Mini Project Report on

TWITTER SENTIMENT ANALYSIS

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Student Name: Ajay Pratap Kandari **University Roll No: 2016599**

Under the Mentorship of
Dr. Vishan Kumar Gupta

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
January 2023

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"Twitter Sentiment Analysis"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of Dr. Vishan Kumar Gupta, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: Ajay Pratap Kandari University Roll no: 2016599

Table of Contents

Chapter 1

Introduction

In the following sections, a brief introduction and the problem statement for the work has been included.

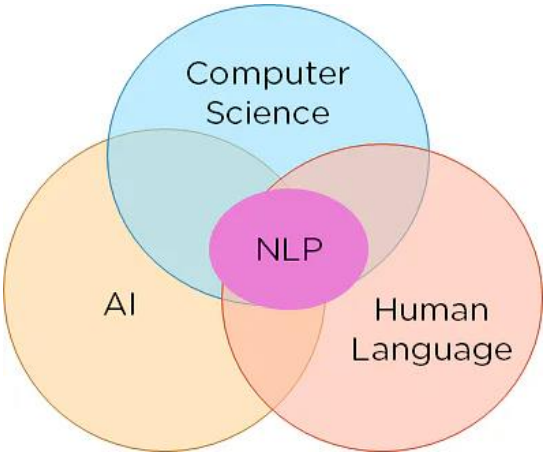
1.1 Introduction

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to automatize the machines and imitate the way that humans learn, gradually improving its accuracy.

Twitter Sentiment Analysis

It's a tool to analyze and determine the emotional aspects behind the tweets made by user on twitter platform. It's a technique which focuses on the positive, negative, neutral and intermediate tone of the tweets which allows the enterprises to analyze customer reviews, feedback and surveys related to quality and services provided.

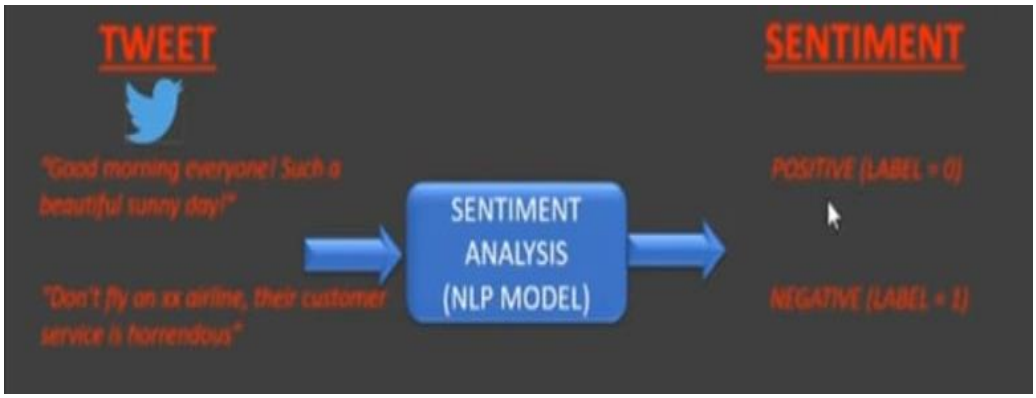
It allows companies to make further decisions and therefore an important tool for the organizations.



It uses Natural Language Processing which a subset of ML and these are the machines which respond to text and speech int the same way humans do. It uses various mathematical, statistical and deep learning models.

Python uses Natural Language Toolkit (NLTK) which provides the tools and libraries for tasks like tokenization, word segmentation, stemming and lemmatization etc.

In this particular model the words are preprocessed and converted into numerical equivalent and uses the **Naïve Bayes** classifier to classify the tweets.



The Naive Bayes classification algorithm is basically a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. The independence assumptions often do not have an impact on reality.

In this project I have used the python libraries like ***pandas*** for fast, easy-to-use data structures, and data analysis tools for manipulating and the library like ***numpy*** for scientific computing. ***Numpy*** provides high-performance multidimensional arrays and tools to deal with them.

The libraries ***matplotlib*** and ***seaborn*** are used for data visualization. Various plots and graphs are plotted with the help of these libraries for better understanding of data.

A very powerful and robust library ***scikit-learn*** has been used for data pre processing and implementing the classification algorithm.

Python NLTK provides the essential tools like wordcloud and stopwords for data formatting and also provides the tool for ***vectorization and tokenization*** of the tweets.

PROBLEM STATEMENT

Generating statistical information regarding emotions, sentiments out of analysis of user's opinions from tweets, which can be used as an inference to understand how users feel thereby improving users experiences regarding. Despite the availability of software to extract data regarding a person's sentiment on a specific product or service, organizations and other data workers still face issues regarding the data extraction. 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 sentiment analysis. This translates to a huge volume of information from a human viewpoint which make it difficult to extract a sentence, read them, analyse tweet by tweet, summarize them and organize them into an understandable format in a timely manner

Chapter 2

Literature Survey

Sentiment analysis is the excellent tool used by companies for better answer the pertinent questions and gain valuable business insights. It's is a term that refers to the use of text analysis, NLP and computational linguistics to know the tone of user in a particular form of speech or text. Basically, it helps to determine whether a text is expressing sentiments that are positive, negative, neutral or somehow intermediate. The origin of sentiment analysis can be traced to the 1950s, when sentiment analysis was primarily used on written paper documents. Today, however, sentiment analysis is widely used to mine subjective information from content on the Internet, including texts, tweets, blogs, social media, news articles, reviews, and comments. This is done using a variety of different techniques, including NLP, statistics, and machine learning methods. Organizations then use the information mined to identify new opportunities and better target their message toward their target demographics. Today Administration even uses sentiment analysis to predict public response to its policy announcements**.** Sentiments refer to attitudes, opinions, and emotions. In other words, they are subjective impressions as opposed to objective facts. Different types of sentiment analysis use different strategies and techniques to identify the sentiments contained in a particular text. With consumers able to share their opinions across the web so easily, online opinions have become a valuable currency for businesses and companies trying to cultivate their digital reputations, identify new opportunities, and

successfully market their products. Furthermore, social media monitoring is also an excellent way to better identify your brand's influencers and promoters. For example, let's say you need to figure how where negative content about your brand is coming from. You could identify, say, 50 of the major influencers in your industry and then analyze sentiment of their tweets about your brand, to figure out who has negative perceptions. From there, you can reach out to the influencers individually, and hopefully change their perception. **Public Relations**: Sentiment analysis can also help companies develop and refine their public relations strategy. For example, companies can use sentiment analysis to identify sales leads and spot industry trends. As previously mentioned, sentiment analysis can also be used to identify influencers in your industry with positive sentiments toward your brand, which can be leveraged in a PR strategy. **Marketing**: Companies are increasingly using the information found in customer-generated content on product reviews and social media sites. For example, let's say that Samsung wants to know how consumers feel about its new Galaxy phone. Instead of conducting a survey, analysts can go online and evaluate the comments customers have left on major online ecommerce sites like Amazon. Samsung can analyze the content of these reviews. For example, Samsung could determine the tone of comments being left to gain insight into the emotions consumers feel toward the product or analyze the comments to figure out how much knowledge customers have about the product. **Data Mining**: Sentiment analysis can also be used for data mining, or gathering competitive intelligence about your competitors. For example, a brand could easily track social media mentions or mentions of competitors in other places across the web, and analyze how consumers feel about the competitors and their

products. This is an excellent way to gain a competitive edge in today's highly competitive marketplace. **Political Analysis**: Studies of sentient analysis of tweets and microblogs have shown that such analysis can accurately indicate political sentiment. "The sentiment of Twitter messages closely corresponds to political programs, candidate profiles, and evidence from the media coverage of the campaign trail."

Methods

1.Using a Heterogeneous Dataset for Emotion Analysis in Text : A supervised machine learning approach was adopted to recognize six basic emotions (anger, disgust, fear, happiness, sadness and surprise) using a heterogeneous emotion-annotated dataset which combines news headlines, fairy tales and blogs. The Support Vector Machines classifier (SVM) performed significantly better than other classifiers, and it generalized well on unseen examples. Five data sets were considered to compare among various approaches. In bag of words each sentence in the dataset was represented by a feature vector composed of Boolean attributes for each word that occurs in the sentence. If a word occurs in a given sentence, its corresponding attribute is set to 1; otherwise it is set to 0.

2. Multiclass Emotional Analysis on Social Media Posts: The models they have built SVM has outperformed with greatest accuracy. After considering around 13,000 examples per emotion they had split 63%for training set a hold out cross validation set (27%), and a final test set (10%).The first model trained and optimized for the task was Multinomial Naive Bayes.

3. Analyzing Sentiment of Twitter Data using Machine Learning Algorithm: Tweets posted on twitter are freely available through a set of APIs of twitter. At first, we collected a corpus of positive, negative, neutral and irrelevant tweets from twitter API. Then pre-processing done by removing stop words, negations, URL, full stop, commas etc. to reduce noise from tweets and to prepare our data for sentiment classification. After that, we apply machine learning algorithms to our dataset and compare their results. Results helps to identify which machine learning algorithm is best suited for classification

of SA. The stages involved in this process are: 1.Data Collection : obtain training data of twitter 2.Pre-processing Setup: removing unrelated contents 3.Sentiment Classifier : various machine learning algorithms are used 4.Evaluation: produces result. And each step is further classified like in the pre-processing step some sub-steps like stemming, stop word extractor are also included. The efficiency of an classifier usually depends on the pre-processing step 5. Methods for Sentiment Analysis: In this paper, various approaches to sentiment analysis have been examined and analyzed,Techniques such as Streaming API SVM etc., discussed. These techniques all have different strengths and weaknesses.

4.Sentiment Analysis on Twitter using streaming API: It uses NLP where it helps in tokenization, stemming, classification, tagging, parsing and sentiment reasoning Its basic feature is to convert unstructured data into structured data. It uses Naive Bayes for classification which requires number of linear parameters.

Chapter 3

Methodology

We use tweepy an API to stream live tweets from Twitter. User based on his interest chooses a keyword and tweets containing that keyword are collected and stored into a csv file.Then we make it a labeled dataset using textblob and setting the sentiment fields accordingly. Thus our train data set without preprocessing is ready. But for sake of concept understanding and learning I have used Kaggle dataset which is a csv file with the users and tweets. Therefore we perform preprocessing to clean, remove unwanted text, characters out of the tweets. Then we train our classifier by fitting the train data to the classifier ,there after prediction of results over unseen test data set is made which there after provides us with the accuracy with which the classifier had predicted the outcomes.There after we present our results in a pictorial and graphical manner which is the best way to showcase results because of its easiness to understand information out of it.

We can divide the process as :

Task 1: Understanding the problem statement

Task 2: Import Libraries and datasets

```
#Import
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

```

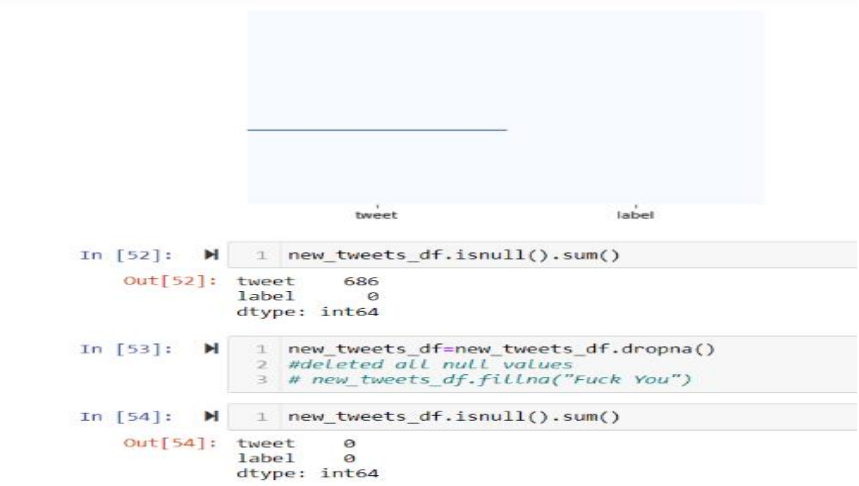
```
tweets_df=pd.read_csv('twitter_training.csv')
new_tweets_df=tweets_df
```

```
tweets_df.info()


```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74681 entries, 0 to 74680
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   id       74681 non-null  int64 
 1   company  74681 non-null  object
 2   label    74681 non-null  object
 3   tweet    73995 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.3+ MB
```

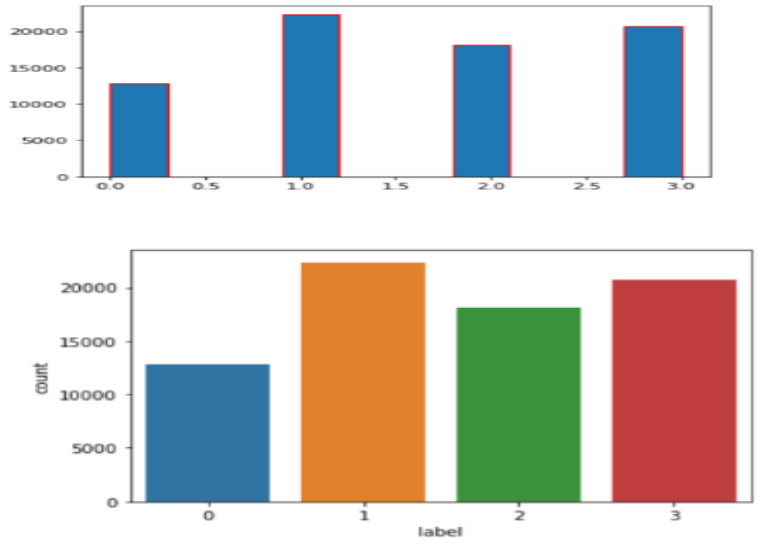
Task 3: Explore Dataset



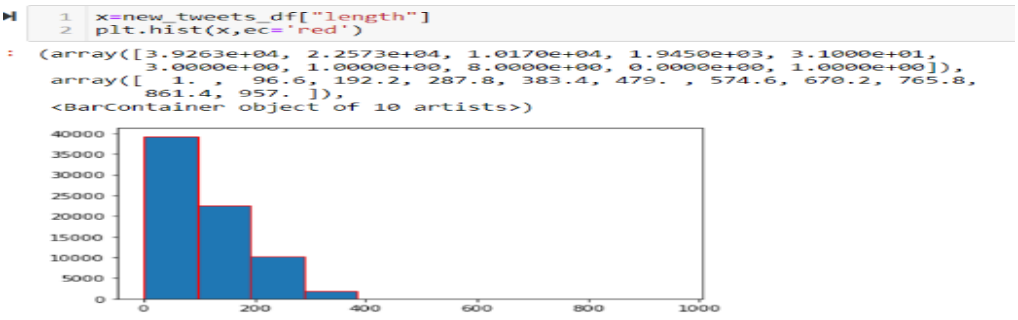
Here it shows many null values in dataset and how null values are dealt

1 : One method is dropping all the rows where null values are present

2: Another method is assigning all null values a default argument



The four classes positive, negative, neutral and intermediate.



```
positive=new_tweets_df[new_tweets_df['label']==0]
intermediate=new_tweets_df[new_tweets_df['label']==1]
neutral=new_tweets_df[new_tweets_df['label']==2]
negative=new_tweets_df[new_tweets_df['label']==3]
```

	tweet	label	length
0	I am coming to the borders and I will kill you...	3	51
1	im getting on borderlands and i will kill you ...	3	50
2	im coming on borderlands and i will murder you...	3	51
3	im getting on borderlands 2 and i will murder ...	3	57
4	im getting into borderlands and i can murder y...	3	53
...	...	...	...
74676	Just realized that the Windows partition of my...	3	128

Assigning Classes…

Task 4: Implementing Wordcloud

```
from wordcloud import WordCloud
```

```
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(one_string_tweets))
```

```
<matplotlib.image.AxesImage at 0x1f922266bb0>
```

Text 5: Data Cleaning (Removing stopwords and punctuations)

Cleaning The Text

```
import string
string.punctuation
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stopwords.words('english')
from sklearn.feature_extraction.text import CountVectorizer
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\ashup\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
def  tweet_cleaning(message):
    punc_removed = [char for char in message if char not in string.punctuation]
    punc_removed_join = ''.join(punc_removed)
    punc_removed_join_clean = [word for word in punc_removed_join.split() if word.lower() not in stopwords.words('englis
    return punc_removed_join_clean
```

```
tweets_df_cleaned = new_tweets_df['tweet'].apply(tweet_cleaning)
```

```
# print(tweets_df_cleaned[5])
```

Task 6 : Vectorization and Tokenization

```
vectorizer = CountVectorizer(analyzer = tweet_cleaning)
tweet_countvectorizer = CountVectorizer(analyzer=tweet_cleaning, dtype='uint8').fit_transform(new_tweets_df['tweet']).to
```

```
tweet_countvectorizer.shape
```

```
(73995, 51233)
```

```
X = tweet_countvectorizer
```

```
Y = new_tweets_df['label']
```

Task 7 : Creating a Classifier

Creating A Naive Bayes Classifier

```
1 X.shape
```

```
0]: (73995, 51233)
```

```
1 Y.shape
```

```
1]: (73995,)
```

```
1 from sklearn.model_selection import train_test_split
2 X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.2)
```

```
1 from sklearn.naive_bayes import MultinomialNB
2 NB_classifier = MultinomialNB()
3 NB_classifier.fit(X_train,Y_train)
```

```
3]: MultinomialNB()
```

Task 8 : Creating a Confusion Matrix and making

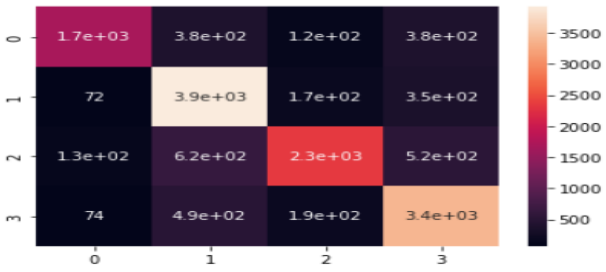
Predictions

```
1 from sklearn.metrics import classification_report, confusion_matrix
```

```
1 Y_Predict_Test = NB_classifier.predict(X_test)
```

```
1 cm = confusion_matrix(Y_test,Y_Predict_Test)
2 sns.heatmap(cm, annot=True)
```

```
]: <AxesSubplot:>
```




```
1 print(classification_report(Y_test,Y_Predict_Test))
```

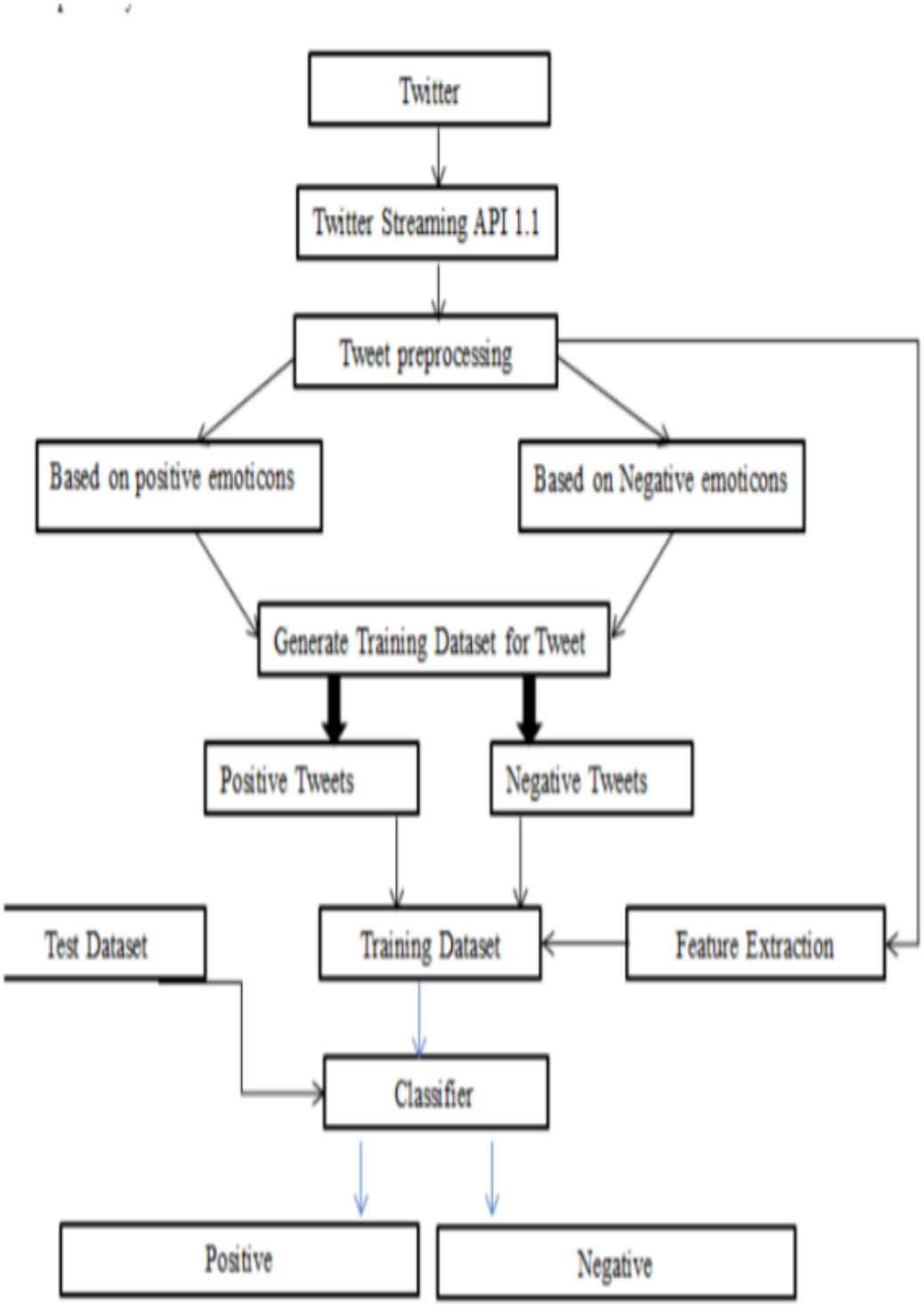
```
              precision    recall  f1-score   support

           0       0.86      0.65      0.74      2556
           1       0.72      0.87      0.79      4502
           2       0.83      0.65      0.73      3606
           3       0.73      0.82      0.77      4135

    accuracy                           0.76     14799
   macro avg       0.79      0.75      0.76     14799
weighted avg       0.77      0.76      0.76     14799
```

```
1
```

Structure Chart



Chapter 4

Result and Discussion

4.1 SYSTEM CONFIGURATION

Intel I7 evo 11th gen processor with 16 GB Ram

Inbuilt Graphic Processor

It also has 4 cores which runs at @ 2.80GHz each

Tokenizer take about 4 to 5 minutes to execute.

Classifier takes around 6-7 minutes to execute.

4.1.1 SOFTWARE REQUIRMENTS

1. Anaconda
2. Jupiter Note Book
3. Nltk
4. Scikit-learn
5. Matplotlib
6. Tweepy
7. Pandas
8. Numpy
9. TextBlob
10. Re(Regular Expressions)
11. Wordcloud
12. StopWords
13. Windows

Chapter 5

Conclusion and Future Work

5.1 CONCLUSION

We furnished results for Sentiment and Emotional Analysis on twitter data . On applying Naive Bayes for sentiment analysis with **76.0% accuracy at test_split=0.2**.

Much practical values we achieved the **precision 73 % recall 82 % and the f1 score of 77%**.

5.2 FUTURE WORK

In future work , we aim to handle emoticons , dive deep into emotional analysis to further detect idiomatic statements .We will also explore richer linguistic analysis such as parsing and semantic analysis.

References

[1] Youtube Tutorials by simplilearn

[2] Youtube Tutorials by edureka

[3] Youtube Tutorials on statistics and Maths by statquest

[4] developer guide of pandas, numpy, nltk, python, wordcloud, stopwords, sk-learn.

[5] javatpoint site for further refining of concepts

[6] tutorials by codewithharry

[7] w3schools tutorials

[8] gfg tutorials