Application Of Text Classification And Clustering Of Twitter Data For Business Analytics

A Project work submitted to
Department of Computer Science and Engineering
University College of Sciences
Acharya Nagarjuna University

In partial fulfillment of the requirements for
The award of the degree of

Master of Computer Applications

by

GUBBALA DURGA PRASAD
Regd. No. Y23MC20018

Under the guidance of

Dr. U. SURYA KAMESWARI., M.Sc., M. Tech., Ph.D.
Assistant Professor
Department of computer science & engineering
University College of Sciences
Acharya Nagarjuna University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY COLLEGE OF SCIENCES
ACHARYA NAGARJUNA UNIVERSITY
Nagarjuna Nagar, Guntur,
Andhra Pradesh, India

April 2024

DECLARATION

I hereby declare that the entire thesis work entitled " **APPLICATION OF TEXT CLASSIFICATION AND CLUSTERING OF TWITTER DATA FOR DATA ANALYTICS"** is being submitted to the Department of **Computer Science and Engineering, University College of Sciences, Acharya Nagarjuna University,** in partial fulfillment of the requirement for the award of the degree of **Master of Computer Applications (MCA)** is a Bonafide work of my own, carried out under the supervision of **Dr. U. SURYA KAMESWARI,** Assistant Professor, Department of Computer Science & Engineering, Acharya Nagarjuna University.

I further declare that the Project, either in part or full, has not been submitted earlier by me or others for the award of any degree in any University.

GUBBALA DURGA PRASAD

Reg. No. Y23MC20018

ACHARYA NAGARJUNA UNIVERSITY

NAGARJUNA NAGAR, GUNTUR.

Department of Computer Science & Engineering.

CERTIFICATE

This is to certify that this project entitled "APPLCIATION OF TEST CLASSIFICATION AND CLUSTERING OF TWITTER DATA FOR DATA ANALYTICS" is a Bonafide record of the project work done and submitted by G. DURGA PRASAD (Y23MC20018) during the year 2023 - 2024 in partial fulfillment of the requirements for the award of degree of Master of Computer Applications (MCA) in the department of Computer Science & Engineering. I certify that he carries this project as an independent project under my guidance.

Head of the Department
(Prof. K. Gangadhara Rao)

Project Guide
(Dr. U. Surya Kameswari)

External Examiner

ACKNOWLEDGEMENTS

Undertaking this Project has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

I would like to first say a very big thank you to my supervisor **Dr. U. SURYA KAMESWARI** for all the support and encouragement he gave me. Her friendly guidance and expert advice have been invaluable throughout all stages of the work. Without her guidance and constant feedback this Project work not have been achievable.

I would also wish to express my gratitude to **Prof. K. Gangadhara Rao** for extended discussions and valuable suggestions which have contributed greatly to the improvement of the thesis.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department which helped us in successfully completing our project work. Also, I would like to extend our sincere regards to all the non-teaching staff of the department for their timely support.

I must also thank my parents and friends for the immense support and help during this project. Without their help, completing this project would have been very difficult.

ABSTRACT

In the recent years, social networks in business are gaining unprecedented popularity because of their potential for business growth. Companies can know more about consumers' sentiments towards their products and services, and use it to better understand the market and improve their brand. Thus, companies regularly reinvent their marketing strategies and campaigns to fit consumers' preferences. Social analysis harnesses and utilizes the vast volume of data in social networks to mine critical data for strategic decision making. It uses machine learning techniques and tools in determining patterns and trends to gain actionable insights. This paper selected a popular food brand to evaluate a given stream of customer comments on Twitter. Several metrics in classification and clustering of data were used for analysis. A Twitter API is used to collect twitter corpus and feed it to a Binary Tree classifier that will discover the polarity lexicon of English tweets, whether positive or negative. A k-means clustering technique is used to group together similar words in tweets in order to discover certain business value. This project attempts to discuss the technical and business perspectives of text mining analysis of Twitter data and recommends appropriate future opportunities in developing this emerging field.

TABLE OF CONTENTS

Chapter 3: FEASIBILITY STUDY

Chapter 4: RESULTS AND DISCUSSIONS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In today's world, most companies around the globe are consistently turning to social media to strategize and create their business policies, rules as well as make use of the information for decision making. The social media contain a large volume of unstructured data (ex: tweets, comments, blogs, forum discussions, a user post, and reviews) done by different users with different or polarizing opinions as well as different perspectives. Most companies or organizations use the information derived from all this data for business intelligence, such as customer profiling and content analytics.

1.1 Introduction to Twitter

Twitter is a popular social networking online service that is mainly used as a marketing and promotion tool by most companies. Twitter data or tweets consist of not only user information, but also opinionated information. Ex: The different and varying opinions, views, feelings of different users etc. towards a particular topic, trend, or issue. These tweets range from users sharing or posting positive, negative, or neutral thoughts on a current topic, product, or trending matter. They can also be unbiased or biased based on the interests of the user. From a business perspective, the increasingly large volume and range of tweets are enough for companies to get enough information about their marketing products, goods, facilities, features from their customers in a cost-effective as wells as time-saving manner.

1.2 Sentiment Analysis

Sentiment analysis alludes to the utilization of natural language processing (NLP), text analysis, etymology, and insights to reliably set up extricate measure and study emotive states as well as natural or individual information or opinion. It is broadly applied to the voice of the customer or client materials like audits, reviews, reactions on the web and web-based life and social insurance materials for applications like customer administration.

1.3 Overview of Project

The purpose of this project is to extract a list of tweets posted in real time about a specific company and build machine learning algorithms that can accurately classify these Tweets as positive tweets, negative tweets, or neutral tweets after calculating the polarity and subjectivity of the respective tweet. Our theory is that we can acquire high precision on classifying sentiment in Twitter messages utilizing these procedures. In order to achieve this, we have followed the development phases listed below:

1. **Preparing The Test Set**: This stage includes getting the authentication credentials that are required by the Twitter API for authenticating our Python script. This step is followed by generating or implementing a function to extract live tweets for building the Test set.
2. **Collection of Data**: This stage focuses on the retrieval of recently posted live tweets using the Twitter API Cursor method to get an object containing tweets that refer to the particular company. They are then extracted to a CSV file as a dataset for further analysis and pre-processing.
3. **Pre-processing of Data**: This stage focuses on data pre-processing or cleaning of extracted tweets to make the twitter data easy to understand and analyze their subjectivity, polarity, and sentiment.
4. **Model Preparation**: This stage focuses on preparing the machine-learning model for classifying the twitter data. We need to specify the testing and training set after data is pre-processed and choose the appropriate algorithms that will yield the best accuracy or result.
5. **Training of Data**: This stage focuses on implementing the different machine learning algorithms, i.e., Naïve Bayes (NB) Classifier, Support Vector Machine (SVM) Classifier, and Random Forest Classifier (RF) and K Nearest Neighbors (KNN) Classifier for training the data that we have used as a training set for this project.

6. **Testing the Model**: The final stage focuses on experimenting with the model by providing an input of our own and check the accuracy of the prediction for the different algorithms used for this project.

1.4 Use of Sentiment Analysis for Business Analytics

In the present condition, because of information overload, organizations typically have enormous volumes of client input gathered to a fault. Thus, companies these days usually have large volumes of customer feedback collected to gain an understanding or overview of their facilities, services, marketing strategies for new products, etc. Lamentably, it is still hard for us to examine and break down the data physically without mistake or inclination

Sentiment analysis yields some resolutions into what the most pressing issues are, from the attitude or point of view (POV) of consumers, at any rate. The decisions made as a result of sentiment analysis are often dependent on a lot of information rather than a person's untrustworthy instinct. A large amount and assortment of information can cause hurdles or lead to roadblocks when trying to find patterns in text. However, by using analytical methods and procedures, such difficulties are feasible. As a result, conducting social media research is becoming widely common practice and has also become a regular occurrence for companies, sites, apps to gain an understanding of their current quality level. In his report, [1] deduced that the expanding development pattern is credited to the need of organizations to understand the perspective of their clients by employing social media analytics on their services and then proceeding to use the information derived from the analysis to determine new policies and work on optimizing performance of their own applications in a logical and rational manner. This helps them keep track of the quality level of their products as well as develop an understanding of the satisfaction level of their customers. Due to this, companies have started becoming fully aware of the advantages of text mining using sentiment analysis and reap its benefits for their own purpose [2]. They

use it for further advertising or marketing of their products and also improve their standing in society.

Recently, Halibas [3] investigated the utilization of sentiment analysis in business applications. He exhibited that hidden information can be utilized for exploring the overall assessment of customers towards a specific brand and gain insight into overall product quality after the completion of investigation.. He tried to refine and improve his own methodology of performing this analysis, despite the lack of research in this domain [4].

Thus, I have decide to go with a simpler solution of applying Sentiment Analysis to Twitter data in order to extract feedback or information from it that it may be useful to give insights on the popularity of a particular brand or company and may pave the way forward for future decision making and policies, etc.

1.5 Problem Statement

This project introduces Sentiment Analysis and its application to gain insights into customer feedback. It highlights the importance of polarity of different views, perspectives or opinions of people in different topics, trends or products and how useful it could be for companies or business organizations to proceed forward in decision making or policies

CHAPTER 2

REVIEW OF LITERATURE

This chapter aims to provide detailed information on development of Sentiment Analysis projects by various published journals and conference proceedings of related field. This chapter includes the comparisons on the efficiency and accuracy of various works and studies to support directly or indirectly helps to carry out the present research work. This review contains the information on various steps of sentiment analysis, different machine learning techniques that were used previously for sentiment analysis, such as Naïve Bayes, SVM, Decision Tree, etc. as well as sources, applications and recent research work.

2.1 Related Work done on Sentiment Analysis

Pang et.al. took into account the idea of sentiment type being primarily based on categorization research, with good and bad sentiments [5]. They have undertaken the test with 3 exceptional machine-learning algorithms, including, NB, SVM, and ME. The classification method is commenced with the usage of n-gram approach and aggregation of its techniques if necessary. They implemented bag-of-words functions framework to enforce each algorithm. According to their evaluation, NB indicates terrible outcome and SVM yields more favorable outcome. Salvetti et.al.[6], presented the "Overall Opinion Polarity" (OvOp) idea that emphasizes the employing of machine-learning algorithms including NB and Markov model for distinguishing data. As inferred from his article, the hypernym furnished with the aid of wordnet and Part Of Speech (POS) tag acts as lexical clear out filter for sorting data. Their test indicates that the final outcome achieved with the aid of WordNet filter is less correct in contrast with that of POS filter. In the sector of OvOp, accuracy is given extra significance in contrast with that of sensitivity. In their article, the authors provided a device in which they rank critiques primarily by taking likelihood into account. As per their observations, their method indicates better performance in the presence of internet information. Beineke et.al.[7] , have implemented

NB for determining sentiment type. They have extracted pair of derived functions which become linearly combinable to estimate the sentiment. For enhancing the accuracy outcome, they have delivered extra derived functions to the model and used categorized information to estimate relative influence. They have used Turneys method as a reference which efficaciously generates a brand new corpus of label report from the prevailing report [8]. Mullen and Collier have implemented SVM set of rules for sentiment evaluation in which values are assigned to few designated phrases after which, they are mixed to form model for grouping [9]. Along with this, an exceptional set of functions having proximity to the subject are assigned with the favorable values which assist in determining class of data. The authors have provided a contrast in their arrangement with information, having subject matter annotation and hand annotation. It was proven bring about higher contrast with that of subject matter annotation while the consequences require improvement, at the same time as evaluating with hand annotated information.

Dave et. al. have used a device for synthesizing critiques, then shifted them and ultimately grouped them with the usage of aggregation sites [10]. These based critiques are used as training data out and testing data. From those critiques, important features are recognized and ultimately scoring techniques are used to decide whether or not the critiques are good or bad. They have used a classifier to categorize the sentences retrieved from internet-searches by using product as the search condition. Matsumoto et.al. , have used the syntactic correlation amongst phrases as a foundation of document level sentiment evaluation [11]. In their article, common phrase subsequence and dependency sub-trees are extracted from sentences, which act as functions for SVM. They extract necessary features from each input in the dataset. They used unique datasets for undertaking the task. In case of first dataset (IMDb), the training and testing data are provided one at a time however in the other dataset (Polarity) 10-fold cross validation approach is used. In his paper, Zhang et.al. advocated using word2vec and SVM to correctly group feedback comments [12] . Their method is primarily based on two parts. In first part, they implemented word2vec device to cluster comparable features in order to gain the semantic functions in chosen domain. Next, the lexicon-based and POS-based totally characteristic selection method is followed to obtain the training information. Word2vec device adopts Continuous Bag-of-Words (CBOW) version and non-stop skip-gram version to analyze the vector illustration

of phrases [13]. SVM is implemented for multi-variate overall performance measures, which is then used for binary classification by following backup method of employing structural components of SVM optimization [14]. Luo et.Al. , identified a method to transform the textual content information into low measurement emotional space (ESM) [15]. They have annotated small length phrases that have precise and clean which means. They have extensively utilized Ekman Paul's studies to categorise the phrases into different classes ranging from anger to surprise or fear, etc. [16]. By using emotional tags, they were able to assign different weights to phrases using exceptional strategies. The messages are categorized based on the overall weight of all the tags. This process managed to yield a fairly good outcome for inventory message board,

Schukla provided a device which judges the standard of textual content that is primarily based on annotations on research papers [17]. Its technique collects sentiments of annotations in two strategies. It counts all of the annotation produced in the documents and calculates overall sentiment scores. Correlation among annotations however, is complex. This method requires large amount of metadata. Kasper &Vela proposed a "Web Based Opinion Mining device" for inn critiques [18].

(Zhang, et-al) in [19] performed an investigation on cell phone surveys. This approach proved helpful in comparing accuracy. It is beneficial in a judgment of the product grade and standing in the society [19]. They used three machine-learning algorithms (Naïve Bayes, KNN, and Random Forest) to calculate the opinion accuracy with the RF method showing an acceptable performance. There are a few approaches in reading sentiments and opinions. (Godbole, et-al) analyzed information sentiments and blogs [20]. It splits previous data in the context of their precise venture into classes. First class regards the strategies for routinely growing sentiment lexicon and the second pertains to structures that examine sentiment for complete files.

Further, Esuli & Srinivasiah's studies splits associated task into different types [21]: the primary one works with detecting the phrase orientation and the alternative works with detecting the phrase subjectivity. These divisions mostly refer to investigation on time period/phrase degree type, and now no longer report-degree type. This was done for making the task of determining sentiment label for data easier. It focused on labeling the

opinion polarity (good, bad, or impartial) for textual critiques data and then proceeded to perform comparison of the sentiment rating. The post is then split into single sentences ("sentence- based") and words ("words-based") or very brief texts from one source.

The preceding studies with Hearst on opinion-based categorization of the given files has implicated both the usage of models stimulated frequently with the aid of cognitive linguistics [22] or the guide or semi-guide production of discriminant-phrase lexicons that Das & Chen [23] proposed in their work. Turney [24] came up with a brand new approach for sentiment extraction in actual time in the field of economics; that is operating primarily based on messages that were extracted from online message boards, try and routinely label every such message as a "buy", "sell" or "impartial" recommendation. It resulted in chosen algorithm having 62% accuracy. Their approach is however, more time-consuming and requires more memory allocation as it focuses a great deal of manual selection and phrase tagging present in numerous thousand messages.

Understanding the relations between words with similar or different meanings has also become a topic of interest for researchers. Subjectivity detection studies usually assume the data to be opinionated or express emotion in some manner. They determine the sentiment class of text based on assessed polarity value. It is imperative that we carefully evaluate whether the given report consists of subjective data or not. It is also necessary that we selectively identify and make note of parts of the report that are subjective. The previous research done by (Etzioni, et-al) in [25] on this topic inspected the influences of adjective orientation and reliability of sentence subjectivity. From this operation, we could understand if a given sentence is subjective or not based on adjectives included in the sentence.

A subjective sentence can express opinions, viewpoints, thoughts, outlooks, perspectives, etc. With sentence-level subjectivity, every sentence in a survey or report is checked for subjective content. This sentence can be categorised into good or bad semantic orientation. Pang and Lee in [26] use a subjectivity detector to discard unbiased sentences from a given report. Then, with the usage of minimum cuts design, they combine inter-sentence degree contextual data with conventional bag-of-words functions. Further enhancements using baseline phrase vector classifier were documented and consequently

reported to their superiors. They proposed a recursive neural model that has following features in common: word vector representations and categorization. Online sentiments were analyzed and compared using a semantic model. Their approach can be implemented on data based on many topics as long as there is more training and assessment resources.

Researchers used machine-learning strategies to analyze labelled information that was extracted from the Internet and detect sentiment polarity in them. As humans are mostly biased and rely on their own intuition rather than always following common view point, they are not adept in selecting the appropriate discriminating words. They proceeded to experiment on the data by using the research previously done by Brody & Elhadad [27] as a reference.

Studies in opinion mining provides people with newer and refined means of improving their knowledge on this field. Despite these improvements there are still few hurdles challenges to take on, especially in the case textual content evaluation of critiques/files and analysis of sentiment scores.

2.2 The Need for Lexicon-Driven Methodology

Wiebe, et-al percieved the emotions of the collected tweets in his paper by making use of the MPQA lexicon vocabulary [28]. He managed to categorize tweets and count them to check if the data includes more positive or negative polarities of words based on the sentiment lexicon. Despite the fact that this methodology is straightforward, they found a significant comparison between the total sentiment in tweets and opinion surveys in Gallups work.

Cui, et-al constructed SentiStrength in [29] as a vocabulary-based calculation which indicates a sentiment polarity (positive/negative) and indistinguishable quality incentive somewhere in the range of 1 and 5 to a given book. Moreover, the paper provided a list that includes 298 positive and 465 negative terms, each with their own extremity and qualities. SentiStrength is used for making a choice operation by making use of different lists and words.

To handle emphatic lengthening, the authors present a three-step method for reducing words of the quality form. They made a comparison between various classifiers on particular microblog site comments by SentiStrength. This approach only worked better when classifying text with negative sentiments.

A rule-based technique for performing sentiment analysis on Twitter data was put forward by Kumar & Sebstian [30]. They assessed a sentiment score for every substance relying upon its literary vicinity to words from a supposition lexicon. The algorithm differentiates between unique sentences. It can also understand and handle similar sentences, negations, and text that include but clauses. To improve performance of these approaches, the researchers recognize that extra tweets get posted regularly with varying lengths and opinions expressed. It is imperative that they train a SVM classifier to accurately label data in the entities based on calculated score.

Nielson [31] identified the challenges of opinion examination on social media messages. Its motivation is to develop semantic investigation consisting of named element extraction and event acknowledgment. It works for determining an assumption based on polarity score for a given tweet. The SentiWordNet was implemented by Kamps, et-al in [32] to incorporate four Part-of-Speech (POS) tags namely adjectives, adverbs, verbs, and nouns having 2 million words with a small part of them being adjectives. The sentiment polarity classification classifies each word in one from three scores positive, negative, and neutral with a range of <0 to maximum of +1.

Kim & Hovy [33] implemented a four step approach for analyzing sentiment and opinions. The first step was recognizing the opinion and understanding it. They then used a machine-learning algorithm to classify it. Next, source word is analyzed and synonyms of words with the similar sentiment are collected. To avoid synonyms, they investigated the similarity of a word to every class of sentiment.

Rao & Ravichandran [34] attempted the challenge of calculating sentiment polarity of words. Data is classified as either positive or negative. Data is denoted with semi-supervised labels during polarity detection in data.

There are challenges that need to be overcome for the sentiment evaluation of such as spam & fake detection, Implicit & Explicit Negation, etc. Saifee & Jay describe related work on these challenges in their work [35]:

1) **Spam and Fake Detection:** The Internet usually consists of authentic as well as spam contents. This content is unrequired and should be removed before processing. This is one of the major issues people face when viewing surveys or reviews. User bias leads to a lot of false comments posted. There are three levels of the challenge spam:

- **The duplicate reviews:** These reviews are usually fake. Online surveys can sometimes consist reviews that are carbon copies of previously posted review by the same reviewer. This can lead to untrustworthy assessment of reviews due to the presence of these fakes. For example, people with different ids regularly post duplicate or similar reviews on the same product or different products.
- **The empty reviews:** This problem can be resolved by counting and assessing the quantity factor for all investigated items.
- **Content of certain reviews have polarity but have no connection the topic:** This is an issue that is prevalent in Emails. Emails have additional features that are capable of detecting and spam. This becomes a challenge in sentiment review issues. This can be resolved by analysing each post for spam and designation it with a probability value as shown by (Radulescu , et-al) in [36].

Researchers a collection of tweets for to categorize data based on the presence of spam. They were able to attain a viable outcome based on features of data. For instance Review 1: "The event was fun" & Review 2: "Soap operas are way too overdramatic". The word [fun] indicates a positive sentiment for review 1 while the word [overdramatic] which suggests negative polarity for review 2.

2) **Implicit and Explicit Negation:** Another important factor to consider in this domain is the use of negative words or negation. They can be expressed explicitly and implicitly.

- **Explicit negation** is easy to detect and report. For example: "I do [not like+] – this movie". The presence of the word "not" in this example suggests negative sentiment.
- **Implicit negation**: These are comparatively difficult to detect due to lack of negative word indication. For example: "I [hope to [improve] +] - your research". In spite of the presence of the word "improve" indicating positivity, the "hope" word it refers to gives us only a vague idea and requires more clarification, thus the polarity for this example is only declared as negative based on implicitly negative.

Some sentences can also have bi-polar values. They are mostly classified as positive despite containing negative terms like "Nobody", etc." This problem is somewhat resolved by keeping track of different combinations of words like "not only" and not reversing polarity of statement simply by encountering negative terms like "not".

3) Domain-independence: The main problem faced in this exercise. It can have unique consequences to the overall performance and outcome of sentiment analysis. It requires determining essential features of a particular topic. Based on the features selected the output can either be desirable or disastrous based on the domain it is collected on.

- **Topic domain:** Adsod & Chopde [37] implemented the Dependency-Sentiment-LDA, to investigate the sentiment dependency on joint sentiment and topic analysis. They studied the dependencies among sentiments using their approach. They analyzed the sentiment reviews for a cancer network in America. This helped them determine the class of a post after evaluating the polarity of emotion shown in the data. The reviews are then classified using binary classification.
- **Multi-topic domain:** Classifier has to be trained using multiple domains or source domains to determine the features that are related to the target domains. Classifier should also be employed on a site isn't included in the source domains. According to researchers, this problem can be resolved by using a system that learns to use the data that determines how the features of both target and source domain are associated.

- **For example:** *Review 1:* "*This is bad*" *& Review 2: "The quality of this translation is horrible"*

Observation: Review 1 consists of word "bad", without specifying precisely *what is bad*. This can refer to any subject or topic. However, the presence of word "bad" which indicates negative polarity for the review. Review 2 at least refers to one subject, i.e., "translation" with a feature of this domain "quality" so "horrible" indicates negative polarity with a particular topic.

The proposed model for Balahur & Turchi [38] is akin to that of a NB model of a word that investigates the document polarity by taking into account all the important features of the document. This model can extract words with polarity based on the domain it belongs to. This allows it to create unique domain-dependent word polarity dictionaries for every domain. This results in better performance rate when applying model in target domains.

4) World knowledge: More often world knowledge has to be incorporated within the system for detecting sentiments. Consider the subsequent examples: "He may be a Frankenstein". The primary sentence depicts a negative sentiment. But one needs to understand the term "Frankenstein" to determine the sentiment. This phase requires construction of word lexicons that can categorize the statement into respective class. The values for the words are predetermined. It can be created by starting with some seed words and so using some linguistic heuristics to add more words to them.

- **For example:** "*The author of this paper writes like an Einstein*"

Observation: It is necessary to have at least a basic idea or knowledge of world figures or information to understand the previous review. Defining someone as "Einstein" can be a positive sentiment. However, this concept is difficult to grasp by merely using machine-learning algorithms.

5) Construct Sentiment Lexicon: This phase is required for extracting features and keywords of a subject or multi-topic domain.

- **Extracting features and keywords:** This stage focuses on extraction of keywords and relevant features of every topic. This problem can be resolved by using surveys to classify all domain features and keywords.
- **Entity identification & extracting features and keywords:** The data can have multiple entities. This stage requires searching out the entity towards which the particular opinion is directed. Consider the subsequent example: "Kane is a better player than Ross." In this example Kane is portrayed in a positive light while Ross is portrayed a negative light.
- **For example: "**This mobile has excellent graphics."

  Observation: The term "graphics" is one feature from the mobile domain.

- **Grouping synonyms:** Words with similar meanings are constantly posted in reviews or posts in social media. This is not a new development. Identifying and grouping of such words can give us better accuracy results. They are not easy to identify however, as people often use different words to explain similar features. For example, the terms "voice" and "sound" both specify the same characteristic in phone review.

6) Natural language processing overheads: Natural language processing overheads can provide several challenges to overcome when performing sentiment analysis. The following example gives us an idea of co-reference: "I want that one!" only implies that the user wants something, rather than a specifying the target of *what* he wants. It is difficult to assess the sentiment of such sentences.

Emotions and opinions are often expressed in an explicitly or implicitly manner. Computers have an easier task in recognizing explicit emotions comparatively as it is difficult for people to spot unexpressed or indirect emotions correctly. This also applies to humour, sarcasm, irony, etc. Another problem is **Inference**, the method of concluding on the basis of knowledge, evidence, hypothesis, etc.

Bipolar sentiments: These are words in our input text that have bipolar values that prevent from properly understanding meaning for given input to classify. An approach was presented in which, bipolar words are separately kept when a word switches polarity. They

are then analyzed from the polarity lexica. This approach was based on human understanding of polarity and was shown to give satisfying result for sentiment detection algorithms.

2.3 Challenges Faced When Applying Machine Learning in Sentiment Analysis

When machine learning algorithms to analyze sentiment of the tweets for the respective training set, the final output depends on the algorithm implemented and also the length and quality of the data used. Some of the challenges faced during this project are:

- As tweets are usually small or varying in size with some only being expressed in vague or general terms, especially those reviews that can be fake or spam, our approach may encounter a lot of problems while categorizing these reviews.
- People often turn to social media to cope with their personal feelings and vent their emotions by posting about their status or opinions on Twitter, Thus, they often have a tendency repeat or exaggerate their opinions like “Amaziiing”. Words like these are still analyzed despite them not having a proper meaning.
- One of the major issues of this project is that the accuracy of our algorithms would constantly change if we were to run the tweet collecting function as tweets get posted in real time by different users at every second in varying lengths. Using automated sentiment analysis could prove to be optimal as long as we have the necessary resources.

CHAPTER 3

MATERIALS AND METHODOLOGY

This chapter deals with various materials used and methodologies adopted for this project of analysis of Twitter Data and the algorithms that will be used for this process.

3.1 System Requirements

3.1.1 Software Requirements

- Operating System: Windows 7,Windows XP, Windows Vista, Windows 8 or higher versions of Windows
- Programming Language: Python

3.1.2 Hardware Requirements

- RAM: 1 GB RAM and more.
- Processor: Any Intel Processor.
- Hard Disk: 6 GB and more.
- Speed: 1 GHZ and more

3.2 Library Installation

The Python programming language libraries imported and used for this project are the following:

1. NumPy library
2. Pandas library
3. Sklearn library
4. Matplotlib library
5. NLTK Library
6. Tweepy Library
7. Textblob Library
8. RegEx Library

3.2.1 NumPy Library

NumPy is a python library which is used for array work. Numpy stands for Numeric Python. It also has functions in linear algebra, Fourier transform, and matrix domain. It is an open source facility, and can be used for free.

3.2.2 Pandas Library

Pandas is a high-level data manipulation tool used for creating and manipulating DataFrames. DataFrames allows us to store and manipulate data in the form of tables with rows and columns. It allows us to perform data processing and read different types of files, ie, CSV files. We make use of this library to create DataFrames to contain all the extracted tweets and consequently concatenate them once data collection is done and write them into a CSV file to use as our data for this operation. We also use it to add or remove columns to the dataset as well as group the data based on their categories.

3.2.3 Sklearn Library

Scikit-learn is the de facto Python machine-learning library that allows us to perform data analysis using machine-learning algorithms.

We use it to divide the testing and training set for our data. We also use it to import all the algorithms that we wish to implement for this project. We also import all the performance parameters that are necessary for evaluating each model in this project.

3.2.4 Matplotlib Library

Matplotlib is the python library that is mainly used for plotting. We use it in this project to plot bar charts, scatterplots, confusion matrix as well as a classification report after analysis of data and implementing of algorithms on the data.

3.2.5 NLTK Library

NLTK stands for Natural Language Toolkit. It provides us with interfaces and lexical resources, etc. for classification, tokenization, etc. This library to import stop words for the data pre-processing phase to remove unnecessary data and save space and time.

3.2.6 Tweepy Library

Tweepy library that allows us to access Twitter platform in Python language and provides us with functions that we can use to extract live tweets.

3.2.7 TextBlob Library

TextBlob is a Python library that we use for processing textual data. It allows us to perform natural language processing (NLP) tasks like sentiment analysis, tokenization of data, etc.

We make use of its features to calculate the polarity and subjectivity of all the tweets in dataset.

3.2.8 RegEx (re) Library

They are accessed using the re module. They are used for handling and matching regular expressions content.

This library is used in this project to perform data cleaning by removing unwanted, dirty or unrequired data, such as @, hash tags, retweets and hyperlinks to simplify each tweet for classification.

3.3 Methodology

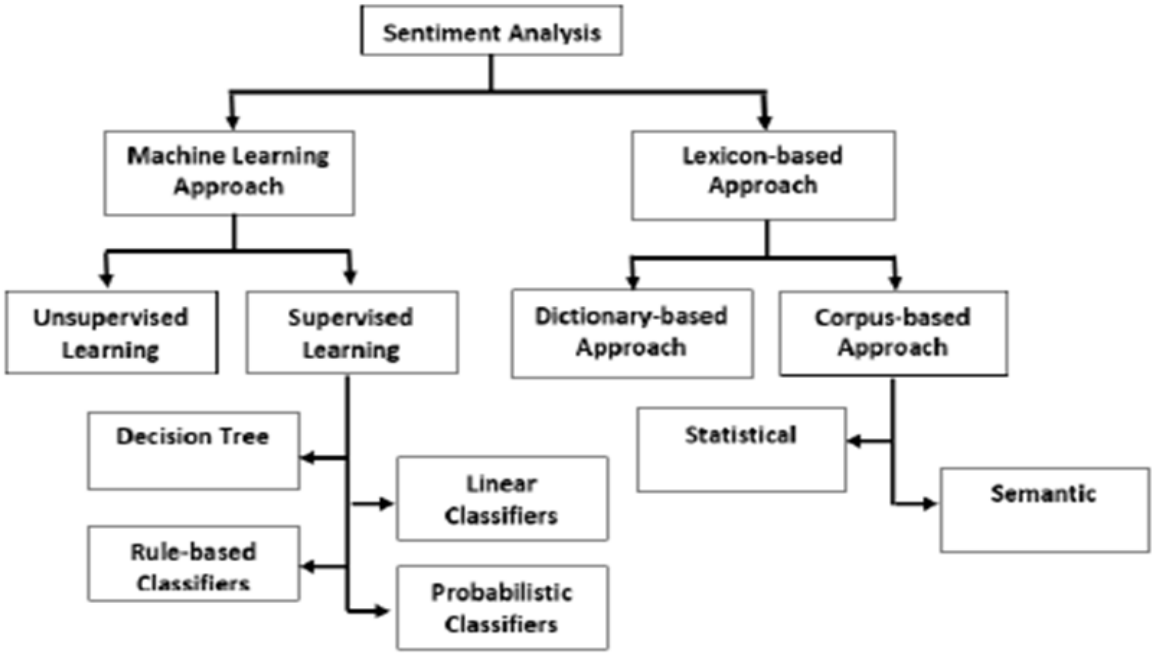

Figure 3.1: Sentiment Analysis

3.3.1 Preparing the Test Set

Sentiment Analysis relies mainly on text processing. The testing and training data will only contain text.

The first phase is to prepare the test set to extract tweets from csv file. This can be done by following the steps below:

1. The first step is to get personal credentials after registering on Twitter App.
2. Using the credentials that we have stored in a safe file, we then proceed to authenticate our code using Tweepy to gain access to Twitter.
3. Download tweets by implementing function that makes use of a search using the Cursor method.

Generating our authentication credentials is essential for us to progress to the next phase of our project. After this step is finished, we proceed to specify a search term for downloading our tweets. These tweets act as our training and testing data.

Stage 1: Getting the authentication credentials

1. Firstly, it is essential that we have a Twitter Developer account.
2. Once the account is created, the next step will be to create an App before proceeding.
3. Next, select the "I am requesting access for my very own personal use" option.
4. After step 3 is completed, we enter your account details before continuing.
5. Once our choice is made, we then fill the rest of the details. Decide on what sort of project will this application be used for.
6. Next, submit the application after accepting the terms and conditions.
7. Next, verify your Twitter Developer account through the link generated in your email.
8. We then wait for approval of your application from the administrator.
9. Click on the login link in the approval email to begin the process of creating our app.
10. After clicking on "Create an app", fill in all the app details. Click on "Create".
11. In the "Keys and tokens" tab, make a note of all the necessary keys and copy into a secure file, as they're necessary for authorizing our Twitter access.
12. Next, click "Create" to get the required credentials. Then, copy the tokens into a secure file. The credential acquisition part is now complete.

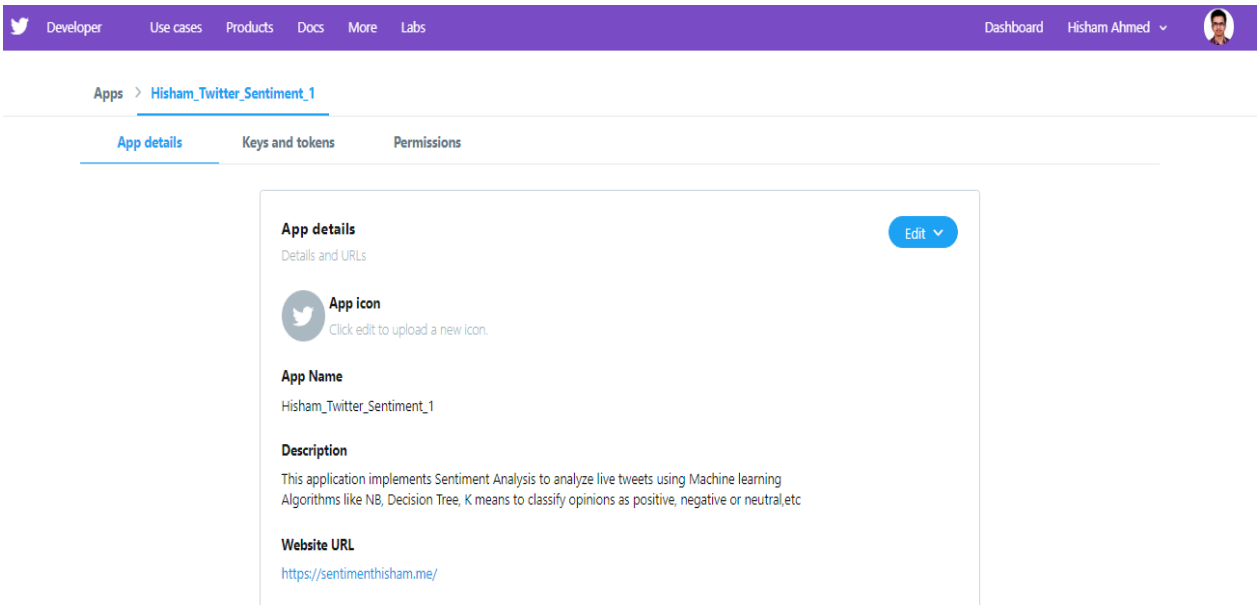

Figure 3.2: My App Details for Twitter Developer Account

Stage 2: Code Authentication

Using our login credentials, we authenticate our code to gain access to the Twitter data. To do this, we import the Twitter library, then refer the credentials from the secure file to create a Twitter API object. To do this we upload the file containing all the required keys, then enter the credentials as variables.

Next, we authenticate to Twitter by using an authentication object created by setting the required access tokens.

3.3.2 Collection of Data

The next stage is collecting the data. This steps performed in this phase is shown in Figure 3.3 as a Block Diagram

We start by searching Twitter for recent tweets. In this case we extract the recent tweets that use #[insert company name here] hashtag. Next, we create an object to collect the tweets containing the hashtag #[company name]. For this, the .Cursor method is very useful.

To implement this query, we need to specify the following attributes:

1. A *search term* that refers to the topic on which we wish extract tweets - in this case #[insert company name here].

2. The *start date* from which we wish to collect tweets. It can only access tweets from a few weeks prior rather than much older time period

By using Cursor() method, we are able to access twitter and find tweets referring to the specified search term. We can control the number of tweets extracted by specifying a numerical parameter in the items() function. Ex: Specifying items(200) will return 200 recent tweets of that particular topic.

Cursor() returns an object that consists of attributes for each tweet like the content, id of the tweeter and date. It allows us to loop through all the tweets that were collected.

After collecting 1000 tweets referring to a particular company from a single search, we extract it to a Data Frame. This process is done three times in this project with three different search words, thus totaling 3000 tweets.

The next step is to create a Data Frame that'll contain the tweets that were extracted, and then show the first five rows. As we have used three different search words, we can use three different Data Frames to extract the respective tweets as well as creating a column specifying which organization or company the tweet is related to. After this, the individual Data Frames are combined into a single Data Frame to make analysis easier and less time consuming.

The tweets' subjectivity and polarity is then calculated using functions. For assessing subjectivity of the tweet, getting 0 score implies that the sentence is a fact, and getting +1 score implies that the tweet is conveying an opinion or expressing emotion. When assessing polarity score of the tweet, getting score <0 means that the tweet is considered as negative, and a score of >0 means that the tweet can be considered as positive.

The next step is to extract the resulting Data Frame into a CSV file. The first stage of this project is complete. Next, we move on to the processing of data phase.

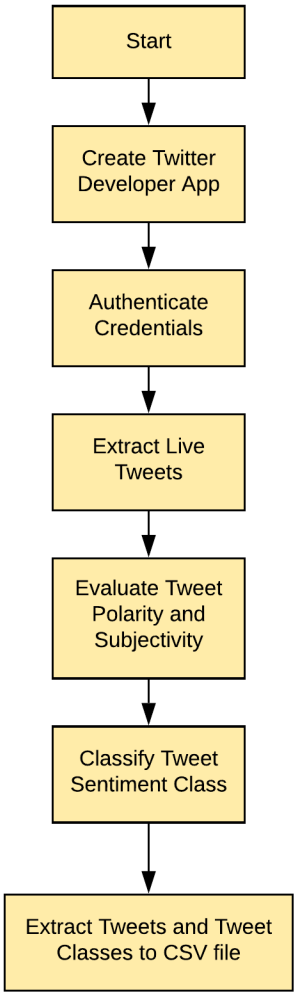

Figure 3.3: Block Diagram for Data Collection Phase

3.3.3 Pre-processing of Data

This is an essential phase in Machine Learning because the nature of information and valuable data that can be acquired from it can influence model's capacity to learn. It is imperative that we prepare our information before incorporating it into the model.

The primary objective of this research is to predict and classify tweets into positive, negative, or neutral based on the word features extracted from the dataset. These features are then used in the processing and for training the machine learning model. The techniques to identify the features and categories for each class of tweets is shown in Figure 3.4.

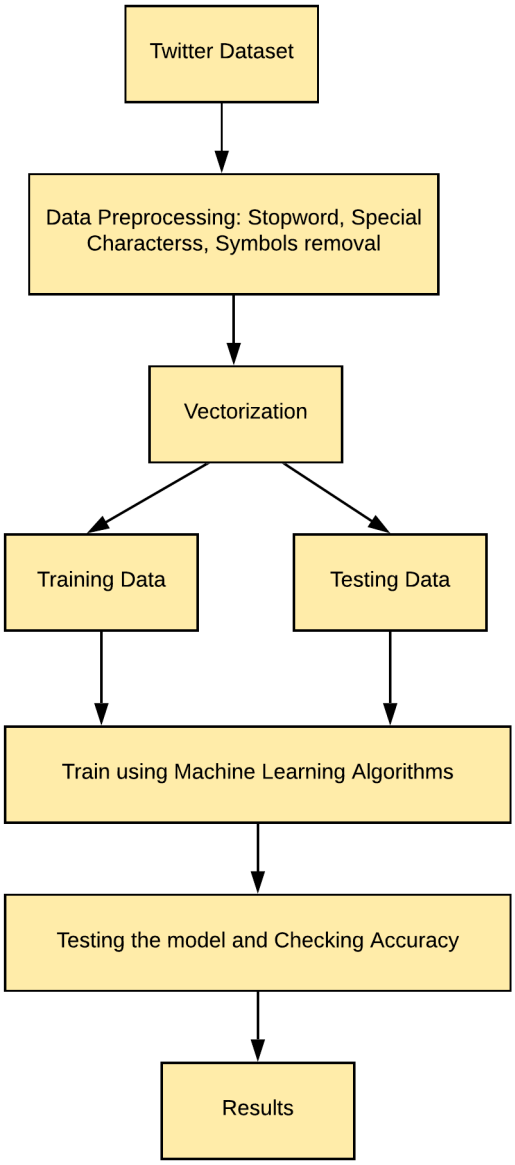

Figure 3.4: Block Diagram for Methodology

For this phase, will be using the Training set from the respective CSV file which we extracted all our data to.

As machines cannot read images or categorical data, and only read binary data, it would for our machine learning model to get trained by processed data.

We can create a new Data Frame to retrieve the data. The resulting Data Frame contains all the 3000 tweets that were extracted.

Here is a sample of the data that we selected for this project.

	Tweets	Company	Subjectivity	Polarity	Sentiment_Analysis
0	How/when will this end? \nDo Patriots gather...	Amazon	0.0	0.0	Neutral
1	This book reminds me of The Fault in our Stars...	Amazon	0.0	0.0	Neutral
2	'In this day and age, you have to be what isn...	Amazon	0.0	0.0	Neutral
3	Amazon usa deals\nRazer Gaming Laptop \n $2,12...	Amazon	0.0	0.0	Neutral
4	LETTERS TO THE CHIEF - An enchanting and unfor...	Amazon	1.0	0.8	Positive

Figure 3.5: Sample of Data Set

We start off by importing the required libraries, such as the re module, which is for managing regular expressions in the text, as well as parsing. The ntlk library is used to take care of any text processing, form changing, component extraction and stop word removal. We have also imported all the required algorithms from the sklearn module.

The first step is to check if there are any empty values in Data set. We then follow it by counting the number of tweets based on Company.

The next step is to plot the subjectivity and polarity as a scatter plot, followed by visually showing the number of tweets based on their classification as positive, negative or neutral as well as plotting another one that categorizes them based on company.

We then distinguish the tweets into two categories by using lambda function to label tweets with negative sentiment with value 0 and labelling tweets with non-negative (positive or neutral) sentiment with value 1.

Since the data extracted contains @ symbols, hyperlinks, RTs, and hash tags, it needs to be processed and cleaned. This can be done employing a function to remove those symbols form the data. First, the text is converted to lower-case format. This is done to

ensure proper interpretation of words by Python. Tweets also consist of links, addresses and usernames which also need to be cleaned. Hash tags are also removed to ensure reliable data processing. Next, the duplicate characters are removed to ensure that all important words are processed irrespective of spelling (e.g. “yesssss” becomes “yes”). Finally, the data is tokenized to ensure better processing for the proceeding phases.

While we removed the duplicate characters in words, we did not remove duplicate words from the text, as these duplicate words can also help in evaluating polarity of the text.

To use the Twitter data for predictive modeling, the text must be tokenized. Certain words are then removed which are then encoded as integers, to provide as inputs for the selected machine-learning algorithms. This process is called feature extraction.

We use CountVectorizer from Scikit-learn library to convert text intto a vector of terms or tokens with numerical values specifying their frequency.

The data processing phase is complete. The data obtained is referred to by using a variable, and then subsequently used for training and testing the model.

3.3.4 Model Preparation

This stage focuses on preparing the different machine learning models for classifying the twitter data. As we have specified the testing and training set after data is pre-processed, we need to choose and import the appropriate algorithms that will yield the best accuracy or result for this project.

For this project we have imported the Naïve Bayes Classifier, Random Forest Classifier, Support Vector Machine Classifier and K Nearest Neighbor Classifier from Scikit-learn. The steps we have followed before implementing each algorithm are:

1) Build a bag-of-words of all the words in data set. This vocabulary is later used for tweet comparison.

2) Use CountVectorizer to transform all tweets into numerical tokens or terms.

3) Build our feature vector.

4) Run Classifier using feature vector.

3.3.4.1 Naïve Bayes Classifier

The Bayesian classification model is a probabilistic classifier machine learning technique. We have used multinomial Naive Bayes for this project due to its simplicity and usefulness in categorizing text and detecting spam. It assumes each feature is conditional independent to other features given the class, that is:

$$P(c \mid t) = \frac{P(c)P(t \mid c)}{p(t)}$$

Where "c" is a specific class and "t" is the tweet we wish to classify. P(c) and P(t) are the prior probabilities of class and tweet. And P(t | c) is the probability the tweet appears given this class. In this case, the value of class c might be 0 (Negative) or 1 (Positive or Neutral) and t is a sentence.

The final outcome is to determine which value of "c" can maximize P(c | t):

Where P(wi | c) is the probability of the ith feature in tweet "t" appearing given classification of "c". It is imperative that we obtain and train parameters of P(c) and P(wi | c). They are calculated by determining Maximum Likelihood Estimation of each other.

When predicting class of new tweet "t", it is necessary to determine the log likelihood log P(c) + ΣilogP(wi|c) for different classes, and take the class with highest log likelihood as prediction.

The benefits of using NB algorithm for this project:

- Requires less training data to learn the parameters or features of data.

- Has faster run-time compared to other models.
- Frequently used for text classification.
- Its classification is somewhat accurate even in instances where its model is oversimplified.

The disadvantages of using NB algorithm for this project are:

- It assumes all the features or data are independent.
- It doesn't discriminate between attributes.
- When the probability determined for feature is zero.

3.3.4.2 Random Forest Classifier

Random forests are an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. It produces multi-altitude decision trees at inputting phase and output is generated in the form of multiple decision trees. The correlation between trees is reduced by randomly selecting trees and thus the prediction power increases and leads to increase in efficiency. The predictions are made by aggregating the predictions of various ensemble data sets.

It may be defined as the gathering of tree-dependent classifiers. It performs its function by splitting every node by using the first-rate node amongst randomly selected predictors at that node.

The original data is replaced with newly created data that is used for training. Random characteristic selection is used to grow new trees. These are not pruned.

The random forests algorithm (for both classification and regression) is as follows:

1) Randomly select N records from the input dataset.

2) Make a decision tree based on these N records.

3) Number of trees required for algorithm are specified and the above steps are repeated.

4) Each tree predicts new information through aggregating the predictions of the N trees

5) Finally, the new record is assigned to the prediction class with majority.

Advantages of using Random Forest for this project:

- This algorithm is unbiased as it relies on majority value.
- It has high stability.
- Works well for text classification and when data contains numerical features

. Disadvantages of using Random Forest for this project:

- Highly complex and requires more memory compared to other algorithms.
- More time-consuming.

3.3.4.3 Support Vector Machine Classifier (SVM)

SVM is a group of supervised machine-learning techniques mainly used for highlighting machine learning groups and recurrence. For this project it makes use of the reduced features obtained during processing phase. For the output, the classifier forecasts that the attributes compared to the labels as either negative or non-negative (positive or neutral). SVM is used as a non-probabilistic sampling classifier by that generates the hyper-plane separating the two groups.

By plotting any value in n-dimensional space or graph, support vector algorithm is performed. The total number of data features present here is “n”. The

value of each data is displayed as a different graph coordinate. SVM is an example of technique that can be implemented in problems of classification and regression. The grouping procedure for this isn't straightforward, and, when it depends on the partition utilizing diverse lines, it is known as hyper-plane classifier.

The optimum hyper plane is chosen according to line distance. SVM vector helps divide the class from one side to the other. The difference is called margin and margin refers to the vectors of support as shown in Figure 3.4. Mathematical functions called "Kernels" are used to draw hyper-planes. We use "linear" Kernel for this problem as classification is linear here with labels being negative and non-negative.

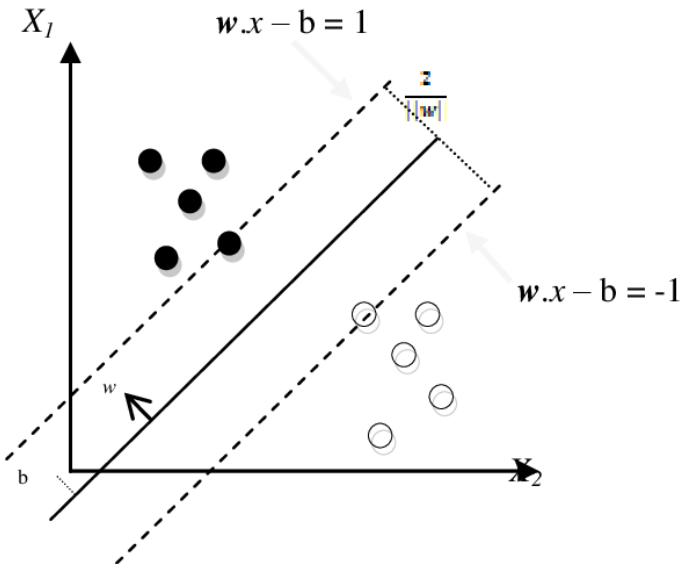

Figure 3.6: Optimal Separating Hyperplane between Two Classes

Advantages of using SVM for this project:

- Useful for text classification.
- Performs well with high dimensional data.

Disadvantages of using SVM for this project:

- It does not work well on large volumes of data.
- It is difficult to implement on data with too much noise or unrequired data

3.3.4.4 K-Nearest Neighbor Classifier

K-Nearest-Neighbors (KNN) is a simple non-parametric classification algorithm. It is widely used for sentiment analysis, machine-learning, text classification, etc.

KNN algorithm classifies data by comparing the unknown data point with the training data points that are in its close vicinity. We use Euclidean distance between the data points to calculate unique similarity. The characteristic values are then normalized. This is done to ensure a balance between longer range attributes and smaller range attributes. New or unlabeled data is assigned the most common or regular class amongst its similar neighbors. In the event there is a tie, the new pattern is assigned to class with least average distance.

Since the KNN classifier performs prediction by identifying the observations closest to sentiment class, a large emphasis is placed on variables scale. Large scale variables have a much larger effect on the final output for KNN than small scale variables. For this project we set the number of nearest neighbors as 3 or more to get an acceptable accuracy.

Advantages of using KNN Algorithm for this project are:

- Easy to implement
- Requires less run-time for execution.

Disadvantages of using KNN Algorithm for this project are:

- It has a high prediction cost for datasets with a large amount of data.

- Doesn't work well with data that requires classification features due to how challenging it is to assess distance between dimensions in this domain.

3.3.4.5 Building a Vocabulary of Words

A vocabulary in this instance is a list of all the words or text present in our data that model uses for Training data. It is a feature provided by NLP.

The words in our input or dataset are broken down into features that are then arranged in list format. These features are a list of distinct words, each with its own unique frequency key.

Next, the vocabulary is compared with the tweets. For this, we loop through all the words in our Training set and compare them all against the respective tweet. Based on this, we associate the following labels to each phrase:

1) Label 1 (true): Tweet contains the Vocabulary phrase
2) Label 0 (false): Tweet lacks or doesn't contain the Vocabulary phrase

The final step is to build our feature vector and then proceed on to training the data.

3.3.5 Training the Data

Next, we proceed to training the data with all the algorithms we have imported by running the classifier training code. The training features and testing features were already specified previously. As we have already imported NLTK, it is helpful in allowing us function calls to train the model as the classifiers are all built into the library.

After having imported all the required Classifier algorithms and performance evaluation parameters from the Scikit-learn, we proceed by implementing the algorithms on training and testing set.

After the algorithms have been implemented, calculate the accuracy of all algorithms and proceed to testing the model.

3.3.6 Testing the Model

The final stage focuses on testing the model by providing an input of our own and testing the accuracy of the different algorithms that were implemented for this project. We also plot the confusion matrix for all the classifiers that were used and plot bar graph displaying the difference in accuracy of all algorithms implemented.

3.4 Performance Evaluation Parameters

These are parameters that help us evaluate the performance of an algorithm. They are determined based on the elements of a confusion matrix.

3.4.1 Confusion Matrix

It is used to describe the performance of an algorithm or model on tested data. It is displayed in a tabular format with four unique combinations of the actual value and the predicted value

An example of a confusion matrix for a binary classifier is shown below.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Figure 3.7: Confusion Matrix Example

From the example, we can infer the following:

- The predicted classes are: "yes" and "no". Ex: If we were to predict a student passing his exam, "yes" would imply they passed, and "no" implies that they failed.
- Total student pass/fail predictions made by the classifier =165.
- No. of times the classifier predicted that the student passed = 110.
- No. of times the classifier predicted that the student failed = 55
- We can infer that from the given data that: There are 165 total students. Out of these, 105 students have passed the exam, while the remaining 60 students failed.

From the confusion matrix, terms such as “True Positive (TP)”, “False Positive (FP)”, “True Negative (TN)”, and “False Negative (FP)” are determined and are used to compare label of classes in this matrix.

- “True Positive (TP)”: No. of positive reviews that are correctly classified as positive.
- “False Positive (FP)”: No. of positive reviews that are falsely classified.
- “True Negative (TN)”: No. of negative reviews that are correctly classified.
- “False Negative (FN)”: No. of negative reviews that are falsely classified.

The above terms are used to calculate performance evaluation parameters such as Precision, Recall, F-Score, Accuracy, etc. These values allow us critically evaluate our algorithms as well as the prediction quality and performance.

3.4.2 Classification Report

A Classification report is used to measure the standard of predictions from a classification algorithm. It is used to check how many predictions are True and how many are False. It is used to calculate the total number of True and False predictions. Terms like “True Positives”, “False Positives”, “True Negatives” and “False Negatives” are used to predict the metrics of a classification report.

3.4.3 Precision

It is characterized as the proportion of number of models effectively named as positive to the total addition of number of emphatically arranged positive model.

$$\text{Precision} = \frac{TP}{(T\,P + F\,P)}$$

3.4.4 Recall

It quantifies the culmination of the classifier result. It is the proportion of complete number of positively labeled model to total number of positive models.

$$\text{Recall} = \frac{TP}{(T\,P + F\,N)}$$

3.4.5 F-Measure

It is the symphonious mean of precision and recall. It is required to streamline the framework towards either precision or recall, which ends up having more impact on conclusive outcome.

$$\text{F} - \text{Measure} = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$$

3.4.6 Accuracy

It is the most widely recognized measure of characterization process. It is determined as the proportion of correctly grouped model to total number of models.

$$\text{Accuracy} = \frac{T\,P + T\,N}{(T\,P + T\,N + F\,P + F\,N)}$$

3.5 Statistical Analysis

Data obtained from Sentiment Analysis of Twitter Data were statistically analyzed by visually plotting scatter plots to display polarity and subjectivity, and bar graphs, etc. to

display the number of tweets per sentiment, number of tweets per sentiment based on company as well as the difference in accuracy of each algorithm used.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Prologue

This project focuses on the implementation of various machine learning algorithms to perform sentiment analysis on real time Twitter data. This chapter deals with the results obtained after implementing the various algorithms.

The data used for this project were at least 3000 real time live tweets that were extracted using Tweepy API Cursor method. As the tweets were extracted real time, the accuracy of various algorithms may vary, as Twitter is updated with posts on a constant basis.

4.2 Classification of Tweets

There were 1000 tweets of each search word extracted three times overall totaling 3000. After calculating the subjectivity and polarity of data, the number of tweets per sentiment were.

1. Positive Tweets: 1207 (40.2 %)
2. Neutral Tweets: 1405 (46.9 %)
3. Negative Tweets: 388 (12.9 %)

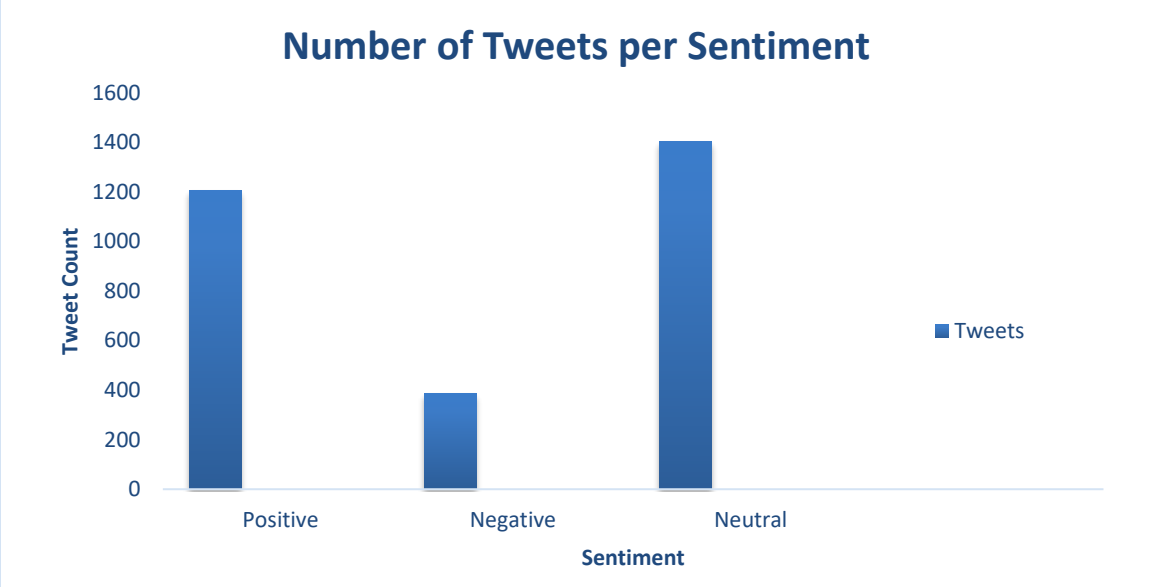

Figure 4.1: Bar Graph of Number of Tweets per Sentiment

4.3 Classification of Tweets based on Company

The next stage is to classify the tweets based on company to get a better understanding of which company had better reviews or sentiment. After performing this step, we got the following results.

1. Amazon: 128 negative, 434 neutral, and 438 positive tweets.
2. Microsoft: 151 negative, 482 neutral, and 367 positive tweets.
3. Samsung: 109 negative, 489 neutral, and 402 positive tweets.

From the above data, we can infer that Microsoft and Samsung had a lot more positive and neutral reviews as compared to Amazon. However, Microsoft had more negative tweets than Amazon and Samsung and Samsung had the least number of negative reviews. Based on this information, tweets posted on Amazon and Samsung have an overall better positive reception compared to tweets posted on Microsoft. It is visually displayed in bar graph in Figure 4.2

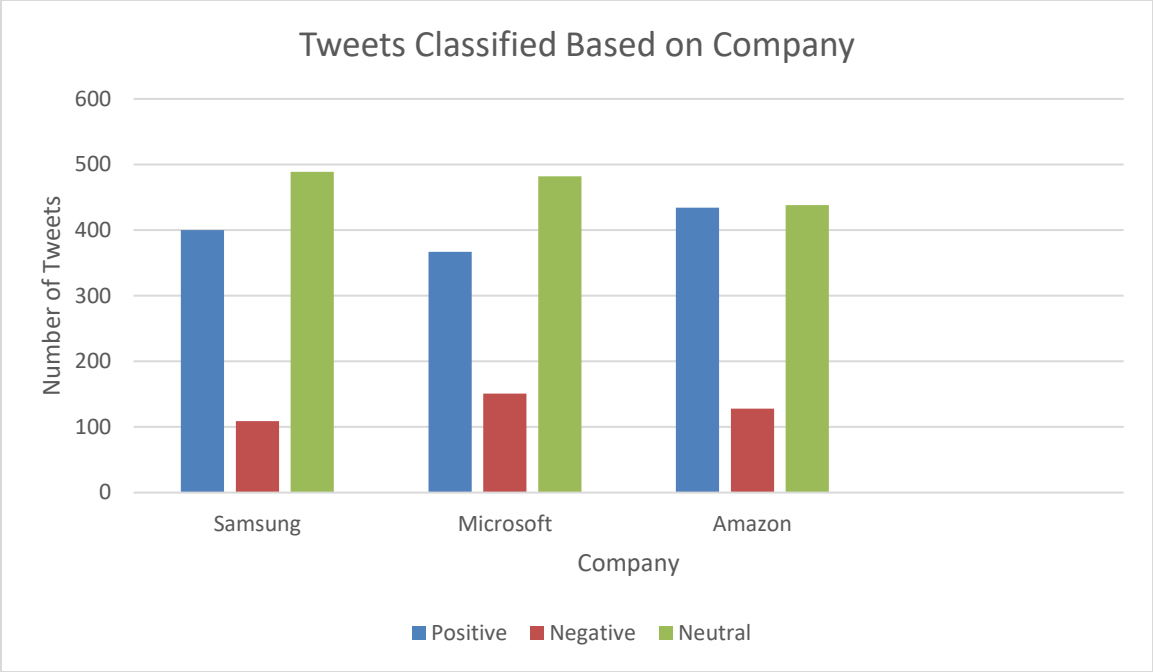

Figure 4.2: Bar Graph of Tweets classified Based on Company

4.4 Polarity and Subjectivity of Tweets

The tweets are assessed for calculating their subjectivity score and polarity score. These scores are then plotted in a scatter plot. If subjectivity of respective tweet is 0, then it can be considered as a fact, and if it is greater than 1, then it can be considered as an opinion. Polarity gives us an idea on what kind of sentiment is shown in the respective tweet. If polarity of text is > 0, the tweet is positive, 0 if neutral and negative if < 0.

As shown in the scatter plot in Fig. 4.3, majority of tweets are either positive or neutral.

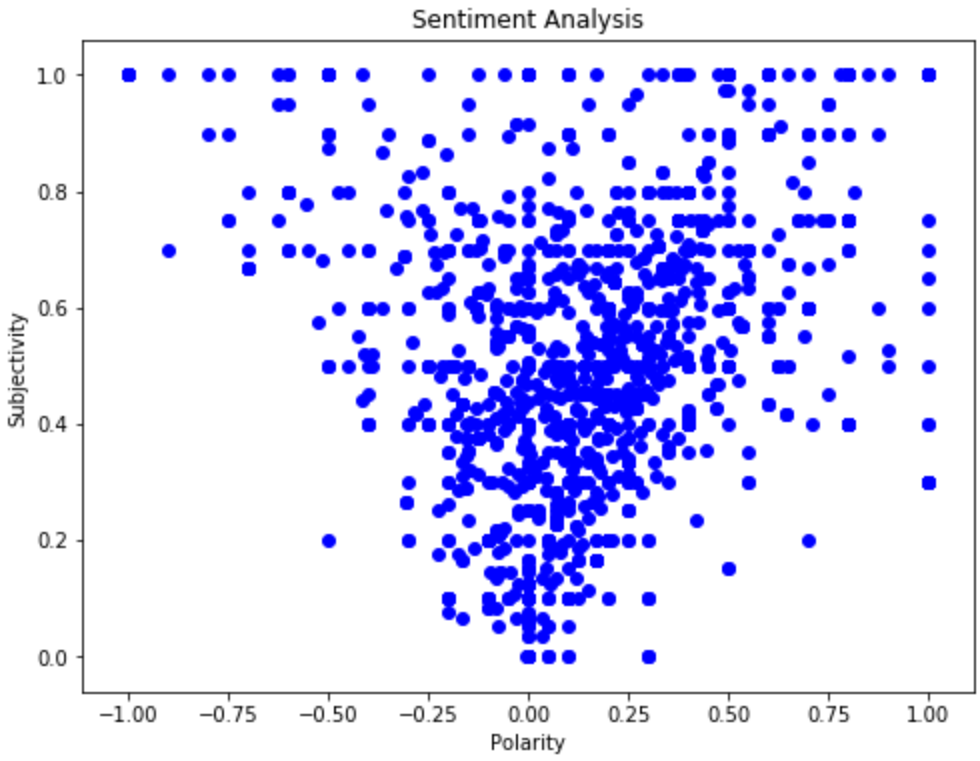

Figure 4.3: Scatter Plot of Subjectivity and Polarity of Tweets

4.5 Performance Evaluation

The next stage is to calculate the accuracy and plot the confusion matrix and classification report of all algorithms after implementing them on all 3000 tweets. We evaluate the performance parameters for two values 0 and 1, with 0 specifying negative sentiment and 1 for non-negative, i.e, positive or neutral sentiment. The model reports of all the algorithms are shown in figures 4.4 to 4.7. They are then summarized in a tabular format in Table 4.1.

```
Accuracy of KNeighborsClassifier is 86.83333333333333%

Confusion Matrix for KNeighborsClassifier model
[[  5  79]
 [  0 516]]

Classification Report for KNeighborsClassifier model
              precision    recall  f1-score   support

           0       1.00      0.06      0.11        84
           1       0.87      1.00      0.93       516

   micro avg       0.87      0.87      0.87       600
   macro avg       0.93      0.53      0.52       600
weighted avg       0.89      0.87      0.81       600
```

Figure 4.4: Model Report for KNN Model

Model 0 has a precision of 1.0 - In other words, when it predicts a tweet as negative, it is correct 100% of the time.

Model 1 has a precision of 0.87 - In other words, when it predicts a tweet as positive or neutral, it is correct 87% of the time.

Model 0 has a recall of 0.06 - In other words, it has low success rate when it identifies a tweet as negative ,i.e., it has many false negatives.

Model 1 has a recall of 1.0 - In other words, it correctly identifies 100% of all tweets that are positive or neutral, it is correct 100% of the time.

F1 Score is needed for achieving balance between Precision and Recall. KNN Model has 0.11 f1 score for model 0 (negative tweet) and 0.93 f1 score for model 1 (positive or neutral tweet).

In Summary, the KNN Model has low recall and F1 score when predicting negative tweets, but high recall and F1 score when predicting non-negative tweets.

```
Accuracy of SVC is 90.0%

Confusion Matrix for SVC model
[[ 36  48]
 [ 12 504]]

Classification Report for SVC model
              precision    recall  f1-score   support

           0       0.75      0.43      0.55        84
           1       0.91      0.98      0.94       516

   micro avg       0.90      0.90      0.90       600
   macro avg       0.83      0.70      0.74       600
weighted avg       0.89      0.90      0.89       600
```

Figure 4.5: Model Report for SVM Model

Model 0 has a precision of 0.75 - In other words, when it predicts a tweet as negative, it is correct 75% of the time.

Model 1 has a precision of 0.91 - In other words, when it predicts a tweet as positive or neutral, it is correct 91% of the time.

Model 0 has a recall of 0.43 - In other words, it has low success rate when it identifies a tweet as negative.

Model 1 has a recall of 0.98 - In other words, it correctly identifies 98% of all tweets that are positive or neutral, it is correct 98% of the time.

F1 Score: SVM Model has 0.55 f1 score for model 0 (negative tweet) and 0.94 f1 score for model 1 (positive or neutral tweet).

In Summary, the SVM Model using “linear” kernel has low recall and F1 score when predicting negative tweets, but high recall and F1 score when predicting non-negative tweets.

```
Accuracy of RandomForestClassifier is 90.33333333333333%

Confusion Matrix for RandomForestClassifier model
[[ 26  58]
 [  0 516]]

Classification Report for RandomForestClassifier model
              precision    recall  f1-score   support

           0       1.00      0.31      0.47        84
           1       0.90      1.00      0.95       516

   micro avg       0.90      0.90      0.90       600
   macro avg       0.95      0.65      0.71       600
weighted avg       0.91      0.90      0.88       600
```

Figure 4.6: Model Report for Random Forest Model

Model 0 has a precision of 1.0 - In other words, when it predicts a tweet as negative, it is correct 100% of the time.

Model 1 has a precision of 0.90 - In other words, when it predicts a tweet as positive or neutral, it is correct 90% of the time.

Model 0 has a recall of 0.31 - In other words, it has low success rate when it identifies a tweet as negative.

Model 1 has a recall of 1.0 - In other words, it correctly identifies 100% of all tweets that are positive or neutral, it is correct 100% of the time.

F1 Score: Random Forest Model has 0.47 f1 score for model 0 (negative tweet) and 0.95 f1 score for model 1 (positive or neutral tweet).

In Summary, the Random Forest Model has low recall and F1 score when predicting negative tweets, but high recall and F1 score when predicting non-negative tweets.

```
Accuracy of MultinomialNB is 87.5%

Confusion Matrix for MultinomialNB model
[[ 16  68]
 [  7 509]]

Classification Report for MultinomialNB model
              precision    recall  f1-score   support

           0       0.70      0.19      0.30        84
           1       0.88      0.99      0.93       516

   micro avg       0.88      0.88      0.88       600
   macro avg       0.79      0.59      0.62       600
weighted avg       0.86      0.88      0.84       600
```

Figure 4.7: Model Report for Multinomial NB Model

Model 0 has a precision of 0.70 - In other words, when it predicts a tweet as negative, it is correct 100% of the time.

Model 1 has a precision of 0.88 - In other words, when it predicts a tweet as positive or neutral, it is correct 87% of the time.

Model 0 has a recall of 0.19 - In other words, it has low success rate when it identifies a tweet as negative.

Model 1 has a recall of 0.99 - In other words, it correctly identifies 100% of all tweets that are positive or neutral, it is correct 99% of the time.

F1 Score: Multinomial NB Model has 0.30 f1 score for model 0 (negative tweet) and 0.93 f1 score for model 1 (positive or neutral tweet).

In Summary, the Multinomial Naïve Bayes Model has low recall and F1 score when predicting negative tweets, but high recall and F1 score when predicting non-negative tweets.

Each model has high precision but low recall for negative tweets, due to imbalance between number of negative and non-negative tweets. This could be resolved by extracting tweets from a much more balanced dataset.

Table 4.1: Comparison Table of Algorithm Performance

Method	No. of Tweets	Value	Precision	Recall	F-Score	Accuracy
Linear SVM	3000	0	0.75	0.43	0.55	90%
		1	0.91	0.98	0.94	
Multinomial Naïve Bayes	3000	0	0.70	0.19	0.30	87.5%
		1	0.88	0.99	0.93	
Random Forest Classifier (n=200)	3000	0	0.90	0.90	0.90	90.33%
		1	0.90	1	0.95	
K Nearest Neighbors (n=3)	3000	0	1	0.06	0.11	86.83%
		1	0.87	1	0.93	

From the details in the table above, it is shown that both Random Forest classifier algorithm and Linear SVM algorithm have greater than 90% accuracy while KNN algorithm and Multinomial Naïve Bayes have around 87% accuracy. This shows that Random Forest Algorithm is the most accurate and reliable of all of them.

Random Forest has the highest accuracy of all algorithms used due to its unbiased approach of predicting the sentiment class of tweet by relying on majority value of class. As this project focuses on text classification, it is the perfect algorithm to run for this project in spite of its runtime.

SVM also has high enough accuracy due to the class labels in this case being only negative and non-negative (positive or neutral), and also due to the fact the data was properly processed or cleaned before it was trained. It might not have as much accuracy if volume of data was substantially increased.

Although Naïve Bayes is widely implemented for text classification and has faster run-time, its independence assumption can cause problems in classification, especially in the case of tweets with negative words preceding adjectives. Despite its relatively high accuracy and simplicity, it might prove to be unreliable with lack of proper features or input data.

Although the KNN Algorithm is easy to run and has fast execution, it can struggle with data that requires classification features. Despite also showing acceptable performance and accuracy, the final accuracy of this algorithm would be severely reduced if the data and dimensions or features of input were to drastically increase.

Figure 4.8 visually displays the accuracy percentage of all algorithms in a bar graph.

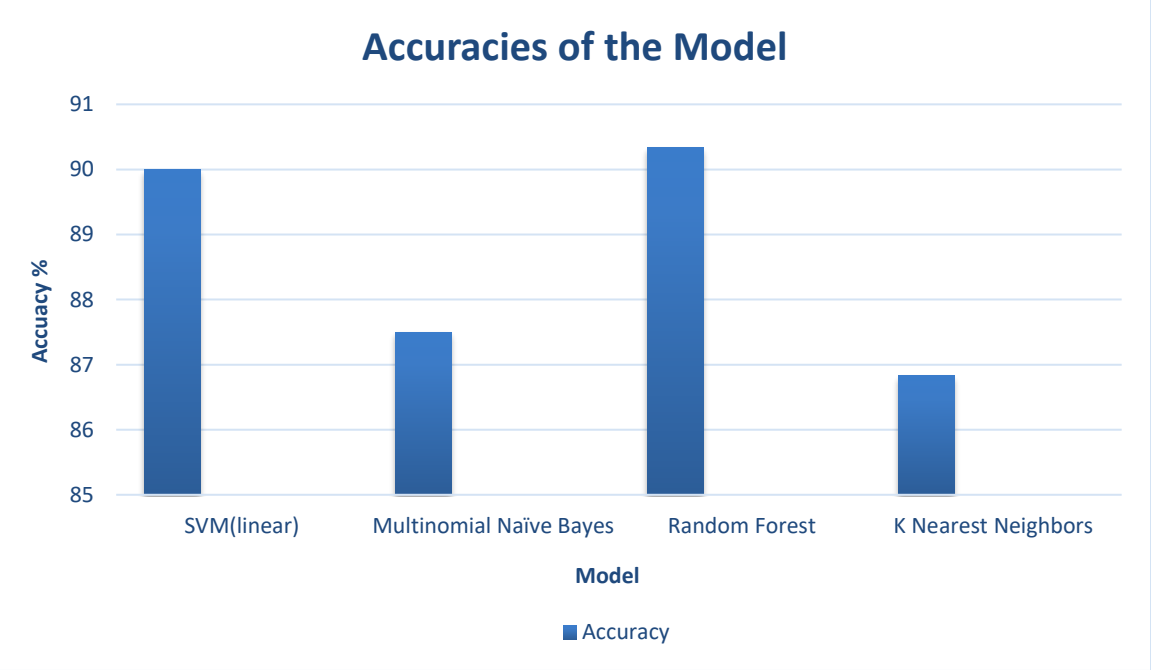

Figure 4.8: Bar Graph showing Accuracies of all Models

The accuracies of the models can vary and change regularly if we get more data or if we run the program again as Twitter data gets updated constantly. Accuracy of any algorithm can change drastically if program is rerun an hour or two later after initial execution.

CHAPTER 5

SUMMARY AND CONCLUSION

5.1 Summary

In this project, real time Twitter data or tweets were extracted based on a specific company using Tweepy Cursor method and different machine learning algorithms were used to accurately classify these Tweets as a positive tweet, a negative tweet, or a neutral tweet based on the polarity and subjectivity of the respective tweet. The goal was to develop a model with optimal performance when classifying Twitter messages using these techniques.

Three different search words were specified in Tweepy Cursor method and each search returned 1000 tweets each. The tweets were all added together and appended into a single Dataset along with the company they were classified as.

The data obtained was then cleaned to remove any unwanted data and its subjectivity and polarity was calculated, based on which they were categorized.

The data was then preprocessed and split into training set as well as a testing set using Vectorizer and stop words. We then implemented linear SVM, Multinomial Naïve Bayes and Random Forest and K Nearest Neighbors on the testing and training set.

Various results obtained during this investigation as given in Chapter 4 are presented here under in summarized form.

- Total Number of Positive Tweets= 1207 (40.2 %)
- Total Number of Neutral Tweets= 1405 (46.9 %)
- Total Number of Negative Tweets= 388 (12.9 %)
- Categorization of Tweets for Amazon: 128 negative, 434 neutral, 438 positive tweets.
- Categorization of Tweets for Microsoft: 151 negative, 482 neutral, 367 positive tweets.
- Categorization of Tweets for Samsung: 109 negative, 489 neutral, 402 positive tweets.
- Accuracy of K Nearest Neighbors Classifier = 86.83333333333333%
- Accuracy of Linear SVM Classifier = 90.0%

- Accuracy of Random Forest Classifier = 90.33333333333333%
- Accuracy of Multinomial Naïve Bayes = 87.5%
- Each model has high precision but low recall for negative tweets, due to imbalance between number of negative and non-negative tweets.
- Samsung, in comparison to the other companies, has the least number of negative tweets, the highest number of neutral tweets and the second highest number of positive tweets respectively.
- Amazon, in comparison to the other companies, has the highest number of positive tweets, the least number of neutral tweets and second lowest number of negative tweets
- Microsoft, in comparison to the other companies, has the highest number of negative tweets, the second highest number of neutral tweets as well as the least number of positive tweets.
- Based on the above analysis, Tweets posted on Amazon and Samsung have an overall better positive reception compared to Tweets posted on Microsoft.

5.2 Conclusion

In this endeavor, we managed to conclude that sentiment analysis can be carried out on Twitter data using several methodologies.

As Twitter data continuously gets updated due to many people posting on the same time on the same topic, data changes and varies the more we run this program. Thus the accuracy of classifiers constantly varies the more data is received as well as being time consuming.

From the results, it was shown that linear SVM and Random Forest Classifier had the highest accuracy (>90%) of all algorithms that were implemented.

Of the three companies that tweets were extracted on, Samsung has the least number of negative tweets and highest number of neutral tweets, while Amazon has the highest number of positive tweets and Microsoft has the highest number of negative tweets respectively. From this information, we can infer that the tweets posted on Amazon and Samsung have an overall better positive reception as compared to those posted on

Microsoft. This implies that Amazon and Samsung are providing better services and are doing and are dong a much better job at satisfying their customers as compared to Microsoft.

5.3**Future Scope**

Even though the classification accuracy for this project is somewhat acceptable by most standards, it still needs to be improved for future work with better data collection and preparation. We also need to experiment on these tweets by using better suited deep learning algorithms.

There was an imbalance between number of negative and non-negative tweets and even negative and positive tweets which led to low precision and sensitivity when predicting negative tweets. This problem could be resolved by extracting tweets from a much more balanced dataset for future work, though extracting them at run-time could prove difficult depending on the popularity of topic

One of the major issues of this project is that the accuracy of our algorithms would constantly change if we were to run the tweet collecting function as tweets get posted in real time by different users at every second in varying lengths. Using automated sentiment analysis could prove to be optimal as long as we have the necessary resources.

It would also help to highlight and underline the reasons for why negative tweets were posted on these company sites or trends to gain an understanding of why the specific customer or user is dissatisfied with the services provided by respective organization or company.

REFERENCES

1. B. Liu, "Sentiment Analysis and Opinion Mining", Synth. Lect. Hum. Lang. Technol., vol. 5, no. 1, pp. 1-167, 2012.

2. Text Analytics Market by Component (Software Services) Application (Customer Experience Management Marketing Management Governance Risk and Compliance Management) Deployment Model Organization Size Industry Vertical Region - Global Forecast to 20, 2017, [online] Available: marketsandmarkets.com.

3. F. N. Ribeiro, M. Araújo, P. Gonçalves, M. AndréGonçalves and F. Benevenuto, "SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods", EPJ Data Sci., vol. 5, no. 1, 2016.

4. A. Moreno and T. Redondo, "Text Analytics: the convergence of Big Data and Artificial Intelligence", Int. J. Interact. Multimed. Artif. Intell., vol. 3, no. 6, pp. 57, 2016.

5. V. A. Kharde and S. S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques", Int. J. Comput. Appl., vol. 139, no. 11, pp. 975-8887, 2016.

6. W. Medhat, A. Hassan and H. Korashy, "Sentiment analysis algorithms and applications: A survey", Ain Shams Eng. J., vol. 5, no. 4, pp. 1093-1113, 2014.

7. L. Ziora, "The sentiment analysis as a tool of business analytics in contemporary organizations", Stud. Ekon., pp. 234-241, 2016.

8. S. Yaram, "Machine learning algorithms for document clustering and fraud detection", Proceedings of the 2016 International Conference on Data Science and Engineering ICDSE, 2016, 2017.

9. N. Yussupova, M. Boyko and D. Bogdanova, "A Decision Support Approach based on Sentiment Analysis Combined with Data Mining for Customer Satisfaction Research", Int. J. Adv. Intell. Syst., vol. 1&2, 2015.

10. S. K. Markham, M. Kowolenko and T. L. Michaelis, "Unstructured Text Analytics to Support New Product Development Decisions", Res. Technol. Manag., vol. 58, no. 2, pp. 30-39, 2015.

11. O. Muller, I. Junglas, S. Debortoli and J. Von Brocke, "Using Text Analytics to Derive Customer Service Management Benefits from Unstructured Data", MIS Q. Exec., vol. 15, no. 4, pp. 64-73, 2016.

12. P. Khobragade and V. Jethani, "Sentiment Analysis of Movie Review", Int. J. Adv. Res. Comput. Sci., vol. 8, no. 5, pp. 19411948, 2017.

13. S. M. Kamruzzaman, F. Haider and A. R. Hasan, "Text Classification using Data Mining", Science (80)., pp. 19, 2010.

14. T. Pang-Ning, M. Steinbach and V. Kumar, Introduction to data mining., 2006.

15. E. Alpaydin, Introduction to Machine Learning., 2004.

16. K. M. Sreerama, "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey", Data Min. Knowl. Discov., vol. 2, no. 4, pp. 345-389, 1998.

17. R. C. Barros, A. C. P.L. F. de Carvalho and A. A. Freitas, Automatic Design of Design-Tree Induction Algorithms., Springer International Publishing, 2015.

18. A. Jain and P. Dandannavar, "text analytics framework using apache spark and combination of lexical and maching learning techniques", Int. J. Bus. Anal. Intell., vol. 5, no. 1, pp. 36-42, 2017.

19. M. Allahyari et al., A Brief Survey of Text Mining: Classification Clustering and Extraction Techniques, vol. 1707, no. 2919, pp. 1-13, 2017.

20. A. Trevino, "Introduction to K-means Clustering", Datascience.com., 2016.

21. M. Hofmann and R. Klinkenberg, "RapidMiner: Data Mining Use Cases and Business Analytics Applications", Zhurnal Eksp. i Teor. Fiz., 2013.

22. K. S. Rawat, "Comparative Analysis of Data Mining Techniques Tools and Maching Learning Algorithms for Efficient Data Analytics", JOSR J. Comput. Eng., vol. 19, no. 4, pp. 56-60, 2017.

23. H. Kaur and V. Mangat, "Dictionary based Sentiment Analysis of Hinglish text", Int. J. Adv. Res. Comput. Sci., vol. 8, no. 5, pp. 816-822, 2017.

24. M. H. Peetz, M. De Rijke and R. Kaptein, "Estimating Reputation Polarity on Microblog Posts", Inf. Process. Manag., vol. 52, no. 2, pp. 193-216, 2016.