T.C.
SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

SENTIMENT ANALYSIS OF TWITTER TEXTS USING MACHINE LEARNING ALGORITHMS

M.Sc. THESIS

Hawar Sameen Ali BARZENJI

Department : COMPUTER AND INFORMATION ENGINEERING

Supervisor : Asst. Prof. Dr. Veysel Harun ŞAHİN

July 2021

T.C.
SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

SENTIMENT ANALYSIS OF TWITTER TEXTS USING MACHINE LEARNING ALGORITHMS

M.Sc. THESIS

Hawar Sameen Ali BARZENJI

Department : COMPUTER AND INFORMATION ENGINEERING

This thesis has been accepted unanimously by the examination committee on 09/07/2021

Head of Jury **Jury Member** **Jury Member**

DECLARATION

I declare that all the data in this thesis was obtained by myself in academic rules, all visual and written information and results were presented in accordance with academic and ethical rules, there is no distortion in the presented data, in case of utilizing other people's works they were refereed properly to scientific norms, the data presented in this thesis has not been used in any other thesis in this university or in any other university.

Hawar BARZENJI

25.5.2021

ACKNOWLEDGMENTS

1. Above all; the one and only almighty *Allah;* the most merciful, and the most gracious.
2. First of all, I would like to express my full thanks to my most generous supervisor Dr. Veysel Harun ŞAHİN; without his help, this research was impossible. It was his precision, professionalism, and guidance that put me in the right path of my study.
3. It is the right time and the right place to thank most of the teachers of Sakarya University, during two semesters, they taught me the most basic material that is required for finishing and getting a master degree with a high quality. If I mention some of them: Dr. Ahmet ZENGIN, Dr. İsmail Hakkı CEDIMOĞLU, Dr. Berrin DENIZHAN, and Dr. Muhammed Fatih ADAK.
4. Someone I owe him my *weltanschauung* in the world of open-source society (i.e. GNU/Linux community), he is Sia NARIMAN. Once upon a time, he was my ideal person.
5. The first and only Math teacher of my entire life forever; Mr. Nizar MAJID. He who shaped my mathematical philosophy.
6. Also, I would like to thank these friends, for their beings in my life and their amazing friendships; Rawa HAJAR, Dilshad STAR, Karzan OSMAN, Rizgar SHAWKAT, Omid MUHAMMED, Muhsin JAF, Fariq AHMED, Dilshad RAHEEM, and Ms. Monica.
7. Finally, my gratitude to the most perfect parents (Sameen and Maliha) ever in the world, who educated me and taught me the most basic morality for a good living life (i.e. *Virtue*). Also, my BARZENJI family; uncle Safa and aunt Zina; and all brothers and sisters, especially Sharan, Sardar, Harem, Dyar, and Hedi.

DEDICATION

Dedicated to Sheikhzada in friendship and admiration.
Sakarya, Turkey, May. 2021.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

ACC	: Accuracy
AI	: Artificial Intelligence
BoW	: Bag of Words
CM	: Confusion Matrix
CSV	: Comma Separated Values
DIP	: Digital Image Processing
FN	: False Negative
FOSS	: Free and open-source software
FP	: False Positive
GNB, GaussianNB	: Gaussian Naïve Bayes
HMM	: Hidden Markov models
HTML	: Hypertext Transfer Protocol
IC	: Computer Science
KNN	: K-Nearest Neighbors
ML	: Machine Learning
NLP	: Natural Language Processing
NLTK	: The Natural Language Toolkit
OS	: Operating System
PL	: Programming Language
POS	: Parts of speech
PPL	: Python Programing Language
PR	: Pattern Recognition
PREC	: Precision
RE, RegEx, RegExr	: Regular Expression
REC	: Recall
RFC	: Random Forest Classifier

sklearn	: Scikit-learn
SMN	: Social Media Network
SQL	: Structured Query Language
SR	: Speech Recognition
SVM	: Support Vector Machine
TF-IDF	: Term Frequency – Inverse Document Frequency
TN	: True Negative
TP	: True Positive
TTD	: Trump's Tweets Dataset
VADER	: Valence Aware Dictionary and sEntiment Reasoner
WWW	: World Wide Web

LIST OF FIGURES

LIST OF TABLES

SUMMARY

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning

In this thesis, sentiment analysis as the use of natural language processing and machine learning classifiers have been studied on Trump's tweets scraped web page, which is saved in the form of dataset. After data preparation, the most important sentiment analysis procedures have been applied to the host dataset. Also, other natural language processing strategies have been processed, like cleaning the dataset in order to be ready for text vectorization. In cleaning the textual data, all the required techniques like removing stopwords, word lemmatization, regular expression, and tokenization have been used to remove undesired words. We succeeded in reducing the size of the "content" feature in the dataset with the target of taking fewer capacity. Since the two last decades with the development of social media networks, hateful activities have become a phenomenon, this became a challenging task to know the subjective polarities of each one's published text; therefore, each sentence has been judged-on regarding their polarities whether they are positive, negative or neutral. At the end, by using machine learning algorithms like (Random Forest classifier, Gaussian Naive Bayes, and Support Vector Machine), the cleaned data has been trained and tested to see the accuracy of the prediction results, the comparison shows 88%, 72%, and 89% respectively for each classifier.

MAKİNE ÖĞRENME ALGORİTMALARI KULLANILAN TWITTER METİNLERİNİN DUYGU ANALİZİ

ÖZET

Anahtar Kelimeler: Duygu Analizi, Doğal Dil İşleme, Makine Öğrenimi

Bu tezde, veri seti şeklinde kaydedilen Trump'ın tweet'leri kazınmış web sayfası üzerinde doğal dil işleme ve makine öğrenmesi sınıflandırıcılarının kullanımı olarak duygu analizi incelenmiştir. Veri hazırlandıktan sonra, ana bilgisayar veri setine en önemli duygu analizi prosedürleri uygulanmıştır. Ayrıca, metin vektörleştirmeye hazır olmak için veri kümesini temizlemek gibi diğer doğal dil işleme stratejileri de işlenmiştir. Metinsel verilerin temizlenmesinde, istenmeyen kelimelerin kaldırılması için stopwords kaldırma, kelime lemmatization, düzenli ifade ve tokenization gibi gerekli tüm teknikler kullanılmıştır. Daha az kapasite alma hedefi ile veri setindeki "içerik" özelliğinin boyutunu küçültmeyi başardık. Son yirmi yılda sosyal medya ağlarının gelişmesiyle birlikte nefret dolu faaliyetler bir fenomen haline geldi, bu, her birinin yayınlanan metninin öznel kutuplarını bilmek zorlu bir görev haline geldi; bu nedenle, her cümle, olumlu, olumsuz veya tarafsız olup olmadığı kutuplarına göre yargılanmıştır. Sonunda (Random Forest sınıflandırıcı, Gaussian Naive Bayes ve Support Vector Machine) gibi makine öğrenme algoritmaları kullanılarak temizlenen veriler eğitilmiş ve tahmin sonuçlarının doğruluğunu görmek için test edilmiştir, 88%, 72% ve her sınıflandırıcı için sırasıyla 89%.

CHAPTER 1. INTRODUCTION

In this thesis study, gathered dataset of Trump's tweets has been used in order to apply the most important stages of NLP on it. Starting from the basic level of data cleaning and selection; after that, different text cleaning techniques has been applied like removing stopwords, word lemmatizing, and regular expression for the target of getting a clean textual data. In the result section, the size of the dataset before and after cleaning has been tested in Apple's macOS operating systems, to make sure that the cleaned stages are working well on the dataset or not. Two more steps have been applied on the cleaned tweets. First text polarities for classifying each tweet's subjectivity which are positive, negative, and neutral. Then, the cleaned texts have been changed into numerical data by using BoW's text vectorization. Finally, Random Forest classifier, Gaussian Naive Bayes and SVM have been applied in order to train the dataset and testing its results by using confusion matrix, and classification report table which covers Accuracy, Precision, Recall metrics, and F1 score, and etc.

Before going to start our research study. There are some questions we need to deal with it. Philosophy, as the mother of all sciences, has given its attention to most of our daily life's problems, from art to techniques, and others. If we want to know where all of these techniques like NLP and ML came from, we have to go back to Ancient Greek, especially its philosophy.

1.1. Philosophical Background

What is the philosophical background of language? What is the nature of language? And how does it work? Can we make a machine that is capable of understanding our language? Are machines capable of generating languages? If yes, are they understand the semantic meaning of language or not?

These were the questions of philosophy, but nowadays we are dealing with them in the fields of NLP and machine learning. The development of machine learning leads to a place that we can give most of our daily life's problems to them, and get it back in a glimpse, not only fast, but also intelligence.

We can find the Plato's (427 – 347 B.C.) dialogue of *Cratylus* (dated about the year 388 B.C.), as the first text that talks about language and its nature. He discusses the question of whether names of things are correct by of convention or natural. In another way, the topic of *Cratylus* is about the names and their relations with the objects. Simply, it is a critique on the subject of naming the objects. Is the relation of name with its object is natural or arbitrary?

This question of whether language is a set of randomized (i.e. arbitrary) signs or whether names have a rational relation to the objects they signify, comes to the modern linguistic. The founder of modern linguistics Ferdinand de Saussure (1857 - 1913), a Swiss linguist, puts his most attention to this problem. In his famous lectures Course in General Linguistics (1916) argues that the action of naming the object is arbitrary. If we accept Saussure's claim, it will be a good argument for those who are claiming that semantic meaning (which is usually uses against AI) is not that much important for the robots, since in the theory of studying signs and symbols in their use or interpretations, there are not any logical relations between signifier and signified.

The target of making a machine that thinks and behaves like a human again goes back to the Greek. Ancient Greek mythology told of a bronze robot more than before 2500 years ago. The first robot (*Talos)* was forged by the god of invention (*Hephaestus*), for patrolling the island of Crete.

From Greek to German idealism the idea of making a machine that acts and thinks like a human being didn't fade away. Along with Isaac Newton (1643 - 1727); the father of differential and integral calculus Gottfried Wilhelm Leibnitz (1646 - 1716) in his

phenomenon book (*Monadology*, fragment number17, 1714) talks about (*Mill Argument*).

His argument is nothing but an imagination. He supposes a machine constructed in a way that is capable of three human properties (*thinking*, *feeling*, and *perception*). He continues his argument and talks about increasing its size, with keeping the same proportions, so that one might enter into it, but what we see and we experience in examining its interior, are only parts which work one upon each another, and never anything about explaining what perception is.

From ancient Greek's philosophy and German's idealism to our modern world, the target of making a machine with human characteristics still did not fade away. Nowadays the computers with the help of ML algorithms become super smart in a way that we give them our own language (especially English) to parse, tokenize, clean and other related processes that belong to text mining on our natural language.

1.2. Social Media As The Modern World's Phenomena

With the rise of the modern era, our life faced a new way of communicating, a new way of social interaction [1]; which is a social media platform. Willy-nilly if we accept it or not, SM became a phenomenon in our daily life; it became an essential part of our recent lifestyle. If in old Greek the Athenian citizens were sharing their opinions with the philosophers about a particular problem in *agora*, nowadays SM can be called *the new agora*; with the difference of the first was in the physical objective world, but the latter is in virtual life.

Nowadays, people use SM not only for sharing their emotional, desire, and ideas about a particular subject [2]; but also, they use it for marketing [3], political messages, and etc. A huge attention for the latter mentioned category is Twitter's platform. It is clear that most of the politicians around the world are using Twitter as their number one's favorite platforms, but we should not forget that this platform has its own positive and

negative effects on the decision of the people, and what is familiar among the people about all of the SM platforms is that its negative side is more than its positive side.

In [4] they are calling the users of the world to unite. This is the same calling of Karl Marx (1818 - 1883) in his famous *Manifesto* (published in 1848) in which he was calling the labors of the word to unite. Not only this, but also, they adapted Karl Marx's famous attention to the world, "The specter of social media."

For the all above important reasons of SM, it's the right time to put our focus on this new phenomenon, from philosophy to machine learning, NLP and other evaluating techniques in data analysis, can be used to check the activities on this platform.

1.2.1. Why trump's tweets?

The Twitter's account of President Donald Trump has about 88 million followers [5] of Twitter users, and many of them are using his account as a way of news and information resource [6]. His frequent use of the social media account and his influence as President of the US, has made his tweets an essential source in a variety of scientific and research studies, like evaluating his Tweets [7, 8].

1.3. Natural Language Processing

NLP is a subfield of Computer science and AI (particularly the field of machine learning). It deals with the language of the human being and how it understood by the computer. This technique can be obtained with the help of the computational linguistics. To understand the natural language needs a lot of information about lexicon, semantics, syntax, and information about our real world [9, 10, 11].

We can talk about NLP as a synthesis of philosophy of linguistics, computer science, and artificial intelligence. This branch of science deals with the interactions between human language and computers (Robot agent). This field cares about how to code and program computers, in order to process and analyze the data of natural language.

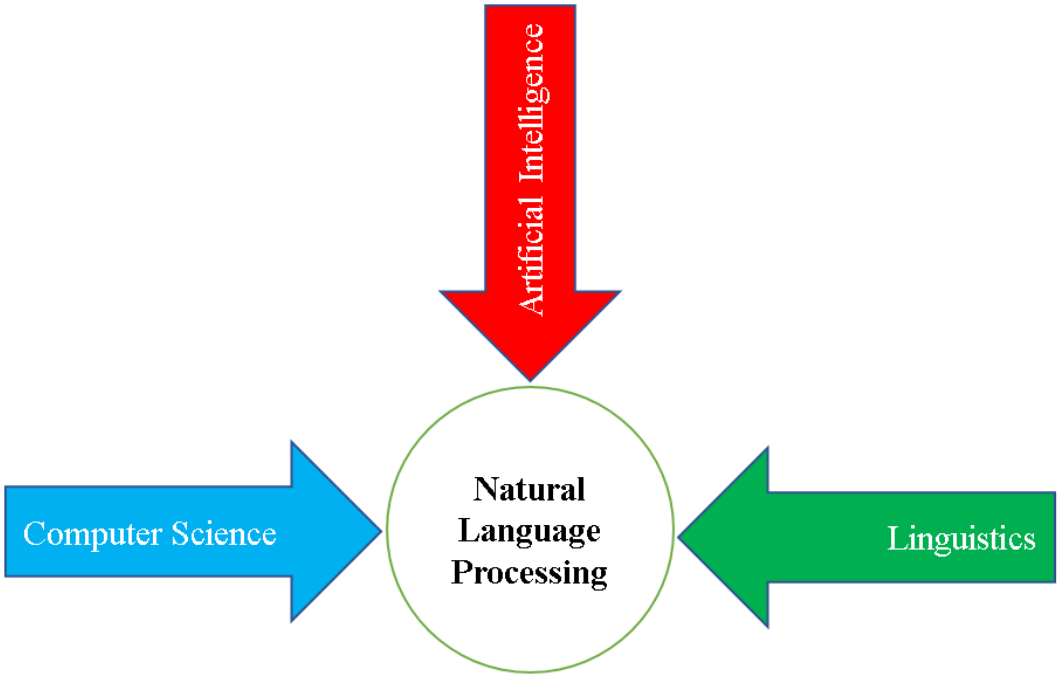

Figure 1.1. The synthesis of NLP

1.4. Machine Learning Classifiers

Machine learning is about taking out knowledge from raw data, and learning from past experience in order to predict the next upcoming data. This research field is an intersection of AI, statistics, and computer science. The usage of machine learning methods in recent years is very useful in nowadays life. Starting from automatic suggestions of which videos to be watched, or what type of fast-food to order or which items to buy, and for customizing the podcasts; most of the modern portals and devices have machine learning algorithms at their kernel; and ML can do all of these based on learning from experiences, the more training the classier, the more accurate the prediction is. For example, Twitter, Amazon, or Netflix (use Random Forest in personalizing the movies) use ML algorithms in most of their parts, in order to customize user's experience [12]. In our thesis, three classifiers have been used which are Random Forest, GNB, and SVM to train and test the dataset. More details are given in Chapter 3.

Originally Random Forest derives from Decision Tree, this means, it shares all the benefits of decision trees, but historically it refers back to an American computer scientist at IBM Watson Health (Tin Kam Ho) in 1995 with the term of random decision forests [13]. After a while Leo Breiman coined the Random Forest term in 2001 [14].

Naive Bayes is a kind of probabilistic or statistical supervised ML algorithm. It builds a probability model on the category description for all feature vectors in the training set. It works based on Bayes theorem [15], which calculates conditional probability. Gaussian distribution, is one of the most usual and main technique in calculating statistics and probability field, stating the "naive" supposition of conditional independence between every pair of attributes given the value of the class variable [16].

Support Vector Machine (SVM) or originally Support Vector Networks (SVN), is a type of supervised ML algorithm that was coined by both C. Cortes and V. Vapnik in 1995 [17]. It can be used in both classification and regression tasks. This prediction tool uses ML theory to maximize required accuracy and avoids overfitting of the data. This supervised learning ML uses in two group classification problems. It can solve linear and non-linear. This algorithm is efficient when dealing with high dimensional data such textual data. The idea of SVM is simple, its objective is to find a hyperplane that has the largest edge (side), i.e. the decision boundary that separates the support vectors to the farthest, and it is in charge of for finding the decision boundary to discrete different classes and maximize the edge.

1.5. Problem Definitions, Scopes, And Suggested Solutions

This study deals with web page scraping in social media in the form of textual data. Since there are a huge amount of written texts every day, we face many problems, starting from the need for more extended capacity, to uncategorized and messy data which is a hard task for later on indexing and searching purposes. Text analyzing is a hard task to do, since it deals with non-numerical data, which is hard for machines to

process on it. It will need to be changed into numerical data (by using Text Vectorization), cleaned, and prepared in a scientific way.

The suggested approach is a combination system of NLP and ML algorithms. First, by using NLP techniques, the text has been cleaned, parsed, and removed all the stopwords, hashtags, links, and other unnecessary information, have been applied on it in order to get a plain text which contains only the most important tokens for the ML classifier, and a text with less capacity required (This is what most of the earlier works did not take care of it.). Then, the cleaned text changed into a bag of words (BoW) in order to be ready for the last process which is training and testing our data.

Finally, since every approach is measured by its outcomes, the Random Forest classifier has been applied to detect and predict the next tweets. The following targets; *Accuracy*, *Precision, F1 Score*, and *Recall* metrics are given for the performance measurement. All of those above outcomes have been talked about in detail. Also, comparing the size of the cleaned text with its original source in both macOS and Windows Oss have been considered and measured; the results have been demonstrated in tables.

The suggested thesis can have a good impact in many scientific areas and can have a good contribution to science. The proposed system can be used in pre and post scraping processes. It can be used during web page scraping before the crawled data is going to be saved in the form of csv file; or after saving the crawled page.

It can be used in various fields; It can be used in Digital Image Processing (DIP) and Pattern Recognition (PR); since there are images that contain textual data, it can be extracted by one of the mentioned proceedings, and our system can be applied on it, of course, the more gathered data, the more accurate it will.

Also, it can be used in AI and robotics; the agents can use the suggested method in converting speech-to-texts, which is famous as Speech Recognition (SR) for cleaning

the noisy speech in order to understand and perform the required given orders, with the help of time, the agent can get more experience and gradually becomes smarter.

Since philosophy is one of the closest fields to AI and NLP, it can be used in social sciences, like the philosophy of linguistics; as we mentioned before, we shall not forget that philosophy as the mother of all sciences, raised the first and most early questions about language and a thinker machine.

Also, our mathematical results in confusion matrix can be used in the field of data analysis and statistical purposes.

The framework of this master's thesis study will be as follows: In chapter 1; an introduction about the whole study (i.e. the problem definitions) has been given, which includes the philosophical background of the problem, the importance of social media, and the reason of choosing Trump's tweets. In addition, NLP and ML classifiers have been given briefly. At the end, the problem definition regarding the scopes, and the suggested solutions. In chapter 2; the closest studies in the same area (i.e. related work) have been cited. In chapter 3; focuses on NLP with its steps, and the ambiguity of NLP. In chapter 4; all three ML classifiers have been discussed, which are Random Forest, GNB, and SVM. Chapter 5; is the methodology (i.e. materials and method); is a full detailed explanation about and pre-processing steps, starting from gathered data, data cleaning, and text cleaning procedures with all of its included steps: like removing stopwords, word lemmatizing, regular expression, and tokenization. After detecting text polarity, and text vectorization (BoW) have been applied. In chapter 6; the results and discussion in two parts, the confusion matrices and classification reports are given for both criterion for performance evaluation and evaluating the experimental results. Finally, the conclusion with the authors' recommendations for the upcoming studies are given.

CHAPTER 2. RELATED WORK

Nowadays Twitter sentiment analysis gained most of the researcher's attention [18]. These concise texts are used as a raw material for data analysis. By using text polarities (positive, neutral, and negative), emotions (angry, sad, loved, etc.) are judging on each text's subjectivities.

Before going deeper into our own study, we will give a brief overview about the previous articles (i.e. Literature Review) that have been done in the same area which is the combination of NLP and machine learning.

In [19] they proposed a study for detecting fake news spread through images from SM like Facebook, Twitter, etc. They proposed K-means clustering (based on issuing day) to get a general outline of how the images were used throughout the time.

In [20] they introduce a hybrid method which is a combination of NLP and ML techniques to guess and recognize hateful speech from social network websites. After gathering the hateful speech, steaming, tokenizing, unwanted character removal was applied on it. Finally, they classified the texts into neutral, offensive, and hate (in our study, we classified the tweets into positive, neutral, negative) language. The performance of the system is then evaluated using overall accuracy, f1 score, and precision and recall metrics. The system achieved an accuracy of 98.71%.

In [21] they applied NLP techniques to analyze tweets with regarding to their mental health. They used deep learning (DL) models to classify each text with regarding of the following emotions: angry, anticipation, disgust, frighten, delight, sadness, surprise, and confidence.

In [22] a group of researchers made a comparison study of the Naïve Bayes algorithm and Natural Language Processing on the dataset of Twitter. Their comparison is in two categories: *accuracy* and *speed.* Their experimental results showed that the Naïve Bayes algorithm got 63.5% accuracy, which is lower than that achieved by the NLP method. But in the processing speed analysis, the machine learning method performance is 5.4 times higher than that of the NLP method.

In [23] they used sentiment analysis to extract human feeling and evaluate whether it's negative, positive or neutral. Through unconstructed text by using NLP. They also Machine learning in order to train and test the dataset. They compared the results using different ML classifier, like Naïve Bayes, Random Forest, Support Vector Machine, and etc.

From all above research studies and articles, we can notice that most of them have hybrid methods, which means a combination of NLP and ML algorithms. It seems that without combining those two fields, the work would be impossible. In our research, after applying NLP techniques on the texts, Random Forest classifier as one of the ML algorithms has been used to train and test the cleaned tweets.

In [24] USSAMA YAQUB applied sentiment analysis on trump's tweets during the early appearance of the coronavirus pandemic (i.e. COVID-19) in the United States. Statistically, he discovered a negative correlation between the sentiments of his tweets and the number of cases in the United States. One thing which is very important in his study research is that he noticed a gradual shifting in his tweets from positive to negative sentiments polarities while he is mentioning China and COVID-19 together. What USSAMA did is amazing, but his study is not a hybrid method, which means he didn't apply machine learning classifier after his sentiment analysis, this makes his research stay in the domain of data analysis and NLP techniques.

In [25] In this paper, they analyzed the relationship between the tweets written by POTUS (stands for the President of the United States) and his acceptance classification using sentiment analysis and data representation tools. They applied all the NLP

requirements on the tweets of POTUS; they extracted, cleaned, and gave a numerical measurement based on the content, which they named the “sentiment score". By setting side by side the tweets before, during, and after the election, they found that the “sentiment score” of Trump’s tweets has been increased with a mean in time by an amount of 60%. By using cross-correlation analysis, they find a preparatory causing relationship between POTUS Twitter activity and approval rating. Still, their study is one-sided research, it seems something is missing. What we do with sentiment analysis and NLP techniques, somehow leaves the problem unsolved. By using machine learning methods, we can train our data in a way that can recognize the next upcoming data which gives to the system, so the robot can predict it.

In [26] this paper, they used social media content to forecast real-world results. In particular, they used the chatter from Twitter platform to predict box office incomes for movies. They revealed that the tweets which are generated about specific movies can perform better in market-based predictors. They applied sentiment analysis on the extracted Twitter data, but they didn’t mention which method they did the forecasting.

In [27] they did most of the essential processes on the Trump’s tweet dataset. From sentiment polarities to word cloud, sentiment analysis, and machine learning methods. They got the used dataset by downloading it from Twitter API using Python script and MongoDB. Then by using Python programing language within most of the packages for gathering, cleaning, and processing data for the required format processing. Also, for the extraction of the tweets from Twitter platform, they have used a package in Python which is called Tweepy. This library includes a huge amount of functions with minimal lines of scripts. They used the NoSQL database which is MongoDB.

In the next steps, they built the sentiment analysis models using the available libraries on NLTK and Scikit-learn. What is good about this research paper is that they used more than one classifier algorithm in order to compare each classifier’s performance and accuracy by training and testing the dataset. Once the classifier within the highest accuracy is determined, the remainder of the tweets will be categorized by using this

classifier, regarding their being positive, negative, or neutral. Finally, the cleaned and classified tweets will be used for analysis purposes.

In [28] they picked up two types of information, which they categorize them according to their subjective polarities: Pro-Trump (threshold class value with 0.52), and anti-Trump (threshold class value with 0.56), with indicative regular expressions and hashtags. Tweets that have at least one positive or one negative pattern, and do not have both positive and negative patterns (i.e. neutral), are considered as the initial positive and negative instances. As a result, this gave them a noisy dataset; for example, the tweets like "*#MakeAmericaGreatAgain, #Tag is a bummer.*" is against Donald Trump, incorrectly labeled favorable. Their suggestion is a quantitative analysis for the impact of this type of noise, and the goodness of initial patterns, can be done in the future works through a supervised learning approach.

Tweets with "neither" class neither positive nor negative ranges from news about the target of interest, to tweets totally irrelevant to him, made it difficult for them to collect neutral tweets (this problem has been solved in our study by using a threshold subjective polarity, details will be given in later sections), and they classified those kinds of tweets to be in that class based on a heuristic method (which is a randomized algorithm) algorithm.

The difficulties and challenges in their studies is in the limited number of seeds, they needed to collect more training instances in building the classifier. Since the original noise in the labels and the imposed fragmentary view of data performs poorly. As a reason and solution for this, instead, they augment the dataset with tweets that their relational model classifies as positive or negative with a minimum confidence values; which means the class value of 0.52 assigned for pro-Trump and 0.56 assigned for anti-Trump.

For cleaning the dataset, they convert the tweets to lowercase, and they remove all the stopwords and punctuation marks (we did all of these procedures in our study). For

classifying the tweets, they used SVM (our used classifier is Random Forest), which they employed these features below like normalized to unit length after conjoinment.

The results showed in metrics used are the macro-average of the F1-score for favor, against, and average of these two. For completing the system for the used task is a deep neural network for training both pro and against instances, which were collected through linguistic patterns. Finally, at the time of test, they randomly assigned the instances, about which the classifier was less confident, to the "neither" class. Another baseline is an SVM, trained on another stance classification dataset, using a combination of n-gram features.

In [29] they analyzed the direct effect of President Trump's tweets on two US selling markets. They compared the Trump's tweets and the data of intra-day market, in order to study market's reactions using a hybrid system of sentiment analysis, and both types ML algorithms which are regression and classification. ML regressors have been used on both stock data and tweets to in order to forecast the incoming tweet market stock index mean, and incoming tweet stock trends.

We can see from their results show a notable negative impact response when President Trump posted tweets in the open market time. But they claim that tweets with a strong positive or strong negative sentiment polarities had a good market reaction.

CHAPTER 3. NATURAL LANGUAGE PROCESSING

In this chapter, NLP and ML classifiers have been studied in details. Previously in Chapter 1 an introduction for both subjects have been given, but here once more, more details will be given of both subjects. First NLP will be studied and other proposed ML algorithms like Random Forest, GNB, and SVM will be applied on the dataset.

Natural languages are differing from computer programming languages. They are not intended to be interpreted (i.e. assembled or compiled) into a sequence set of mathematical operations, like how programming languages are, but there are no compilers or interpreters for natural languages such as English and Deutsch. Natural languages are what we as human beings are using it to share and say our opinions with each other. But in the case of PL, we don't use PL to tell each other feelings about a specific problem. A computer program written with a PL tells a computer what to do. Concisely, NLP is a link between the human being to human being; but PL is a link between human beings to machines [30].

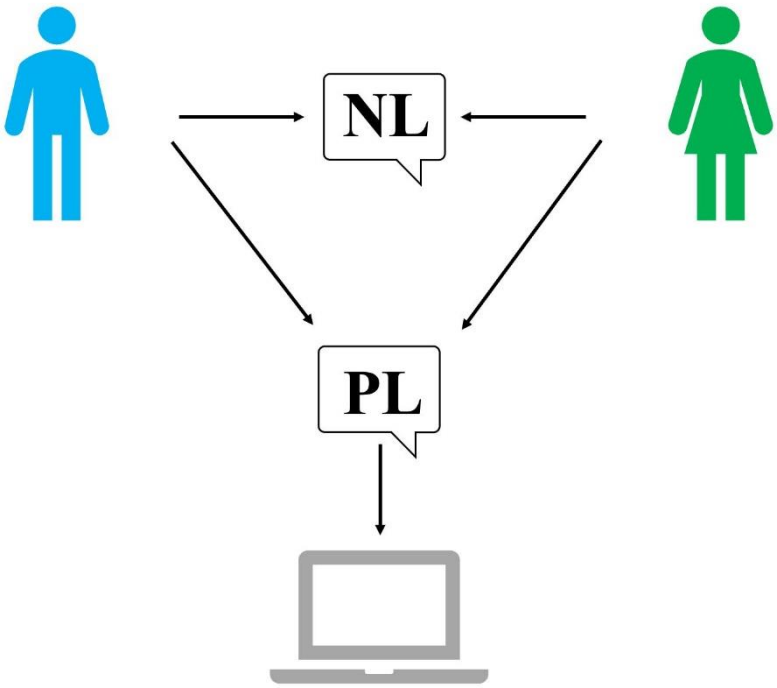

Figure 3.1. NL vs LP

If we go back to the history of NLP, we can notice Alan Turing's efforts during the early 1950s. Alan Turing, who was a mathematician and considered as the father of modern Computer science, in his phenomenon article (*Computing Machinery and Intelligence*, 1950) arguments that, if a computer is able deceive a human during a chatting text, the human cannot recognize the one who is chatting with is a robot; it means the machine has passed the Turing Test.

In 1981, John Searle coined "Chinese room" in his famous article (*Minds, brains, and programs,* 1980) argument. Simply Searle claims that it is impossible to get semantics from syntactic, which means the machines cannot think and they don't get those letters and the symbols the same as the way we are getting them.

3.1. Steps Of NLP

The general steps of NLP will start from the very basic analysis which is lexical analysis, followed by syntactic analysis, then semantic analysis, disclosure integration, and pragmatic analysis. In order to apply NLP in any texts, it has many stages that

needs to go through. The figure below, explains the steps of NLP from the starting point (Input text) to the desired text (Output text).

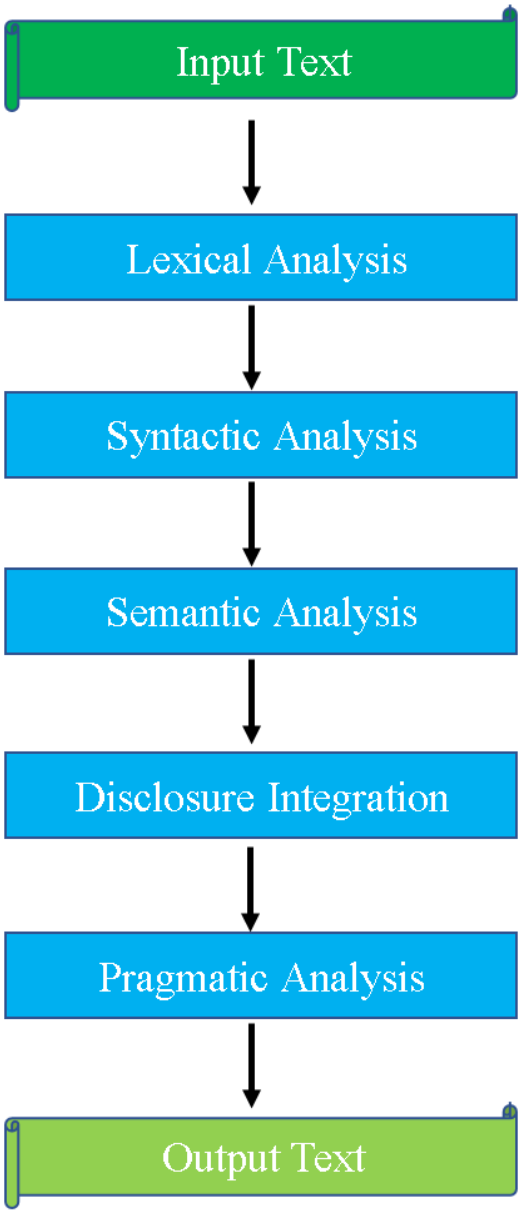

Figure 3.2. The NLP steps

Briefly, in the proposed thesis, the following actions have been done in each step. In the lexical analysis, word lemmatization like parsing and tokenization have been performed. In syntactic analysis stopwords, hashtags, links, and other unnecessary words have been removed. In disclosure integration the sentences have been integrated after removing unwanted tokens. In the semantic analysis, on each cleaned text,

subjective sentiment polarities have been applied depending on the meaning of each sentence. Finally, in the pragmatic analysis, the proposed ML algorithms have been applied to train and test the results, for the target of predicting the next tweets subjectivity.

3.2. Ambiguities Of Natural Language Processing

Dealing with NLP causes some ambiguities and confusions, when for a single word (or token) in one single passage there exist more than two possible meanings; is named as an ambiguity. The given passage is considered as ambiguous, if a variety of replacement linguistic structures can be construct for it. These ambiguities come from the three level of NLP (i.e. three categories):

1. Lexical ambiguity (relating to *words* or vocabs in language).
2. Syntactic ambiguity (relating to the *rules* (i.e. the structure) of sentences in language).
3. Semantic ambiguity (relating to the *meanings* in language).

3.2.1. Lexical ambiguity

In this type of ambiguity in which a word or a number of words are having dissimilar meanings in different circumstances. In these cases, an individual word may have different meanings in the language to which it refers to. For example, the word "Square" has several different lexical meaning; in Geometry is "a plane figure with four equal sides and four right angles"; and it can be "a mathematical operation which is power (multiplying a number by itself)"; also, "an open 4-sided place in the road where it divides the road into three or four directions".

Similarly, the word "can" have many meanings is this sentence: "*I can can the can, but the can cannot can me*".

Here, the word “can” is being used in three different definitions. First, the word “can” with blue color refers to the verb of abilities; second, the “can” with green color refers to a technique (a work) which is used in industry to put things (fish or beans) into a cylinder-shaped metal which is called can; third, the “can” with red color used as nouns which is a geometrical cylinder shape.

Another ambiguity can be found in this sentence: *“I saw the saw”*. Here we have used the word “saw” twice, but it can have three different meaning, the first “saw” is an action which is the past tense of the verb “see”; while the second “saw” is a noun, which is an instrument for cutting wood or other hard materials. Also, “saw” can be an American horror movie (2004 - 2010).

So, the words “can” and “saw” are ambiguous regarding their forms and syntax, either noun or verb. Similarly, the term ‘Pound’ can also create lexical ambiguity, since the term ‘Pound’ can be used as a measure of heaviness; and it can be currency.

3.2.2. Syntactic ambiguity

The primary reason which causes a syntactic ambiguity to arise is the formation of a sentences, in which a sentence can have many possible interpretations. The occurrence of syntactic ambiguity often happens, when we add an expression, such as a function word expression, the use of which is not defined clearly. For example, we will take the below sentence: *“Did you talk to the girl, with the mobile phone?”*

This sentence can have two possible meanings, and makes it unclear. This type of obscurity appears from the prepositional phrase. The two possible meaning of the phrase “with the mobile”, would be as follows:

“Did you speak with the girl, who is holding the mobile phone?”
“Did you speak to the girl, by using the mobile phone?”

In NLP, tasks like part of speech (POS) tagging, can be used to address and solve this obscurity.

3.2.3. Semantic ambiguity

In a plain text this type of ambiguity is identifies as word confusions. Also, this type of obscurity is demanding more efforts in contrast to the two other mentioned types' disambiguation. We shall not forget that even John Searle's claim against Turing's test refers to the same problem; which claims that it is impossible to get semantic meanings from syntactic. This obscurity occurs in those situations when in a sentence there exists equivalent words or a series of words that have got many correlate meanings. For example, imagine you are writing a message in a social media chat groups like Facebook, and your message is this statement: "*Dear friends, could you please inbox me the required information? Regards*"

From the above given example, the term "inbox" is not clear enough. The word 'inbox' can represent G-mail, Y-mail (Yahoo mail), and also for Facebook accounts. It means, we are not exactly clear about it, and not sure of which inbox we mean.

Machine learning technique is a powerful analysis tool when cooperating with the field of NLP or in computational linguistics. Due to the reason of linguistic knowledge, which may contain ambiguities; therefore, various ML techniques are used for solving the ambiguity in speech and language processing. The current techniques for directing several NLP tasks, in supervised learning are the following techniques [31]:

1. Hidden Markov Models (HMM),
2. Support Vector Machines (SVM),
3. Naïve Bays (NB),
4. Deep Learning (DL),
5. Random Forest.

Some of the researchers published a paper [31] with the goal of handling the dilemma of these ambiguities in speech and language processing, to supply and give a summary of fundamental classification of linguistic knowledge, and arguing various and different existing ML algorithms and their classification into types and finally to give an understandable analysis of various types of the status quo ML algorithms. The researchers' target is to test these approaches and to hang on those advanced procedures.

CHAPTER 4. THE CLASSIFICATION ALGORITHMS

This chapter has been put for explaining the mathematical background of the three used classifiers. First, we start with Random Forest, then we will give a brief overview of the other two algorithms which are GNB and SVM.

4.1. Machine Learning

We can say that machine learning is the implementation side of artificial intelligence (AI). More philosophically, AI is potential (being in itself), and ML is the actual (being for itself). ML provides the systems with the capability of learning and advancing from experience by itself, with or without human intervention.

There are two types of ML algorithms for both *regression* and *classification* [32]:

1. *Supervised Learning*
2. *Unsupervised Learning*

The most popular supervised machine learning classifiers are:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM (Support Vector Machine)
5. Naïve Bayes: (GNB)
6. KNN (K- Nearest Neighbors)
7. K-Means
8. Random Forest

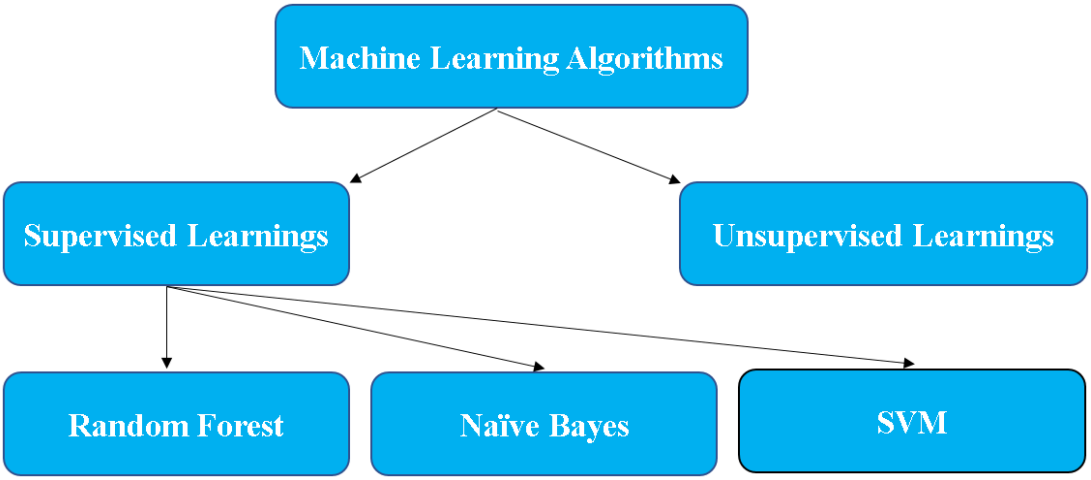

Figure 4.1. The diagram of ML approach

4.1.1. Random forest classifier

While the decision trees have the problem of overfitting the training data, Random Forest is a solution to label and solve this problem. A Random Forest is basically a group of decision trees, where each tree is a little bit differ from the other one. The idea behind this algorithm is that each tree may do a relatively good job of predicting, but will likely over fit on part of the data. If we construct many trees, all of which work well and over fit in different ways, in order to resolve this issue, we have to decrease the amount of overfitting by averaging their outcomes. In order to implement this procedure, many decision trees need to be constructed. Where each tree is supposed to do an allowable work of predicting the object, and also it must be different from the other trees. Random Forests get their name from managing randomness into the tree building to make sure each tree is distinct [33].

Trees in the Random Forest classifier are randomized in two ways:

1. Either by choosing the data points used to construct a tree,
2. Or, by choosing the attribute in each splitting test technique.

This type of algorithm uses in both *classification* and *regression* types that utilize by building a multiple of decision trees during the training procedure and producing a class that is the mode of the classes or average prediction of the single trees.

As we said earlier, historically it refers back to an American computer scientist at IBM Watson Health (Tin Kam Ho) in 1995 with the term of (random decision forests). After a while (Leo Breiman) coined the Random Forest term in 2001.

Since it uses many trees, one of the biggest pros of Random Forest is its high accuracy. Because of its formation of forests which are based on a lot of trees, lots of decision trees can ensure a better and higher accuracy than other classifiers.

As we see from its name, Random Forest is a tree-based group where each tree depends on a combination of random variables. This means, for a *p*-dimensional random vector, $X = (X_1, ..., X_p)^T$ Representing the real-valued input or predictor variables, and a random variable Y is representing the real-valued response, if we suppose an unknown join distribution $P_{XY}(X, Y)$. The target is to detect and get a prediction function $f(X)$ for predicting Y. The prediction function is calculated by a loss function, $L(Y, f(X))$, and defined to minimize the expected value of the loss function [34, 35]. The final result of this system is drowned by ordinary majority vote, the decision function which is

$$f(x) = arg\,\max_{Y} \sum_{j=1}^{\infty} I(y = h_j(x)) \tag{4.1}$$

In our approach, we saw that Random Forest works much better than other algorithms. For example, we applied the Gaussian Naïve Bayes classifier on the cleaned tweets, but the accuracy results were much less than Random Forest.

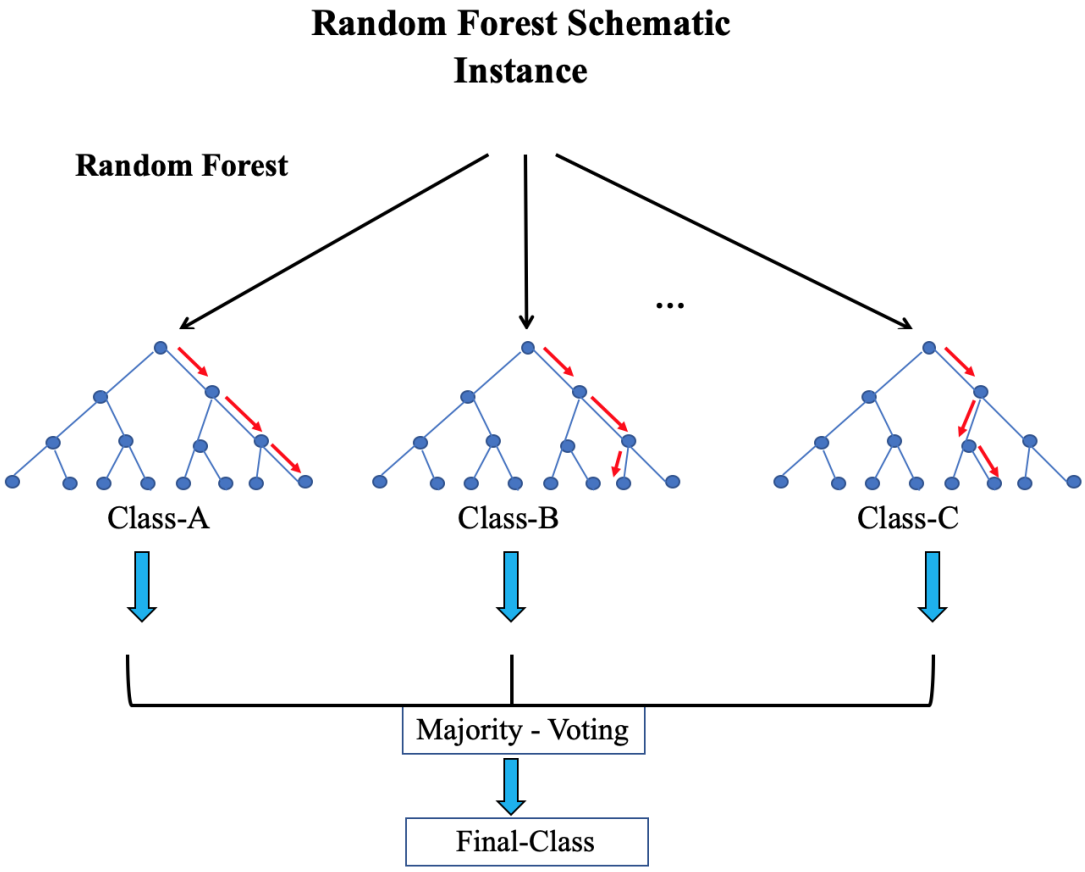

Figure 4.2. Random Forest schematic

The figure above illustrates the Random Forest schematics; as we can see it uses different types of paths (i.e. classes) for the instance, it crosses many trees in order to vote for the final class.

4.1.2. Support vector machine

As we mentioned earlier, this classifier algorithm has been introduced by Cortes and Vapnik with the name of Support Vector Networks (SVN) and later changed to SVM term. This predictive supervised ML tool is used in both classification and regression methods. It is trying to maximize predictive accuracy with avoiding overfitting to the data.

The target of the SVM classifier is to find a hyperplane in N-dimensional space (N the number of features) that separately categorize the data points using a hyperplane. This

Line is decision boundary that puts and divides the data points in 2-dimensional spaces. In another way, it is a line that divides the plane into two parts, where each class (group of data) lies in one side of the hyperplane.

The dimension of the hyperplane depends on the number of features. In the below table, the number of input feature with the output hyperplane is given [36],

Table 4.1. The number of input features with the required number of hyperplanes

Input features	Output hyperplane
2-features	1-line hyperplane
3-features	2-dimentional plane

In SVM, we are trying to find those points which are the closest to the line from both the classes, the points are called *support vectors*. Then, the distance between the line and the support vectors will be calculated, this distance is called the margin [37]. The goal of SVM is to maximize the margin, as it shown in below figure.

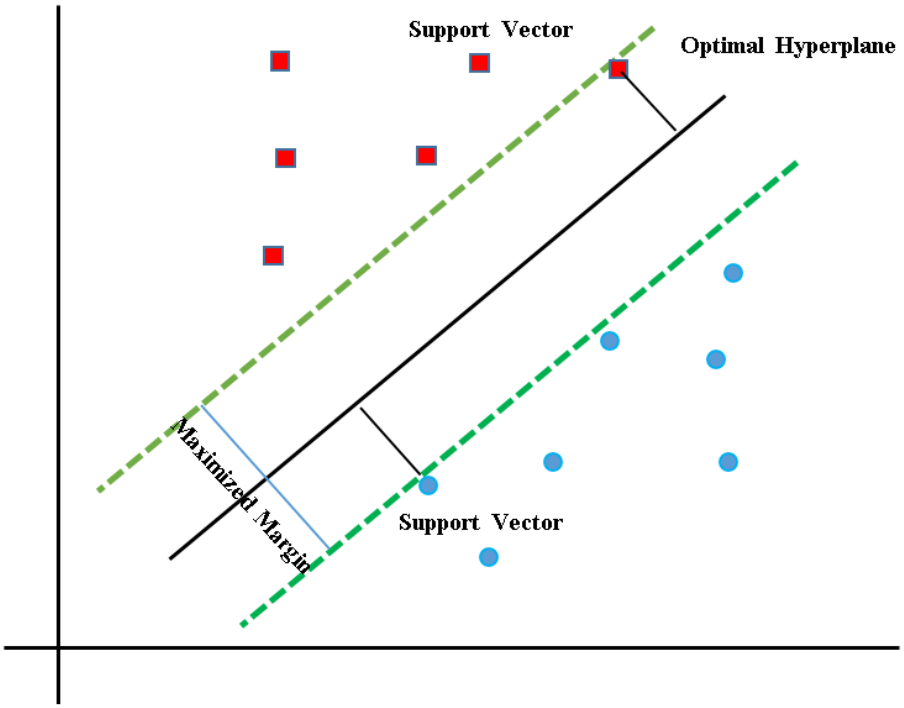

Figure 4.3. Optimal Margin

The advantages of using maximum margin, will be [37],

1. Having better classification performance in the experimental results.
2. To avoid and prevent the least chance of causing a misclassification, even if we have made a small error in the location of the boundary.

Here, the mathematical computation can get by the dot products between them,

$$a \cdot b = x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b \tag{4.2}$$

$$a \cdot b = x_a \cdot x_b + y_a \cdot y_b + (x_a \cdot x_b) \cdot (y_a \cdot y_b) \tag{4.3}$$

For example, when we want to try this algorithm for classifying text, first we have to change the text into a vector of numbers in order to run SVM on them. One example of this kind of transformation is Bag of Words (BoW) [38], we will talk about BoW in details, in the next sections.

The SVM have two tuning variables (i.e. parameters) which are C and Gamma. The C variable controls the trade-off between smooth decision boundary and classifying training points correctly. A C with bigger value means more training points, and this leads the need of more required time. In Linear SVM, the Gamma variable, has been neglected [39].

4.1.3. Gaussian naive bayes

Naive Bayes (NB) is a type of probabilistic and statistical supervised ML algorithm, which works based on Bayes theorem that calculates conditional probability. This type of algorithm provides a fast model building and scoring (faster than other two classifiers). It can be used in both binary and multi-class classification tasks. Naive Bayes is a stable algorithm, which means, a small changing in the training data will not make a big change in the model.

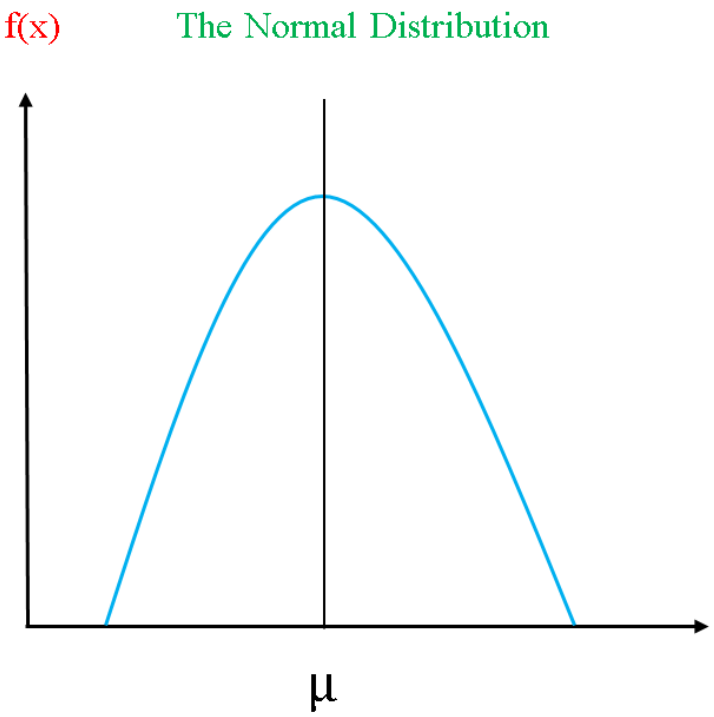

Figure 4.4. The Gaussian distribution

The Gaussian distribution equation is as follows:

$$P(x)\frac{1}{\delta\sqrt{2\pi}}exp\frac{(x-\mu)^2}{2\delta^2} \tag{4.4}$$

Where, μ is the mean and δ is the standard deviation. The value of μ and δ are computed as:

$$\mu = \frac{\sum_{i=1}^{n} xi}{n} \tag{4.5}$$

$$\delta^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n-1} \tag{4.6}$$

The conditional probability of A when B happened, is denoted by P(A|B) [40]. And Bayesian Theorem is:

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \quad (4.7)$$

Where,

P(A|B): The conditional probability that event A occurs, given that B has occurred.

p(A) and P(B): Probability of A and B without regard of each other.

p(B|A): The conditional probability that event B occurs, given that A has occurred.

CHAPTER 5. METHODOLOGY

In this chapter, which covers the most important part of our study, we will talk about the most required methods and algorithms that need to be applied to our dataset in order to get the target results. Actually, it is considered as the backbone of our study.

As we already mentioned above, the aim of our study is to apply sentiment analysis on Twitter's textual data and performing text polarities on it. At the final step, the Random Forest, GNB, and SVM classifiers have been used to train and test the data. The accuracies of the used classifiers are compared to see the best approach. The results showed the proposed cleaning method is working well. The details of the results will be given in a classification report in the result section.

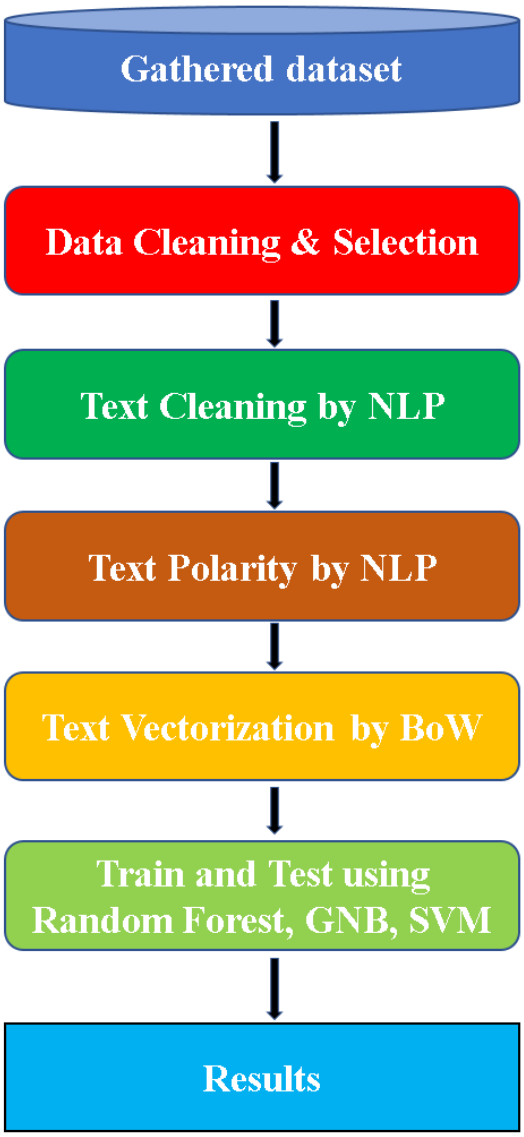

Figure 5.1. A quick overview of the study

The above diagram displays the most essential steps in our research. It starts with the collected tweets and applies the most required processes which is the data cleaning and selection level. In text cleaning which represents all the NLP techniques that have been used in order to prepare the text for conversion. At text polarity level, each cleaned tweet has been judged regarding their subjectivity. At text to number level, BoW has been applied for converting the categorical data (i.e. textual data) into numerical data. Finally, in the ML level, Random Forest classifier and has been used to train and test the data for getting the required results.

5.1. Gathered Dataset

Imagine all of that information (Hypertext transfer protocol) that is being put into the net every single minute. Collecting this information and organizing them in one single file, became a required demand for most of the developers. For this reason, most programming languages have their own libraries to deal with and support scraping, some of them are PHP, Python, etc. Recently Python became a beloved programming language for most of the developers. From data sciences to web programming, machine learning, it has been used.

5.2. Web Scraping

There are two big problems that face web scraping which are *Time* and *Security*. The first problem that is facing any web scraping techniques is the need of time. For example, if we want to scrape someone's tweets, we may need one or two year to save most of his posts. Why? Because if we want to apply machine learning classifiers, we need a lot of data to train our dataset, less data, will make our learning (i.e. Testing) less accurate. For example, in [41] they scrapped more than 500000 of users' feedbacks and application rate from Google Play Store. The gathered results have been measured with different ML classifiers such as the Random Forest classifier, Multinomial Naïve Bayes (MNB), and Logistic Regression. With the target of getting different parameters; including the accuracy, to evaluate the working and precision of the used algorithms.

In most of the websites, scraping their web pages is considered as hacking attacks. So, they prevent the developers from crawling their websites. Every website has their own policy which is stored in robots.txt file. This file tells the one who is going to crawl their website, which part of their website is allowed to be crawled and which part is forbidden. For example, in Twitter which is an American social networking service there is a special tool (Twitter API) for the developers, if they want to crawl their websites for analyzing purposes. It means you cannot scrape twitter by your own programming languages, if you do this, they will block your IP address.

For the two above mentioned problems, we used a prepared dataset in kaggle, which is called trump's tweets. The data set can be found in the cited link [42].

Table 5.1. The first five records of the saved dataset

index	id	link	content	date	retweets	favorites
0	1698308935	https://twitter.com/realDonaldTrump/status/169...	Be sure to tune in and watch Donald Trump on L...	2009-05-04 20:54:25	500	868
1	1701461182	https://twitter.com/realDonaldTrump/status/170...	Donald Trump will be appearing on The View tom...	2009-05-05 03:00:10	33	273
2	1737479987	https://twitter.com/realDonaldTrump/status/173...	Donald Trump reads Top Ten Financial Tips on L...	2009-05-08 15:38:08	12	18
3	1741160716	https://twitter.com/realDonaldTrump/status/174...	New Blog Post: Celebrity Apprentice Finale and...	2009-05-08 22:40:15	11	24
4	1773561338	https://twitter.com/realDonaldTrump/status/177...	"My persona will never be that of a wallflower...	2009-05-12 16:07:28	1399	1965

5.3. Data Cleaning And Selection

Sentiment analysis which is also well-known as opinion mining, deals with the use of natural language processing to recognize, extract, quantifying, and studying the subjective polarities. This procedure needs many pre-processing techniques that need to be prepared. In the next following sections, we are going to talk about them.

Whenever there is a dataset, there should be data analysis too, for the reason that any dataset needs some special commands to manipulate them. Data analysis performs most of the actions that need to be done on any dataset, including importing the dataset, performing most of the actions on its columns and rows, appending and deleting the records, and etc. Without data analysis, applying NLP and ML algorithms would be impossible. Deciding which features should be used and which one should be eliminated, is in this step of any studying research.

In the Python programming language, there are two important libraries for data analysis, and those are Pandas and NumPy. These two libraries contain all of the commands that are required for manipulating any dataset.

To bring these two libraries, simply write the following code in Anaconda Jupyter notebook:

```
import pandas as pd
import numpy as np
```

Table 5.2. The statistical information of our dataset

	id	retweets	favorites	geo
count	4.112200e+04	41122.000000	41122.000000	0.0
mean	6.088909e+17	5455.590657	22356.899105	NaN
std	3.027946e+17	10130.076661	41501.859711	NaN
min	1.698309e+09	0.000000	0.000000	NaN
25%	3.549428e+17	25.000000	28.000000	NaN
50%	5.609149e+17	291.000000	247.000000	NaN
75%	7.941218e+17	8778.000000	32970.750000	NaN
max	1.219077e+18	309892.000000	857678.000000	NaN

From our data analysis experimental results, we found that there are some features in our dataset that have the most NULL values. These NULL values do not have any effect on our results. So, we have dropped them out. See below table.

Table 5.3. NULL values in the dataset

id	link	content	date	retweets	favorites	mentions	hashtags	geo	dtype
0	0	0	0	0	0	18655	35312	41122	int64

5.3.1. Wordcloud

Data visualization (like graphs, charts, infographics, etc.) is giving a valuable way to represent the important information, but what if your raw data is text-based. The solution for this, is using a Wordcloud which is available in Python programing language. Wordcloud or Tagcloud refers to a figure like a cloud, fill up with many words in different shapes and sizes, the size of each of the word represents the frequency or the importance of each word; bigger size, means more repeated word. From the below figure, you will see the Wordcloud of our dataset for the feature of "content".

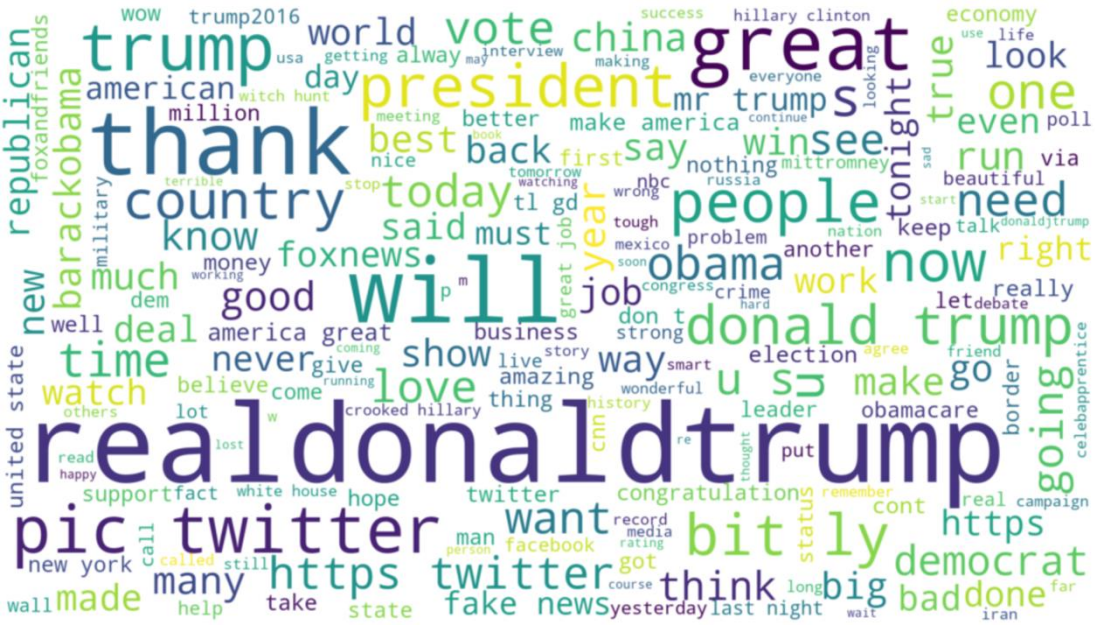

Figure 5.2. Wordcloud for Trump's tweet

5.4. Text Cleaning

The data (in our case Twitter texts) needs to be fully cleaned before applying any classifier algorithms. In every text, there are many (mentions, hashtags, emoticons, unconventional punctuation, spaces, symbols), that do not have any value on classifying, have to remove (filter out) them. One of the biggest advantages of this step is that it makes our data smaller which saves our storage capacity. This step is one of the most differences between our study and the previous studies in the same area. As

far as we know, from our searches, none of the previous work considered the size of the dataset. Decreasing the size of the hosted dataset can have a good effect on the performance of the work (Details have been given in result section).

Text cleaning includes the following steps:

1. Removing stopwords
2. Word lemmatizing
3. Regular Expression
4. Tokenization

5.4.1. Removing stopwords

This a list of words (or tokens) that are removed out before or after the NLP is processed. Stopwords are a list of English words which does not add any meaning to our sentences. Some of the *stopwords* are: 'a', 'an', 'the', 'and', 'but', 'if', ‘or’ ...etc.

Georg Wilhelm Friedrich Hegel (1770 - 1831) in his book, The Phenomenology of Spirit (dated 1807) puts these terms in the first stage of consciousness. The first stage is Sense-certainty which holds only the temporary terms like (*here*, *now*, and *this*). According to Hegel, Sense-certainty which is an immediate knowledge is the most abstract and the poorest knowledge.

To bring *stopwords*, simply write the following code in Anaconda Jupyter notebook,

```
from nltk.corpus import stopwords
from collections import Counter
def remove_stopword(text):
stop_words = stopwords.words('english')
stopwords_dict = Counter(stop_words)
text = ' '.join([word for word in text.split() if word not in stopwords_dict])
return text
```

This bunch of codes will download a list of English stopwords from The Natural Language Toolkit (NLTK) which includes libraries for statistical natural language processing of English writing, in the Python scripting language.

From our experimental result, the textual data in the column of "content", with index(0) has this statement: "*Be sure to tune in and watch Donald Trump on Late Night with David Letterman as he presents the Top Ten List tonight.*"

But after applying removing *stopwords* functions on our dataset, the statement becomes: "*sure tune watch donald trump late night david letterman presents top ten list tonight*".

We can see that; the following stopwords have been removed: {be, to, in, and, on, with, as, he, the}. This is because of their non-meaningful which does not participate in the whole meaning of our sentences.

5.4.2. Word lemmatizing

There is an essential difference between stemming and lemmatization. Stemming eliminates or stems the last few letters of a word, but most of the time, leads to inaccurate meanings or spelling. While lemmatization studies the words in its circumstances and converts them to their significant base forms, which is called Lemma. From the below examples, we will show the difference between them.

Table 5.4. The different outcomes of stemming and lemmatization

Input	Stemming	Lemmatization
Changing	Chang	Change
Changed	Chang	Change
Changes	Chang	Change
Change	Chang	Change
Changer	Chang	Change

As we can see from the table above, each method has the same output for different kinds of forms for the same word, but lemmatization provides a better and more meaningful word than stemming. For the reason above, we applied lemmatization process not stemming.

One thing is important to mention is that, we cannot use stemming and lemmatization procedures at the same time. Since the outcomes are somehow near in both methods; otherwise, it will give inconvenient results.

To bring *lemmatization*, simply write the following code in Anaconda Jupyter notebook:

```
from nltk.tokenize import word_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()
def lexicon_normalization(text):
words_lem = [lem.lemmatize(w) for w in text]
return words_lem
```

In the dataset, in the column of "content" with index (0), contains this tweet: "*sure tune watch donald trump late night david letterman presents top ten list tonight*",

But after applying word lemmatization function, we got the below sentence: "*sure tune watch donald trump late night david letterman present top ten list tonight*". Here, the word *presents* changes to its root which is *present*.

5.4.3. Regular expression

Regular expression is a set of letters or characters that builds a search sample (scheme). This technique uses a special class of rules in formal language, which is called a regular grammar. From the companies "Amazon Alexa" and "Google Now" are mainly engines with sample based, which depend on regular grammars.

Usually this procedure is used to catch a specific textual data. Every character in RE has a special meaning. For example, "**.**" is a very general character that matches all of the alphabetic letters, numbers (all in all, every character except a new line). In this link [43] you can find most rules of regular expression related to Python programming language.

Since in our study research we need a cleaned textual data, we have to remove all the metacharacters, emails, hashtags, links and the website domains. For example, in our practical research we used the below code that removes any *https* links in our dataset.

```
https?://(www\.)?(\w+)(\.\w+)
```

The code above will interpret that any text that matches the pattern above is a host domain and it has to be removed by the Python codes.

Another use of RE is eliminating emails which may be found in our dataset. The below code is removing any email that may be found in the tweet.

```
[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+
```

As we can see from the two texts above, our text had a domain link, but after applying RE pattern, the link was removed, and there were two (**:**)s, they were removed also, and the entire text changed into lowercase.

For example, in our dataset, in the column of "content", index(3), you can find the text below: "*New Blog Post: Celebrity Apprentice Finale and Lessons Learned Along the Way: http://tinyurl.com/qlux5e'* "

But after applying RE, the outcome of the text will be: "new blog post celebrity apprentice finale and lessons learned along the way".

5.4.4. Tokenization

Usually, pre-processing the original dataset decreases the size of the input the host documents. The techniques that are taking place and required in this step are, *stopwords* elimination, tokenization and stemming. From the mentioned procedures, the most crucial and main technique is the tokenization. Tokenization divides the textual data into individual tokens [44]. In order to clarify this step, an example will be given from the experimental result section.

The first record of index (0), after cleaning and tokenization will be as follows: [‘sure’, ‘tune’, ‘watch’, ‘donald’, ‘trump’, ‘late’, ‘night’, ‘david’, ‘letterman’, ‘present’, ‘top’, ‘ten’, ‘list’, ‘tonight’].

5.5. Detecting Text Polarity

This part is one of the main goals of our thesis. What we do from the beginning until the final step, is to prepare our text for subjective sentiment polarities (or in some resources, sentiment score).

Text polarity is a method to detect each tweet’s subjectivity. Since the tweets have been written by human as a subject, and he is tweeting his own ideas about a specific event or anything else, so the tweets are not objective. It needs to be detected in order to be classified into three levels, which are Positives, Negatives, and Neutrals.

In our experimental work, we judged in each sentence after being cleaned by NLP techniques. Each textual data (In our case is Twitter tweet) is labeled with three possible values: negative, positive or neutral. In this study, sentiment polarity of each tweet will be established by applying below measurement [45],

$$Sentiment\ Score\ (C) = \frac{Positive - Negative}{Positive + Negative + 2} \tag{5.1}$$

Where,

Positive represents total number of the positive words; and negative counts the negative words in the tweet. We represent it by a separate two valued with variable C, which represents the sentiment class:

$C \in \{-1, 1\}$.

Where,

C can hold three values, since of having different thresholds,

$$C = \begin{cases} 1\ (Positive) & if\ Sentiment\ score \geq 0.1 \\ -1\ (Negative) & if\ Sentiment\ score < 0.1 \\ 0\ (Neutral) & if Sentiment\ score = 0.0 \end{cases} \quad (5.2)$$

In the Python programming language, there are two libraries for text polarity, which are TextBlob and VADER. To bring TextBlob, simply write the following code in Anaconda Jupyter notebook,

from textblob import TextBlob.

And then write,

```
def get_tweet_sentiment(tweet):
analysis = TextBlob(tweet)
if analysis.sentiment.polarity > 0:
return '+1'
elif analysis.sentiment.polarity == 0:
return '0'
else:
return '1'
```

The above code will assign sentiment subjective polarities for all the cleaned tweets in the dataset. Unlike most of the previous work in this area, in which they neglect neutral; our work puts neutral texts in its own category.

From our experiment result, we saw that from the total of 41122 records, the distribution of sentiment polarities will be as follows:

Table 5.5. Sentiment polarities of the dataset

Sentiment polarities	number of occurrences	Percentages
Positive	22274	54.16%
Negative	7148	17.38%
Neutral	11700	28.45%
Total	41122	100%

As we can see from the result table above, most of the tweets are positive, and we got the least number of negative tweets, and about 11700 neutral tweets. The graphical distribution of the tweets will be as follows,

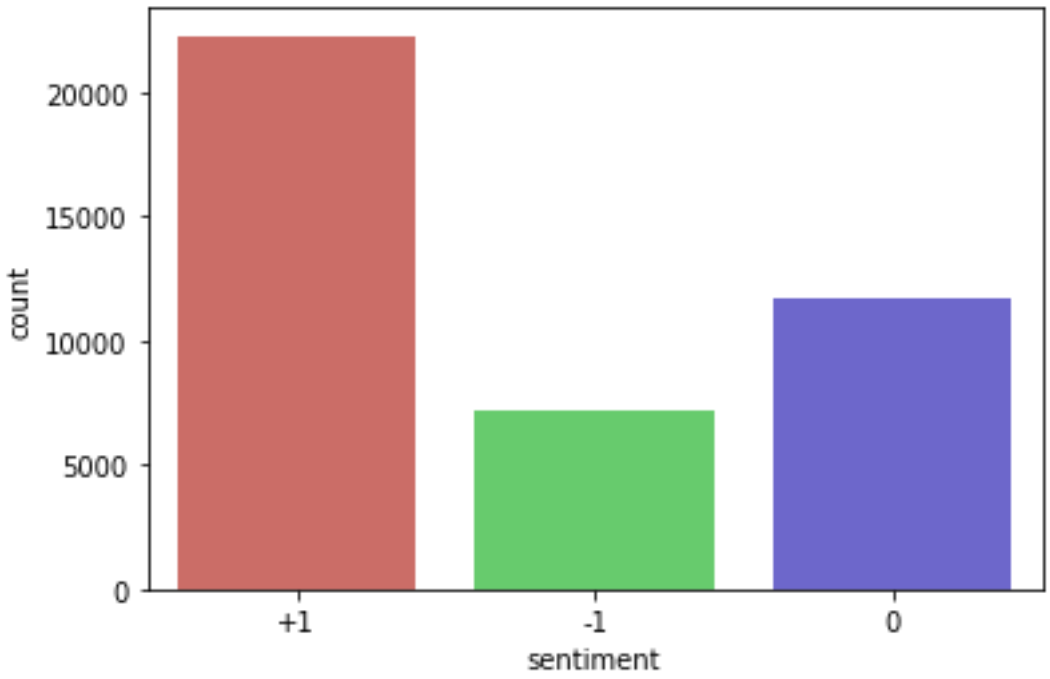

Figure 5.3. Sentiment polarity distribution of the tweets

There is another text classification which more detailed, and is that the tweets can be classified into 7 levels of subjective polarities in TTD with 0.25 degrees for each;

starting from negatives (strong, moderate, weak); neutral; positive (strong, moderate, weak). In the table below, we will put the threshold of each polarities.

Table 5.6. Sentiment analysis for 7 classes

Sentiment Class	Range of threshold
Strong negative	$-1.00 \leq VS < -0.75$
Moderate negative	$-0.75 \leq VS < -0.50$
Weak negative	$-0.50 \leq VS < -0.25$
Neutral	$-0.25 \leq VS < 0.25$
Weak positive	$0.25 \leq VS < 0.50$
Moderate positive	$0.25 \leq VS < 0.50$
Strong positive	$0.75 \leq VS \leq 1.00$

But since our target is only recognizing only three levels polarities, we didn't apply this much of details. It can be a suggestion for the upcoming studies to do it (i.e. future works in the same area).

5.6. Text Vectorizations

Since all of the machine learning classifiers are dealing with numbers only, we need to change our cleaned text into a matrix of numbers, and the field will be ready for training and test process. Text vectorization is a technique of changing texts into quantitative data. The most popular types of text vectorization are:

1. Binary Term Frequency
2. Bag of Words (BoW) Term Frequency
3. (L1) Normalized Term Frequency
4. (L2) Normalized TF-IDF
5. Word2Vec

5.6.1. Bag of words

Simply, BoW is a method for representing text in the form of numbers. This model is used for simplifying representation which is used in NLP and information retrieval (IR). In this method, a list of all the text will be considered as a bag of its words, with ignoring the grammar and even the order of the words, but protecting its varieties.

Simply, BoW is a link between natural language processing and the machine learning classifier. It connects NLP techniques to ML.

Figure 5.5. BoW as a link between NLP and ML

In order to clarify BoW concept, in the example below, let's take three sentences:

Sentence 1: "*The wolf sat*"
Sentence 2: "*The wolf sat on the hill*"
Sentence 3: "*The wolf with the hill*"

We will construct a vector form, from all the unique words in the above three sentences. This vector contains six words which are: 'The', wolf', sat, 'on', 'hill', 'with'. Finally, we will make a table for the results,

Table 5.7. The BoW table

	the	wolf	sat	on	hill	with
Sentence 1	1	1	1	0	0	0
Sentence 2	2	1	1	1	1	0
Sentence 3	2	1	0	0	1	1

Vector 1: [1, 1, 1, 0, 0, 0]

Vector 2: [2, 1, 1, 1, 1, 0]

Vector 3: [2, 1, 0, 0, 1, 1]

So, the sentence similarities between the vectors will be as follows:

Vector 1: [1, 1, 1, 0, 0, 0]

× × × × × ×

Vector 2: [2, 1, 1, 1, 1, 0]

2+1+1+0+0+0 = 4

We can gather everything together in one table,

Table 5.8. Sentence similarities

Vectors			Similarities		
Vector 1	Vector 2	Vector 3	V1 × V2	V1 × V3	V2 × V3
1	2	2	2	2	4
1	1	1	1	1	1
1	1	0	1	0	0
0	1	0	0	0	0
0	1	1	0	0	1
0	0	1	0	0	0
Sum of Similarities for each			4	3	5

From the results above, we can say that the similarities between vector 2 and vector 3 are the highest similarities which are 5; while, the similarities between vector 1 and vector 2 are 4; and, the similarities between vector 1 and vector 3 is 3 which are the lowest similarities.

5.7. Splitting Into Train And Test

Every used machine learning algorithm needs to a technique which is a kind of division. In this process, the whole dataset will be divided into two parts: Training and Testing. It's up to the researcher divides each part into how many percentages. It can 80%, 20% for both training and testing respectively. Also 70%, and 30%. This method is used to evaluating the performance of the used machine learning algorithm. As we said earlier, this process requires dividing the dataset into two subsets:

Training set: Used to fit and train the model,
Testing set: Used to evaluate and fit the model.

This technique is an important step in any supervised learnings. While the agent does not have any default information about the environment, this procedure gives that ability to the agent to learn from the experiences by training more than half of the data. In most of the cases, 70% of the dataset is given to the agent in order to learn from the training; and the remaining part which is 30% is put for the test to see the accuracy of the used classifier, in order to check whether it works good or not. In case if the suggested ML algorithm is not doing well, another classifier has to be applied on the hosted data. The figure below will explain the procedure of splitting in ML.

In our case, the total records of the dataset equal 41122 records. The mathematical calculating of the splitting method for Random Forest classifier will be as follows:

Training set: $70\% \times 41122 = \frac{70}{100} \times 41122 = 0.7 \times 41122 = 28785$ records

Testing set: $30\% \times 41122 = \frac{30}{100} \times 41122 = 0.3 \times 41122 = 12337$ records

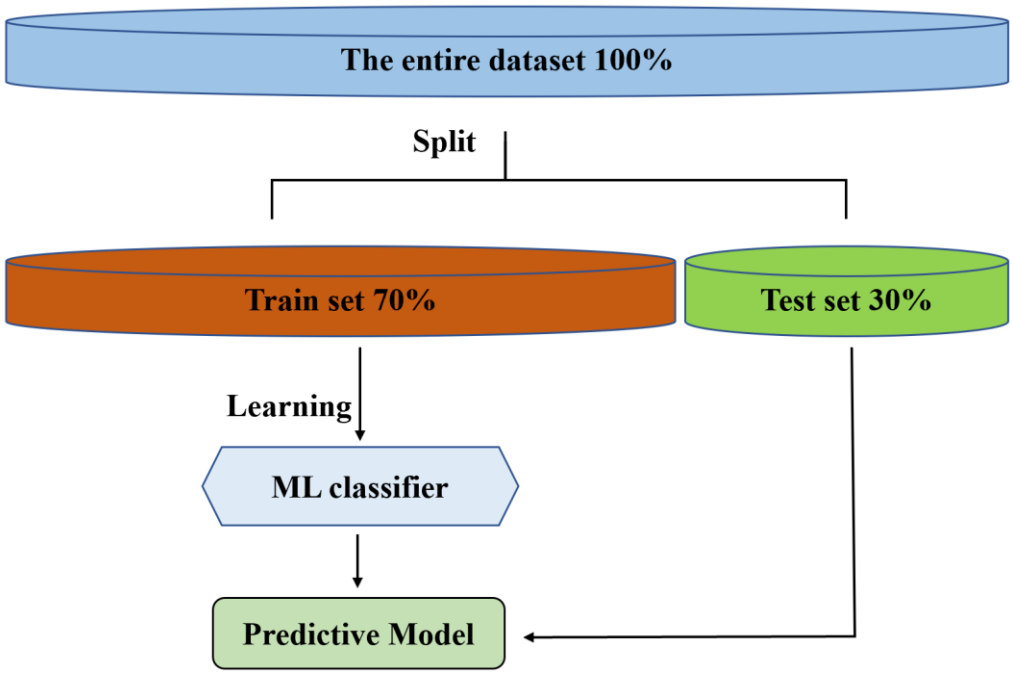

Figure 5.5. The procedure of splitting dataset into training test and testing set in ML

CHAPTER 6. RESULTS AND DISCUSSION

This section has been divided into two part. In the first part (6.1); the criterion for performance evaluation of each rule has been given, and examples have been given in order to clarify each used formula. In the second part (6.2), the experimental results have been discussed and the output of the confusion matrix, classification report have been evaluated, with checking the size of the dataset before and after cleaning.

6.1. The Criterion For Performance Evaluation

Every research study in the field of machine learning can be evaluated on its performance results. This section is specified for *Confusion matrix, and Classification report* which give the outcomes of the classification's method, in order to see whether the suggested model fits the actual data well or not. The metrics measurements hold the following results:

1. Accuracy,
2. Precision,
3. Recall metrics,
4. F1 score.
5. Training and predicting time
6. Data size

In the upcoming suctions, we will discuss all of those terms regarding their mathematical formulas, and at the end, we will put our practical results.

Before talking about the mentioned terms, we need to talk a little bit about the rules of *Logic*. Simply, Logic is the study of truth-preserving arguments [46], which is the

study of what considers as a good reason for question of what, and why [47]. Here, every judgment can be categorized into four states:

1. True Negative (TN): correct prediction
2. True Positive (TP): correct prediction
3. False Positive (FP): incorrect prediction
4. False Negative (FN): incorrect prediction

The above four sentences, can be summarized in easier way like the table below,

Table 6.1. Confusion matrix standards

	Actual True/ False	
Predicted Positive/ Negative	True Positive False Negative	False Positive True Negative

From the above table, there are two types of error:

Error type 1: False Positive; since it has only one false (F).
Error type 2: False Negative; since it has two falses (F, F).

A famous example for *Confusion matrix* is pregnancy test result for two patients; a man and a woman,

TP for the woman: You are pregnant. Her pregnancy test result predicted positive, and she actually is.
FP for the man: You are pregnant. His pregnancy test result predicted positive, and he actually is not.
FN for the woman: You are not pregnant. Her pregnancy test result predicted negative, while she is actually pregnant.
TN for the man: You are not pregnant. His pregnancy test result predicted negative, while he is actually not pregnant.

As has been said earlier, classification report is a table that holds the most essential results regarding the performance of the machine learning algorithm. Classification report contain the following results for the suggested classifiers: [48, 49]

1. Accuracy
2. Precision
3. Recall
4. F1 Score

In the following sections, we will give brief information about each one of them, and we will give each one's formula with an example. Finally, we will give our experimental results.

6.1.2. Accuracy

Accuracy is calculated as the total number of correct predictions, over the total number of the dataset (i.e. all correct / all). Accuracy work well when our data is balanced. The rule of accuracy is,

$$Accuracy = \frac{True\ Positive\ (TP) + TrueNegative\ (TN)}{All\ (TP + TN + FN + FP)} \quad (6.1)$$

a metric measurement for evaluating your classification models. It shows how good the model is working on your data. In our work, with the used classifier (which is Random Forest) we approached a great result with accuracy of 88% which is a good approach.

6.1.3. Precision

The rule of precision is,

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (6.2)$$

Precision works on the vertical lines (i.e. the columns) of our table.

6.1.4. Recall

The rule of recall is,

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (6.3)$$

Recall works on the horizontal lines (i.e. the rows) of our table.

6.1.5. F1 score

The solution for the misleading performance of accuracy on imbalanced data, is F1 score. We use F1 score when our data is imbalanced. F1 score is the average of precision and recall. We can decide whether an ML classifier works well or not on the imbalanced data, by looking at accuracy and f1 score. A good approach is obtaining when the nearer to each other they are; which means our model is working well on the given data.

The rule of F1 score is,

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6.4)$$

In order to understand the above rules, we will take an example.

Consider a classification system that has been trained to classify (or recognize) the pictures of three types of animals: phoenix, owl, and wolf. The system gave the results in a confusion matrix. Assume that the number of animals are given to the system which are 30 animals; there where 10 phoenixes, 8 owls, 12 wolves [50].

Table 6.2. Confusion matrix for three samples of animals

		Predicted class			Total no. of each
		phoenix	owl	wolf	
Actual class	phoenix	6	3	1	10
	owl	1	5	2	8
	wolf	0	1	11	12

In this confusion matrix 3 by 3 table, out of 10 actual phoenixes, the system predicted that 3 were owls, and 1 was a wolf; and of the 8 owls, it predicted 1 was a phoenix, and 2 were wolves; and out of 12 wolves, predicted 1 was owls. The green colors are the true actual values for each class.

Considering the confusion matrix above, the corresponding table of confusion which is (Table 6.2), for the phoenix, owl, and wolf classes, would be as follows:

Table 6.3. Confusion matrix for phoenix class

6 true positive (actual phoenixes that were correctly classified as phoenixes)	1 false positive (owls that were incorrectly labeled as phoenixes)
3, 1 false negative (phoenixes that were incorrectly marked as owls, and wolf respectively)	19 true negative (all the remaining animals, correctly classified as non-phoenixes)

Table 6.4. Confusion matrix for owl class

5 true positive (actual owls that were correctly classified as an owl)	3, 1 false positive (phoenixes and wolf that were incorrectly labeled as owl respectively)
1, 2 false negative (owls that were incorrectly marked as phoenix and wolf respectively)	18 true negative (all the remaining animals, correctly classified as non-owls)

Table 6.5. Confusion matrix for wolf class

11 true positive (actual wolves that were correctly classified as wolves)	1, 2 false positive (phoenix and owls that were incorrectly labeled as wolves)
1 false negative (wolf that was incorrectly marked as owls)	15 true negative (all the remaining animals, correctly classified as non-phoenixes)

The results of (Table 6.2) will be as follows,

$$Overall\ accuracy = \frac{(5+6+11)}{30} = \frac{22}{30} = 0.73 \times 100\% = 73\%$$

$$Precision\ for\ phoenix\ class = \frac{6}{(6+1+0)} = \frac{6}{7} = 0.85 \times 100\% = 85\%$$

$$Precision\ for\ owl\ class = \frac{5}{(5+3+1)} = \frac{5}{9} = 0.55 \times 100\% = 55\%$$

$$Precision\ for\ wolf\ class = \frac{11}{(11+2+1)} = \frac{11}{14} = 0.78 \times 100\% = 78\%$$

$$Overall\ average\ precision = \frac{(0.85+0.55+0.78)}{3} = \frac{2.18}{3} = 0.72 \times 100\% = 72\%$$

$$Recall\ for\ phoenix\ class = \frac{6}{(6+3+1)} = \frac{6}{10} = 0.6 \times 100\% = 60\%$$

$$Recall\ for\ owl\ class = \frac{5}{(5+2+1)} = \frac{5}{8} = 0.62 \times 100\% = 62\%$$

$$Recall\ for\ wolf\ class = \frac{11}{(11+1+0)} = \frac{11}{12} = 0.91 \times 100\% = 91\%$$

$$Overall\ average\ recall = \frac{(0.60+0.62+0.91)}{3} = \frac{2.13}{3} = 0.71 \times 100\% = 71\%$$

$$F1\ score = 2 \times \frac{(0.72 \times 0.71)}{(0.72+0.71)} = 2 \times \frac{0.5112}{1.43} = 2 \times 0.357 = 0.71 \times 100\% = 71\%$$

6.2. Evaluating The Experimental Results

For our proposed system, we used macOS operating system; with the following hardware technical specification:

Table 6.6. The technical specifications of the used computer

Computer manufacture	Type of OS	Processor	Amount of RAM
MacBook Pro	macOS, Big Sur, 2020	Intel Core i5, ~2.6 GHz	8GB

From the tables below, all the results from the classification report for the three algorithms will be as follows:

1. The classification report for GNB is,

Table 6.7. Classification report for Gaussian naïve bayes classifier

Types of polarity	Precision	Recall	F1-score
Positive	0.95	0.60	0.74
Negative	0.54	0.73	0.62
Neutral	0.65	0.96	0.78
Accuracy	72%		

2. The classification report for SVM is,

Table 6.8. Classification report for SVM classifier

Types of polarity	Precision	Recall	F1-score
Positive	0.96	0.91	0.93
Negative	0.90	0.73	0.80
Neutral	0.81	0.98	0.88
Accuracy	89%		

3. The classification report for Random Forest is,

Table 6.9. Classification report for Random Forest classifier

Types of polarity	Precision	Recall	F1-score
Positive	0.93	0.91	0.92
Negative	0.88	0.67	0.76
Neutral	0.81	0.95	0.87
Accuracy	88%		

As it can be sees from the tables above, the overall accuracy result is approximately 88%; and other results will be as follows:

From the results above, it can be noticed from all of the classifiers work on both positive and neutral polarities better than negative tweets, this due to the total number of negative tweets which lesser that other polarities. As it's given in (Table 5.2) the total number of the whole records are 41122 tweets; from this number, 33974 tweets (which is about 83% of the dataset) are both positive and neutral, only 7148 tweets are negative; which is equal to,

$$\frac{7148}{41122} \times 100\% = 0.1738 \times 100\% = 17.38\%$$

17% of the whole dataset; that's why we notice from the classification report table, the proposed algorithms are not working well on negative polarities. Since the agent does not have any previous knowledge about the dataset, it has to be trained a lot, the more data for training, the more accuracy results will get.

One last comment about the classification report tables; in some cases, we may face imbalanced data, which means the data in the host dataset are not coherent. Due to this reason, measuring the accuracy alone is not enough. It has to be compared with the result of f1-score. If their results are near to each other, it means it performs well. In the case of Random Forest, for positive and neutral tweets, the results of F1 score are

92% and 87% respectively; with regarding to overall accuracy result which is 88% it means they are near in each other.

Also, to obtain and test the performance of the suggested classifier, the overall accuracy of Random Forest classifier with Gaussian Naive Bayes and SVM have been compared as it shows in the table below,

Table 6.10. The accuracy comparison between the classifiers with their required time

ML Classifier	Accuracy	Required time	
		Training	Prediction
Random Forest	88%	7min 5s	2.12 s
GaussianNB	72%	1.64 s	367 ms
SVM	89%	15min 37s	1min 42s

The result shows 88%, 72%, and 89% respectively. From the comparison, it seems that the Ransom forest classier works better than GNB. So, the authors are suggesting Random Forest over Gaussian Naive Bayes, but in the case of Random Forest with SVM, we noticed one-degree difference in their inaccuracies, but SVM has the problem of elapsing time. The table above which contains the required time for each classifier, on our macOS system (the hardware specification is given in table 6.6). The results show that GNB needs the least amount of time, while SVM needs the most amount of time, which is a huge difference from the two other classifiers.

Another way for testing the proposed system is to check the size of the dataset before and after cleaning processes. Reducing the size of the data, means the used cleaning method has worked well on the dataset; also, it causes and essential impact on the performance of the work, the less and more cleaned data, means the faster system is.

Thus, cleaning-out the noisy, or wrong samples in the original training dataset; is a very important step for the training dataset methods in enhancing the classification accuracy.

In the macOS system is, the differences can be noticed between both before and after CSV files,

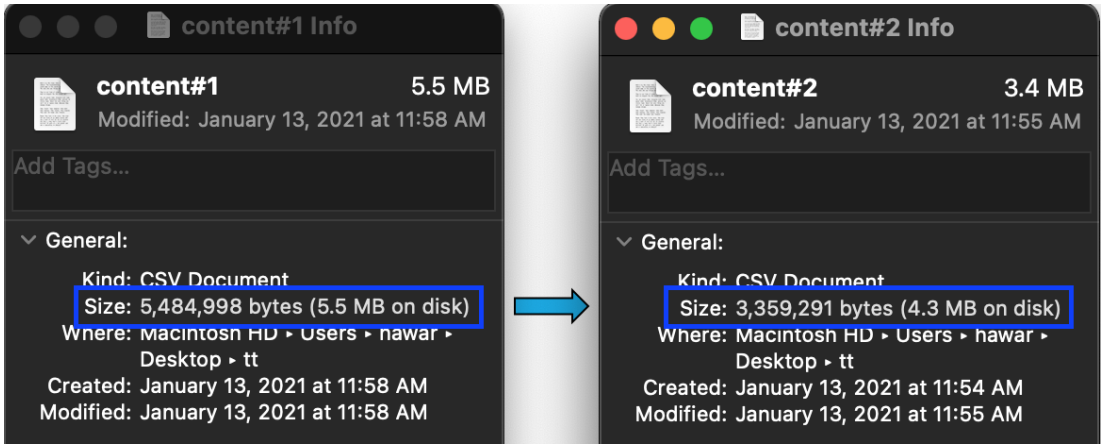

Figure 6.1. Both saved “content” with “cleaned content” features in macOS

One of the benefits of cleaning textual data is reducing capacity. We succeeded to decrease the size of our dataset about 1.2 MB (from 5.5 MB to 4.3 MB) for macOS. To see the results more understandable, from the chart below, the size of both tweets (before and after cleaning) for 20 records have been put,

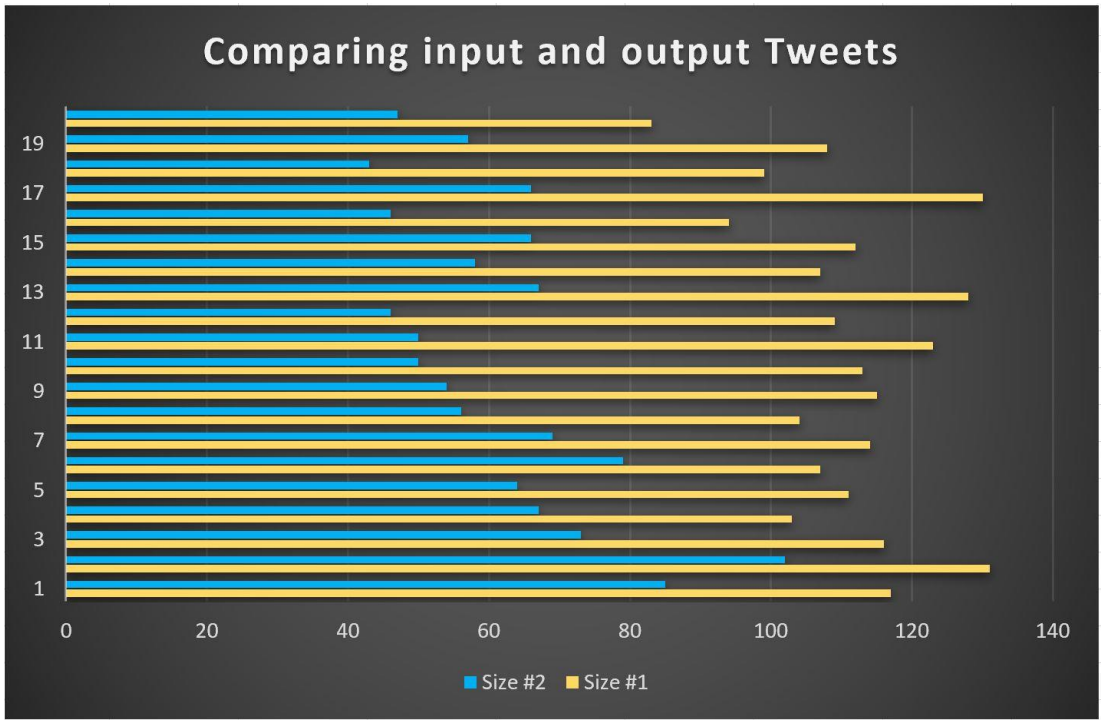

Figure 6.2. Comparing the sizes of In/Out texts

It can be noticed, the size of record number1 in terms of (length of its words) equals 117, but after cleaning, it became 85.

This decreasing of size leads to apply the ML algorithms faster. For example, in the case of Random Forest, the training and testing technique on the macOS took about 7min. This is useful in those situations that have a dataset with a huge amount of information and a small with a determined amount of capacity.

CHAPTER 7. CONCLUSION

In the proposed research study, sentiment analysis as the use of use NLP and ML classifiers have been applied on Trump's tweet dataset. After data preparation, the most important sentiment analysis procedures have been applied on the host dataset, like cleaning the dataset in order to be ready for text vectorization. Cleaning the dataset, which includes removing stopwords, word lemmatization, regular expression and tokenization. We succeeded in reducing the size of "content" feature with the target of taking fewer capacity. With the rapid growth of social media networks, hateful activities have become a phenomenon in this platform, this became a challenged task to know the subjective polarities of each one's tweet; therefore, we judged each sentence regarding their polarities whether they are *positive*, *negative* or *neutral*. At the end, Random Forest classifier with both Gaussian Naive Bayes and SVM have been compared. Other related results to the confusion matrix and classification report have been given in the result and dissection section.

For the future studies, detecting text polarities can be classified into 7 levels (strong, moderate, weak) each with 0.25 degrees of threshold. Also, we recommend the same system for not only on texts, but for speech recognition for cleaning noisy data in practical AI and Robotics. The proposed method can be used in AI industries, and applied linguistics.

REFERENCES

[1] Duncombe, C., 2019. The politics of Twitter: emotions and the power of social media. International Political Sociology, 13(4), pp.409-429.

[2] Akram, W. and Kumar, R., 2017. A study on positive and negative effects of social media on society. International Journal of Computer Sciences and Engineering, 5(10), pp.347-354.

[3] Ajjoub, C., Walker, T. and Zhao, Y., 2020. Social media posts and stock returns: The Trump factor. International Journal of Managerial Finance.

[4] Kaplan, A.M. and Haenlein, M., 2010. Users of the world, unite! The challenges and opportunities of Social Media. Business horizons, 53(1), pp.59-68.

[5] https://socialblade.com/twitter/user/realdonaldtrump (Access Date: Dec. 7, 2020).

[6] Wells, C., Shah, D., Lukito, J., Pelled, A., Pevehouse, J.C. and Yang, J., 2020. Trump, Twitter, and news media responsiveness: A media systems approach. New Media & Society, 22(4), pp.659-682.

[7] Clarke, I. and Grieve, J., 2019. Stylistic variation on the Donald Trump Twitter account: A linguistic analysis of tweets posted between 2009 and 2018. PloS one, 14(9), p.e0222062.

[8] Yaqub, U., Chun, S.A., Atluri, V. and Vaidya, J., 2017. Analysis of political discourse on twitter in the context of the 2016 US presidential elections. Government Information Quarterly, 34(4), pp.613-626.

[9] Deshmukh, K.V. and Shiravale, S.S., Ambiguity Resolution in English Language for Sentiment Analysis. In 2018 IEEE Punecon (pp. 1-6). IEEE.

[10] Verma, M.T.S.R., 2018. Natural Language Processing (Nlp): A Comprehensive Study.

[11] Vasiliev, Y., 2020. Natural Language Processing with Python and SpaCy: A Practical Introduction. No Starch Press.

[12] Müller, A.C. and Guido, S., 2016. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

[13] Ho, T.K., 1995, August. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282). IEEE.

[14] Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

[15] https://scikit-learn.org/stable/modules/naive_bayes.html (Access Date: May 2, 2021).

[16] Syafie, L., Indra, D., Hamrul, H., Anraeni, S. and Ilmawan, L.B., 2018, November. Comparison of Artificial Neural Network and Gaussian Naïve Bayes in Recognition of Hand-Writing Number. In 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT) (pp. 276-279). IEEE.

[17] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

[18] Elbagir, S. and Yang, J., 2019. Twitter sentiment analysis using natural language toolkit and VADER sentiment. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 122, p. 16).

[19] Li, I., Li, Y., Li, T., Alvarez-Napagao, S., Garcia-Gasulla, D. and Suzumura, T., 2020, December. What Are We Depressed About When We Talk About COVID-19: Mental Health Analysis on Tweets Using Natural Language Processing. In International Conference on Innovative Techniques and Applications of Artificial Intelligence (pp. 358-370). Springer, Cham.

[20] Al-Makhadmeh, Z. and Tolba, A., 2020. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing, 102(2), pp.501-522.

[21] Vishwakarma, D.K., Varshney, D. and Yadav, A., 2019. Detection and veracity analysis of fake news via scrapping and authenticating the web search. Cognitive Systems Research, 58, pp.217-229.

[22] Back, B.H. and Ha, I.K., 2019. Comparison of sentiment analysis from large Twitter datasets by Naïve Bayes and natural language processing methods. Journal of information and communication convergence engineering, 17(4), pp.239-245.

[23] Fitri, V.A., Andreswari, R. and Hasibuan, M.A., 2019. Sentiment analysis of social media twitter with case of anti-lgbt campaign in indonesia using naïve bayes, decision tree, and random forest algorithm. Procedia Computer Science, 161, pp.765-772.

[24] Yaqub, U., 2020. Tweeting During the Covid-19 Pandemic: Sentiment Analysis of Twitter Messages by President Trump. Digital Government: Research and Practice, 2(1), pp.1-7.

[25] Sahu, K., Bai, Y. and Choi, Y., 2020, January. Supervised Sentiment Analysis of Twitter Handle of President Trump with Data Visualization Technique. In 2020 10th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0640-0646). IEEE.

[26] Huang, X., King, I., Raghavan, V. and Rüger, S., 2010. Proceedings: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology—Workshops (Vol. 3). IEEE Computer Society.

[27] Oh, C. and Kumar, S., 2017. How trump won: the role of social media sentiment in political elections. In Pacific Asia Conference on Information Systems (PACIS). Association for Information Systems.

[28] Ebrahimi, J., Dou, D. and Lowd, D., 2016, November. Weakly supervised tweet stance classification by relational bootstrapping. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 1012-1017).

[29] Kinyua, J.K., Mutigwe, C., Cushing, D.J. and Poggi, M., 2021. An analysis of the impact of President Trump's tweets on the DJIA and S&P 500 using machine learning and sentiment analysis. Journal of Behavioral and Experimental Finance, 29, p.100447.

[30] Hapke, H.M., Lane, H. and Howard, C., 2019. Natural language processing in action.

[31] Khan, W., Daud, A., Nasir, J.A. and Amjad, T., 2016. A survey on the state-of-the-art machine learning models in the context of NLP. Kuwait journal of Science, 43(4).

[32] Géron, A., 2019. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.

[33] Müller, A.C. and Guido, S., 2016. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

[34] Cutler, A., Cutler, D.R. and Stevens, J.R., 2012. Random forests. In Ensemble machine learning (pp. 157-175). Springer, Boston, MA.

[35] Liu, Y., Wang, Y. and Zhang, J., 2012, September. New machine learning algorithm: Random forest. In International Conference on Information Computing and Applications (pp. 246-252). Springer, Berlin, Heidelberg.

[36] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (Access Date: May 1, 2021).

[37] Jakkula, V., 2006. Tutorial on support vector machine (svm). School of EECS, Washington State University, 37.

[38] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/ (Access Date: May 5, 2021).

[39] https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989 (Access Date: May 24, 2021).

[40] https://datacadamia.com/data_mining/naive_bayes (Access Date: May 24, 2021).

[41] Farhan, M., Latif, R.M.A., Qureshi, A.A., Alruily, M., Bajahzar, A. and Aldabbas, H., 2020. Google Play Content Scraping and Knowledge Engineering using Natural Language Processing Techniques with the Analysis of User Reviews.

[42] https://www.kaggle.com/austinreese/trump-tweets (Access Date: Nov. 7, 2020).

[43] https://www.w3schools.com/python/python_regex.asp (Access Date: Aug. 10, 2020).

[44] Vijayarani, S. and Janani, R., 2016. Text mining: open source tokenization tools-an analysis. Advanced Computational Intelligence: An International Journal (ACII), 3(1), pp.37-47.

[45] Ruz, G.A., Henríquez, P.A. and Mascareño, A., 2020. Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. Future Generation Computer Systems, 106, pp.92-104.

[46] Cryan, D., Shatil, S. and Mayblin, B., 2014. Introducing Logic: a graphic guide. Icon Books Ltd.

[47] Priest, G., 2017. Logic: A very short introduction (Vol. 29). Oxford University Press.

[48] Kulkarni, A., Chong, D. and Batarseh, F.A., 2020. Foundations of data imbalance and solutions for a data democracy. In Data Democracy (pp. 83-106). Academic Press.

[49] Tharwat, A., 2020. Classification assessment methods. Applied Computing and Informatics.

[50] Patro, V.M. and Patra, M.R., 2015. A novel approach to compute confusion matrix for classification of n-class attributes with feature selection. Transactions on Machine Learning and Artificial Intelligence, 3(2), p.52.

RESUME

Name and surname : Hawar BARZENJI

EDUCATIONAL STATUS

Degree	Education Unit	Graduation Year
MSc	T.C. Sakarya University/ Department of Computer and Information Engineering	Ongoing
Undergraduate	Salahaddin University – Erbil/ Department of Computer Science	2011
High School	Sulaymaniyah, Iraq	2007

WORKING EXPERIENCE

Year	Location	Task
2020-Present	Sakarya University	Master Student
2012-2019	Sulaymaniyah, Iraq	Teacher

FOREIGN LANGUAGE

English, Arabic, Persian, Turkish(A1)

WORKS (articles, papers, projects, etc.)

1. Sentiment analysis of Twitter texts using Machine learning algorithms (Accepted article)

HOBBIES

Reading, Writing, Physical Exercises (Fitness and MMA)