TWITTER SENTIMENTAL ANALYSIS

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF COMPLETION
OF
INTERNSHIP PROJECT

BACHELOR OF TECHNOLOGY
IN
COMPUTER ENGINEERING

Submitted by:

SUNIL SAHU

2K18/CO/363

Under the supervision of

Ram Murti Rawat

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-110042

CANDIDATE'S DECLARATION

I, ,Sunil sahu 2K18/CO/363, student of B.Tech (COMPUTER Engineering), hereby declare that the project Dissertation titled "Twitter sentimental analysis" which is submitted by me to the Department of Computer Science and Enginnering, Delhi Technological University, New Delhi in partial fulfilment of the requirement for the award of the certificate of completion of INHOUSE internship . It is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Place: New Delhi **SUNIL SAHU**

2K18/CO/363

Date: **31st July 2021**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Twitter sentimental analysis" which is submitted by Sunil sahu, 2K18/CO/363, Department of Computer Science and Enginnering, Delhi Technological University, New Delhi in partial fulfilment of the requirement for the award of the certificate of completion of INHOUSE internship , is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: New Delhi

Date: 31st July 2021

Ram Murti Rawat

SUPERVISOR

Assistant Professor

Department of CSE

Delhi Technological University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-110042

ACKNOWLEDGMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

I am highly indebted to DTU and especially Asst.Prof. Ram Murti Rawat, my Faculty Advisor, for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

Thank you

ABSTRACT

Social networking sites are the obsolete assets to collect information regarding beliefs and sentiments of different peoples living in different aspects of the society towards different subjects as many of them spend hours daily on social medias and share their opinion. In this specialized paper, we will clarify the utilization of nostalgic investigation and how we can associate with Twitter and run wistful examination inquiries. We review investigates various inquiries from governmental issues to mankind and display the entrancing result. We have understood that the target feeling for tweets is amazingly high from which we can unmistakably see the restrictions of the current situation in this field.

Keywords—Twitter, Sentiment Analysis (SA), text mining, Machine learning, Naive Bayes (NB), Bayesian Algorithm, Social Media, Maximum Entropy

List of content

CHAPTER 1

1.1 INTRODUCTION

Microblogging sites have modernized the cycle through which individuals express their musings, assessment and offer their day by day lives. It is currently getting refined through various microblogging sites like Twitter, Facebook, Google in addition to and so forth.

Sentimental investigation has numerous applications for various spaces for instance in organizations to get inputs for items by which organizations can become familiar with clients' criticism and audits on social medias and gives a stage to various organizations to interface with their clients through commercials. Individuals over yonder primarily trust on client created substance generally for getting to a choice. For example in the event that any individual is eager to purchase an item or needs to overcome a help, above all else they search for the items audits on the web, examine about it on long range interpersonal communication destinations prior to getting to a choice.

The tremendous substance produced by an extraordinary number clients is hard to break down by a typical. Along these lines, unmistakably there is an extraordinary need to automatize this strategy, different slant investigation procedures are generally acknowledged. Note the primary test is to design an innovation which won't just distinguish yet will likewise sum up the notions of various clients.

In this paper, we will construct a Naive bayes model for arranging "tweets" into positive and negative supposition. We manufacture a model for arrangement: a double errand of ordering assumption into positive and negative classes. And furthermore to examine existing supposition investigation techniques for Twitter information and give hypothetical correlations of the condition of-workmanship draws near.

1.2 Definition

Sentiment analysis is a procedure for checking appraisals of various individuals or groups; for instance, a part of a brand's adherents or an individual client in correspondence with a client service agent. Relating it with a scoring instrument, assumption investigation screens countless conversations shared by countless clients and evaluates discourse and voice gestures to examine temperaments, emotions and conclusions, particularly of the individuals who are related with a business, item or administration, or topic

1.3 Motivation

Sentiment analysis is a path through which one can give composed or communicated in dialects to review whether enunciation is positive or negative. The current existing and acknowledged examination apparatuses in the market can manage enormous volumes of client analysis dependably and precisely.

Tremendous volumes of customer made electronic interpersonal interaction correspondences are in effect relentlessly conveyed and acknowledged in various structures like overviews, online diaries, remarks, talks, pictures, and accounts. These correspondences offer surprising occasions to acquire and grasp the perspectives of customers on subjects, for example, interest and format information outfitted for explaining and foreseeing business and social news, for example, item offers, stock returns, and the aftereffects of political choices. Intrinsic with these assessments is the appraisal of the ideas traded between the customers in their substance exchanges fiery customers in a given month, likewise including 100 million customers day by day. Customers beginning from various corners of the world, with 77% arranged outside of the US, producing in excess of 500 million tweets every day. The Twitter site got a twelfth situation for movement in 2017 worldwide and responded to in excess of 15 billion API calls every day. Twitter content what's more turn up in more than 1,000,000 non-part locales. In adaptation with this huge turn of events, Twitter has been the subject of much investigation from the exceptionally past, as Tweets often pass on customer's estimation and conclusions on disputable issues. In the web-based media setting, feeling investigation and mining assessments are exceptionally demanding errands, and this is a result of the enormous data created by people and machines

1.4 PROPOSED SYSTEM

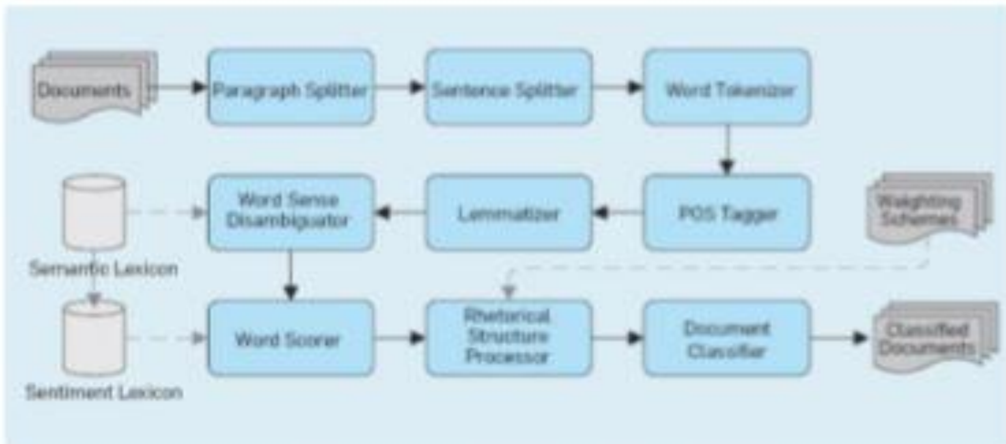


Fig 1.steps in sentiment classification

1.5 ARCHITECTURE OF TWITTER SENTIMENTAL ANALYSIS

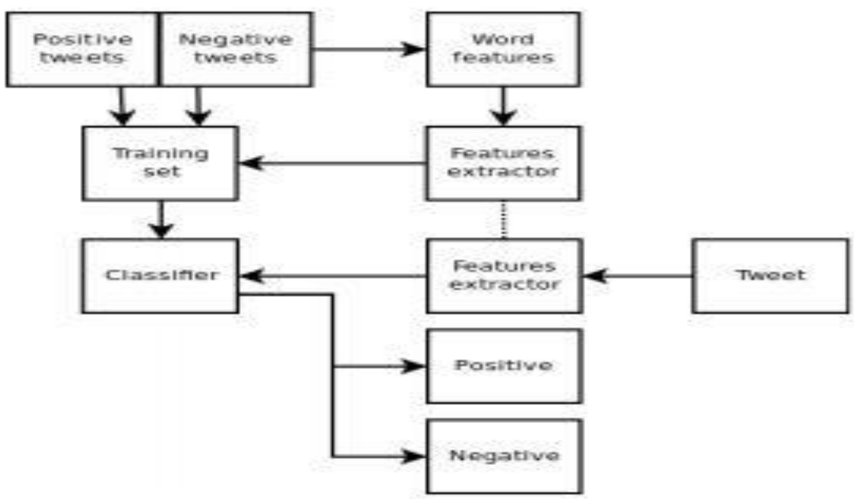


Fig.1. Sentiment Analysis Architecture

Following are the phases required for sentiment analysis of twitter data,

1.6 Understanding the problem statement

TASK #1: UNDERSTAND THE PROBLEM STATEMENT AND BUSINESS CASE

TASK #2: IMPORT LIBRARIES AND DATASETS

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
# Load the data
tweets_df = pd.read_csv('train.csv')
```

In [3]:
```
tweets_df
```

Out[3]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
...	...	...	...
31957	31958	0	ate @user isz that youuu?ðððððð...
31958	31959	0	to see nina turner on the airwaves trying to...
31959	31960	0	listening to sad songs on a monday morning otw...
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...
31961	31962	0	thank you @user for you follow

31962 rows × 3 columns

```
In [4]: tweets_df.tweet
```

```
Out[4]: 0         @user when a father is dysfunctional and is s...
1        @user @user thanks for #lyft credit i can't us...
2                                   bihday your majesty
3        #model   i love u take with u all the time in ...
4                  factsguide: society now    #motivation
                              ...
31957    ate @user isz that youuu?ðððððð...
31958      to see nina turner on the airwaves trying to...
31959    listening to sad songs on a monday morning otw...
31960    @user #sikh #temple vandalised in in #calgary,...
31961                    thank you @user for you follow
Name: tweet, Length: 31962, dtype: object
```

```
In [5]: tweets_df = tweets_df.drop(['id'] , axis=1)  #asis means that we are droping the entire column
```

```
In [6]: tweets_df
```

Out[6]:

	label	tweet
0	0	@user when a father is dysfunctional and is s...
1	0	@user @user thanks for #lyft credit i can't us...
2	0	bihday your majesty
3	0	#model i love u take with u all the time in ...
4	0	factsguide: society now #motivation
...	...	...
31957	0	ate @user isz that youuu?ðððððð...
31958	0	to see nina turner on the airwaves trying to...
31959	0	listening to sad songs on a monday morning otw...
31960	1	@user #sikh #temple vandalised in in #calgary,...
31961	0	thank you @user for you follow

31962 rows × 2 columns

1.7 data description

In the given program, we gather the dataset from Kaggle prepared model on the given datasets only.

Dataset Resource:- https://www.kaggle.com/arkhoshghalb/twitter-sentiment- analysis-hatred-speech

As clarified Twitter is a social webbing and microblogging site that licenses account holders to share ongoing messages, otherwise called tweets. Tweets can be characterized as little messages, of around 140 letters long. As a result of the class of the above-clarified microblogging facility(fast and little messages), clients regularly use abbreviations, commit a great deal of spelling errors, utilizes a quantity of emojis, and various different highlights which displays explicit implications. After this data in this para, we have a concise wording identified with tweets.

1 Emoticon: Can be clarified as outward appearances pictorially delineated utilizing accentuation and character that is to state they communicate the client's temperament.

2 Target: Account holders of Twitter utilize the "@" image to relate or nail it to some other record holder to the stage. Referencing other record holders in this manner alarms the individual holder all alone.

3 Hashtags: Holders frequently use hashtags to light up the connected subjects. This is generally done to grow the permeability of their tweets.

For the given Assignment we require 2 Parameters:

1. The tweets itself
2. The labels or matter associated with each tweet. Now We plotted a graph for the amount of the tweets to further understand the dataset

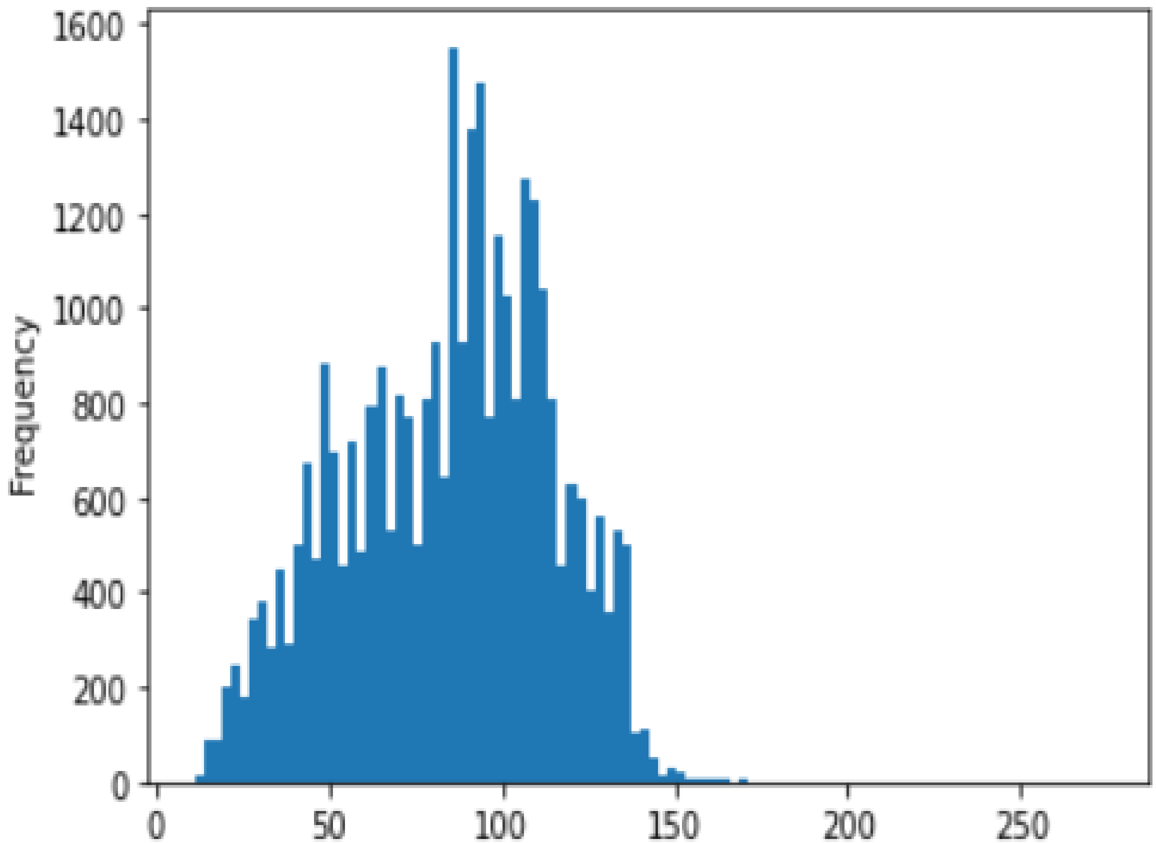


Fig 1: Histogram : showing us the range of the amount of tweets in the dataset

Our dataset had approximately 32000 tweets, which were labelled either as ‘0’ or ‘1’, where ‘0’ showing positive tweets and ‘1’ showing negative tweets.

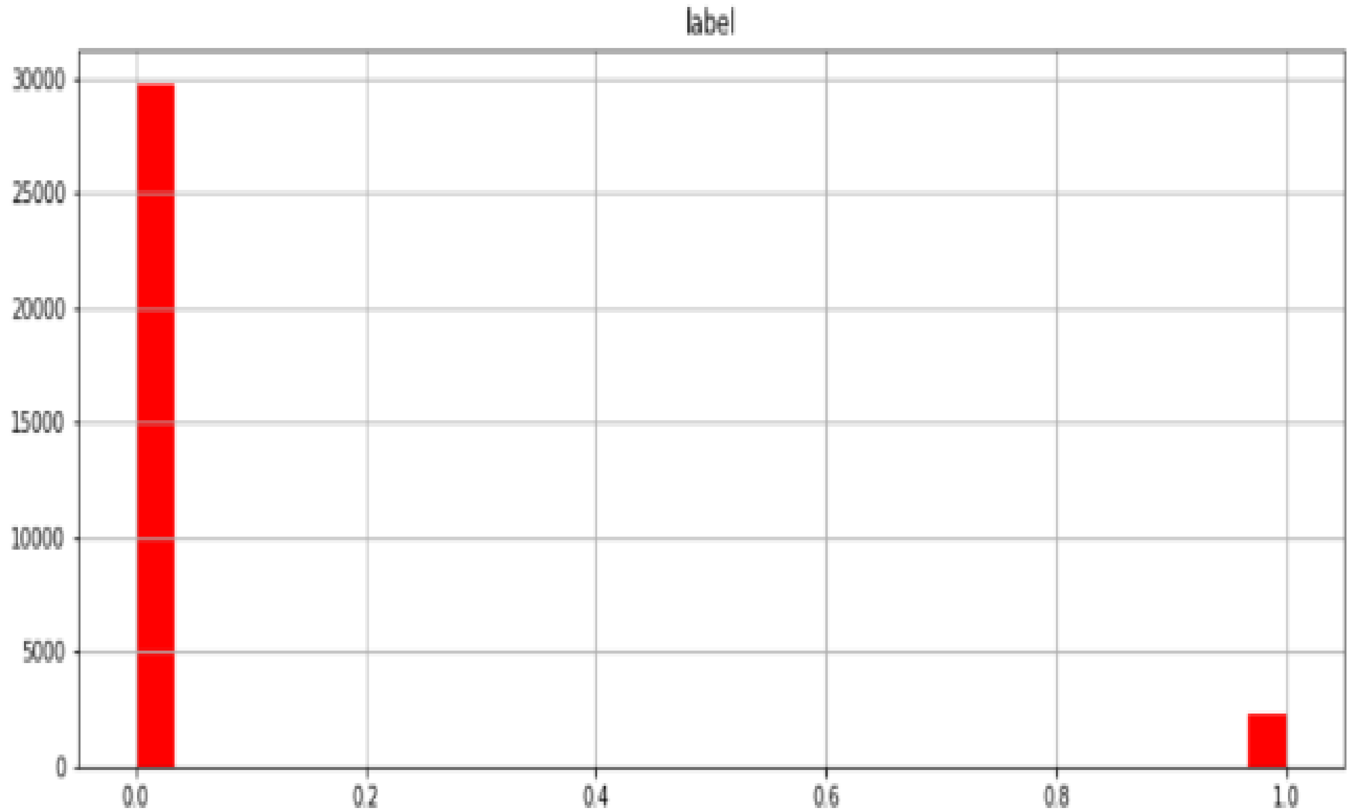


Fig 2: Histogram : Showing us the range of tweets in the dataset, tweets with label of 0 are positive and the tweets with label of 1 are negative

Table 1 is showing the distribution of tweets as positive and negative

Distribution of tweets	**%**
29720 (0)	92.98
2242 (1)	7.02

Table 1. Distribution of tweets

Table 2 shows the sample positive tweets

S.NO.	LABEL	TWEET	LENGTH
1	0	@user when a father is dysfunctional and is s...	102
2	0	@user @user thanks for #left credit i can't us...	122
3	0	bidet your majesty	21
4	0	#model I love u take with u all the time in ...	86
5	0	facts guide: society now #motivation	39

Table 2. Sample positive(0) tweets.

Figure 3 shows the word cloud plot of positive tweets

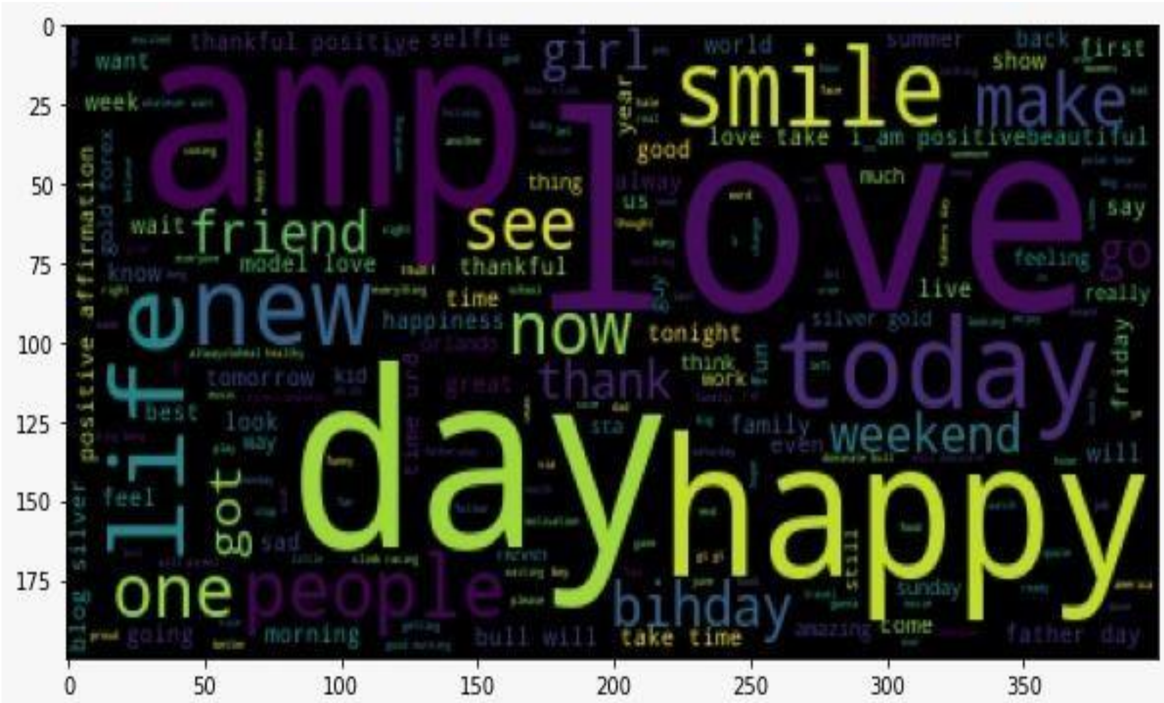


Fig. 3: Word cloud plot of positive tweets

Table 2 shows the sample positive tweets.

s.no	label	tweet	length
1	1	@user #cnn calls #michigan middle school 'buil...	74
2	1	no comment! in #australia #opkillingbay #se...	101
3	1	retweet if you agree!	22
4	1	@user @user lumpy says i am a . prove it lumpy.	47
5	1	it's unbelievable that in the 21st century we'...	104

Table 3. Sample Negative(1) tweets.

Figure 3 shows the word cloud plot of negative tweets

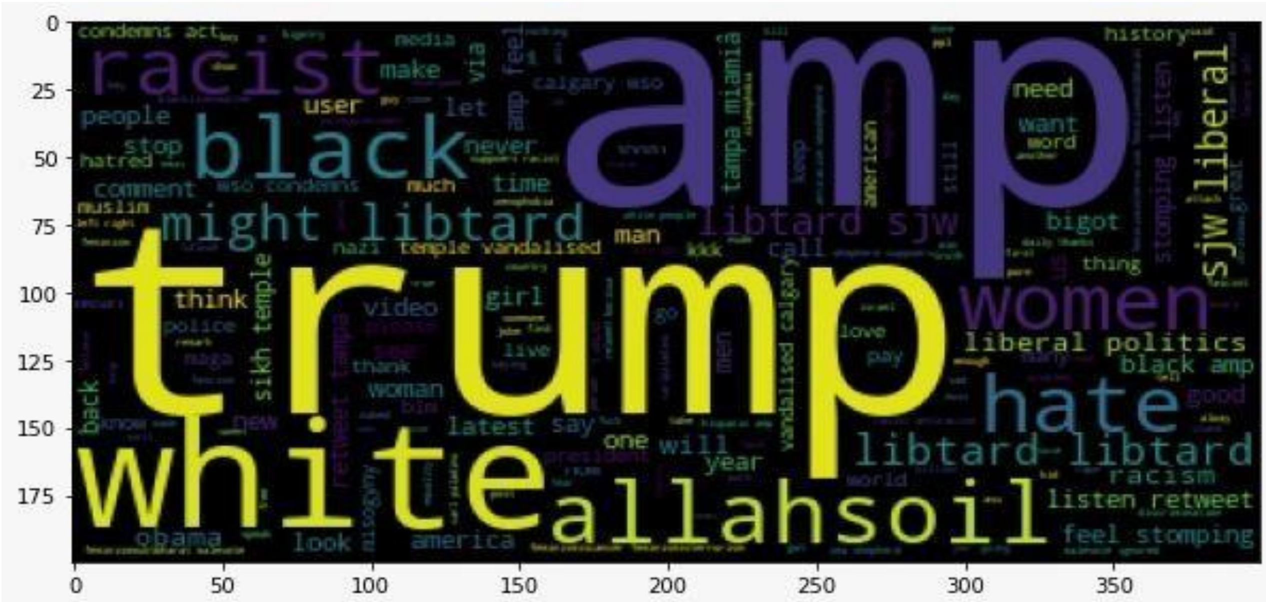


Fig. 4: Word cloud plot of negative tweets

Task #4 : This histogram shows us the range of the dataset The tweets with label of 0 are positive and the tweets with label of 1 are negative

```
In [ ]:

In [73]: # fig, ax = plt.subplots(figsize=(10,10))
         tweets_df.hist(bins = 30, figsize = (13,7), color = 'r')

Out[73]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000016B172477C0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000016DEB40EDC0>]],
               dtype=object)
```

In [9]:
```
# Let's get the length of the messages
tweets_df['length'] = tweets_df.tweet.apply(len)
```

In [10]:
```
tweets_df
```

Out[10]:

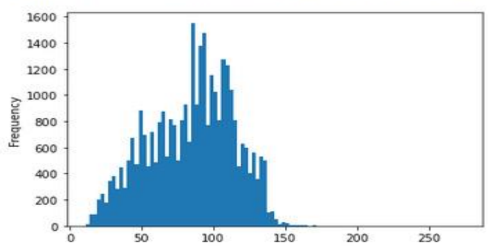
	label	tweet	length
0	0	@user when a father is dysfunctional and is s...	102
1	0	@user @user thanks for #lyft credit i can't us...	122
2	0	bihday your majesty	21
3	0	#model i love u take with u all the time in ...	86
4	0	factsguide: society now #motivation	39
...	...	...	...
31957	0	ate @user isz that youuu?ðððððð...	68
31958	0	to see nina turner on the airwaves trying to...	131
31959	0	listening to sad songs on a monday morning otw...	63
31960	1	@user #sikh #temple vandalised in in #calgary,...	67
31961	0	thank you @user for you follow	32

31962 rows × 3 columns

Task #5 : This histogram gives us the range of the lengeth fo the tweets in the dataset

In [74]:
```
# fig, ax = plt.subplots(figsize=(20,20))
tweets_df['length'].plot(bins=100, kind='hist')
```

Out[74]: <matplotlib.axes._subplots.AxesSubplot at 0x16decf0df40>



2.1 Experimental design

2.1 A) Holdout Validation Technique -

We broke the dataset into

1.Training set of 25,570(80%) tweets and

2.Testing set of 6,392(20%) tweets

2.1 B) Performance Metrics -

Area Under Curve (AUC), Precision, Recall shows better instinct of results in contrast with accuracy. We can calculate Precision, Recall, F1-score by the beneath indicated equations once Confusion Matrix is framed.

We utilized 4 strategies for assessment measurements for our Dataset.

a) Accuracy is characterized as the proportion of genuine expectation to the all outnumber of tweets in the set.

$$acc = (TP + TN)/Total\ Population$$

b) Precision is the portion of anticipated extremity that are accurately perceived. Precision is characterized as –

$$P = TP/(TP + FP)$$

c) Recall just is the division of no. of the right prediction of a specific polarity over the real no. of tweets with that polarity. Recall is characterized as

$$-R = TP/(TP + FN)$$

d) F1-score is characterized as the harmonic mean of Recall(R) and Precision(P) and numerically equation is given by –

$$f1 - score = (2 \times R \times P)/(R + P) \text{ TP}$$

TP is no. of true positives i.e., number of right ordered tweets, FN is no. of false negatives, and FP is the no. of false positives.

2.1C) Pre-processing of the dataset

A tweet has an extraordinary number of discernment comparable to the information which are communicated in isolated manners by an incredible number of clients. The twitter dataset clarified in the given overview appraisal is as of now marked into two classes that is negative and positive extremity and accordingly examination of the feelings of the information came out to be anything but difficult to consider the consequence of different highlights. The crude information having extremity is exceptionally defenseless to incontinence and repetition.

Preprocessing of tweet include the following points,,

- Removing frequent Words
- Removing Stop words
- Removing Punctuations

```
tweets_df.tweet[0]    # Before Cleaning
```

```
' @user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction.   #run'
```

```
tweets_df_clean[0]     # After Cleaning
```

```
['father', 'dysfunctional', 'selfish', 'drags', 'kids', 'dysfunction', 'run']
```

We can Clearly see the difference between the tweet before and after cleaning

2.1 D) ALGORITHM

NAÏVE BAYES

Naive Bayes classifiers are a gathering of order calculations dependent on Bayes' Theorem. It is simply not a solitary calculation but rather a major gathering of calculations where the entirety of the calculations have a typical guideline, that is each pair of highlights being characterized is f ree of one another.

Bayes' Theorem look through the odds of a result happening given the odds of another result that has just happened.

Bayes' hypothesis is expressed numerically as the accompanying condition:

$$p(a/b) = \{ p(b/a) * p(a) \} / p(b)$$ (Where a and b are events)

- In short, we are trying to search for the chances of outcome A, given the event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e. Chances of outcome before evidence is seen). The evidence is actually an attribute value of an instance which is not known(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

TASK #10: TRAIN A NAIVE BAYES CLASSIFIER MODEL

```
In [46]: X.shape
Out[46]: (31962, 47386)

In [47]: Y.shape
Out[47]: (31962,)

In [48]: from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)

In [49]: from sklearn.naive_bayes import MultinomialNB

         NB_classifier = MultinomialNB()
         NB_classifier.fit(X_train, Y_train)
Out[49]: MultinomialNB()
```

Count Vectorization:

When we find out any model in data science, they can only be expressed in numeric inputs.

Count Vectorization does the same function, it transforms a collection of words into an array of binary numbers.

Count Vectorization involves counting the exact number of outcomes each character appears as a document (i.e., separate text for example as an article, book, even a paragraph!).
Python's Sci-kit learn library has a tool called Count Vectorizer to illustrate this, which we have used in this project.

Example sentence: "The weather was wonderful today and I went outside to enjoy the beautiful and sunny weather".

the	weather	was	wonderful	today	and	I	went	outside	to	enjoy	beau
2	2	1	1	1	2	1	1	1	1	1	1

Table 4: Array of numbers of the sample sentence

This is the first paper. [[0 1 1 1 0 0 1 0 1]

This paper is the second paper. [0 2 0 1 0 1 1 0 1]

And this is the third one. [1 0 0 1 1 0 1 1 1]

Is this the first paper? [0 1 1 1 0 0 1 0 1]]

	'and'	paper'	'first'	'is'	'one'	'second'	'the'	'third'	'this'
Training Sample #1	0	1	1	1	0	0	1	0	1
Training Sample #2	0	2	0	1	0	1	1	0	1
Training Sample #3	1	0	0	1	1	0	1	1	1
Training Sample #4	0	1	1	1	0	0	1	0	1

TASK #8: CREATE A PIPELINE TO REMOVE PUNCTUATIONS, STOPWORDS AND PERFORM COUNT VECTORIZATION

In [33]:
```
import string
import nltk # Natural Language tool kit
nltk.download('stopwords')
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
```

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

In [34]:
```
# Let's define a pipeline to clean up all the messages
# The pipeline performs the following: (1) remove punctuation, (2) remove stopwords

def message_cleaning(message):
    Test_punc_removed = [char for char in message if char not in string.punctuation]
    Test_punc_removed_join = ''.join(Test_punc_removed)
    Test_punc_removed_join_clean = [word for word in Test_punc_removed_join.split() if word.lower() not in stopwords.words('eng
    return Test_punc_removed_join_clean
```

In [35]:
```
# Let's test the newly added function
tweets_df_clean = tweets_df['tweet'].apply(message_cleaning)
```

In [36]:
```
print(tweets_df_clean[5]) # show the cleaned up version
```

```
['22', 'huge', 'fan', 'fare', 'big', 'talking', 'leave', 'chaos', 'pay', 'disputes', 'get', 'allshowandnogo']
```

In [37]:
```
print(tweets_df['tweet'][5]) # show the original version
```

```
[2/2] huge fan fare and big talking before they leave. chaos and pay disputes when they get there. #allshowandnogo
```

In [38]:
```
from sklearn.feature_extraction.text import CountVectorizer
# Define the cleaning pipeline we defined earlier
vectorizer = CountVectorizer(analyzer = message_cleaning)
tweets_countvectorizer = vectorizer.fit_transform(tweets_df['tweet'])
```

In [39]:
```
tweets_countvectorizer.shape
```

Out[39]:
```
(31962, 47386)
```

In [40]:
```
X = tweets_countvectorizer
```

In [41]:
```
Y = tweets_df['label']
```

In [42]:
```
print(vectorizer.get_feature_names())
```

3.1 Training the Model

Parameters That go in the model:

1. X = the count vectorizer of the tweets
2. Y = the labels associated with each tweet

Validation Technique used: Holdout with test size as 0.2

Multinomial NB Classifier Used:

NB_classifier = MultinomialNB()

NB_classifier.fit(X_train, Y_train)

TASK #11: ASSESS TRAINED MODEL PERFORMANCE

In [50]:
```
from sklearn.metrics import classification_report, confusion_matrix
```

In [75]:
```
# Predicting the Test set results
# fig, ax = plt.subplots(figsize=(18,20))
Y_predict_test = NB_classifier.predict(X_test)
cm = confusion_matrix(Y_test, Y_predict_test)
sns.heatmap(cm, annot=True)
```

Out[75]: <matplotlib.axes._subplots.AxesSubplot at 0x16decbe1040>

3.2 RESULT

We took 25,570 tweets to prepare our model in the teaching set, and experimented on 6,392 tweets All the representing measures calculated by the experimented set. And the precision reported on the experiment set was around 94% which is pretty awesome in relation to others.

3.2A) CONFUSION MATRIX

On the dataset It is seen from Fig.5, that our model is showing amazing results on both the polarities. Hence this proves the high precision gained on our dataset.Our model achieved a test accuracy of 94%. Hence it is much precision and doesn't confuse between classes.

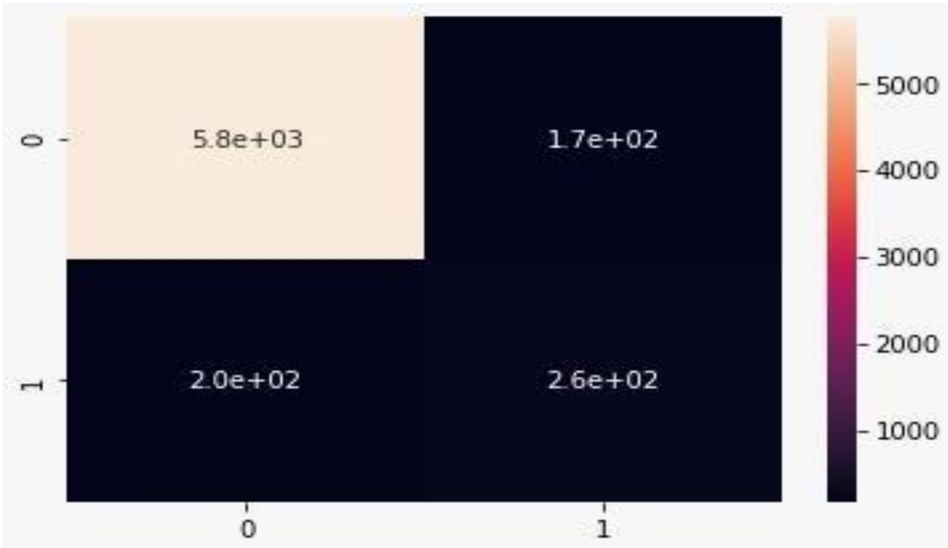


Fig. 5: The confusion matrix on dataset (testing set)

3.2.B) AREA UNDER CURVE

AUC is the best possible way to access a trained machine learning model..Basically it is the plot between the True Positive Rate and the False Positive Rate, And the AUC Score is Calculated by the area under that curve.

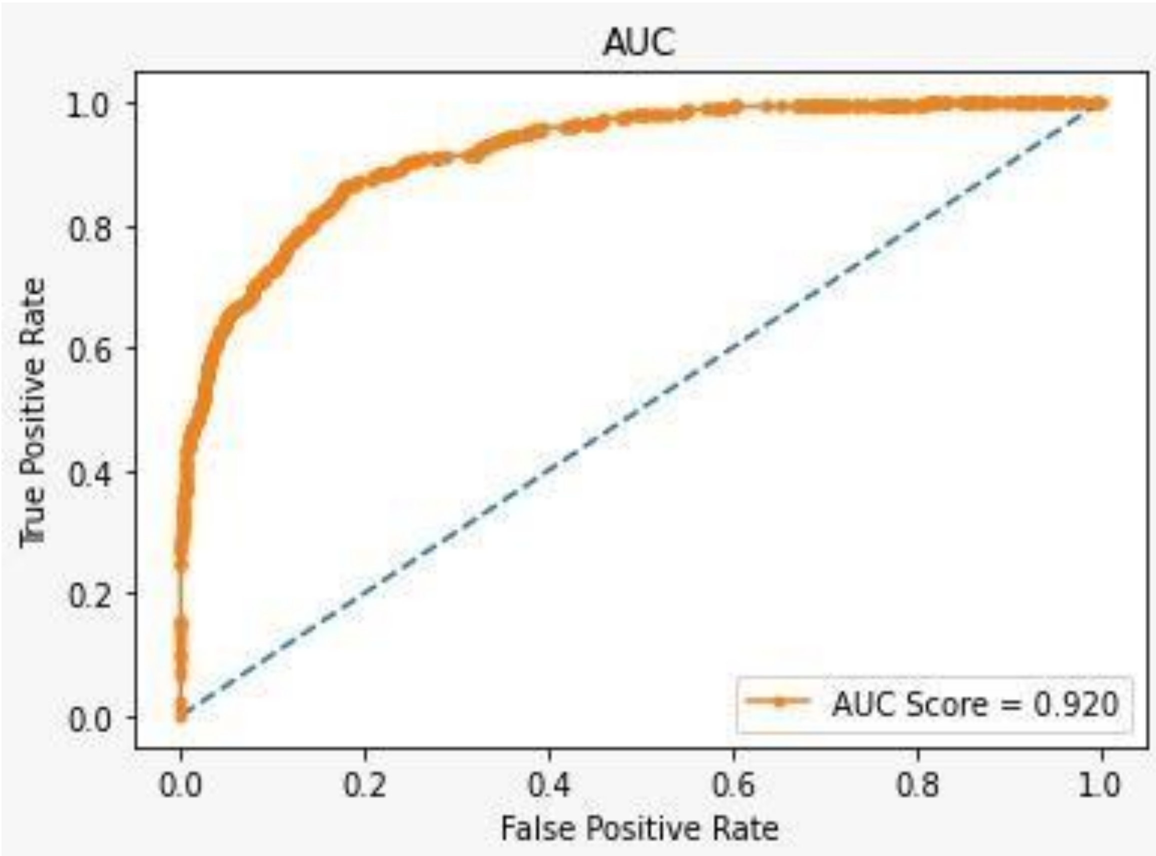


Fig. 6: AUC (Area Under Curve)

So here for our model we have an AUC Score of 0.920 which is considered to be extremely good

3.2. C)COMPLETE CLASSIFICATION REPORT

	precision	Recall	F1-score	Support
0	0.97	0.97	0.97	5925
1	0.60	0.56	0.58	468
Accuracy			0.94	6393
Macro avg	0.78	0.77	0.78	6393
Weighted avg	0.94	0.94	0.94	6393

TABLE 5: Classification accuracies, precision, recall and f1-score for dataset model.

4.1 CONCLUSION

In the given paper, we clarified the need of long-range interpersonal communication examination and its applications in isolated class. We aimed Twitter and have engendered Naive Bayes algorithm to clarify wistful examination. Thus, program will arrange assessment into either certain or negative, which is illustrated in Histogram fig 2. We were able to cross benchmark precision and gained a test accuracy of 94% with our algorithm , which is wonderful in comparison with different papers. Research results show that machine learning methods, such as SVM and naive Bayes have the highest precision and can be viewed as the standard learning techniques. We likewise examined the impacts of various characters on classifier. We can infer that more the cleaner data, more précised result can be acquired…

4.2 REFRENCES

1. A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326

2. R. Parikh and M. Movassate, "Sentiment Analysis of User- Generated Twitter Updates using Various Classi_cation Techniques",CS224N Final Report, 2009

3. Go, R. Bhayani, L.Huang. "Twitter Sentiment Classification Using Distant Supervision". Stanford University, Technical Paper,2009

4. Aliza Sarlan, Chayanit Nadam and Shuib Basri, "Twitter Sentiment Analysis", 2014 International Conference on data Technology and Multimedia (ICIMU), Putrajaya, Malaya Gregorian calendar month eighteen – twenty, 2014.

5. Alexander Pak, Apostle Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion

Mining”, Proceedings of the International Conference on Language Resources and analysis,
6. LREC 2010, 17-23 could 2010, Valletta, Malta,2010
7. Mining Twitter information with Python (Part half-dozen – Sentiment Analysis Basics), marcobonzanini 2015, *https://marcobonzanini.com/2015/05/17/miningtwitter- data-* link "*https://marcobonzanini.com/2015/05/17/miningtwitter- data-with-python-part-6-sentiment-analysisbasics/*" with- python-part-6-sentiment-analysis- basics/
8. Naive Bayes for Sentiment Analysis, Medium Corporation,2018, *https://medium.com/@martinpella/naivelink"https://mediu m.com/@martinpella/naive-bayesfor-sentiment-analysis- 49b37db18bf8*" bayes-forsentiment-analysis-49b37db18bf