92	▲
clae-ranker	57.75	57.75	▼	57.63	▼
cse-gs	62.7	62.7	▼	62.76	▲
grae-ranker	63.31	63.34	▲	63.34	▲
igae-ranker	63.18	63.19	▲	63.05	▼
suae-ranker	62.69	62.8	▲	62.85	▲
wse-bf	62.61	62.6	▼	62.37	▼
Min Accuracy	**57.75**	**57.75**	▼	**57.63**	▼
Max Accuracy	**63.31**	**63.34**	▲	**63.34**	▲
Average Accuracy	**61.845**	**61.8825**	▲	**61.8425**	▼
Standard Deviation	**1.98**	**1.99**	▲	**2.01**	▲

Table 9: J48 additional configuration results

5.6 Summarized Results and Evaluation

Based on the above results, it can be obtained that few algorithms and feature selection options gives highest accuracy as follows;

Top Performers

Table 10 table shows the algorithms and selector/evaluator combinations in feature selection which gave overall highest accuracy of the prediction of stock price trend.

Accuracy %	Algorithm	Test Strategy	Data Set
63.73	Random Forest	80% Split	grae-ranker
63.6	J48	80% Split	grae-ranker
63.54	J48	66% Split	grae-ranker
63.47	Random Forest	10-fold	grae-ranker
63.44	Random Forest	66% Split	grae-ranker
63.31	J48	10-fold	grae-ranker

63.17	Bagging	80% Split	grae-ranker
63.07	Bagging	66% Split	grae-ranker
63.06	SMO	80% Split	grae-ranker
63.02	Bagging	10-fold	grae-ranker

Table 10: Top performance by accuracy

Performance By Algorithm

Table 11 and the graph in Figure 27 show how each algorithm performs in prediction with their highest accuracy data sets.

Algorithm	Test Strategy	Accuracy %
Naive Bayes	80% Split	62.36
Naive Bayes Multinomial	10-fold	60.27
SMO	80% Split	63.06
Bagging	80% Split	63.17
J48	80% Split	63.6
Random Forest	80% Split	63.73

Table 11: Top Performance by algorithm

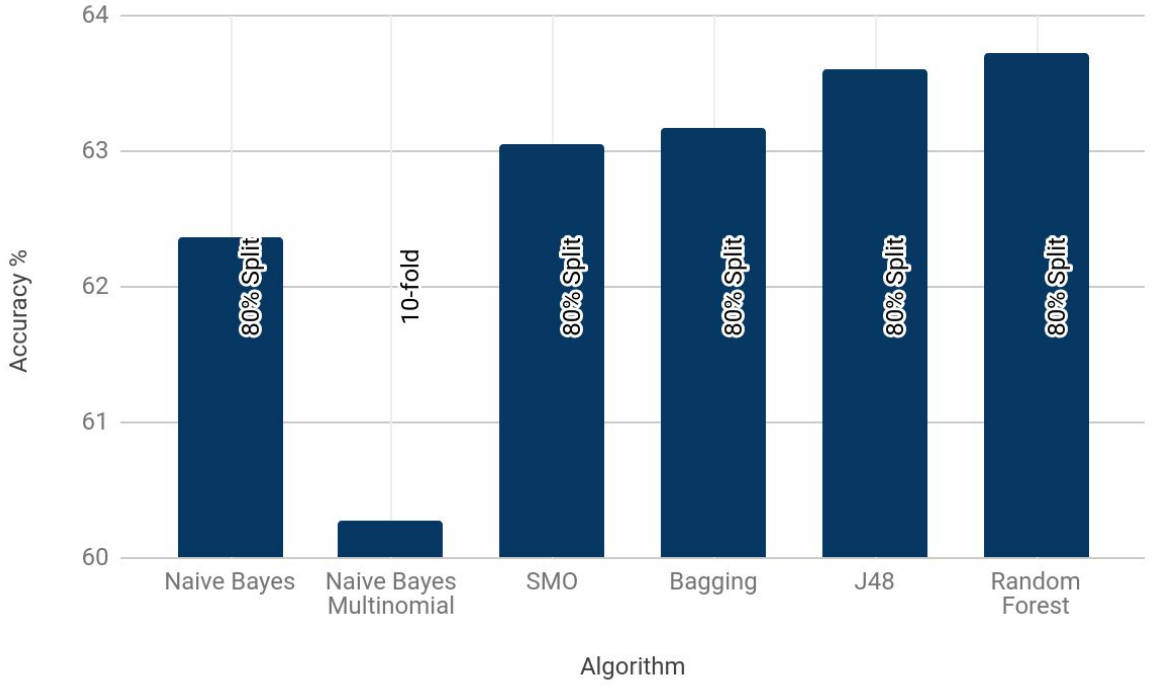

Figure 27 : Top performance by algorithm

With above results, it is clearly shown that Random Forest shows highest success rate out of all algorithms and the test data set has 80% split.

Performance By Data Set

Table 12 and the graph in Figure 28 show how each feature selection method contributed for the highest accuracy within the set of algorithms.

Data Set	Search Method	Evaluator	Accuracy
cfs-bf	CfsSubsetEval	Best First	60.27
clae-ranker	ClassifierAttributeEval	Ranker	58.03
cse-gs	ClassifierSubsetEval	Greedy Stepwise	62.96
cae-ranker	CorrelationAttributeEval	Ranker	62.71
igae-ranker	InfoGainAttributeEval	Ranker	63.18
suae-ranker	SymmetricalUncertAttributeEval	Ranker	63.16
grae-ranker	GainRatioAttributeEval	Ranker	63.73
wse-bf	WrapperSubsetEval	Greedy Stepwise	62.61

Table 12: Top performance by dataset

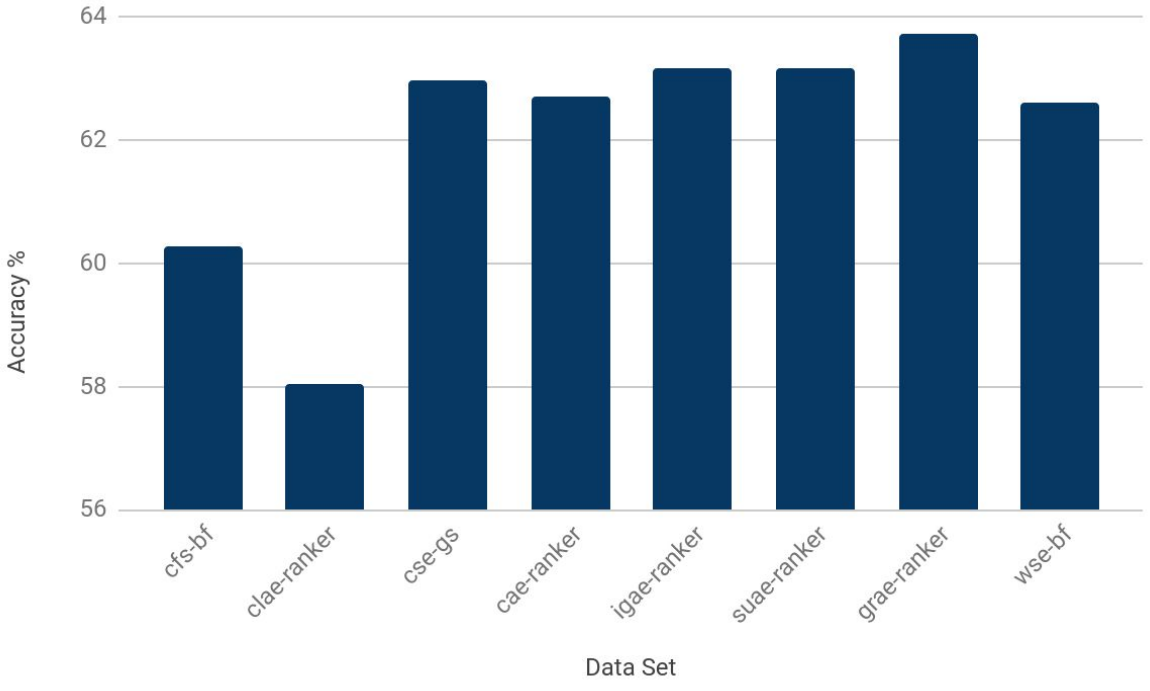

Figure 28 : Top performance by dataset

Highest accuracy was given by the data set names “grae-ranker”. It was created using the “GainRatioAttributeEval” as the search method and “Ranker” as the evaluator.

Performance By Algorithm with 10-Fold Cross Validation

With the use of 10-Fold cross validation for the training the data, Table 13 and Figure 29 were the observations of the performance of each algorithm.

Algorithm	cfs-bf	clae-ranker	cse-gs	cae-ranker	igae-ranker	suae-ranker	grae-ranker	wse-bf
Naive Bayes	59.29	56.56	61.72	61.61	61.8	61.59	62.06	61.6
SMO	59.81	57.46	61.96	62.52	62.64	62.57	63.01	62.11
Bagging	59.76	57.55	62.75	62.51	62.81	62.57	63.02	62.35
J48	59.83	57.75	62.7	62.69	63.18	62.69	63.31	62.61
Random Forest	59.99	57.65	62.67	62.71	63.15	62.72	63.47	62.51

Table 13: Top performance by algorithm with 10-Fold cross validation

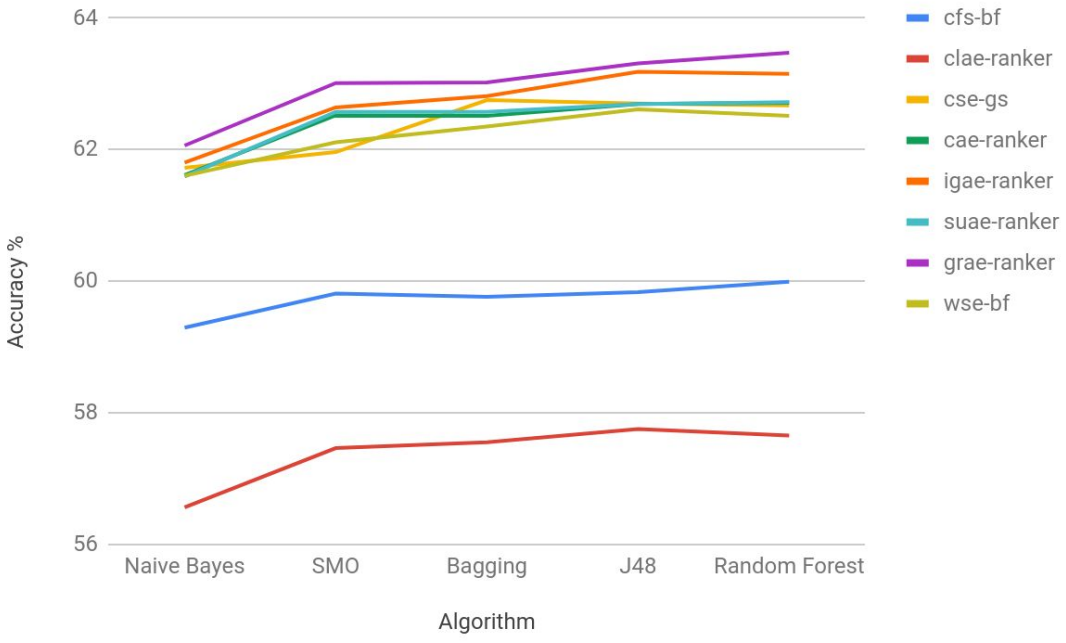

Figure 29 : Top performance by algorithm with 10-Fold cross validation

Performance By Algorithm with 66% Split

With the use of 66% Split for the training the data, Table 14 and Figure 30 were the observations of the performance of each algorithm.

Data Set	cfs-bf	clae-ranker	cse-gs	cae-ranker	igae-ranker	suae-ranker	grae-ranker	wse-bf
Naive Bayes	59.4	56.73	61.64	61.56	61.73	61.75	62.15	61.55
SMO	59.86	57.44	61.9	62.36	62.4	62.62	62.81	61.85
Bagging	59.99	57.62	62.69	62.37	62.48	62.67	63.07	62.16
J48	60.13	57.86	62.74	62.49	62.85	62.89	63.54	62.3
Random Forest	60.15	57.86	62.54	62.49	62.81	62.91	63.44	62.2

Table 14: Top performance by algorithm with 66% split

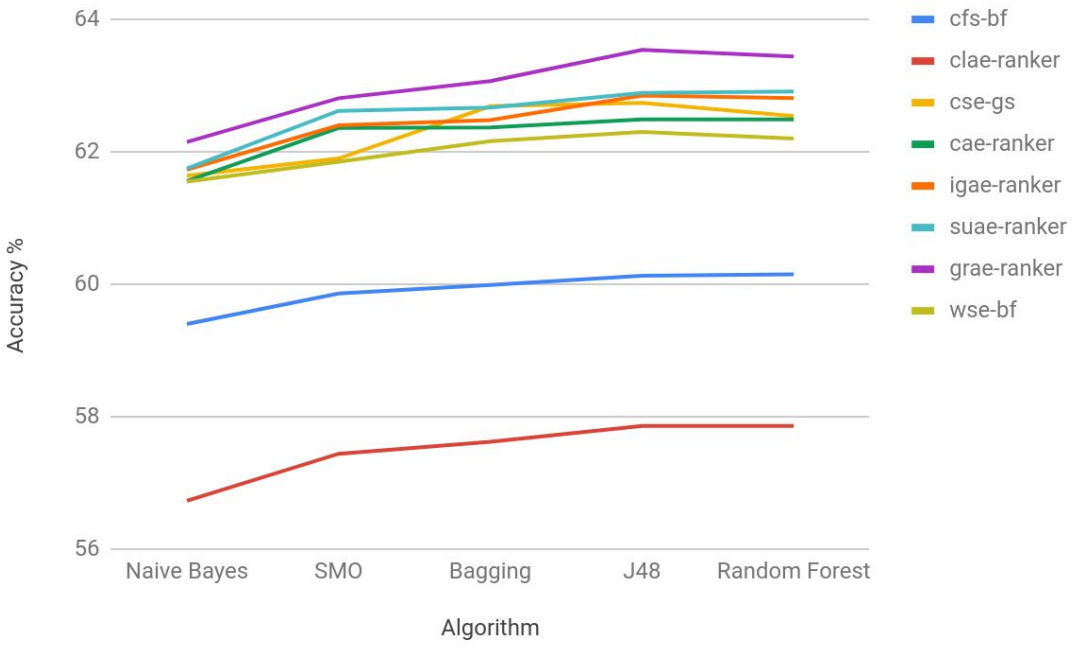

Figure 30 : Top performance by algorithm with 66% split

Performance By Algorithm with 80% Split

With the use of 80% Split for the training the data, Table 15 and Figure 31 were the observations of the performance of each algorithm.

Data Set	cfs-bf	clae-ranker	cse-gs	cae-ranker	igae-ranker	suae-ranker	grae-ranker	wse-bf
Naive Bayes	59.51	56.84	62.25	61.27	61.43	61.98	62.36	61.43
SMO	60.04	57.74	62.16	62.27	62.34	62.97	63.06	61.81
Bagging	60.1	57.96	62.96	62.29	62.52	62.99	63.17	62.15
J48	60.13	58.03	62.92	62.31	62.67	63.16	63.6	62.43
Random Forest	60.27	57.96	62.72	62.38	62.78	63.16	63.73	62.28

Table 15: Top performance by algorithm with 80% split

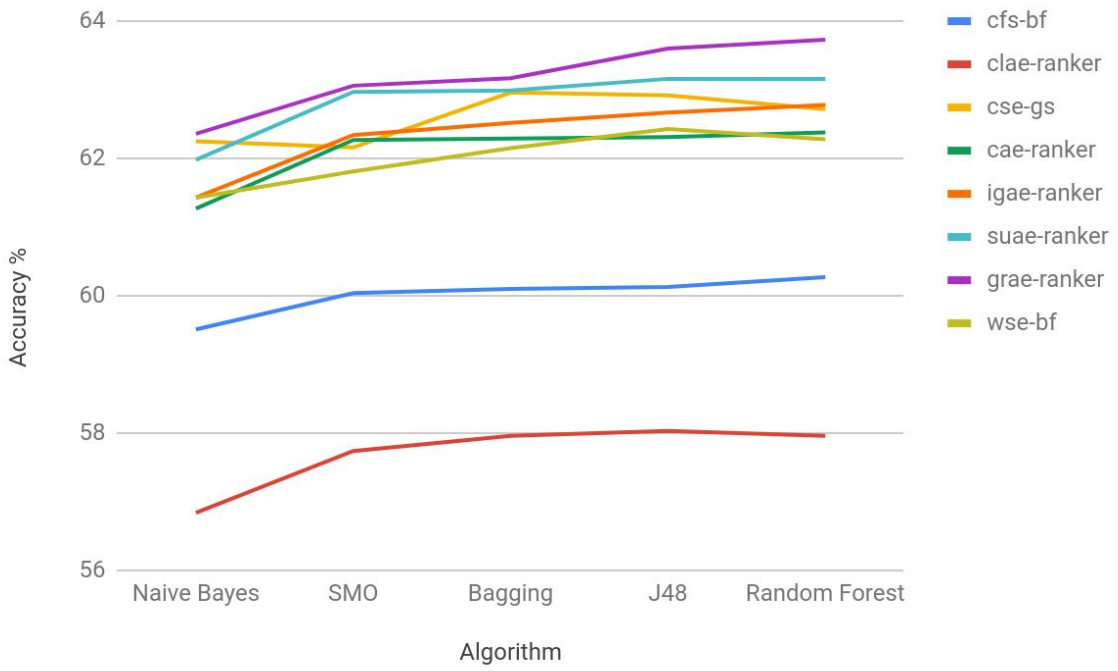

Figure 31: Top performance by algorithm with 80% split

5.7 Evaluation of Final Data Model

By looking at above results obtained from different algorithms and data sets, the most accurate results were given by Random Forest algorithm when 80% split of training data is used. The feature set used for this sample was obtained with "GainRatioAttributeEval" as the evaluator and "Ranker" as the search method.

The model built from this data set was used to evaluate sample data set with empty class as shown in Figure 32. Testing sample was created with 600 news items, 100 items from each category. Class was not included in the data set which allowed it to predict from the built model.

```
@relation test

@attribute Document string
@attribute class-name {one-up,two-up,three-up,one-down,two-down,three-down}

@data

"adx fall lowest level bank sector loss ",?
"abnic profit ",?
"abu dhabi aviat make mln increase",?
"abu dhabi aviat board propos benefit",?
"abu dhabi aviat board propos awarded",?
"abu dhabi aviat gener mln profit undertake",?
"abu dhabi aviat board meet earli novemb ",?
```

Figure 32: Sample arff file format with empty class value

Also to avoid data type conflicts against the data set used in building the model and testing, "*FilteredClassifier*" was used to build the model with all other filters and classification algorithms in place.

Random Forest algorithm was used to evaluate the model. Table 16 contains the result obtained from the evaluation.

Class	Total News Items in Test Data Set	Correctly Classified	Accuracy
one-up	100	74	74%
two-up	100	72	72%
three-up	100	63	63%
one-down	100	75	75%
two-down	100	59	59%
three-down	100	64	64%
Total	**600**	**408**	**68%**

Table 16: Evaluation results for the best data model

Below Figure 33 shows how Weka gives the output of the evaluation.

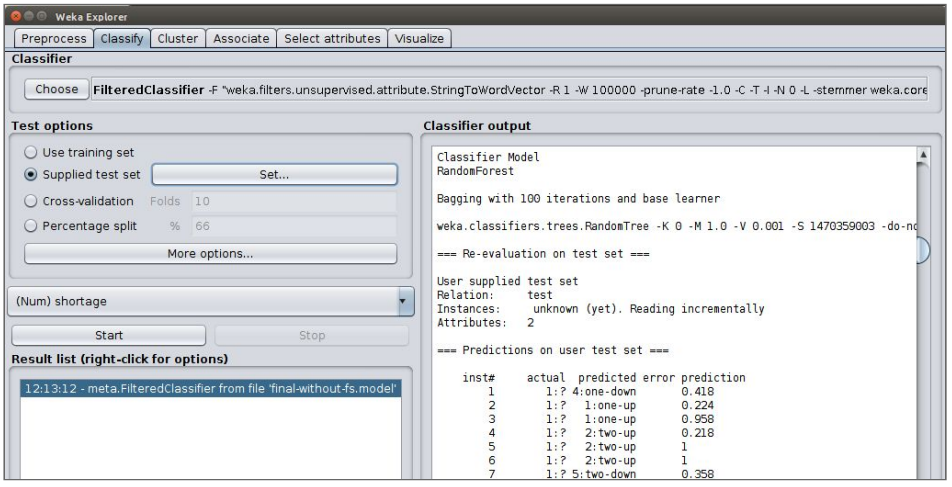

Figure 33 : Sample results with predicted class value

Chapter 6: Conclusion and Future Work

6.1 Conclusion

There were five algorithms were applied and tested with seven data sets prepared from actual stock market data within this project. Data sets were created using different combination of feature selection search methods and evaluators.

Random Forest algorithm gave the best results, around 63% accuracy followed by J48, Bagging and SMO with next best accurate prediction percentages.

It was observed that, the raw data contains massive set of unwanted words and characters which needs to be removed before going into feature selection and classification. Some of those data can be removed from standard data cleansing methods such as stop word removal and stemming. However, due to the nature of the data obtained for this project, it was not effective when those standard methods were used to clean data. Therefore, some manual intervention was required to identify very frequent data within news items and remove those from the data set. This was done by including those words also in the stop words list and applied it through Weka.

Most of past researched were done to predict only the price change direction of stocks in the stock market. However, within this project, it was able to predict number of days which prices change persists with the given direction. It gave considerably good accuracy in prediction, which can always be improved in many ways. It will be discussed in next section.

Weka was the tool used for all feature selection and classification tasks. It was an excellent open source software which was having rich set of features for text analysis. There were lot of other options which were not used within this project, which would be helpful to increase the accuracy of the result.

6.2 Future Work

As mentioned in previous topic, there are several ways to improve the accuracy of this method in different data sets. The algorithm used here can be combined with algorithms used in previous researches which uses standard financial word master proven to be giving best results. Test can be conducted by using that standard word master instead of creating a word master from the data set itself to check whether it provides more accuracy to the Trend and Weight prediction scenario.

The data set used in this project is based on Middle East based stock markets. But those are actual data disseminated form stock exchanges without applying any formatting or refactoring. A test can be conducted by applying a data set from different region to verify the applicability of the algorithm and data model as a common model globally. It would be beneficial to have a global data model instead of region specific data model and algorithms.

Another option to improve the efficiency is using TF-IDF techniques which are used in text classifications. This technique is mostly used to categorize and tagging documents based on key words and their occurrences within the text item as well as within the whole set of text items. Therefore, it provides the most suitable words to represent a text item rather than providing a data model to predict the category of a new item. However, this may be used with some of already used methodologies to add an advantage of predicting data.

Also there are improvements to do by altering parameters of search methods, evaluators and classification algorithms rather than using default values configured in Weka. It will provide more set of results which might contain more accurate prediction models.

References

[1] "Bloomberg news" [Online]. https://www.bloomberg.com/asia.(25-Feb-2019).

[2] "Dow Jones" [Online]. https://www.dowjones.com/.(25-Feb-2019).

[3] "Yahoo Finance: [Online]. https://finance.yahoo.com/.(25-Feb-2019).

[4] "Predicting Stock Prices from News Articles- Jerry Chen, Aaron Chai, Madhav Goel, Donovan Lieu, Faazilah Mohamed, David Nahm, Bonnie Wu". [Online]. https://www.stat.berkeley.edu/~aldous/Research/Ugrad/chen_USA.pdf. (8th-Dec-2018).

[5] "News versus Sentiment: Predicting Stock Returns from News - Steven L. Heston and Nitish R. Sinha". [Online].https://www.federalreserve.gov/econresdata/feds/2016/files/2016048pap.pdf. (4th-Jan-2019).

[6] "Stock Price Forecasting Using Information from Yahoo Finance and Google Trend - Selene Yue Xu (UC Berkeley)". [Online]. https://www.econ.berkeley.edu/sites/default/files/Selene%20Yue%20Xu.pdf. (4th-Jan-2019).

[7] "Stock trend prediction using news sentiment analysis - Joshi Kalyani, Prof. H. N. Bharathi, Prof. Rao Jyothi". [Online]. https://arxiv.org/ftp/arxiv/papers/1607/1607.01958.pdf.(8th-Dec-2018).

[8] "A Stock Prediction System Based on News and Twitter - Kibum Kim, SeungminYang, Dongyoung Kim, Jeawon Park, Jaehyun Choi". [Online]. Available: http://www.sersc.org/journals/IJSEIA/vol10_no6_2016/6.pdf. (15th-July- 2018).

[9] "Textual Analysis of Stock Market Prediction Using Financial News Articles - Robert P. Schumaker and Hsinchun Chen". [Online]. https://www.federalreserve.gov/econresdata/feds/2016/files/2016048pap.pdf.(8th-Dec-2017).

[10] "Stop Words" [Online]. https://www.ranks.nl/stopwords.(25-Feb-2019)

[11] "Naive Bayes Classifier" [Online]. https://en.wikipedia.org/wiki/Naive_Bayes_classifier.(25-Feb-2019)

[12] "jsoup: Java HTML Parser" [Online]. https://jsoup.org/.(25-Feb-2019)

[13] "Simple java library to filter the stopping,stemming words from input data or file" [Online]. https://github.com/uttesh/exude.(25-Feb-2019)

[14] "Apache OpenNLP - Machine learning based toolkit for the processing of natural language text." [Online]. https://opennlp.apache.org/.(25-Feb-2019)

[15] "PtnPlanet - A java classifier based on the naive Bayes approach complete with

Maven support and a runnable example." [Online]. https://github.com/ptnplanet/Java-Naive-Bayes-Classifier.(25-Feb-2019)

[16] "Naive Bayes Classifier - Probabilistic Model" [Online]. https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Probabilistic_model. (25-Feb-2019)

[17] "Java-ML - Java Machine Learning Library" [Online]. http://java-ml.sourceforge.net/. (25-Feb-2019)

[18] "Stock - Investopedia" [Online]. http://www.investopedia.com/terms/s/stock.asp.(25-Feb-2019).

[19] "Stock Market - Wikipedia" [Online]. https://en.wikipedia.org/wiki/Stock_market.(25-Feb-2019).

Appendices

Appendix A – Tools and Software

Weka

Weka provides set of algorithms for machine learning. It also provides features for data mining and feature extraction. It supports extensions as well for Java, which allows to be integrated with any Java code in addition to the default standalone software for direct use.

https://www.cs.waikato.ac.nz/ml/weka/

Jsoup

Jsoup is a java library specially designed for HTML which contains rich set of functionalities for parse and manipulating HTML data.

https://jsoup.org/

Exude

Exude is a simple Java library which can be used to basic text cleaning operations like stemming, stopping and filtering. It supports different inputs like plain text, file or a web link.

https://github.com/uttesh/exude

PtnPlanet

PtnPlanet provides a basic library for Naïve Bayes classification using categories and features. It determines the category of an object based on the features includes.

https://github.com/ptnplanet/Java-Naive-Bayes-Classifier

OpenNLP

Apache OpenNLP is a machine learning toolkit for natural language processing.
https://opennlp.apache.org/

Java-ML

Java ML library contains a set of machine learning. It does not have a GUI and can be used with any Java program.

http://java-ml.sourceforge.net/

Appendix B – Code Repository

Java code used in this project can be found in below Github URL.

https://github.com/jagathsisira/price-trend-predictor

PredictorMain.java is used as the main class for this implementation and there are several classes used for different kind of data cleansing and categorization activities. The order of executing those methods is the same order as they appears in the main class. If someone wants to run a specific method like news list generation, relevant method can be run independently.

Appendix C – More Classification Results

Performance of Naive Bayes algorithm

Data Set	10-fold	66% Split	80% Split
cfs-bf	59.29	59.4	59.51
clae-ranker	56.56	56.73	56.84
cse-gs	61.72	61.64	62.25
cae-ranker	61.61	61.56	61.27
igae-ranker	61.8	61.73	61.43
suae-ranker	61.59	61.75	61.98
grae-ranker	62.06	62.15	62.36
wse-bf	61.6	61.55	61.43

Table 17 : Performance of Naive Bayes Algorithm

Figure 34 : Performance of Naive Bayes algorithm

Performance of Naive Bayes Multinomial algorithm

Data Set	10-fold	66% Split	80% Split
cfs-bf	56.62	57.05	57.06
clae-ranker	53.62	53.95	53.89
cse-gs	54.76	54.91	54.95
cae-ranker	57.81	57.67	57.18
igae-ranker	59.41	59.32	58.75
suae-ranker	59.07	59.39	59.44
grae-ranker	59.7	60.16	60.15
wse-bf	60.27	59.93	59.47

Table 18 : Performance of Naive Bayes Multinomial algorithm

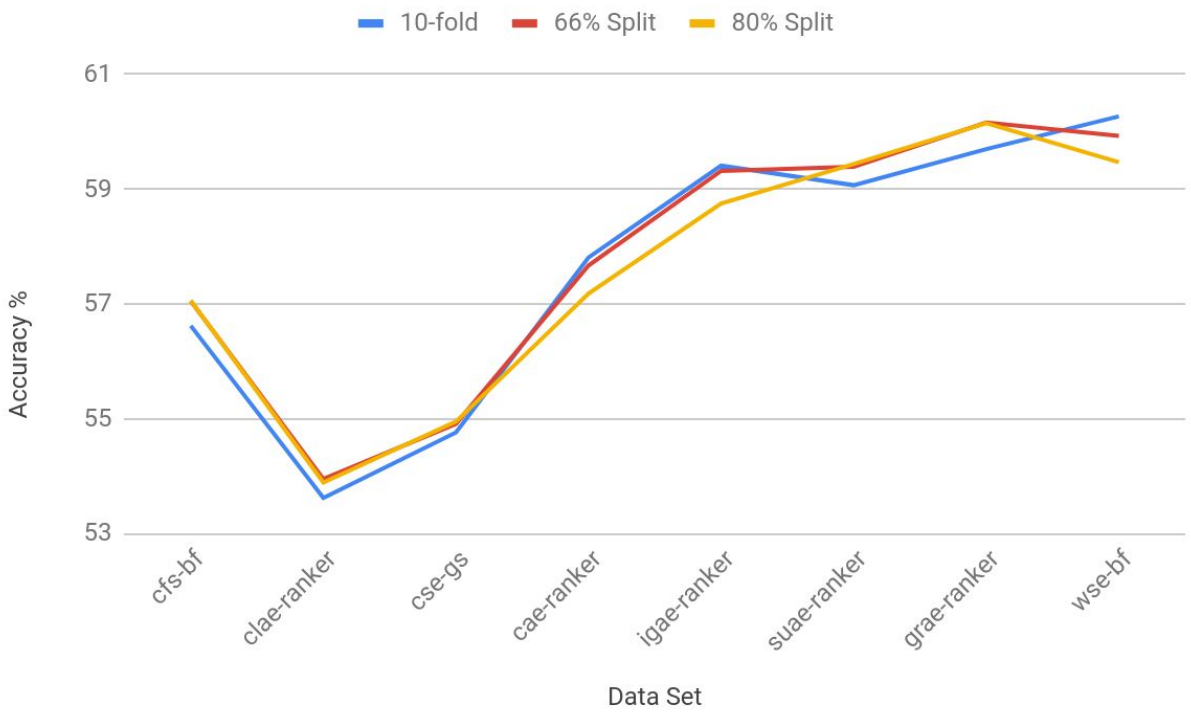

Figure 35 :Performance of Naive Bayes Multinomial algorithm

Performance of SMO (SVM algorithms)

Data Set	10-fold	66% Split	80% Split
cfs-bf	59.81	59.86	60.04
clae-ranker	57.46	57.44	57.74
cse-gs	61.96	61.9	62.16
cae-ranker	62.52	62.36	62.27
igae-ranker	62.64	62.4	62.34
suae-ranker	62.57	62.62	62.97
grae-ranker	63.01	62.81	63.06
wse-bf	62.11	61.85	61.81

Table 19 : Performance of SMO algorithm

Figure 36 : Performance of SMO algorithm

Performance of Bagging

Data Set	10-fold	66% Split	80% Split
cfs-bf	59.76	59.99	60.1
clae-ranker	57.55	57.62	57.96
cse-gs	62.75	62.69	62.96
cae-ranker	62.51	62.37	62.29
igae-ranker	62.81	62.48	62.52
suae-ranker	62.57	62.67	62.99
grae-ranker	63.02	63.07	63.17
wse-bf	62.35	62.16	62.15

Table 20 : Performance of Bagging algorithm

Figure 37 : Performance of Bagging algorithm

Performance of J48

Data Set	10-fold	66% Split	80% Split
cfs-bf	59.83	60.13	60.13
clae-ranker	57.75	57.86	58.03
cse-gs	62.7	62.74	62.92
cae-ranker	62.69	62.49	62.31
igae-ranker	63.18	62.85	62.67
suae-ranker	62.69	62.89	63.16
grae-ranker	63.31	63.54	63.6
wse-bf	62.61	62.3	62.43

Table 21 : Performance of J48 algorithm

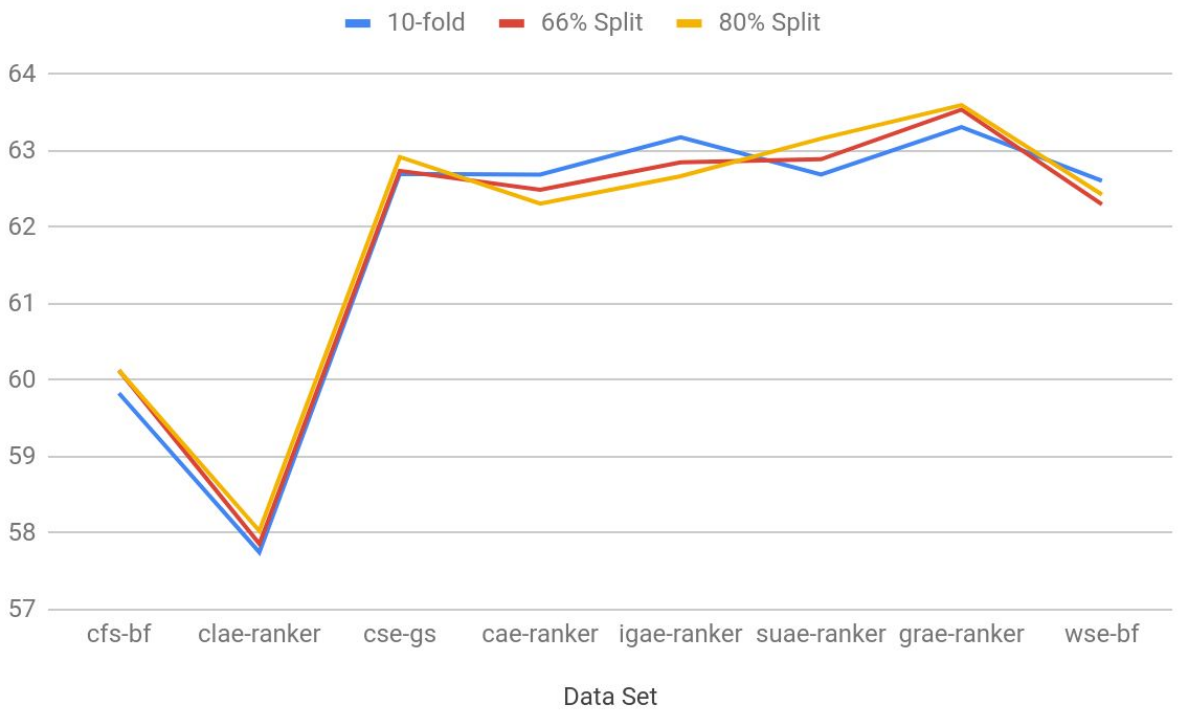

Figure 38 : Performance of J48 algorithm

Performance of Random Forest

Data Set	10-fold	66% Split	80% Split
cfs-bf	59.99	60.15	60.27
clae-ranker	57.65	57.86	57.96
cse-gs	62.67	62.54	62.72
cae-ranker	62.71	62.49	62.38
igae-ranker	63.15	62.81	62.78
suae-ranker	62.72	62.91	63.16
grae-ranker	63.47	63.44	63.73
wse-bf	62.51	62.2	62.28

Table 22 : Performance of Random Forest algorithm

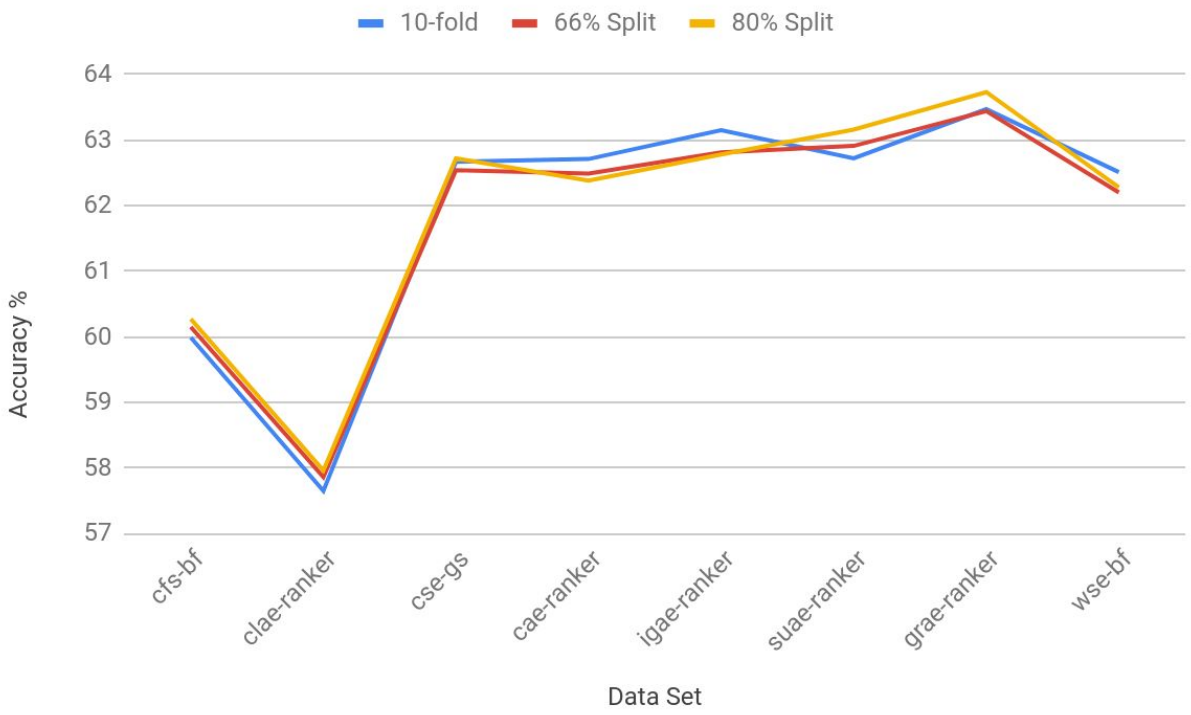

Figure 39: Performance of Random Forest algorithm