Jomo Kenyatta University of Agriculture and Technology (JKUAT)

Department of Computing

ICS2406: Computer Systems Project

REF: JKU/2/83/022

Title: Predicting Stock Market Movement Using an Enhanced Naïve Bayes Model for Sentiment Analysis Classification.

Author:
Name: Mbadi Atieno Sheila Reg. No: CS281-1556/2014

Submission Date: _____ Sign: _____

Course: _____

Supervisor 1:
Name: _____Sign: _____Date: _____

Supervisor 2:
Name: _____Sign: _____Date: _____

Supervisor 3:

Name: _____Sign: _____Date: _____

Table of Figures

Table of Contents

CHAPTER 1: PROJECT PROPOSAL

1.1 Background/ Introduction

Social networks like Facebook and twitter have changed the way people communicate. People use such outlets to express their views and opinions on various topics. These information is beneficial to data analysts, businesses and other institutions that mine various opinions from users as feedback which they use to get insight on a product or service offered.

Sentiment analysis is used to extract such remarks of users and then gives them a polarity of positive, negative or neutral. The neutral case is usually ignored as it normally holds no weight in a study.

Sentiment analysis can be defined as using Natural Language Processing (NLP), statistics, or machine learning methods to extract, identify, or otherwise characterize the opinion content of a text unit. (Introduction to Sentiment Analysis, n.d.)

The stock also called capital stock, of a corporation is constituted of the equity stock of its owners. A single share of the stock represents fractional ownership of the corporation in proportion to the total number of shares. (Stock, n.d.)

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit for an investor. Thus there is need to come up with an accurate means of predicting stock market trend. (Stock market prediction, n.d.)

Naïve Bayes classifier is a fast and simple machine learning classifier that can be used in the sentiment classification of text. However it is not the most efficient classifier thus there is need to improve it so as to yield high classification accuracy.

The efficient market hypothesis suggests that stock prices are a function of information and rational expectations, and that newly revealed information about a company's prospects is almost immediately reflected in the current stock price. This would imply that all publicly known information about a company, which obviously includes its price history, would already be reflected in the current price of the stock. Accordingly, changes in the stock price reflect release of new information and changes in the market generally.

The efficient market hypothesis therefore suggests that using public sentiments to predict stock market movement as well as a stock's price history will yield more accurate predictions.

1.2 Problem Statement

Using only price history to predict stock market movement does not yield the most accurate predictions. There is need to come up with a new model of predicting stock prices.

The Efficient Market Hypothesis (EMH) states that stock market prices are largely driven by new information; thus, it is prudent to incorporate public sentiments when coming up with a prediction model for stock market prices. (Burton, 2003).

1.3 Objectives

1.3.1 Main Objective

1. To develop a stock market prediction model with a high degree of accuracy.

1.3.2 Specific Objectives

1. To research on the enhancement of the Naïve Bayes classifier for sentiment classification of tweets and use it.
2. To research on the best techniques for stock prediction and implement one.

1.3.3 Research Questions

1. How can the current methods of stock prediction be improved to come up with more accurate predictions?
2. How can the Naïve Bayes classifier be enhanced to improve the classification accuracy of tweets?
3. What is the best technique for stock prediction?

1.4 Justification

Stock market prediction is complex as markets are quite hard to understand. There is need for a stock prediction model that will provide predictions with a high degree of accuracy so as to yield significant profit by any investor.

A lot of research has been conducted on the topic of stock prediction but very few actually take public sentiments into consideration. From Bollen et.al. (2010) research work, the incorporation of sentiment analysis in predicting stock prices results in predictions with a higher degree of accuracy when compared to those that only use price history. Therefore it is prudent to come up with a stock prediction model that takes public sentiment into account.

1.5 Schedule

Activity	**Estimated Start and End Date**	**Deliverables**
Project Identification and Proposal Writing	28th September - 6th October 2017	Project Proposal
Project Research	10th October - 12th January 2018	Literature Review Document and Methodology
Project Analysis and Design	5th February - 16th February 2018	Analysis and Design Document
System Implementation	19th February - 30th March 2018	Working Prototype
System Testing and Debugging	2nd April- 13th April 2018	Final Prototype
Project Submission	May 2018	Complete System with Documentation

Figure 1. Table of project schedule

1.6 Budget

Resource	Cost	Justification
Printing and Binding	Kshs 1500	Need for printing and binding of documents
Airtime and Bundle Purchase	Kshs 3500	Researching on the internet and contacting supervisor when there is need
Flash Disk	Kshs 2000	Data transfer when printing and backing up of data
Total	**Kshs 7000**	

Figure 2. Table of project budget

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

A research conducted by Bollen et al. (2010) shows that the collective mood on Twitter, the aggregate of all positive and negative tweets, can predict the Dow Jones Industrial Average (DJIA) index with 87.6% accuracy.

This therefore shows that there is a very strong correlation between twitter sentiments and the stock market movement and will be the basis of this research

2.2 Sentiment Analysis

2.2.1 Introduction

Sentiment Analysis is an area of study within Natural Language Processing that is concerned with identifying the mood or opinion of subjective elements within a text towards an entity (Bhadane, Dalal & Doshic, 2015). The entity can represent individuals, events, topics or a product/service offered.

It is becoming a popular area of research and social media analysis, especially around user reviews and tweets. It is a special case of text mining generally focused on identifying opinion polarity, and while it's often not very accurate, it can still be useful (Text Classification for Sentiment Analysis – Naive Bayes Classifier, 2010).

The tasks involved in sentiment analysis include finding opinions, identify the sentiments they express, and then classifying their polarity.

2.2.2 Sentiment Classification Techniques

There are two main sentiment classification techniques, these are lexical-based approach and machine learning approach. These two have subgroups which are shown in Figure 1 (Medhat, Hassan, & Korashy, 2014).

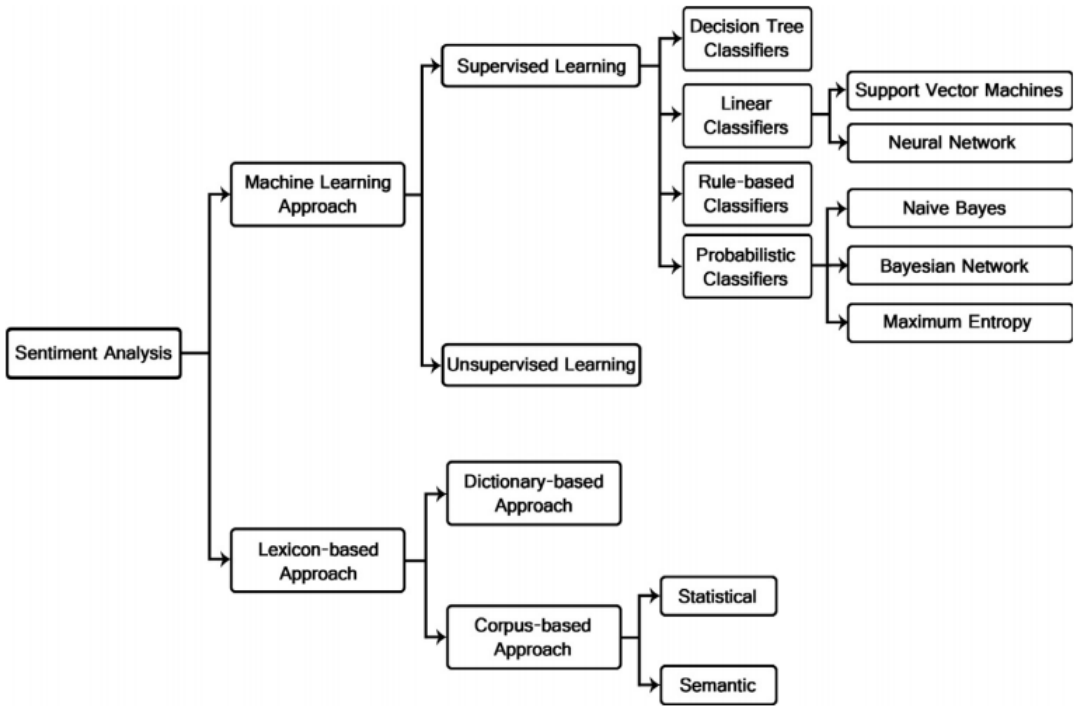

Figure 3. Sentiment Classification Techniques

Lexical and machine learning approaches are sentiment classification techniques, they are used to classify a given piece of natural language text according to the opinions expressed in it.

For the dictionary-based lexical approach, a dictionary is prepared to store the polarity values of lexicons. For calculating polarity of a text, polarity score of each word of the text, if present in the dictionary, is added to get an 'overall polarity score'. If the overall polarity score of a text is positive, then that text is classified as positive, otherwise it is classified as negative. Though this approach seems very basic, variants of this lexical approach have been reported to have considerably high accuracy.

Since the polarity of the text depends on the score given to each lexicon, there has been a large volume of work dedicated to discovering which lexical information is most efficient. A target word along with the word 'good', and a second with the target word with the word 'bad' was used by Turney (2002) in the AltaVista search engine queries. The polarity of the target word was determined by the search result that returned the most hits. This approach resulted in accuracy of 65%. In another research, Turney & Littman (2003) mapped the semantic association between the target word and each word from the selected set of positive and negative

words to a real number. By subtracting a word's association strength to a set of negative words from its association strength to a set of positive words, an accuracy rate of 82% was achieved.

For the machine learning approach, a series of feature vectors are chosen and a collection of tagged corpora is used to prepare a model. The model can then be used to classify an untagged corpus of text. In machine learning approach, the selection of features is crucial to the success rate of the classification. Most commonly, a variety of unigrams, which are, single words from a document, or n-grams, which are, two or more words from a document in sequential order are chosen as feature vectors. Most commonly employed classification techniques of machine learning are Support Vector Machines (SVMs), Naive Bayes and Maximum Entropy algorithm. Other techniques include Neural Networks, Bayesian Networks and Decision Tree Classifiers. The accuracy results for these algorithms greatly depends on the features selected.

2.2.3 Annotation

To check the accuracy of any emotion detecting algorithm, the results need to be compared to a human-labeled text. The process in which humans manually label a text is called annotation. Annotation can be done on multiple levels: word, sentence, paragraph, section, or even the entire document.

Annotation can be based on polarity, emotion and intensity. When annotating on polarity, text is labeled with positive, negative or neutral emotion. Text annotated on emotion is labeled based on some predefined list. The most common lists of emotions used are those suggested by Ekman, Izard and Plutchik (Mulcrone, 2012). Additionally, some studies annotate the text by labeling the intensity of the emotion. Intensity is based on a numeric scale, but there are no standards for this type of annotation. The first two categories are the ones that are applicable for this study, this is because emotions are closely related to the polarity of texts.

In general, studies either use pre-annotated datasets to test an algorithm or undergo a small annotation process. In the latter case, annotators who are qualified to label emotion in text, such as psychologists are used. The text is annotated to the given specification and level of analysis. Sometimes annotators are given an additional word list that consists of words from the original text. These lists help determine which words are attached to a specific emotion and which vary by context. When the annotation process is complete the agreement among the annotators is calculated using a method such as the Kappa Value.

2.3 Steps in Sentiment Analysis

2.3.1 Data Gathering

The data to be studied is collected from the appropriate data source.

2.3.2 Preprocessing

Once the data to be analyzed is collected, the text is split into individual words each word becoming a feature in the feature vector which is stored in a bag-of-words. This breakdown of a sentence into individual words is known as tokenization.

Although many tokenizers are geared towards throwing punctuation marks away, for sentiment analysis a lot of valuable information could be deduced from them. "!" puts extra emphasis on the negative/positive sentiment of the sentence, while "?" can mean uncertainty (no sentiment). ", ', [], () can mean that the words belong together and should be treated as a separate sentence. Same goes for words which are bold, *italic* or <u>underlined</u>. But symbols such as the "@" symbol, and links can be removed as they are regarded as noise generating elements.

Word Normalization should be applied. This is the reduction of each word to its base/stem form (by chopping of the affixes). This is known as stemming or lemmatizing. An example is walking to walk. Capital letters should be normalized to lowercase, unless it occurs in the middle of a sentence; this could indicate the name of a writer, place, brand etc. Words with an apostrophe should also be handled. "George's phone" should obviously be tokenized as "George" and "phone", but I'm, we're, they're should be translated as I am, we are and they are. To make it even more difficult, it can also be used as a quotation mark.

Other preprocessing steps include the elimination of stop words. These are the un-informative words in tweets which include "so", "and", "or" and "the". A stop-word list can be created or searched against a language-specific stop word dictionary. Further, Parts of Speech taggers (PoS) are used to classify words into the English 8-parts of speech. Nouns and pronouns do not contain any sentiment according to (Fang and Zhan, 2015). As such these words are exempted from the classification process.

2.3.3 Feature Extraction and Sentiment Classification

After the text has been segmented into sentences, each sentence has been segmented into words, the words have been tokenized and normalized. We can make a simple bag-of-words model of the text. In this bag-of-words representation you only take individual words into account and

give each word a specific subjectivity score. This subjectivity score can be looked up in a sentiment lexicon. If the total score is negative the text will be classified as negative and if it is positive the text will be classified as positive.

The sentiment lexicon can be created using some simple statistics of the training set. To do this the class probability of each word present in the bag-of-words will be determined.

The sentiment lexicon is simple to make, but is less accurate because it does not take the word order of the grammar into account. A simple improvement on using unigrams would be to use bigrams and trigrams. That is, not to split a sentence after words like “not”, ”no”, ”very”, “just” etc. It is easy to implement but can give significant improvement to the accuracy.

The best words to put in a bag-of-words include salient words that give domain specific information and discriminatory words that help to clear distinctions of polarities (Mulcrone, 2012). The bag-of-words model simply uses a statistical approach to classify polarities.

2.3.4 Train and build model

The above mentioned steps will be carried out in the training set. A test set will then be used to do classifications and to tell the efficiency of the classifier. Overfitting can be checked here.

2.4 Algorithm

2.4.1 Naïve Bayes Classifier

A Naive Bayes classifier is a simple probabilistic model based on the Bayes rule along with a strong independence assumption.

The Naïve Bayes model involves a simplifying conditional independence assumption. That is, given a class (positive or negative), the words are conditionally independent of each other. Due to this simplifying assumption the model is termed as “naïve”. This assumption does not affect the accuracy in text classification by much but makes really fast classification algorithms applicable for the problem.

The maximum likelihood probability of a word belonging to a particular class is given by the equation:

$$P(x_i|C) = \frac{Count\ of\ x_i\ in\ texts\ of\ class\ C}{Total\ number\ of\ words\ in\ texts\ of\ class\ C}$$

The frequency counts of the words will be stored in hash tables during the training phase.

According to the Bayes Rule, the probability of a particular text belonging to some particular class is given by:

$$P(c_i|t) = \frac{P(t_i|c_i) * P(c_i)}{P(t)}$$

If the simplifying conditional independence assumption is used, that is, given a class (positive or negative) the words are conditionally independent of each other. The following equation will be used.

$$P(c_i|t) = \frac{(\prod P(x_i|c_j)) * P(c_j)}{P(t)}$$

Here the x_i's are the individual words of the text. The classifier outputs the class with the maximum posterior probability.

Naive Bayes classifiers are thought to be less accurate than their more sophisticated counterparts like support vector machines and logistic regression classifier. A simple Naive Bayes classifier can be enhanced to match the classification accuracy of these more complicated models for sentiment analysis. The advantages of using Naive Bayes as our classifier are:

- Naive Bayes classifiers due to their conditional independence assumptions are extremely fast to train and can scale over large datasets.
- They are robust to noise and less prone to over-fitting.
- Ease of implementation is also a major advantage of Naive Bayes.

A significantly high accuracy can be achieved by applying the following processes to the simple Naive Bayes classifier:

1. **Bernoulli Naïve Bayes**

Duplicate words are removed from the text as they don't add any additional information, this type of Naïve Bayes algorithm is called Bernoulli Naïve Bayes.

Including just the presence of a word instead of the count has been found to improve performance marginally, when there is a large number of training examples.

The data under study should be distributed according to the multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued variable. Samples are required to be represented as binary-valued feature vectors

The decision rule for Bernoulli naive Bayes is based on

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

In text classification, word occurrence vectors (rather than word count vectors) are be used to train and use this classifier.

2. Laplacian Smoothing

If the classifier encounters a word that has not been seen in the training set, the probability of both the classes would become zero and there won't be anything to compare between. This problem can be solved by Laplacian smoothing

$$P(x_i C_j) = \frac{Count\ (x_i)\ +\ k}{(k\ +\ 1)\ *(No\ of\ words\ in\ class\ C_j)}$$

Usually, k is chosen as 1. This way, there is equal probability for the new word to be in either class. Since Bernoulli Naïve Bayes is used, the total number of words in a class is computed differently. For the purpose of this calculation, each text is reduced to a set of unique words with no duplicates.

3. Negation Handling

Negation handling is one of the factors that contribute significantly to the accuracy of this classifier. A major problem faced during the task of sentiment classification is that of handling negations. Since we are using each word as a feature, the word "good" in the phrase "not good" will be contributing to positive sentiment rather that negative sentiment as the presence of "not" before it is not taken into account.

To solve this problem a simple algorithm for handling negations using state variables and bootstrapping will be devised. This builds on the idea of using an alternate representation of negated forms. This algorithm uses a state variable to store the negation state. It transforms a word followed by a not into "not_" + word. Whenever the negation state variable is set, the

words read are treated as “not_” + word. The state variable is reset when a punctuation mark is encountered or when there is double negation.

Since the number of negated forms might not be adequate for correct classifications. It is possible that many words with strong sentiment occur only in their normal forms in the training set. But their negated forms would be of strong polarity.

This problem is addressed by adding negated forms to the opposite class along with normal forms of all the features during the training phase. That is to say if we encounter the word “good” in a positive document during the training phase, we increment the count of “good” in the positive class and also increment the count of “not_good” for the negative class. This is to ensure that the number of “not_” forms are sufficient for classification. This modification will result in a significant improvement in classification accuracy (about 1%) due to bootstrapping of negated forms during training. This form of negation handling can be applied to a variety of text related applications

4. n-grams

Generally, information about sentiment is conveyed by adjectives or more specifically by certain combinations of adjectives with other parts of speech.

This information can be captured by adding features like consecutive pairs of words (bigrams), or even triplets of words (trigrams). Words like "very" or "definitely" don't provide much sentiment information on their own, but phrases like "very bad" or "definitely recommended" increase the probability of a document being negatively or positively biased. By including bigrams and trigrams, we will be able to capture this information about adjectives and adverbs. Bigrams will be used as the 280 characters limit will hinder the use of more n-grams. The counts of the n-grams will be stored in a hash table along with the counts of unigrams.

5. Feature Selection

Feature selection is the process of removing redundant features, while retaining those features that have high disambiguation capabilities.

The use of higher dimensional features like bigrams and trigrams presents a problem, that of the number of features increasing. Most of these features are redundant and noisy in nature. Including them would affect both efficiency and accuracy. A basic filtering step of removing the

features/terms which occur only once will be performed. The features can then be further filtered on the basis of mutual information.

2.4.2 Predicting classifications

To predict a classification we simply look at which of the two classes give a higher Naïve Bayes probability: That is given a sample X and two classification categories ω1 and ω2, if P (ω1| x)>P (ω2| x) classify sample X as ω1 else classify as ω2. That is the classification opts for the class that gets a high probability with the Naïve Bayes formula.

2.5 Stock Prediction

2.5.1 Introduction

Stock prediction methodologies fall into three broad categories which can and often do overlap. They are fundamental analysis, technical analysis/charting and technological methods.

Technical analysis is the interpretation of the price action of a company's underlying stock (or any tradable financial instrument). It utilizes various charts and statistical indicators to determine price support/resistance, range and trends. It identifies historically relevant price patterns and behaviors to help forecast potential direction of the stock. This methodology focuses only on the price of the shares, not the operations of the company (Technical Analysis, n.d.).

Technical analysis seeks to determine the future price of a stock based solely on the (potential) trends of the past price (a form of time series analysis). Techniques such as exponential moving average (EMA) are employed.

This research work will focus on technical analysis methodology.

2.5.2 Techniques Used in Stock Prediction

2.5.2.1 Time Series Analysis

A time series is an ordered sequence of values of a variable at equally spaced time intervals. It can be taken on any variable that changes over time.

In investing, it is common to use a time series to track the price of a security over time. This can be tracked over the short term, such as the price of a security on the hour over the course of a business day, or the long term, such as the price of a security at close on the last day of every month over the course of five years.

Time series analysis can be useful to see how a given asset, security or economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period. It can therefore be used to determine the relationship between public sentiments of a stock and the movement in price of the mentioned stock.

Time series can be used for quantitative forecasting by using information regarding historical values and associated patterns to predict future activity. Most often, this relates to trend analysis, cyclical fluctuation analysis and issues of seasonality.

Inherent in the collection of data taken over time is some form of random variation. There exist methods for reducing or canceling the effect due to random variation. An often-used technique is "smoothing". This technique, when properly applied, reveals more clearly the underlying trend, seasonal and cyclic components. Therefore, in this research work there will be need for **smoothing functions** that react quickly to changes in the signal, hence the need for **moving averages**.

2.5.2.2 Moving Average (MA)

A moving average (MA) is a widely used indicator in technical analysis that helps smooth out price data by filtering out the “noise” from random price fluctuations and form a trend following indicator. They do not predict price direction, but rather define the current direction with a lag. Moving averages lag because they are based on past prices. Despite this lag, moving averages help smooth price action and filter out the noise.

The two basic and commonly used moving averages are the:

i. **Simple moving average (SMA).**
ii. **Exponential moving average (EMA)**.

2.5.2.2.1 Simple Moving Average

A Simple Moving Average is formed by computing the average price of a security over a specific number of periods. Most moving averages are based on closing prices. A 5-day simple moving average is the five-day sum of closing prices divided by five. As its name implies, a

moving average is an average that moves. Old data is dropped as new data comes available. This causes the average to move along the time scale.

2.5.2.2.2 Exponential Moving averages (EMA)

An exponential moving average (EMA) is a type of moving average that is similar to a simple moving average, except that more weight is given to the latest data. It's also known as the exponentially weighted moving average. This type of moving average reacts faster to recent price changes than a simple moving average.

It is often used where latency is critical, such as in real time financial analysis. In this average, the weights decrease exponentially. Each sample is valued some percent smaller than the next most recent sample. With this constraint the moving average can be calculated very efficiently.

The formula is:

$$avg_t = (alpha * sample_t) + ((1 – alpha) * avg_{t-1})$$

Where alpha is a constant that describes how the simple weights decrease over time. For example if each sample was to be weighted at 80% of the value of the previous sample, you would set alpha = 0.2.

Each new sample needs to be average with the value of the previous average. So computation is very fast. In theory all previous samples contribute to the current average, but their contribution becomes exponentially smaller over time.

This is a very powerful technique, and probably the best in getting a moving average that responds quickly to new samples, has good smoothing properties and is fast to compute.

It is therefore beneficial to apply the Exponential Moving Average in our Time Series Analysis to get better results in predicting the direction of the stock market movement.

2.6 Previous Work

2.6.1 Twitter Sentiment Analysis to Predict the Stock Market Movement

Earlier work done by Bollen, Mao, & Zeng, (2010) shows how collective mood on Twitter (aggregate of all positive and negative tweets) is reflected in the Dow Jones Industrial Average (DJIA) index movements. They investigated whether public sentiment, as expressed in large-scale collections of daily Twitter posts, could be used to predict the stock market.

They used two tools to measure variations in the public mood from tweets submitted to the Twitter service from February 28, 2008 to December 19, 2008. The first tool, OpinionFinder, analyzed the text content of tweets submitted on a given day to provide a positive vs. negative daily time series of public mood. The second tool, Google-Profile of Mood States (GPOMS), similarly analyzed the text content of tweets to generate a six-dimensional daily time series of public mood to provide a more detailed view of changes in public along a variety of different mood dimensions.

The resulting public mood time series were correlated to the DJIA to assess their ability to predict changes in the DJIA over time. The results indicated that the prediction accuracy of a standard stock market prediction models is significantly improved when certain mood dimensions are included, but not others. In particular variations along the public mood dimensions of Calm and Happiness as measured by GPOMS seem to have had a predictive effect, but not general happiness as measured by the OpinionFinder tool.

The research found an accuracy of 87.6% in predicting the daily up and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error by more than 6%. This therefore shows that there is a strong correlation between twitter data and stock market movement.

2.6.2 Sentiment Classification using an Enhanced Naive Bayes Model.

Narayanan, Arora & Bhatia showed that a simple Naive Bayes classifier can be enhanced to match the classification accuracy of more complicated models for sentiment analysis by choosing the right type of features and removing noise by appropriate feature selection. Naive Bayes classifiers were thought to be less accurate than their more sophisticated counterparts like support vector machines and logistic regression. The enhanced Naïve Bayes classifier was able to achieve an 88.8% accuracy from the 73.77% accuracy when using the original Naive Bayes algorithm with Laplacian smoothing. The improvements from applying the various processes are illustrated in Figure 2.

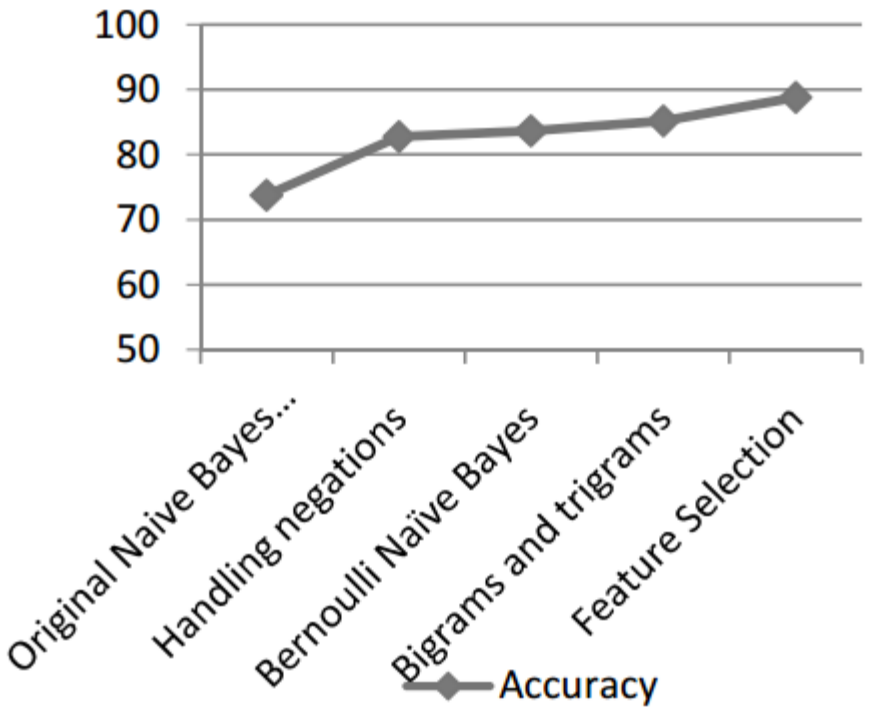

Figure 4. Evolution of classification accuracy.

2.6.3 Forecasting Stock Market Trend using Exponential Moving Average

From the research, Alexander Decker concluded that the curve of the moving average shows the market/ stock trend. If the curve is in an upward direction, the market heads towards Bull Run. If the moving average is going down the market is in Bear phase and a flat moving average shows consolidation.

He further states that there are 3 steps in calculating the EMA:

1. Calculate the simple moving average for the initial EMA value. An exponential moving average (EMA) has to start somewhere, so a simple moving average is used as the previous period's EMA in the first calculation.
2. Calculate the weighting multiplier.
3. Calculate the exponential moving average for each day between the initial EMA value and today, using the price, the multiplier, and the previous period's EMA value.

The formula below is for a 10-day EMA.

Initial SMA: 10-period sum / 10

Multiplier: (2 / (Time periods + 1)) = (2 / (10 + 1)) = 0.1818 (18.18%)

EMA: {Close – EMA (previous day)} x multiplier + EMA (previous day).

By using this technical analysis tool, he was able to find out the trends in different stocks; whether they are going upwards, downwards or stagnant. This increases the chance of investors to predict the prices more accurately by prediction the trend reversals in advance and hence increased profit in the share markets.

In short, EMA when combined with market behavioral analysis, such as Twitter sentiments about a company, then the probability of it being correct for forecasting market trends will increase.

2.7 Conclusion

Opinion mining of people's thoughts on the stock market is very important in the construction of a stock prediction model. When combined with stock prediction techniques such as the exponential moving average, the degree of accuracy improves significantly.

CHAPTER 3: RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

This chapter covers the research design, research methods, data collection tools used to gather information during the research and the development tool used

3.2 Research Design

A research design is an arrangement of conditions or collections (Research design, n.d.).

The research design chosen for this study is the Correlation design type and the subtype under consideration is Observational study. The type of observational study that will be looked at is longitudinal study which is a correlational research study that involves repeated observations of the same variables over long periods of time (Observational study, n.d.).

The longitudinal study is chosen as the twitter feeds as well as the stock data to be analyzed will involve repeated observations of these variables over a long period of time before conclusions are made.

3.3 Research Methodology

3.3.1 Research Methods/ Processes/ Techniques/ Procedure

The works reviewed in this research were selected and analyzed based on the following criteria:

1. Steps that can be taken to enhance the Naïve Bayes classifier for it to be used for sentiment analysis classification with a high degree of accuracy.
2. Techniques that can be used for stock market prediction.
3. Studies that show a correlation between stock market movement and the public sentiment on this stocks

3.3.2 Data Collection Tools

The techniques used for data collection are:

1. **The study of existing literature.**

This is so because the research is a confirmatory one. The research tests priori hypotheses that were set by researches who have studied the proposed solution such as Bollen et.al. Previous studies carried out by various researchers formed a rich source of information and was the basis of all other data collection techniques used.

2. **Primary Data collection**

Twitter data and the stock data to be analyzed comes directly from the source. Twitter feed studied are those that users have typed.

3. Secondary Data collection

Secondary data in form of an annotated stock market corpus is used in this research to train the sentiment classification model.

3.4 Development Tool

3.4.1 Process Model

3.4.1.1 Waterfall Model

This Software Development Lifecycle Model (SDLC) model is used as the requirements are well documented, clear and fixed. This model is simple and easy to understand and use and it has clearly defined stages and milestones which are key in this project.

The sequential phases in this model are:

1. Requirement Gathering and analysis
2. System Design.
3. Implementation.
4. Integration and Testing.
5. Deployment of system
6. Maintenance

CHAPTER 4: PROJECT SYSTEM ANALYSIS

4.1 Introduction

This section will cover the system methodology used, the data sources, feasibility study and the system requirements.

4.2 System Methodology

Objected Oriented Programming (OOP) methodology will be used. This is because the project is complex thus the kind of modularity that OOP provides will be beneficial when it comes to making the program simpler and thus understandable. It will also be easier to troubleshoot problems that may be encountered.

It offers the most effective problem solving solution as its modules are reusable, they scale well and they are relatively easy to maintain.

4.3 Data Sources

The sentiment data to be analyzed will come from Twitter, the social media platform. The platform offers an Application Programming Interface that is free and easy to use and it provides a lot of public sentiment about various products and services that are traded in the stock market.

The stock market data will come from Alpha Vantage Inc. It is a leading provider of free APIs for real-time and historical data on stocks among other financial data. The best part is that they provide this data for free and their Application Programming Interface is well documented making it easy to use.

4.4 Challenge in Sentiment Analysis of Twitter Data

Twitter data costs a lot of money, and if it has not been possible to retrieve or set up a system to retrieve Twitter data within 7 days on a topic of interest, then it becomes difficult to obtain the data. This is because using the free Twitter public API ecosystem it is only possible to retrieve Twitter data going back in time 7 days. However, it is possible to obtain this data through other means such as obtaining them from Twitter at a fee, by using a licensed reseller of Twitter data or by using an existing dataset. Historical Twitter data can range from not that expensive, to very expensive depending on both the query and time of retrieval (Ahmet, 2015).

4.5 Feasibility Study

This is the analysis of a problem to determine if the conditions are right for it to be solved effectively. The results of the study will determine whether the solution should be implemented.

Four aspects of the project will be studied.

4.5.1 Technical Feasibility

The technical feasibility checks whether the right technology exists to solve the problem.

The Twitter streaming API will support all the data requirements when it comes to obtaining public sentiments. All its shortcomings can be easily solved. One major one was only obtaining data from the past 7 days and this will be mitigated by storing our results in our own database for future references.

The financial API from Alpha Vantage Inc. is perfect for our needs and no further modification is needed to it.

Python Programming Language will also provide us with its Natural Language Toolkit (NLTK) library which will be crucial for natural language processing and also scikit-learn library for scientific analysis.

4.5.2 Economic Feasibility

Economic feasibility studies the costs and benefits associated with a project.

The project cost is feasible as all the APIs to be used are free and the software to implement the project is open source. The monies allocated for the project will also be enough for the stationary and Internet charges required in the course of the project.

The benefits that come with implementing this project far outweighs any cost that will be incurred.

4.5.3 Schedule Feasibility

This checks to see if the solution can be implemented in the stipulated time.

The project can be implemented in the stipulated time. The setbacks in terms of the lecturers strike will mean some minor adjustments will be needed in our schedule. But all the objectives will still be achieved.

4.5.4 Operational Feasibility

The operational feasibility seeks to assess if the solution will work and if it will be possible to maintain it.

The interface of the solution will follow all the standards that have been put in place by User Interface experts to ensure it is user friendly. This will mean that the intended users will have an easy time in learning to use it thus ensuring no resistance is faced.

The project will also be well documented ensuring that future maintenance be it corrective or an enhancement of its features will be easy for whomever will be working on it.

4.6 System Requirements

4.6.1 Functional Requirements

The following are the functionalities required from the system:

1. Training the naive Bayes classifier for sentiment classification. The naive Bayes classifier will be enhanced to offer more accurate classification of sentiments as this is key in ensuring the model has a high degree of accuracy.
2. Analyzing stock market data for accurate predictions. Time series and exponential moving averages will be used to ensure a high degree of accuracy in the stock prediction that will be carried out.
3. Reporting statistics for stock market movement. The analyzed sentiments and stock market data will be combined to come up with a model that accurately predicts the movement of stock market data. The model will use graphs to come up with useful visualizations.
4. Historical twitter data and statistical storage: Twitter data will be filtered and stored for future access as well as the statistical data.

4.6.2 Non Functional Requirements

1. User friendliness: The system should be easy to use.
2. Reliability and availability: This two will depend on the Internet availability as tweets need an Internet connection to be fetched.
3. High response time: The trained classifiers will be stored and reused later to ensure that retraining is not needed as the training takes a long time.

4. Interoperability: The system should be used in all major operating systems, which are Microsoft Windows, Linux as well as Mac OS platform.

CHAPTER 5: System Design

5.1 Introduction

This section covers the architecture, modules, components and data for the system to satisfy the specified requirements.

It will be divided into physical and logical design.

5.2 Logical Design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system.

5.2.1 Data Flow Diagram (DFD)

A data flow diagram is a graphical representation of the flow of data through an information system. It shows how information is input to and output from the system, the sources and destinations of that information, and where that information is stored (Data Flow Diagram, n.d.).

In this project data flow diagram will be used to model the flow of data, relationships and storage of messages which will be used in training and retraining of the model.

5.2.1.1 Level 0 DFD/ Context Diagram

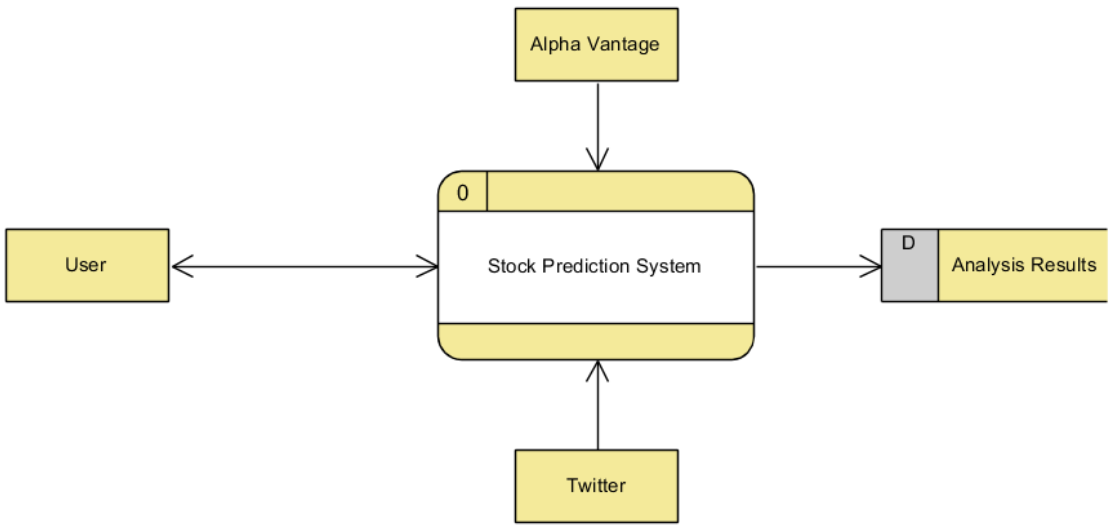

Figure 5. Data flow diagram level 0 showing the relationship between the system user, twitter and alpha vantage systems as well as the storage of analyzed results

5.2.1.2 Level 1 DFD

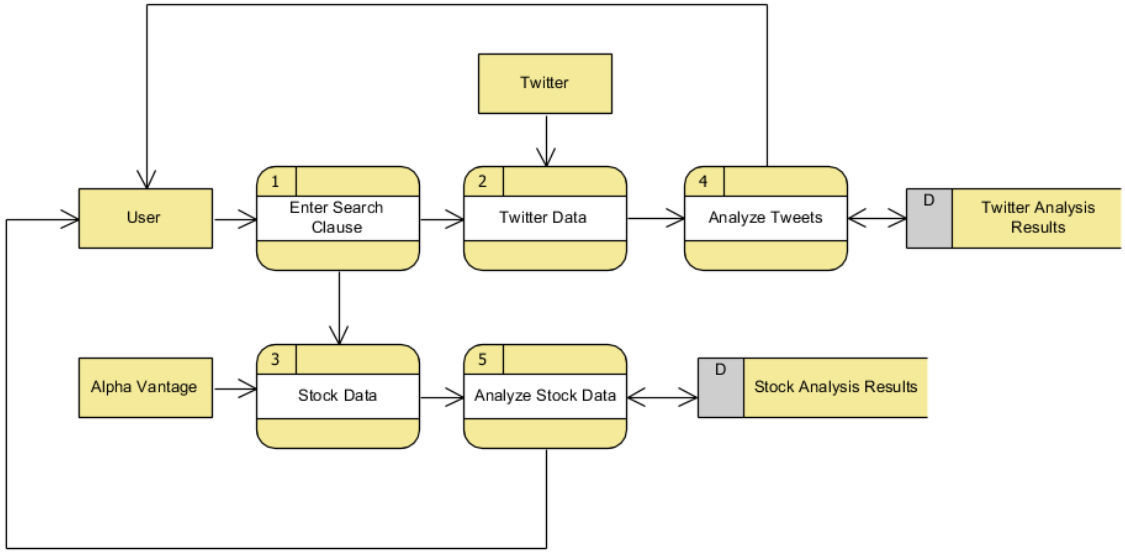

Figure 6. Data flow diagram level 1 showing basic processes of the system and the data flow between the processes

5.2.1.3 Level 2 DFD

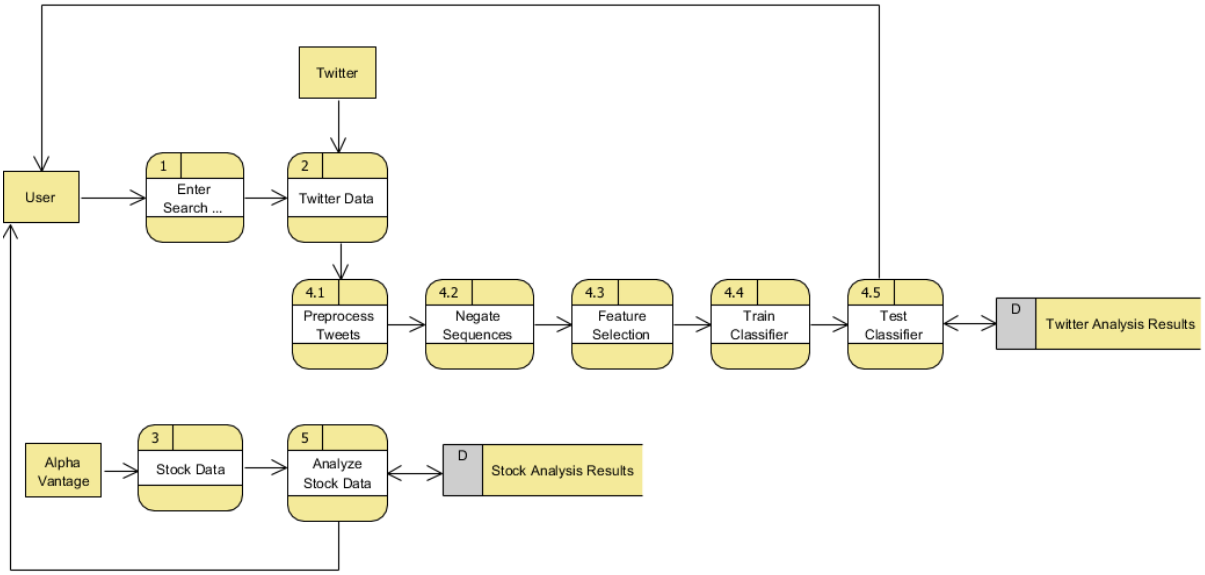

Figure 7. Data flow diagram level 2 showing more detailed look at the processes that make process 4

5.2.2 Activity Diagram

Activity diagram shows the flow of activities in a system. It describes the sequence from one activity to another, and describes the parallel, branched and concurrent flow of the system.

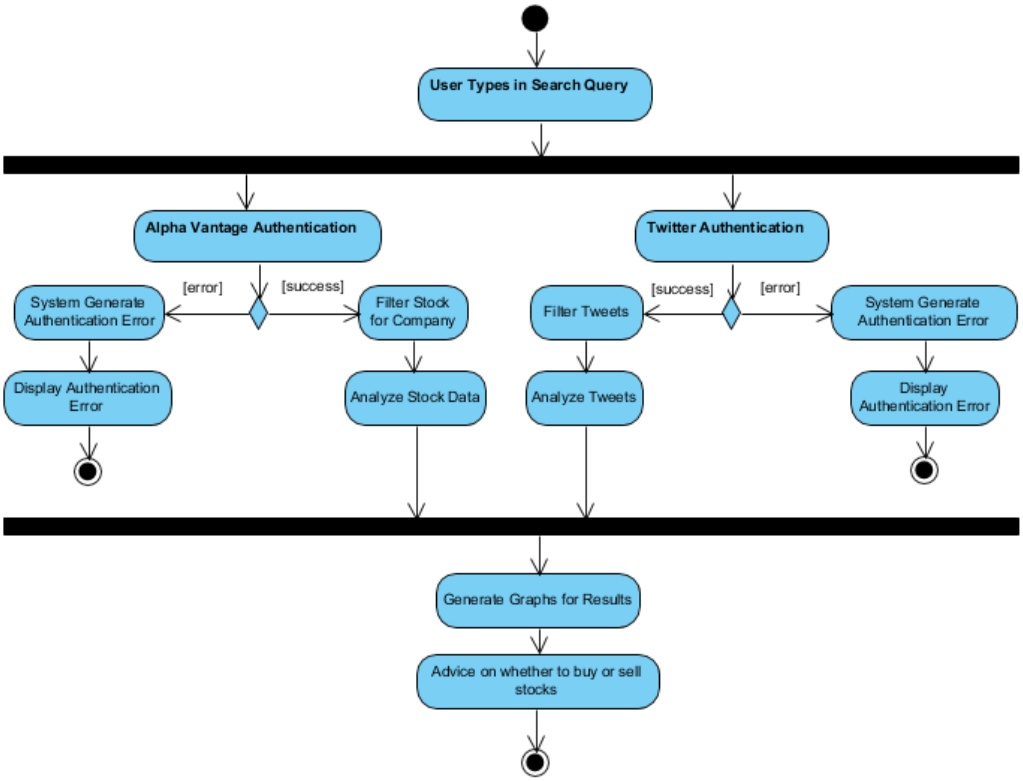

Figure 8. Activity diagram showing the flow of activities in the system

5.2.3 Use Case Diagram

Use case diagram is used to gather system requirements and actors in the system.

Actors define the role played by a user or any other system that interacts with the system being designed. For our case the actors are an investor using the system, twitter and alpha vantage system.

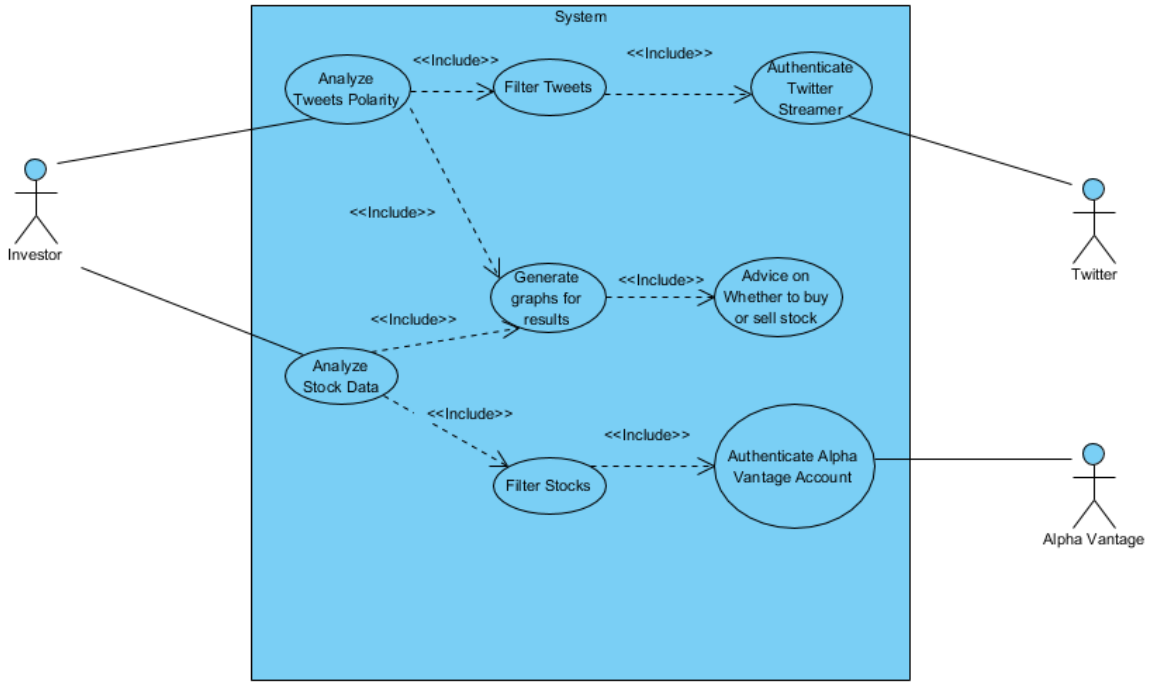

Figure 9. Use case diagram showing the actors of the system and the various use cases

5.2.4 Class Diagrams

A class diagram describes the static aspect of the system. It shows the structure of classes that make up the system and the relationship between them.

The following classes and their relationships were identified.

i. APISetup class: This is the class responsible for setting up the API authentication needed to use the Twitter and Alpha Vantage APIs. It also takes the search clause that the user inputs and filters the tweets and stock data to be analyzed using the search clause.

ii. TwitterClassifier class: This class gets the twitter data that is filtered using the search clause from APISetup classes and carries out sentiment analysis on this data.

iii. StockPredictor class: This class gets the stock data that is filtered using the search clause and analyzes it.

iv. MakePrediction class: This class visualizes the results obtained from the TwitterClassifier and StockPredictor class and then makes the most suitable prediction.

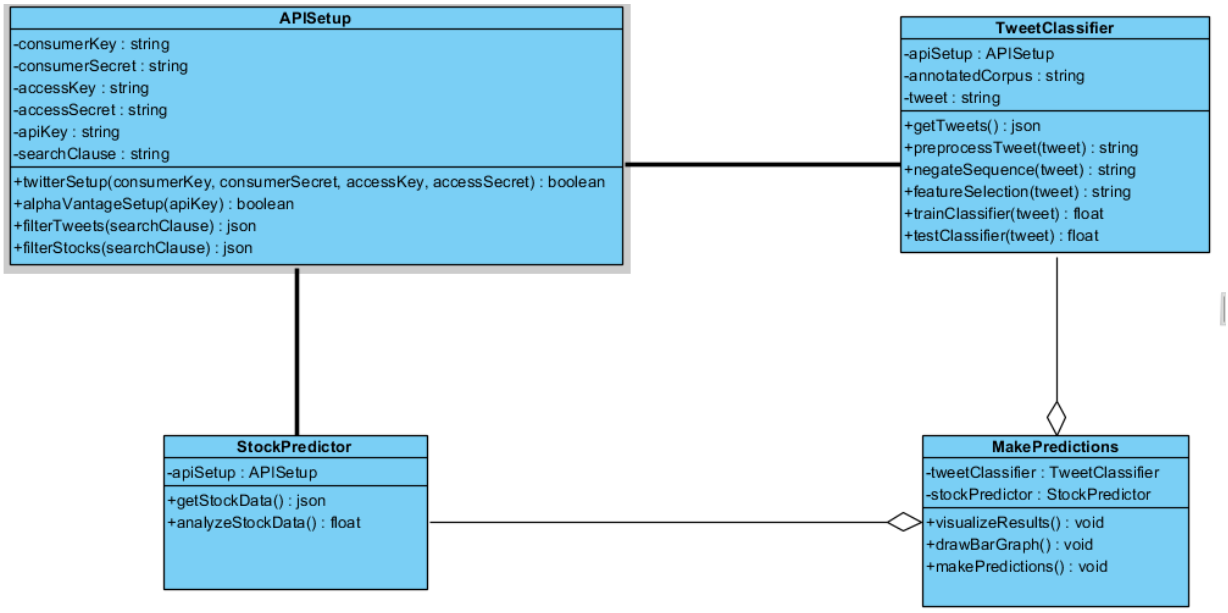

Figure 10. Class diagram showing the attributes and methods of classes and relationship between them

5.2.5 Entity Relationship Diagram (ERD)

Analyzed tweets and their polar label are stored in an SQL database. This is the case due to the 7 day twitter limit imposed by twitter and also to make the process of training the model faster.

The analyzed tweets will be stored using the following database fields: Search clause that generated the tweet, the date the tweet was streamed and the polarity tag and it's score.

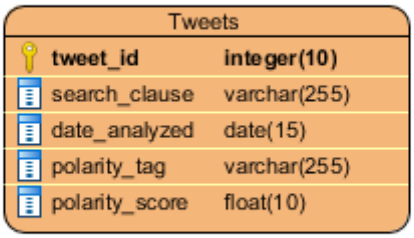

Figure 11. Entity relationship diagram showing attributes of Tweets table

5.3 Physical Design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed.

5.3.1 User Interface Design

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.

The UI is the platform through which users will be able to access the system, thus it will be necessary to make it as simple as possible to make it user friendly for all types of users.

There is need for a search bar in our UI so that a user can search for a company by its ticker symbol to see the stock predictions that will guide them into making the best financial decision.

The result of the search will be visualized by using graphs. There will be line graphs for stock prices movement, another one for the public sentiment on the given stock, there will also be one that will combine the two graphs to visualize their correlation. There will also be a bar graph showing percentage of positive and negative tweets per day. There will also be a textbox at the very top of the webpage that will suggest to the user whether to buy or sell the given stock depending on the results from the sentiment and stock prices graphs.

There will be a tweets section at the very bottom of the site to display tweets related to the query parameter input by the user.

5.4 Systems Architecture

The system architecture is built on 4 key components which play an important role in the success of the system as a whole.

i. The **Interface** where the user interacts with the system
ii. The **Make Prediction Component**. This is the key component of the system. It visualizes results and advices user on whether to sell or buy stocks.
iii. The **Sentiment Classifier Component**. Classifies user sentiments that are later on use in the analysis of whether or not stocks should be bought or sold.
iv. The **Stock Prediction Component.** Analyzes stock data that will influence the Make Prediction component's decision.

Reference

1. *Introduction to Sentiment Analysis*. (n.d.). [EBook] p.5. Retrieved from: https://lct-master.org/files/MullenSentimentCourseSlides.pdf.
2. *Stock market prediction*. (n.d.) Retrieved from https://en.wikipedia.org/wiki/Stock_market_prediction.
3. *Stock.* (n.d.) Retrieved from https://en.wikipedia.org/wiki/Stock.
4. *Technical Analysis* (n.d.) Retrieved from https://www.investorsunderground.com/technical-analysis/
5. Burton G. Malkie (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, *17*(1), 59–82.
6. Johan Bollen, Huina Mao, and Xiao-Jun Zeng (2010). Twitter mood predicts the stock market. *1*(1), 1–8.
7. Bhadane, C., Dalal, H., & Doshic H., (2015). Sentiment analysis: Measuring opinions. *International Conference on Advanced Computing Technologies and Applications (ICACTA-2015)*. 808-814. Retrieved from http://www.sciencedirect.com/science/article/pii/S1877050915003956
8. *Text Classification for Sentiment Analysis – Naive Bayes Classifier*. (2010). Retrieved from https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/
9. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*, 1093-1113
10. Turney, P. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of ACL*. 417-424.
11. Turney, P.D., Littman. 2003. M.L.Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems*. 315-346.
12. Ahmet, T. (2015). *Text classification and sentiment analysis.* Retrieved from http://ataspinar.com/2015/11/16/text-classification-and-sentiment-analysis/
13. Ahmet, T. (2016). *Sentiment analysis with bag-of-words.* Retrieved from http://ataspinar.com/2016/01/21/sentiment-analysis-with-bag-of-words/
14. Mulcrone, K. (2012). Detecting Emotion in Text. University of Minnesota.
15. Fang, X. & Zhan, J. (2015). Sentiment analysis using product review data.

16. Narayanan, V., Arora, I., & Bhatia A. (2007). Fast and accurate sentiment classification using enhanced Naive Bayes model.
17. Naïve Bayes (n.d.). Retrieved from http://scikit-learn.org/stable/modules/naive_bayes.html
18. Time Series (n.d.). Retrieved from https://www.investopedia.com/terms/t/timeseries.asp#ixzz55mTuzkK8
19. Moving Average (MA), (n.d.). Retrieved from. https://www.investopedia.com/terms/m/movingaverage.asp#ixzz55mXdcgu6
20. Moving Averages- Simple and Exponential, (n.d.). Retrieved from. http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages
21. Exponential Moving Averages For Irregular Time Series, (2013) https://oroboro.com/irregular-ema/
22. Research design. (n.d.) Retrieved from https://en.wikipedia.org/wiki/Research_design
23. Observational Study. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Observational_study
24. Alpha Vantage. (n.d.) Retrieved from https://www.alphavantage.co/
25. Feasibility Study. (n.d.) Retrieved from https://www.techopedia.com/definition/19297/feasibility-study
26. Data Flow Diagram. (n.d.) Retrieved from https://www.computerhope.com/jargon/d/data-flow-diagram.htm