

Emotion analysis of Tweets

**A Mini Project Report Submitted in Partial Fulfilment of the Requirement for the
Award of the Degree of**

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

PARDHU GUTTULA

19PA1A04C7

TANGELLAPALLI SAI RAMYA MANASA

19PA1A04F5

VEMULAPALLI BAVYA SRI

19PA1A04H3

THOTE PRASANTH

19PA1A04G1

PITANI NIRANJAN SAI

19PA1A04D2

Under the Esteemed Guidance of

Mr. D. Durga Prasad

Assistant Professor, ECE Department



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISHNU INSTITUTE OF TECHNOLOGY (Autonomous)

**(Accredited by NBA, NAAC, Approved by AICTE & Affiliated to JNTU Kakinada)
Vishnupur, Bhimavaram - 534202.**

2022-2023

VISHNU INSTITUTE OF TECHNOLOGY (Autonomous)

**(Accredited by NBA, NAAC, Approved by AICTE & Affiliated to JNTU
Kakinada) Vishnupur, Bhimavaram-534202**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



CERTIFICATE

This is to certify that the Mini Project entitled “**EMOTION ANALYSIS OF TWEETS**” is being submitted by **PARDHU GUTTULA (19PA1A04C7), TANGELLAPALLI SAI RAMYA MANASA (19PA1A04F5), VEMULAPALLI BAVYA SRI (19PA1A04H3), THOTE PRASANTH (19PA1A04G1), PITANI NIRANJAN SAI (19PA1A04D2)** in partial fulfilment for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering is a record of the Bonafide work carried out by them under my guidance and supervision during academic year 2022– 2023 and it has been found worthy of acceptance according to the requirements of the university.

Mr.D. Durga Prasad
Project Guide

Prof. K. Srinivas
Head of Department

External Examiner

ABSTRACT

In recent years, messages and social media has ended up being a very close representation of a person's life and his mental state. People are willing to share their thoughts, stories and their personal feelings, mental states, desires on social network sites, blogging platforms etc. This is a huge stockpile of data about a person's behavior and can be used for detection of various emotion states. Emotion Analysis, as the name suggests, it means to identify the view or emotion behind a situation. That is done by analyzing expressed opinions, images, sentiments, and other activities. In our case we took mainly posted opinions or tweets as input to implement our NLP model. This project is about constructing machine learning model using NLP (Natural Language Processing) to predict emotion of tweets. In this fast-pacing world as everyone is open in sharing their opinions on any particular product or any movie that they have watched in form of tweets or reviews. A person can easily judge a movie or product by just watching tweets or review. So, this project helps in finding whether the posted tweets create positive impact or negative impact. The leading goal deals with how opinion mining techniques can be accessed to analyze some of the tweets in many reports involving various types of tweet languages on Twitter and classify its polarity.

Keywords:

Twitter, sentiment, opinion mining, social media, natural language processing

CONTENTS

Topic	Page No
ABSTRACT	iii
List Of Figures	vi
List Of Tables	vii
1.INTRODUCTION	1
1.1 Twitter	2-3
1.2 Emotions	3
1.3 Emotion Analysis	4
1.4 Applications	5
1.5 Summary	5
1.6 Emotion Analysis	5
1.7 Problem Statement	6
1.8 Literature Survey	6-7
2.METHODOLOGY	8
2.1 Tools and Frameworks	9
2.2 Algorithms	10
2.3 F1-Score	11
2.4 Bag of words	11-13
2.5 TF-IDF	13-14
2.6 Tasks Required for analyzing emotion of tweets	15-20
2.7 Training Data	20
2.8 Testing Data	20
3. Results and Discussion	21
3.1 Output	22-23
3.2 Data Cleaning	23-27
3.3 Results – Models Accuracy	27-28

4. Conclusion	29-30
5. References	31
6. Appendix	32

List of Figures

Figure No	Figure Name	Page No
Fig 1.2.1	Various Emotions	3
Fig 1.3.1	Emotion Analysis	4
Fig 2.6.1	Flowchart of Emotion analysis of Tweets	15
Fig 3.1.1	Analyze the datasets by importing necessary libraries.	22
Fig 3.1.2	First ten rows from taken dataset.	22
Fig 3.1.3	The count of non-offensive tweets and offensive tweets.	23
Fig 3.2.1	The implementation of Porter Stemmer and creating a function to read and to remove twitter handles, punctuation, short words and stop words.	23
Fig 3.2.2	Creation of word cloud to generate an image of most representative Words.	24
Fig 3.2.3	Clear visualization of all tweets, good tweets, bad tweets.	25
Fig 3.2.4	The downloading package vader_lexicon.	25
Fig 3.2.5	The code for adding 4 columns of positive, negative, neutral and Compound to the dataset.	26
Fig 3.2.6	Classification of emotion of tweets by comparing compound score.	26
Fig 3.2.7	Frequencies of all labels.	26
Fig 3.2.8	Number of positive, negative, neutral tweets present in the given dataset.	27
Fig 3.3.1	The code for implementing SVM algorithm.	27
Fig 3.3.2	The code for random forest model and finding the F1 score of this model.	28

List of Tables

Table No	Table Name	Page No
Table 2.4.1	Bag of words table.	13
Table 2.5.1	Giving values to words in sentences.	14
Table 2.5.2	Calculating TF-IDF values of words.	14

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

1.1 Twitter

Twitter, a social networking site launched in 2006, is undoubtedly one of the most popular social media platforms available today, with 100 million daily active users and 500 million tweets sent daily. Twitter is incredibly easy to use. It can be used to receive news, follow high-profile celebrities, or stay in-touch with old high school friends. It began as an SMS-based platform, so the 140 character limit was initially simply a necessity -- mobile carriers imposed the limit, not Twitter. However, as Twitter grew to become a web platform, they kept the limit simply because it aligned with Twitter's brand -- Twitter is a platform that aims to create highly skimmable content for our tech-heavy, attention-deficit modern world. Twitter has grown exponentially over the past 10+ years. Its purpose is ultimately to spread information fast -- while that information is not always serious

Users broadcast short posts known as tweets. These tweets can contain text, videos, photos or links. Users choose what they want to see on Twitter by following other users and companies and searching topics. Generally, the timeline reflects the users' preferences, but they may see re-tweets from people they follow and promoted Tweets, which are paid advertisements. To help limit fake accounts, Twitter created the verified account symbol, which indicates the account is legitimate and belongs to the person or company. To get verified, Twitter confirms the identity of the individual or company. This helps maintain trust with users. The account must be associated with a popular brand or person and follow Twitter's criteria using an official website, ID or email address to be verified.

Twitter determines what is trending based on an algorithm and users' preferences, locations and interests. This algorithm also determines what is popular now and highlights emerging discussions and topics. In the search bar, users can type in a person, topic or keyword to search. There is also an #Explore function to search for keywords and trending topics. When users post a tweet, the messages are posted on their profile and then appear in followers' feeds. These tweets can also be searched on Twitter. Tweets include jokes, news, random thoughts and sharing articles; however, there is a restriction on length. Originally, Twitter limited tweet characters to 140. The limit is now 280 characters, which includes spaces and punctuation. To weave tweets into a conversation thread or connect them to a general topic, members can add hash tags to a keyword

in their post. The hash tag, which acts like a meta tag, is expressed as #keyword. This makes the tweet searchable under that keyword.

1.2 Emotions

Emotions are mental states brought on by neurophysiology changes, variously associated with thoughts, feelings, behavioral responses, and a degree of pleasure or displeasure. There is currently no scientific consensus on a definition. Emotions are often intertwined with mood, temperament, personality, disposition, or creativity.

Emotions involve different components, such as subjective experience, cognitive processes, expressive behavior, psycho physiological changes, and instrumental behavior. At one time, academics attempted to identify the emotion with one of the components: William James with a subjective experience, behaviorists with instrumental behavior, psycho physiologist with physiological changes, and so on. More recently, emotion is said to consist of all the components. The different components of emotion are categorized somewhat differently depending on the academic discipline. In psychology and philosophy, emotion typically includes a subjective, conscious experience characterized primarily by psycho physiological expressions, biological reactions, and mental states. A similar multi-componential description of emotion is found in sociology.

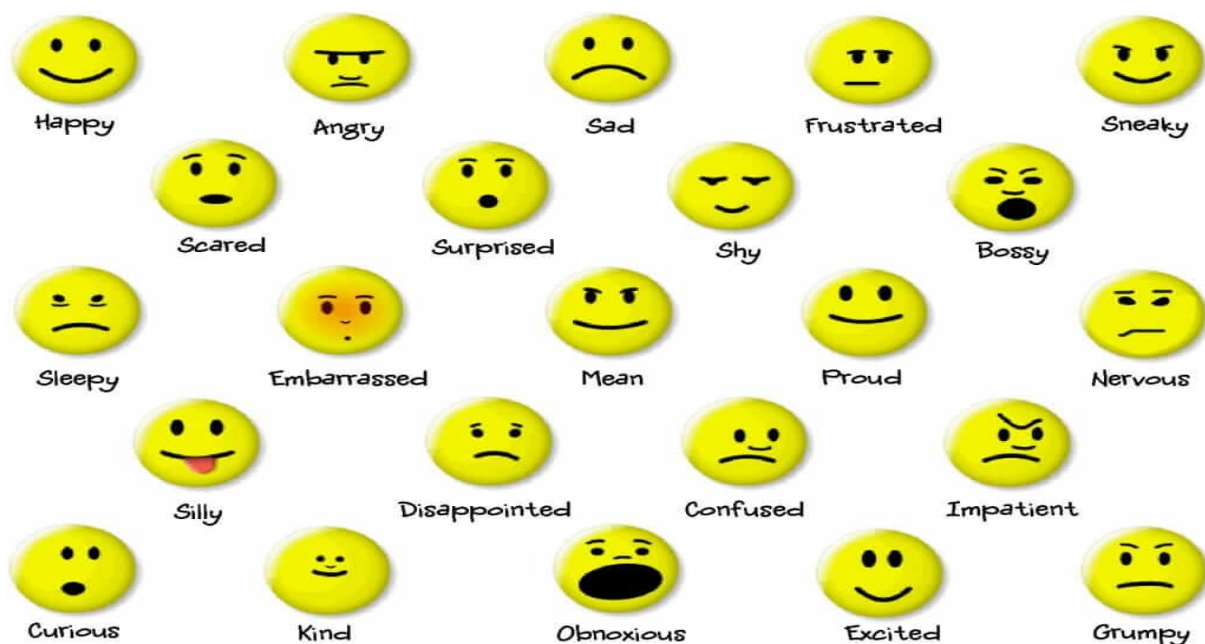


Fig.1.2.1 : Various Emotions

1.3 Emotion Analysis

Emotion analysis is the process of identifying and analyzing the underlying emotions expressed in textual data. Emotion analytics can extract the text data from multiple sources to analyze the subjective information and understand the emotions behind it. Advanced machine learning techniques can help you analyze the emotions expressed by the author in a piece of text. It can be easily done based on the types of feelings expressed in the text such as fear, anger, happiness, sadness, love, inspiring, or neutral.

In recent years, so-called emotional marketing has become a key factor of success for many business-to-consumer companies, especially global ones. Emotional marketing has the basic goal of convincing customers that a brand or a product is not just a brand or a product, but a kind of “friend.” Any emotional marketing strategy starts with an in-depth understanding of customers’ emotional or motivational drivers, and this is achieved either via open question surveys or by analysis of emotional reactions on social media. As mentioned already, sentiment and opinion analysis consists in “classifying a piece of text into positive versus negative classes,” while emotion analysis consists in “multi classifying” text into different types of categories (anger, disgust, fear, happiness, sadness, surprise, etc.).

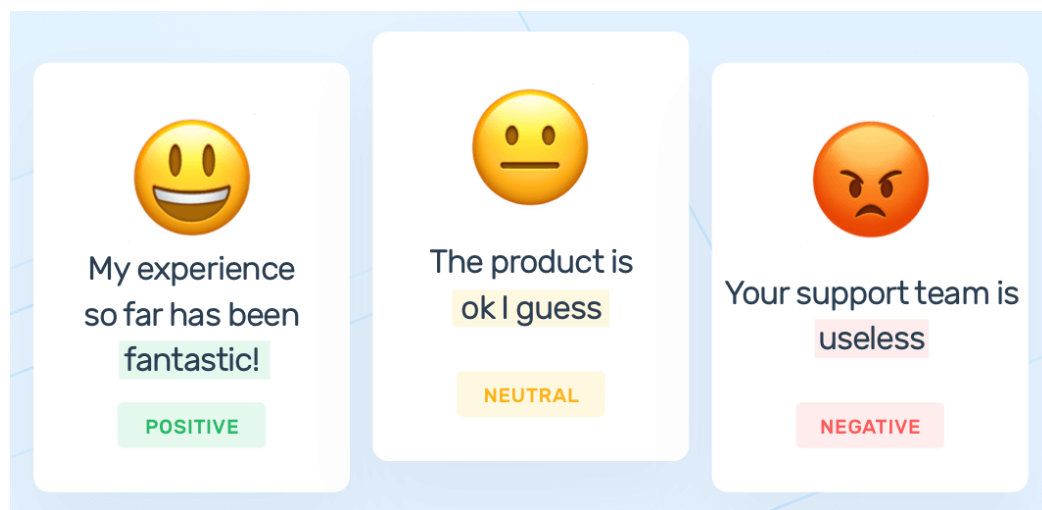


Fig.1.3.1: Emotion Analysis

1.4 Applications

Some popular applications in various fields are listed below.

1. Social media marketing
2. Customer support ticket analysis
3. Brand monitoring and reputation management
4. Voice of customer
5. Product analysis
6. Market research
7. Reputation management

1.5 Summary

In this chapter, what is Twitter, what is an emotion and what is emotion analysis were explained along with applications.

1.6 Overview

In this project, we develop a machine learning model using NLP. The Automated Machine Learning Emotion Analysis Model has been developed to understand customer perception from the data collected from Twitter. And it comes to conclusion that a particular tweet has a positive or negative or neutral emotion.

Emotion analysis of tweets has tremendous use for government and political leaders. It helps them stay abreast of the public opinion about their parties, their actions, and their statements. One wrong statement can sway the public opinion negatively on Twitter and, in an election season, this may prove to be detrimental. This analysis of tweets can help public servants understand how government policies and actions have affected the public psyche. Newsrooms do Twitter analysis to understand the emotions of citizens during elections. Predicting the election results based on public opinion on Twitter is a common use case.

Similarly, sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analysing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm's stock market value.

1.7 Problem Statement

Despite the availability of software to extract data regarding a person's sentiment or emotional opinion on a specific product or service, organizations and other data workers still face issues regarding the data extraction. Emotion Analysis of Web Based Applications Focus on Single Tweet Only. With the rapid growth of the World Wide Web, people are using social media such as Twitter which generates big volumes of opinion texts in the form of tweets which is available for the various emotion analysis. This translates to a huge volume of information from a human viewpoint which make it difficult to extract sentences, read them, analyze tweet by tweet, summarize them and organize them into an understandable format in a timely manner.

Difficulty of various emotion analysis with inappropriate English Informal language refers to the use of colloquialisms and slang in communication, employing the conventions of spoken language such as 'would not' and 'wouldn't'. Not all systems are able to detect a particular from use of informal language and this could hanker the analysis and decision-making process. This NLP algorithm collects tweets and then study it with the help of different statistical computing procedures

1.8 Literature Survey

This section summarizes some of the scholarly and research works in the field of Machine Learning and data mining to analyze sentiments or various emotions on the Twitter and preparing prediction model for various applications. As the available social platforms are shooting up, the information is becoming vast and can be extracted to turn into business objectives, social campaigns, marketing and other promotional strategies as explained in. The benefit of social media to know public opinions and extract their emotions are considered by authors in and explained how twitter gives advantage politically during elections. Further, the concept of the hashtag is used for text classification as it conveys emotion in few words. Tracking on Twitter, authors also predicted the polarity – positive, negative or neutral of tweets by creating a classifier. In addition, they used multiple algorithms and methods to determine the influence of active entity on the tweet patterns of users exhibiting certain emotions. They mined tweets only at the entity level i.e. brand, product, celebrity elements present in tweets rather than the whole sentence in the tweets posted by users. The approach they followed using algorithms to extract features and track the impact and influence made their work different from rest of the literature. The feature extraction process after

preprocessing included constructing n grams along with POS taggers taking care of negation part and improving accuracy of classification. For further analysis and measuring influence, they opted two algorithms – People Rank Algorithm inspired by Page Rank Algorithm used by Google. The main idea behind this algorithm is more the value of People Rank, the more central is the node in the graph means its importance on twitter in terms of followers, retweets and mentions. The other algorithm is Twitter Rank algorithm, an extension to page Rank to determine the influence of users by considering the similarity between users and the structure of nodes i.e. other users they are linked to. They addressed the shortcomings of Page Rank and developed this approach. The influence measure is considered by following the idea that popular/influential people follow you and they act as medium to broadcast specific topic. Following some mathematical computation of ratio of followers/following, retweets, mentions like parameters, they determine the weights and finally derived a mathematical formula to track influence of specific entity. The approach they proposed have potential to determine influence personalities/entities on twitter and can be used for promotional and branding purposes. Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised and semi-supervised approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

CHAPTER 2

METHODOLOGY

2.1. TOOLS AND FRAMEWORKS

The tools and frameworks used in this project are:

- NLP
- Python

1. Natural Language Processing:

NLP techniques are based on machine learning and especially statistical learning which uses a general learning algorithm combined with a large sample, a corpus, of data to learn the rules. Sentiment analysis has been handled as a Natural Language Processing denoted NLP, at many levels of granularity. Starting from being a document level classification task, it has been handled at the sentence level and more recently at the phrase level. NLP is a field in computer science which involves making computers derive meaning from human language and input as a way of interacting with the real world.

2. Python:

Python was found by Guido Van Rossum in Netherland, 1989 which has been public in 1991. Python is a programming language that's available and solves a computer problem which is providing a simple way to write out a solution mentioned that Python can be called as a scripting language. Moreover, also supported that actually Python is a just description of language because it can be one written and run on many platforms. In addition, mentioned that Python is a language that is great for writing a prototype because Python is less time consuming and working prototype provided, contrast with other programming languages. Many researchers have been saying that Python is efficient, especially for a complex project, as has mentioned that Python is suitable to start up social networks or media steaming projects which most always are a web-based which is driving a big data. gave the reason that because Python can handle and manage the memory used. Besides Python creates a generator that allows an iterative process of things, one item at a time and allow program to grab source data one item at a time to pass each through the full processing chain.

2.2 Algorithms

- Support vector machine
- Random Forest

1. Support vector machine:

Support vector machine is a supervised learning system and used for classification and regression problems. Support vector machine is extremely favoured by many as it produces notable correctness with less computation power. It is mostly used in classification problems. We have three types of learning supervised, unsupervised, and reinforcement learning. A support vector machine is a selective classifier formally defined by dividing the hyperplane.

Given labelled training data the algorithm outputs best hyperplane which classified new examples. In two-dimensional space, this hyperplane is a line splitting a plane into two parts where each class lies on either side. The intention of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that separately classifies the data points.

2. Random Forest:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. This algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

This algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

2.3 F1-score

F1-Score or **F-measure** is an evaluation metric for a classification defined as the harmonic mean of **precision** and **recall**. It is a statistical measure of the accuracy of a test or model. Mathematically, it is expressed as follows,

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Here, the value of F-measure(F1-score) reaches the best value at 1 and the worst value at 0. F1-score 1 represents the perfect accuracy and recall of the model.

Now let's see what Recall and precision actually means,

Recall: It tells us what proportion of Data belonging to a certain class say, **class A** is classified correctly as in **class A** by our classifier.

Precision: It tells us what proportion of data that our classifier has classified in a certain class, say **class A** actually belongs to the same **class A**.

Importance of the F1 score: F1-Score (F-measure) is an evaluation metric, that is used to express the performance of the machine learning model (or classifier). It gives the combined information about the precision and recall of a model. This means a high F1-score indicates a high value for both recall and precision. Generally, F1-score is used when we need to compare two or more machine learning algorithms for the same data. We opt for the algorithm whose f1 score is higher.

2.4 Bag of words

The bag-of-words model is one of the simplest representations of textual information used in Natural language processing. In bag of words, the order among the words and grammar is not taken into consideration.

Bag of words is a two-step process.

1) A vocabulary of known words: In this step, all unique words in the document/text is collected.

“It was the best of times and best of wisdom, - Document 1

it was the best and worst of times, - Document 2

it was the age of wisdom, - Document 3

it was the age of foolishness” - Document 4

For this small example, let’s treat each line as a separate “document” and the 4 lines as our entire corpus (Corpus means collection of documents) of documents.

1. “it”

2. “was”

3. “the”

4. “best”

5. “of”

6. “times”

7. “worst”

8. “age”

9. “wisdom”

10. “foolishness”

11. ”and”

2). A measure of the presence of known words:

Now we will have a table where, column corresponds to the index of the unique words and rows corresponds to the number of times a word occurs in a document.

	1 "it"	2 "was"	3 " the"	4 " best"	5 " of"	6 " times"	7 " worst"	8 " age"	9 " wisdom"	10 " foolishness"	11 "and"
D1	1	1	1	2	1	1	0	0	1	0	1
D2	1	1	1	1	1	1	1	0	0	0	1
D3	1	1	1	0	1	0	0	1	1	0	0
D4	1	1	1	0	1	0	0	1	0	1	0

Table 2.4.1 : Bag of words table

This results in a vector with lots of zero scores called a sparse vector or sparse representation. Sparse vectors require more memory and computational resources when modelling and the vast number of positions or dimensions can make the modelling process very challenging for traditional algorithms.

2.5 TF-IDF

Term frequency-inverse document frequency (TF-IDF), is a measure that is intended to reflect how important a word is to a document in a collection of documents (Corpus).

Term Frequency: Term frequency shows how frequently a term/word occurs in a document. It is denoted as

$$\text{Term frequency} = \frac{\text{Number of times the word occurred in document}}{\text{Total number of words in the document}}$$

Inverse document frequency: The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

$$\text{Idf}(t,D)=\log \frac{\text{Total number of documents}}{\text{Number of documents having the word}} = \log \frac{N}{d \in D:t \in d}$$

Where, t is the term, d = a document. D = Total number of documents

TF-IDF is calculated as

$$\text{tfidf} = \text{tf}(t, d) * \text{Idf}(t, D)$$

Example:

A-doc : The car is driven on the road.

B-doc : The truck is driven on the highway.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	$\log(2/2) = 0$	0	0
Car	1/7	0	$\log(2/1) = 0.3$	0.043	0
Truck	0	1/7	$\log(2/1) = 0.3$	0	0.043
Is	1/7	1/7	$\log(2/2) = 0$	0	0
Driven	1/7	1/7	$\log(2/2) = 0$	0	0
On	1/7	1/7	$\log(2/2) = 0$	0	0
The	1/7	1/7	$\log(2/2) = 0$	0	0
Road	1/7	0	$\log(2/1) = 0.3$	0.043	0
Highway	0	1/7	$\log(2/1) = 0.3$	0	0.043

Table 2.5.1: Giving values to words in sentences

From the above table, we can see that the TF-IDF of common words was zero, which shows they are not significant. On the other hand, the TF-IDF of “car”, “truck”, “road”, and “highway” are non-zero. These words have more significance.

	The	Car	Truck	Is	Driven	On	The	Road	Highway
Sentence 1	0	0.043	0	0	0	0	0	0.043	0
Sentence 2	0	0	0.043	0	0	0	0	0	0.043

Table 2.5.2 : Calculating TF-IDF values of words

So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach to 1.

Multiplying tf and idf results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

TFIDF is successfully used by search engines like Google, as a ranking factor for content.

2.6 TASKS REQUIRED FOR ANALYSING EMOTION OF TWEETS

We have proposed a system that explained the process of the gathering of data, analysis of emotion of tweets, and classification of Twitter opinions. Great works and tools are focusing on text mining on twitter. In this project, the wealth of available libraries has been used. We consider the opinion of the current political views by the posted tweet of users in the form of hashtags. Then we store the tweets in the database and pre-process these datasets (set of tweets of users). After that, divide the datasets into the training and testing samples. Here the 31962 tweets are taken as training samples, and 17197 tweets are test samples. After that we apply data classification and cleaning techniques. Then apply the Natural Language processing method to build a score checking module. This module is used to assign and check the sentiment score for each tweet. Then visualize and test the module. The flowchart is as follows:

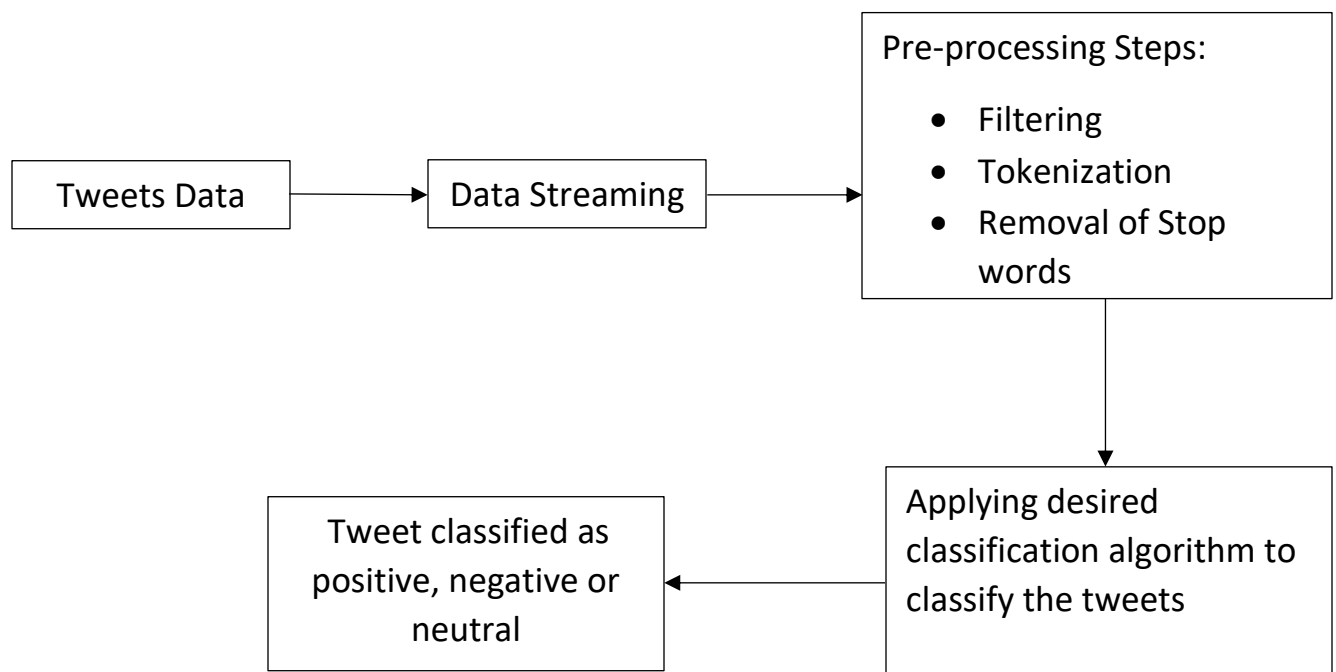


Fig 2.6 Flowchart of Emotion analysis of tweets

1.Data Collection

Data in the form of raw tweets is retrieved by using the Scala library “Twitter4j” which provides a package for real time twitter streaming API. The API requires us to register a developer account with Twitter and fill in parameters such as consumer Key, consumer Secret, access Tokenaccess, and TokenSecret. This API allows to get all random tweets or filter data by using keywords. Filters supports to retrieve tweets which match a specific criterion defined by the developer. We used this to retrieve tweets related to specific keywords which are taken as input from users. But we didn’t used Twitter API in our project. We have taken the dataset which is readily available in online resources and tried to observe the above few tweets corresponding to each labels, it is safe to conclude that label 0 is for non-offensive tweets and label 1 is for offensive tweets. After observing data we applied data preprocessing and cleaning techniques used in NLP.

2. Data Streaming

It is the text normalizing process of reducing a derived word to its root or stem [28]. For example, a stemmer would reduce the phrases “stemmer”, “stemmed”, “stemming” to the root word “stem”. Advantage of stemming is that it makes comparison between words simpler, as we do not need to deal with complex grammatical transformations of the word. In our case we employed the algorithm of “porter stemming” on both the tweets and the dictionary, whenever there was a need of comparison

3.Data Processing and cleaning

Text is a highly unstructured form of data, various types of noise are present in it and the data is not readily analysable without any pre-processing. The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text pre-processing. We will divide it into 2 parts:

- Data Inspection
- Data Cleaning

In data inspection, we will check the label distribution in the train dataset. We seemed to found like we have more non-offensive tweets in our training set than offensive ones. In the train dataset, we have 2,242 tweets labeled as offensive and 29,720 tweets labeled as non-offensive. So, it is an imbalanced classification challenge. An imbalanced dataset is one in which labels are not in equal counts. Often small differences doesn't matter. It often gives rise to accuracy paradox. Let us consider a dataset having test details of 100 cancer patients of which naturally 6 were found to have cancer. If we build a model which predicts whether the patient as cancer or not on the basis of random guessing it is going to be accurate 94% of the time. So does that mean we have a good machine learning model. No, rather it says we are using wrong metrics for evaluating model performance. So in the case of imbalanced datasets we usually resort to metrics such as confusion matrix, F1 score etc rather than accuracy score.

In any natural language processing task, cleaning raw text data is an important step. It helps in getting rid of the unwanted words and characters which helps in obtaining better features. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data. The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don't carry much weightage in context to the text. Before we begin cleaning, we first combine train and test datasets. Combining the datasets will make it convenient for us to pre-process the data. Later we will split it back into train and test data. In the dataset, 31962 tweets are taken as training samples, and 17197 tweets are test samples. Here Data processing involves Tokenization which is the process of splitting the tweets into individual words called tokens. Tokens can be split using whitespace or punctuation characters. It can be unigram or bigram depending on the classification model used. The bag-of-words model is one of the most extensively used model for classification. It is based on the fact of assuming text to be classified as a bag or collection of individual words with no link or interdependence. The simplest way to incorporate this model in our project is by using unigrams as features. It is just a collection of individual words in the text to be classified, so, we split each tweet using whitespace. For example, the tweet "Such an amazing climax!!" is split from each whitespace as follows:

{

Such

an

amazing

climax

!!

}

4.Data Filtering

A tweet acquired after data processing still has a portion of raw information in it which we may or may not find useful for our application. Thus, these tweets are further filtered by removing stop words, numbers and punctuations.

Stop words: For example, tweets contain stop words which are extremely common words like “is”, “am”, “are” and holds no additional information. These words serve no purpose and this feature is implemented using a list stored in stopfile.dat. We then compare each word in a tweet with this list and delete the words matching the stop list.

Removing non-alphabetical characters: Symbols such as “#@” and numbers hold no relevance in case of sentiment analysis and are removed using pattern matching. Regular expressions are used to match alphabetical characters only and rest are ignored. This helps to reduce the clutter from the twitter stream. **Stemming:** It is the process of reducing derived words to their roots. Example includes words like “fish” which has same roots as “fishing” and “fishes”. The library to use stemming is Stanford NLP which also provides various algorithms such as porter stemming. In our case, we have not employed any stemming algorithm due to time constraints.

Stemming: It is the text normalizing process of reducing a derived word to its root or stem [28]. For example a stemmer would reduce the phrases “stemmer”, “stemmed”, “stemming” to the root word “stem”. Advantage of stemming is that it makes comparison between words simpler, as we do not need to deal with complex grammatical transformations of the word. In our case we employed the algorithm of “porter stemming” on both the tweets and the dictionary, whenever there was a need of comparison.

5. Applying Algorithms

After analyzing data we will implement machine learning algorithms like Support vector machines (SVM) and Random Forest. By using these algorithms we will try to fit our model in the

best suitable algorithm. This best algorithm can be identified by observing F1 score of each algorithm. We will come to conclusion by observing the maximum F1 score of a particular algorithm means if an algorithm has higher F1 score compared to another algorithm, we can say that algorithm provide very accurate result.

6. Classification of Tweets

Based on values we given the tweets are classified as three types they are:

- Positive
- Negative
- Neutral

1. Positive

Tweets undergoes into positive tweets which are written when we are happy, excited, enjoy... etc. which give us a good vibe.

Example:

1. I am feeling Happy.
2. I am excited for my birthday.
3. I have a pleasant day.

2. Negative

Tweets undergoes into negative tweets which are written when we are sad, worry, disgusted, frustrated... etc. which give us bad vibe.

Example:

1. I am worry about you.
2. It is disgusting.
3. These people are mean to me.

3. Neutral

Tweets undergoes into neutral tweets which are written generally, daily things, confused, shy etc.

Example:

1. I am confused.
2. My friend felt shy when I introduced her to one of my guy friends.
3. I feel content.

2.7 TRAINING DATA

Training data refers to the initial set of data fed to any machine learning model from which the model is created. The training data contains a pair of input data and annotations gathered from various resources and organized to train the model to perform a specific task at a high level of accuracy. Machine learning models learn the annotations on training data, so that they may apply them to new, unlabeled examples. In supervised learning, humans will label data telling the model exactly what it needs to find.

2.8 TESTING DATA

A test set in machine learning is a secondary (or tertiary) data set that is used to test a machine learning program after it has been trained on an initial training data set. The idea is that predictive models always have some sort of unknown capacity that needs to be tested out, as opposed to analyzed from a programming perspective. In contrast, a program that memorizes the training data by learning an overly complex model could predict the values of the response variable for the training set accurately, but will fail to predict the value of the response variable for new examples. Memorizing the training set is called over-fitting. A program that memorizes its observations may not perform its task well, as it could memorize relations and structures that are noise or coincidence. Balancing memorization and generalization, or over-fitting and under-fitting, is a problem common to many machine learning algorithms. Regularization may be applied to many models to reduce over-fitting.

CHAPTER 3

RESULTS AND DISCUSSION

Firstly, we analyze the datasets which was collected from Kaggle. We will import necessary libraries, read the datasets using pandas and to have a clear idea about the content in the datasets we try to read first 10 rows in datasets in the following manner:

Fig 3.1.1 : Analyze the datasets by importing necessary libraries.

Fig 3.1.2 : First ten rows from taken datasets

By observing the above few tweets corresponding to each labels. We concluded that label 0 is for non-offensive tweets and label 1 is for offensive tweets. The respective count of offensive and non-offensive tweets are shown below:

```
[13] train.label.value_counts()

0    29720
1     2242
Name: label, dtype: int64
```

Fig 3.1.3: The count of non-offensive tweets and offensive tweets.

3.2 Data Cleaning

In any natural language processing task, cleaning raw text data is an important step. It helps in getting rid of the unwanted words and characters which helps in obtaining better features. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data. The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don't carry much weightage in context to the text. We used PorterStemmer class from NLTK (Natural Language ToolKit) library in a function. This function takes raw tweet as input and returns cleaned tweet. It removes twitter handles, punctuation, short words and stopwords which does not contribute much to our analysis. The code given below shows the implementation of PorterStemmer and creating a function to read and to remove twitter handles, punctuation, short words and stopwords.

```
[25] from nltk.stem.porter import PorterStemmer
      stemmer = PorterStemmer()

      def clean_tweet(tweet):
          clean_handle = re.sub(r'@[^\w]*', '', tweet)
          clean_punc = re.sub(r'^a-zA-Z#', ' ', clean_handle)
          clean_short_tokenized = [word for word in clean_punc.split() if len(word) > 3]
          clean_normalize = [stemmer.stem(word) for word in clean_short_tokenized]
          return ' '.join(clean_normalize)
          # Removes twitter handles from tweets
          # Removes punctuation, special characters(except #tags)
          # Remove short words and tokenize
          # Stem tokenized words

[26] clean_tweet(combined.tweet.iloc[3])

'#model love take with time'

[27] combined.tweet = combined.tweet.apply(lambda x : clean_tweet(x))
```

Fig 3.2.1 : The implementation of PorterStemmer and creating a function to read and to remove twitter handles, punctuation, short words and stopwords.

Now, we will explore the cleaned tweets. Exploring and visualizing data, before we begin exploration, we must think about some questions related to the data in hand. A few probable questions are as follows:

- What are the most common words in the entire dataset?
- What are the most common words in the dataset for negative and positive tweets, respectively?
- How many hashtags are there in a tweet?
- Which trends are associated with my dataset?
- Which trends are associated with either of the sentiments? Are they compatible with the sentiments?

To answer all these questions, we will create the word cloud object and able to generate an image that gives us the most representative words (actually the more common) in a chosen set of reviews. Before that We need to remove English stopwords as they don't add anything meaningful in the explanation and they are everyday words that we use. Firstly, we classify the tweets into all tweets, good tweets, bad tweets after removal of stopwords. The code for word cloud is shown below figure.

```
[28] from wordcloud import WordCloud, STOPWORDS
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=110, stopwords=STOPWORDS)

[29] all_tweets = ' '.join(combined.tweet)
good_tweets = ' '.join(combined[combined.label == 0].tweet)
bad_tweets = ' '.join(combined[combined.label == 1].tweet)

[ ] bad_tweets

'cnn call #michigan middl school build wall chant #tcot comment #australia #opkillingbay #seashepherd #helpcovedolphin #thecov #helpcovedolphin retweet agre lumpi say prove lumpi unbeliev that
centuri need someth like thi again #neverump #xenophobia let fight against #love #peac white establish have folx run around love themselv promot great white peopl call peopl white #race #ident #
med #altright use insecur lure into #whitesupremaci interest #linguist that doesn address #race racism about #power #raciolinguist bring mock obama be black #brexit #peopl aren protest #trump be
caus #republican they becaus trump fuher when call #michelleobama gorilla becaus racist have long thought black peopl smaller hand show barri probabl lie about be #knick game suck more than #gol
f point finger million point right back #jewishsupremacist might libtard #libtard #sjw #liber #polit take #trash america vote against #hate vote against vote against hold open door woman becaus
woman becaus nice thing that ev...'

[30] all_tweets

'when father dysfunct selfish drag kid into dysfunct #run thank #lyft credit caus they offer wheelchair van #disappoint #getthank bihday your majesti #model love take with time factsguid societi
#motiv huge fare talk befor they leav chao disput when they there #allshowandnogo camp tomorrow dann next school year year exam think about that #school #exam #hate #imagin #actorslif #revoluti
onschool #girl love land #allin #cav #champion #cleveland #clevelandcavali welcom here #ireland consum price index climb from previou #blog #silver #gold #forex selfish #orlando #standwithorland
o #pulseshoot #orlandosshoot #biggerproblem #selfish #heabreak #valu #love daddi today day #gettingf ouch junior angri #got #junior #yugyoem #omg thank have paner #thank #posit retweet agre #friday smile around user #cooki make peopl know essent
i oil made chemic #euro peopl b...'

good_tweets

'when father dysfunct selfish drag kid into dysfunct #run thank #lyft credit caus they offer wheelchair van #disappoint #getthank bihday your majesti #model love take with time factsguid societi
#motiv huge fare talk befor they leav chao disput when they there #allshowandnogo camp tomorrow dann next school year year exam think about that #school #exam #hate #imagin #actorslif #revoluti
onschool #girl love land #allin #cav #champion #cleveland #clevelandcavali welcom here #ireland consum price index climb from previou #blog #silver #gold #forex selfish #orlando #standwithorland
o #pulseshoot #orlandosshoot #biggerproblem #selfish #heabreak #valu #love daddi today day #gettingf ouch junior angri #got #junior #yugyoem #omg thank have paner #thank #posit #friday smile arou
nd user #cooki make peopl know essent i oil made chemic #euro peopl blame conced goal rooney gave away free kick know bale them from there littl dude #baddy #coneofsham #cat #piss #funni #laugh
product happi #wine tool #weeke...'

```

Fig 3.2.2: Creation of word cloud to generate an image of most representative words

The following code shown in below figure helps in obtaining clear visualization of all tweets, good tweets, bad tweets.



Fig 3.2.3 : Clear visualization of all tweets, good tweets, bad tweets.

To add labels to unlabeled data for analysis of emotion of tweets, we can use the Vader sentiment model which is one of the best approaches for sentiment analysis of tweets. We can access it using the NLTK library in Python by importing the necessary Python libraries and an unlabeled dataset that we need for the task of adding labels to a data for emotion analysis of tweets. The below figure shows the downloading package vader_lexicon.

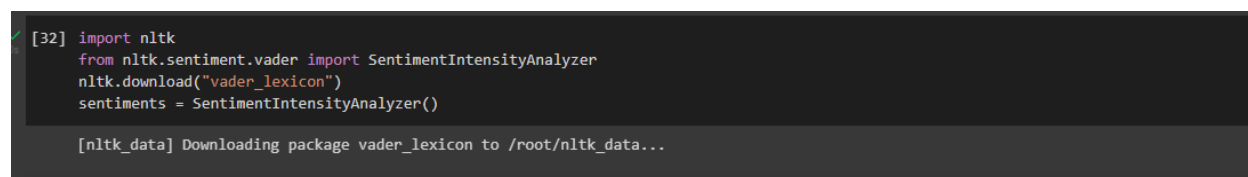
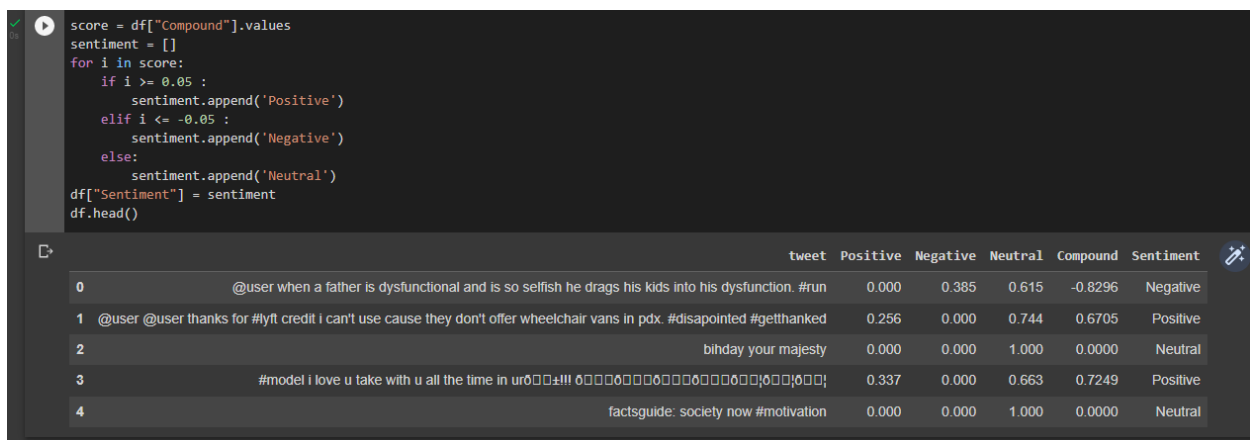


Fig 3.2.4: The downloading package vader_lexicon.

So this dataset contains only two columns, We will now move to the task of adding labels to the dataset. We will start by adding four new columns to this dataset as Positive, Negative, Neutral, and Compound by calculating the sentiment scores of the column containing textual data. The below figure shows the code for adding 4 columns of Positive, Negative, Neutral and Compound to the dataset.



As we can see in the above output, we have added four new columns containing the sentiment scores of the Review column. Now the next task is to add labels by categorizing these scores. According to the industry standards, if the compound score of emotion is more than 0.05, then it is categorized as Positive, and if the compound score is less than -0.05, then it is categorized as Negative, otherwise, it's neutral. So with this information, we will add a new column in this dataset which will include all the emotion of labels. The below figure shows classification of emotion of tweet by comparing compound score.



From the above values we will draw a pie chart to have a better visualization of data. The below figure shows number of positive tweets, negative tweets and neutral tweets present in the given dataset.

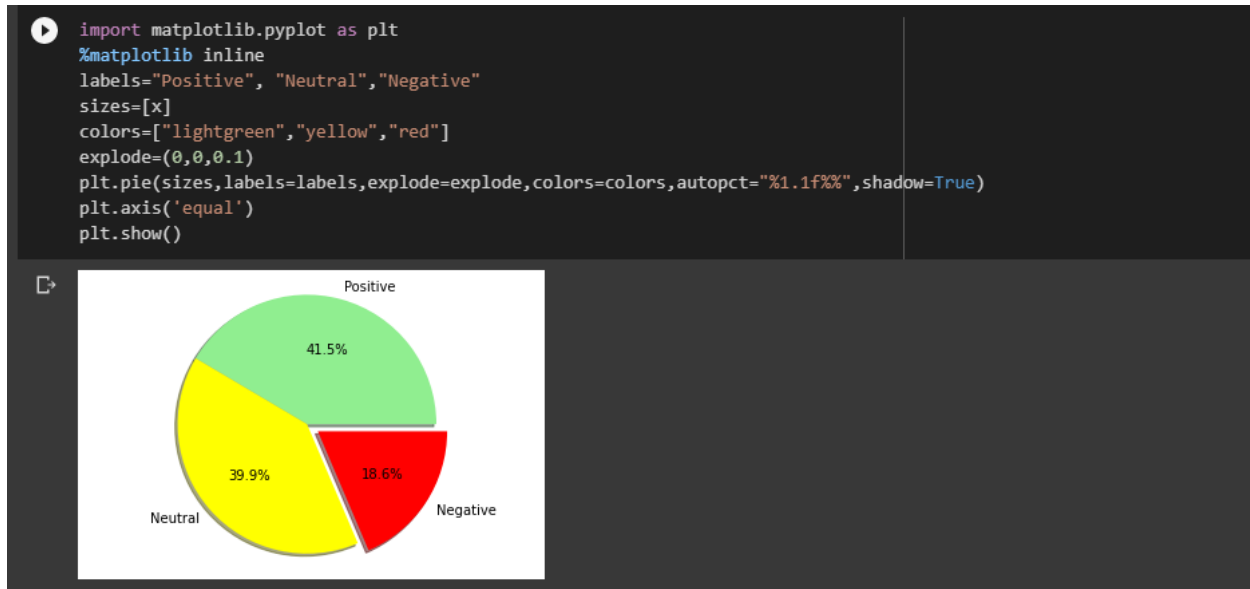


Fig 3.2.8: Number of positive, negative, neutral tweets present in the given dataset.

3.3 Results – Models Accuracy

After cleaning and clea analysis of data, we will implement machine learning algorithms such as Logistic Regression, Support vector machine, Random Forest etc. After implementing algorithms, we will try to find out F1 score of each model to check which model is giving best

```

- Support vector machine

[41] from sklearn.svm import SVC
     svc = SVC(kernel='linear', C=1, probability=True)

[47] svc.fit(x_bow_train, y_bow_train)
     bow_pred_prob = svc.predict_proba(x_bow_test)
     bow_pred_thresh = bow_pred_prob[:, 1] >= 0.3
     bow_pred = bow_pred_thresh.astype(np.int)
     print('F1 Score : ', f1_score(y_bow_test, bow_pred))

F1 Score : 0.5078776645041705

[48] svc.fit(x_tfidf_train, y_tfidf_train)
     tfidf_pred_prob = svc.predict_proba(x_tfidf_test)
     tfidf_pred_thresh = tfidf_pred_prob[:, 1] >= 0.3
     tfidf_pred = tfidf_pred_thresh.astype(np.int)
     print('F1 Score : ', f1_score(y_tfidf_test, tfidf_pred))

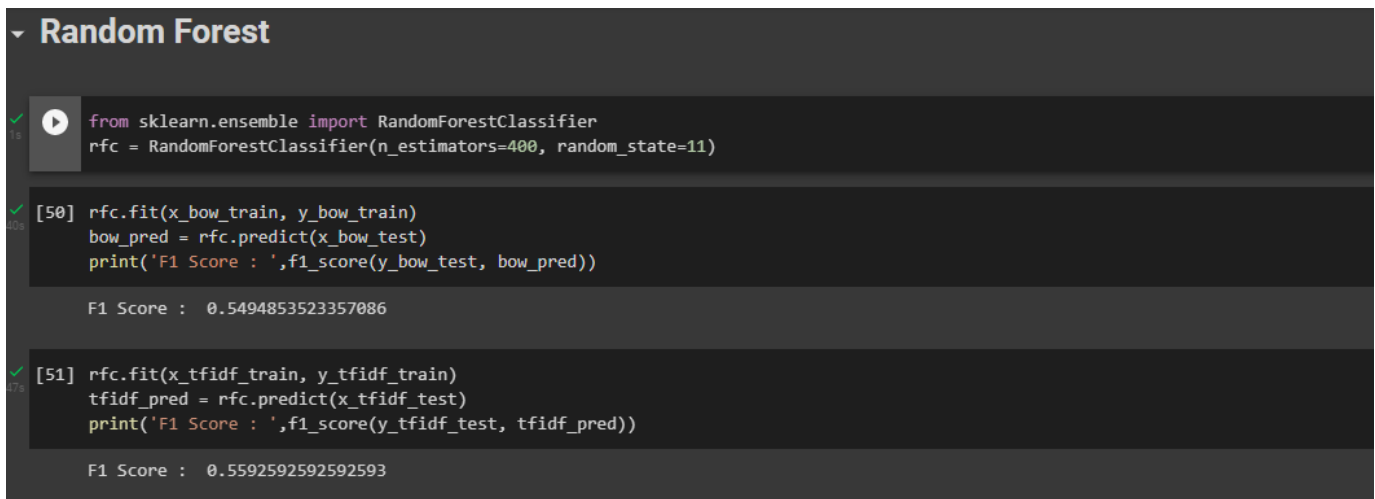
F1 Score : 0.5109489051094891

```

Fig 3.3.1: The code for implementing SVM algorithm.

In our project we considered only Support vector machine and Random Forest to find the best fit model for the given data. The below figure shows the code for implementing SVM algorithm. From the above code we can observe that F1 score of this model is 0.5 which is good. We can use this model for our data.

Next, we will implement Random Forest model and we will find F1 score of this model. The below figure explains the code for Random Forest model and finding the F1 score of this model.



```
Random Forest

from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=400, random_state=11)

[50] rfc.fit(x_bow_train, y_bow_train)
      bow_pred = rfc.predict(x_bow_test)
      print('F1 Score : ',f1_score(y_bow_test, bow_pred))

      F1 Score : 0.5494853523357086

[51] rfc.fit(x_tfidf_train, y_tfidf_train)
      tfidf_pred = rfc.predict(x_tfidf_test)
      print('F1 Score : ',f1_score(y_tfidf_test, tfidf_pred))

      F1 Score : 0.5592592592592593
```

Fig 3.3.2: The code for Random Forest model and finding the F1 score of this model.

From the above code we observe that F1 score of Random Forest algorithm is 0.55 which is near to SVM model. So we can conclude saying that this two works with the same accuracy and we can use any of these two models as F1 score is same.

CHAPTER 4

CONCLUSION

CONCLUSION

Twitter is a source of vast unstructured and noisy data sets that can be processed to locate interesting patterns and trends. Tweet emotion analysis is developed to analyze customers perspectives toward the critical to success in the marketplace. The projected framework gathers data from the twitter and uses natural language processing techniques to extract features. Then natural language processing is applied to the data to classify the sentiments as Positive, Negative and Neutral. Polarity and partiality are also calculated by the dictionary, that consists of a semantic score of a tweet. It is observed that natural language processing is a better method for sentiment analysis as compared to traditional methods. The program is using a machine-based learning approach which is more accurate for analyzing a sentiment; together with natural language processing techniques will be used. As a result, program will be categorized sentiment into positive and negative. After cleaning the data, we will implement machine learning algorithms like Support vector machine and Random Forest. The best model fit is calculated by finding F1 score of these two models and based on the F1 scores we took maximum value of F1 score algorithm and we treat that algorithm as our main priority model. We learned how to approach an emotion analysis problem. We started with preprocessing and exploration of data. Then we extracted features from the cleaned text using Bag-of-Words and TF-IDF. Finally, we were able to build a couple of models using both the feature sets to classify the tweets.

REFERENCES

- Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, "Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011 Workshop on Languages in Social Media, 2011, pp. 30-38
- Pang, B. and Lee, L. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts". 42nd Meeting of the Association for Computational Linguistics[C] (ACL-04). 2004, 271-278.
- Neethu M, S and Rajashree R, "Sentiment Analysis in Twitter using Machine Learning Techniques" 4th ICCCNT 2013, at Tiruchengode, India. IEEE – 31661
- A. Pak, and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," Special Issue of International Journal of Computer Application, France: Universite de Paris-Sud, 2010
- Dasgupta, S. S., Natarajan, S., Kaipa, K. K., Bhattacharjee, S. K., & Viswanathan, A. (2015, October). Sentiment analysis of Facebook data using Hadoop based open source technologies. In Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on (pp. 1-3). IEEE.

APPENDIX

DATA SHEET of any special IC(s) used in the project and link for the code uploaded to GitHub, etc.

Google colab link: NLP-1.ipynb - Colaboratory (google.com)