Machine Learning and Pattern Recognition
Project - Twitter Sentimental Analysis

Team :

Aniket Wani (B19EE009)
Harpal Sahil Santosh (B19CSE107)
Aanchal (B19EE001)

ABSTRACT

In today's era, everyone is expressive in one way or other. Many social websites and android applications whether being Facebook, WhatsApp or Twitter, in this highly advance world is flooded with views and data. One of the most global and popular platforms is Twitter. This is seen as the main source of sentiments where almost every enthusiastic or social person tends to express his or her views in form of comments. These comments not only express the people but also give the understanding of their mood. Text present on these medias are unstructured in nature, so to process them firstly we need to pre-process, pre-processing techniques are used and then features are extracted from the pre-processed data.

There are so many feature extraction techniques such as Bag of Words, TF-IDF,N-Grams, word embedding, NLP(Natural Language Processing) based features like word count, length count, unique words,poistive tweet , negative tweet etc.
Analysis is done using two classification algorithms (Logistic Regression, Naive Bayes) and considering F1-Score , Accuracy, Precision, and Recall performance parameter.

Introduction

Twitter is the most popular microblogging where users create status message called tweets. These tweets express opinion about certain topics. We, here, propose a method to extract sentiment from the tweets automatically. This proves to be very useful because it can be used to provide feedback without manual intervention. Twitter sentiment is used to classify the tweets into neutral, positive, or negative. Text classification is the process of processing the text data

generated on social media for various applications such as email categorization, web search, topic modeling, and information retrieval.

We can undergo sentiment analysis with distant supervision. Our solution is to use distant supervision in which our training data consists of tweets with emoticons. We run the classifier trained on data containing emoticons against the test sets of tweets. We, then, present the result of experiments and our thoughts on how to further improve our results.

Our approach is to use different machine learning classifiers and feature extractors. The machine learning classifiers are Naive Bayes, Logistic Regression. We build a framework that treats classifiers and feature extractors as two distinct components. This framework allows us to easily try out different combinations of classifiers and feature extractors.

Dataset

This is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter api. The tweets have been annotated as (0=negative, 4=positive) and they can be used to detect sentiment.
There are following 6 fields present in database:

i) target :
Gives polarity of the tweet (negative=0, neutral=2, positive=4).

ii) ids:
This gives unique ids of the tweet.

iii) date:
The date of the tweet.

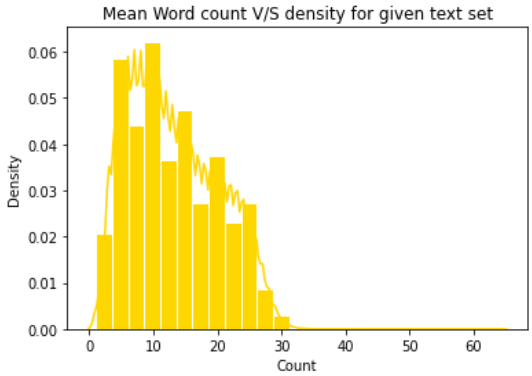
iv) flag:
The query of the tweet. If there is no query, then this value is NO_QUERY.

v) User:
The user that tweeted.

vi) text: the text of the tweet.

Characteristics of Tweets

Twitter messages have many unique attributes, which differentiates our research from previous research:
Length : The maximum length of a Twitter message is 140 characters. From our training set, we calculate that the average length of a tweet is 14 words or 74 characters.



This shows that there are higher number of tweet that has word count of 10.

Data availability : Another difference is the magnitude of data available. With the Twitter API, it is very easy to collect millions of tweets for training. In past research, tests only consisted of thousands of training items.

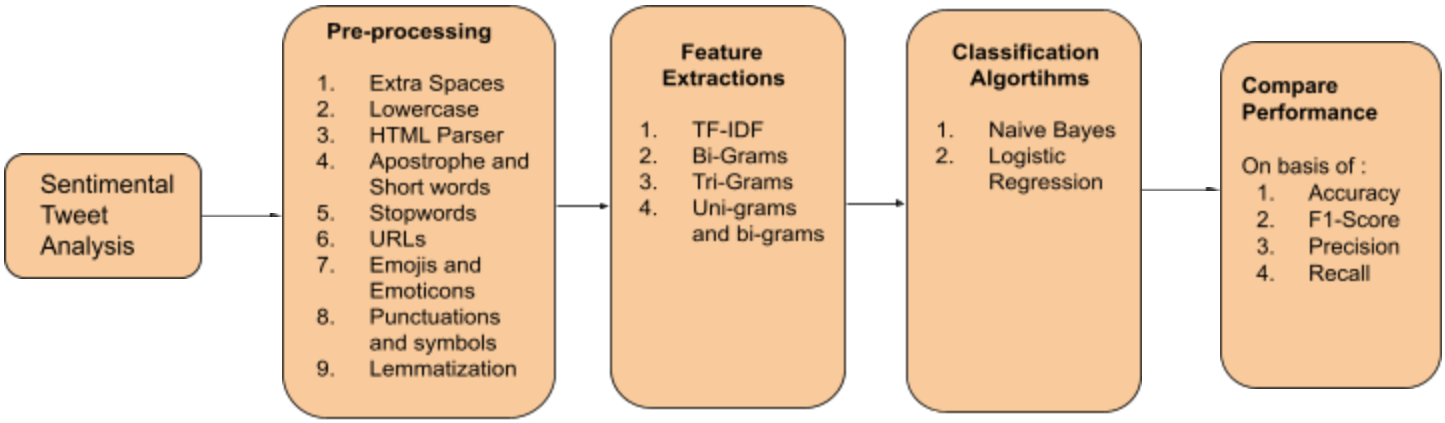
Language model : Twitter users post messages from many different media, including their cell phones. The frequency of misspellings and slang in tweets is much higher than in other domains.

Domain : Twitter users post short messages about a variety of topics unlike other sites which are tailored to a specific topic. This differs from a large percentage of past research, which focused on specific domains such as movie reviews.

Our Approach

Our approach is to use different machine learning classifiers and feature extractors. The machine learning classifiers are Naive Bayes,Logistic Regression. The feature extractors are unigrams, bigrams,tri grams, unigrams and bigrams (1,2) . We build a framework that treats classifiers and feature extractors as two distinct components. This framework allows us to easily try out different combinations of classifiers and feature extractors.

Machine learning Pipeline



Pre-processing

1) Removing Extra Spaces :

"tree yhft hdygd bgft" --> tree yhft hdygd bgft

2) Apostrophe and Short words :

"Can't" --> "cannot"

3) Removing Stop Words :

Stop words refer to most common words in the English language which doesn't have any contribution towards sentiment analysis. Some of the stop words are "are", "of", "the", "at" etc. So, these need to be eliminated.

4) Removing URLs :

"Driverless AI NLP blog post on https://www.h2o.ai/blog/detecting-sarcasm-is-difficult-but-ai-may-have-an-answer/" ----> `Driverless AI NLP blog post on`

5) Removing Emojis and Emoticons :

"progress is on 🔥🔥" -----> progress is on

6) Removing Punctuations :

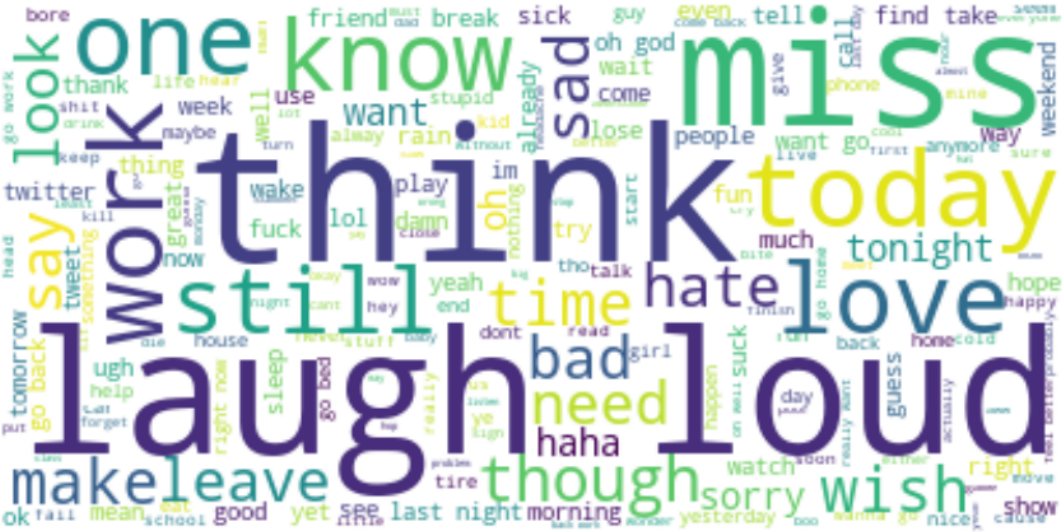
"hey, hello!" ---> hey hello

7) Lemmatization : Lemmatization is the process of merging two or more words into single word. This analyses the word morphology and eliminates the ending of the word like shocked to shock, caught to catch etc.

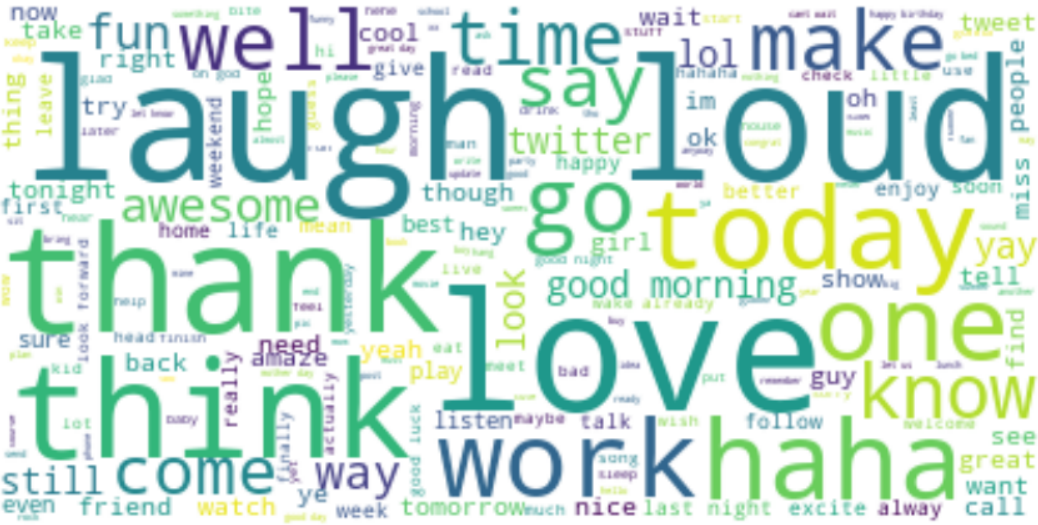
"crying testing ablaze" --> cry test ablaze

Feature Extraction and Visualization

Word Cloud of Negative Class:



Word Cloud of Positive Class:



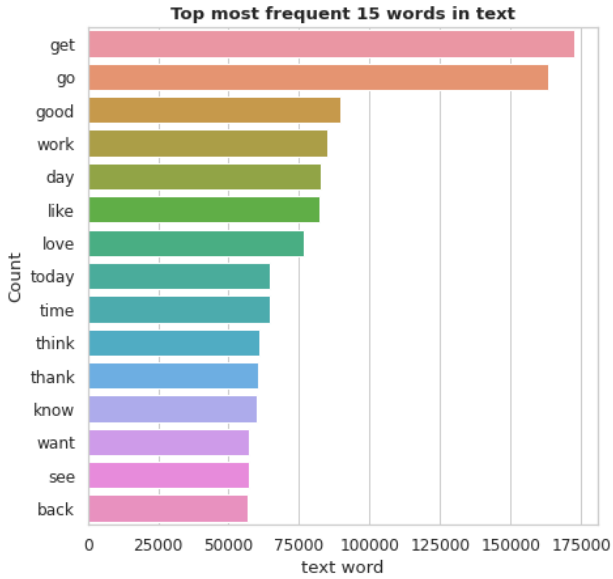
Total Unique words : 753901

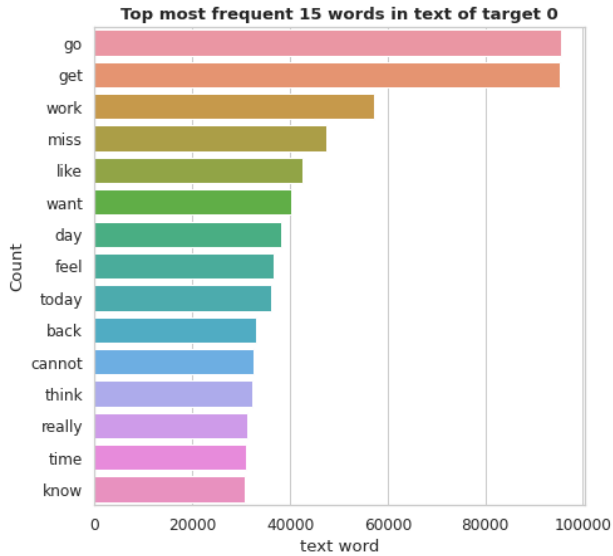
Total Unique words in negative tweets : `418034`

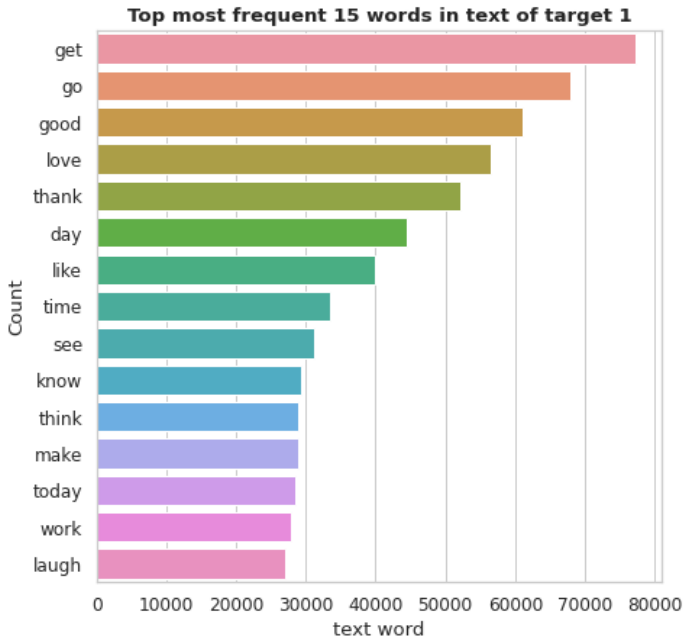
Total Unique words in positive tweets : `487131`

Sum of unique words in target 0 is 418034 and sum of unique words in target 1 is 487131 and there sum is 905165 and this is not equal to the total number of unique words in the whole documents ,as total unique words are 753901.This dismatch arises because there are some words common in target 0 and target 1 .Hence while adding unique words of target 0 and target 1,we counted twice those words that are common in both.

Most Frequent Words :







Total Frequecy of Words :

The total frequecy of word in the whole text column of document is : 12433378
The total frequecy of word in the whole text column of document with target as 0 is : 6286937
The total frequecy of word in the whole text column of document with target as 1 is : 6146441

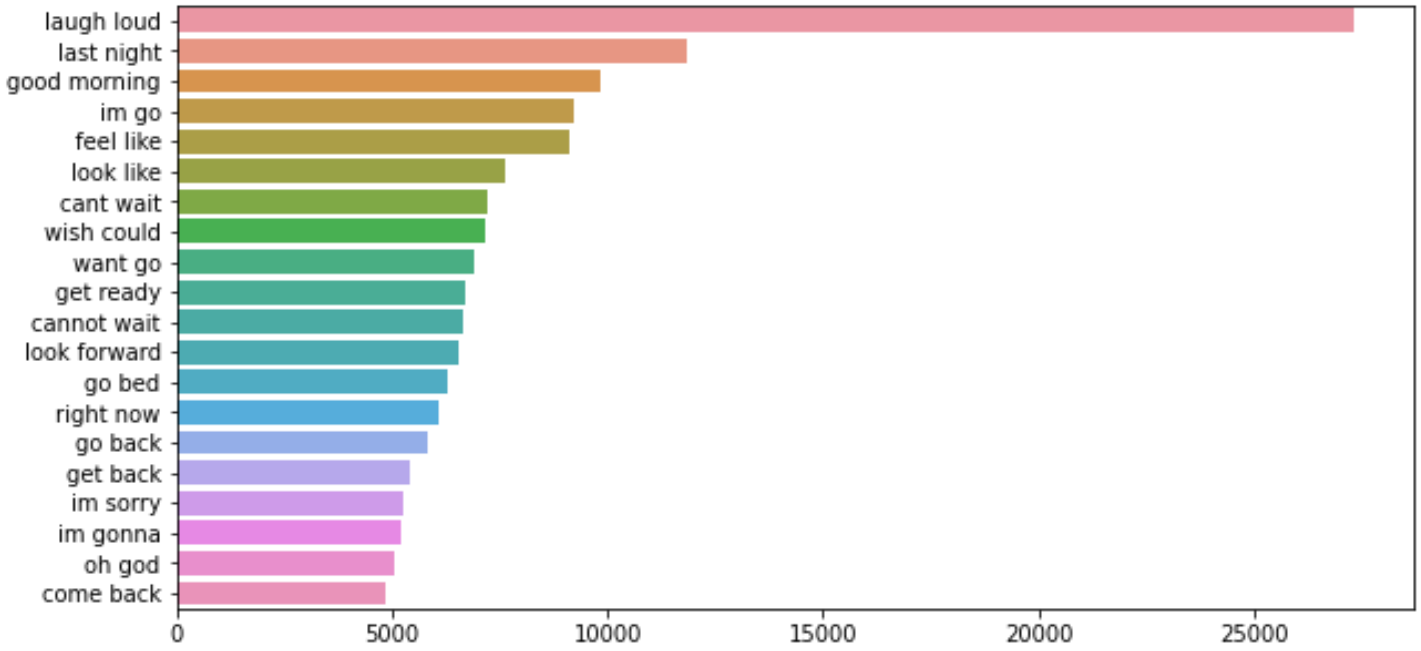
Sum of frequency of both the classes is equal to the Total frequency.

TF-IDF :
TF-IDF is a well-recognized method to evaluate the importance of a word in a document. TF-IDF works by penalizing the common words by assigning them lower weights while giving importance to words which are rare in the entire corpus but appear in good numbers in few documents. Term Frequency of a particular term (t) is calculated as number of times a term occurs in a document to the total number of words in the document. IDF (Inverse Document Frequency) is calculated as IDF (t) = log(N/DF), where N is the number of documents and DF is the number of document containing term t. TF-IDF is a better way to convert the textual representation of information into a Vector Space Model (VSM).

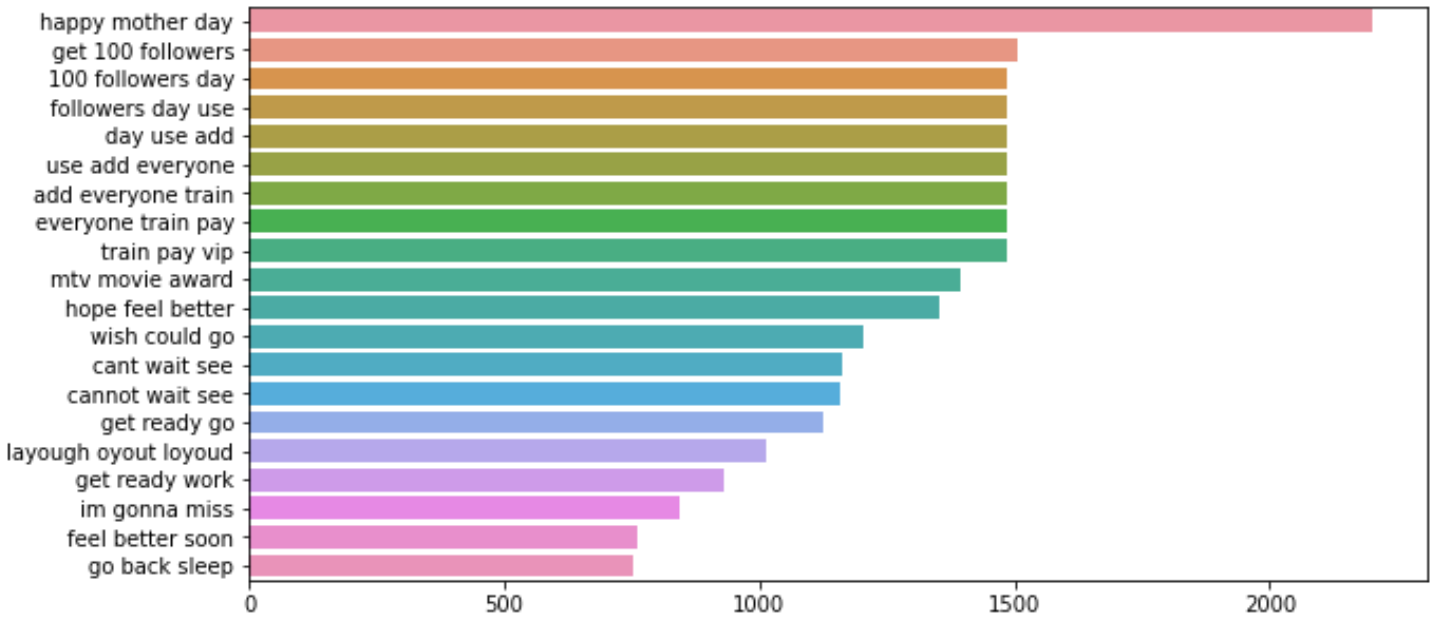
N-gram :
N-Gram will form the features of text for supervised machine learning algorithms. These are sequence of n tokens from the given text. Value of n can be 1, 2, 3, and so on. If we consider the value of n to be 1 it is called unigram, for n=2, bigram and for n=3 trigram and so on. We will be using both bigram and trigram to find which one is the best and then accordingly use the best for our purpose.

Visualization of bi-Gram and tri-gram for whole document for train dataset using Count Vectorizer

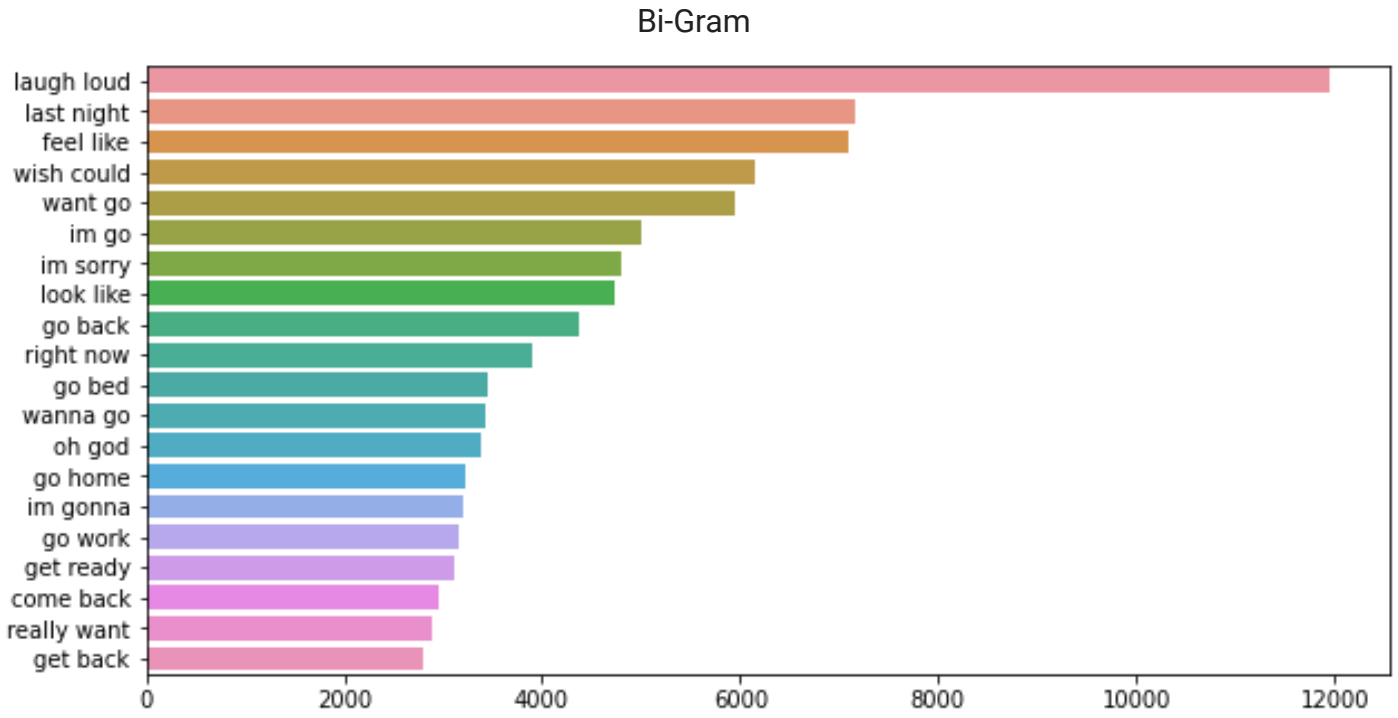
Bi-Gram

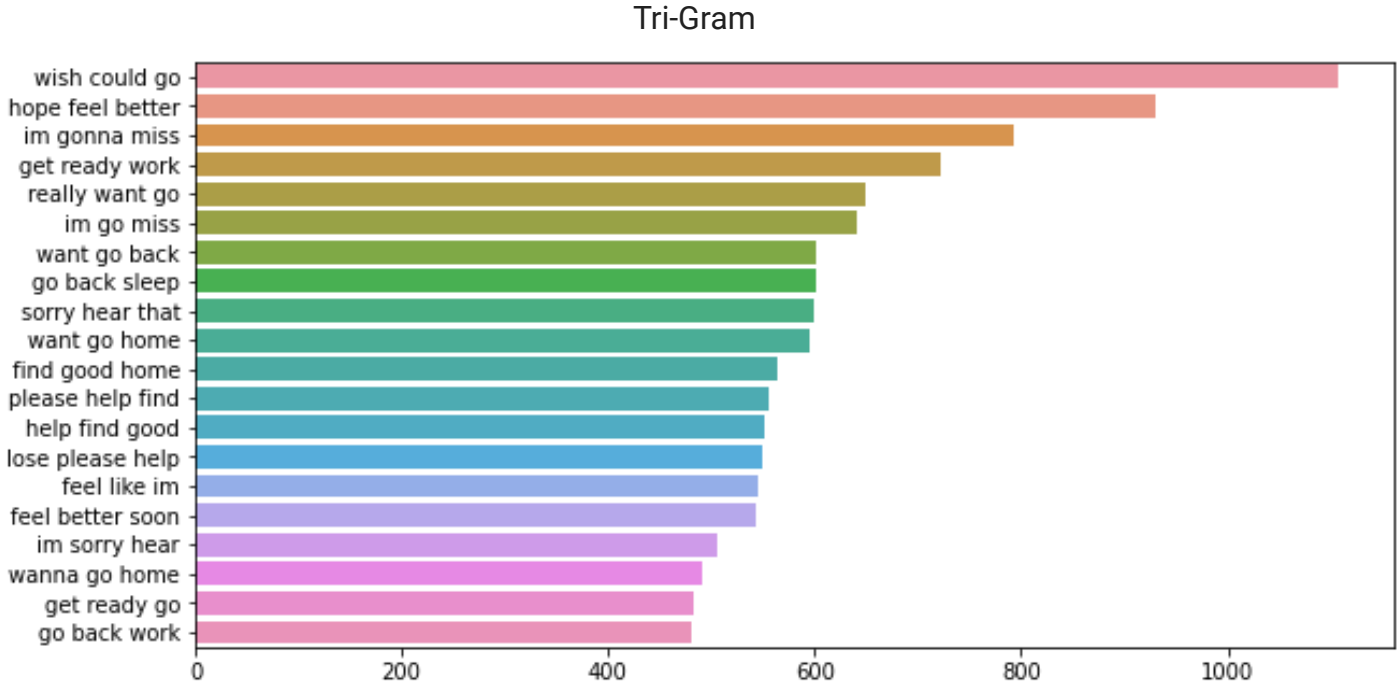


Tri-Gram



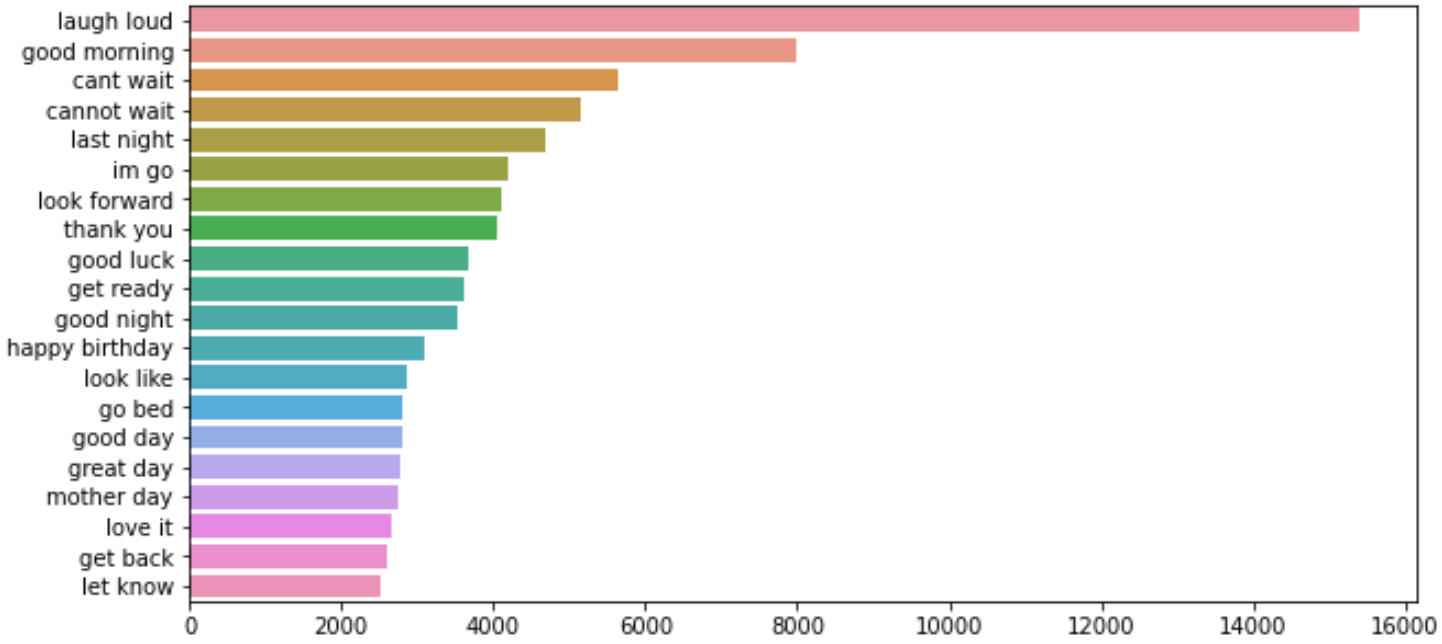
Visualization bi-Gram and tri-gram for document with target 0 using Count Vectorizer :



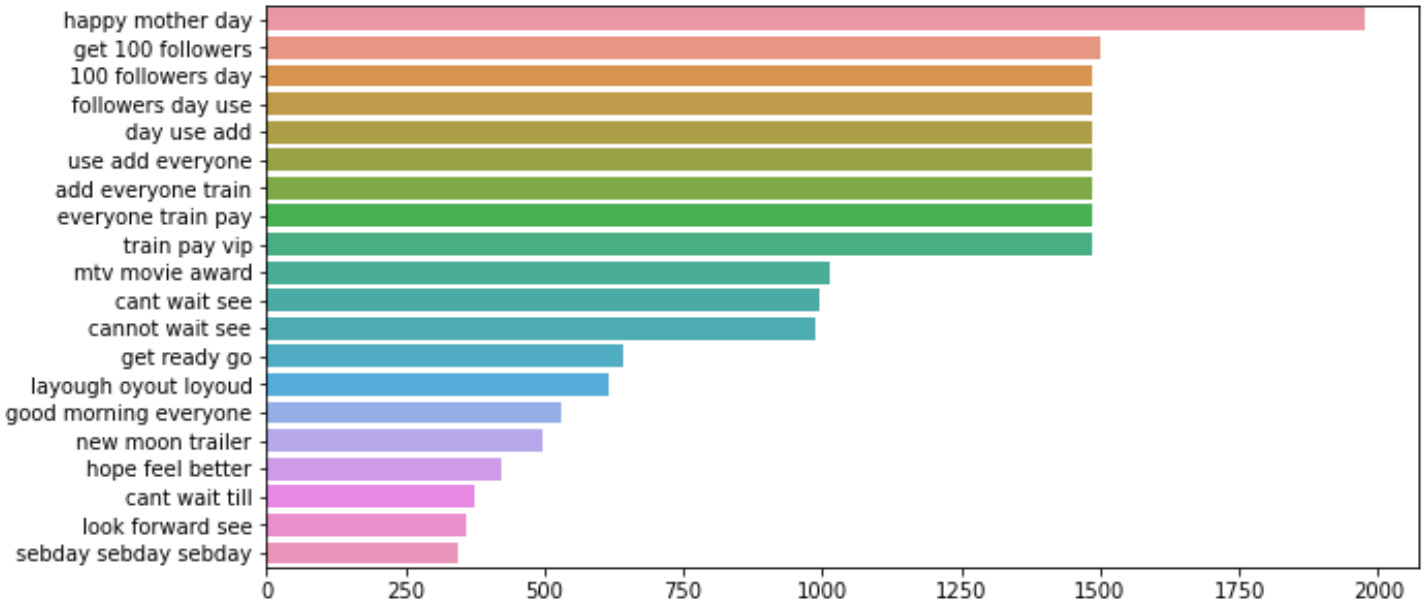


Visualization of N-Gram for document with target 1 using Count Vectorizer :

Bi-Gram



Tri-Gram



Classification algorithm used are described as follow:

LOGISTIC REGRESSION:

This is a popular classification algorithm which belongs to class of Generalized Linear Models. The probabilities describing outcome of a trial is modeled using logistic regression. This algorithm is also called Maximum Entropy.

NAÏVE BAYES:

This is powerful algorithm for classification used for classifying data on basis of probabilities

Naive Bayes classifiers are a collection of classification algorithms based on Baye's theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

The aim here is to get the best model for sentiment classification.

Our experimentation results involve **accuracy, recall, precision ,F1-Score**, **Confusion matrix, Roc curve** of each model and its comparison to get the best model predictions, which is further mentioned in the report.

Evaluation of Performace is done on the basis of following criteria:

TP=true positive, FP=false positive, FN=False negative, TN=true negative

Accuracy :

It calculates how much percentage of cases is correctly classified. The equation to calculate it:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Recall :

Ratio of correctly predicted positive observations to all observations in actual class yes is known as recall. It is computed as follows:

$$Recall = \frac{TP}{TP + FN}$$

Precision

Ratio of predicted positive observations to the total number of positive observation is known as precision. It is computed as follows.

$$Precision = \frac{TP}{TP + FP}$$

F1-Score

Weighted average of recall and precision is called f-score. More important parameter than accuracy when having an uneven class distribution in data. It is calculated as follows:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

CONFUSION MATRIX

Confusion Matrix A much better way to evaluate the performance of a classifier is to look at the confusion matrix. The general idea is to count the number of times instances of class A are classified as class B

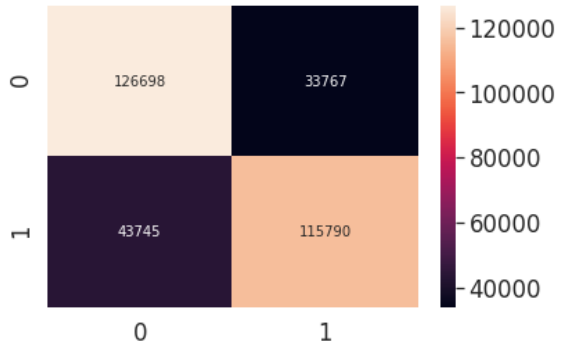
ROC CURVE

A Receiver Operator Characteristic (ROC) curve is a graphical plot used to show the diagnostic ability of binary classifiers. This curve is a commonly used way to visualize the performance of a binary classifier, meaning a classifier with two possible output classes.

Evaluation of Model through TF-IDF :

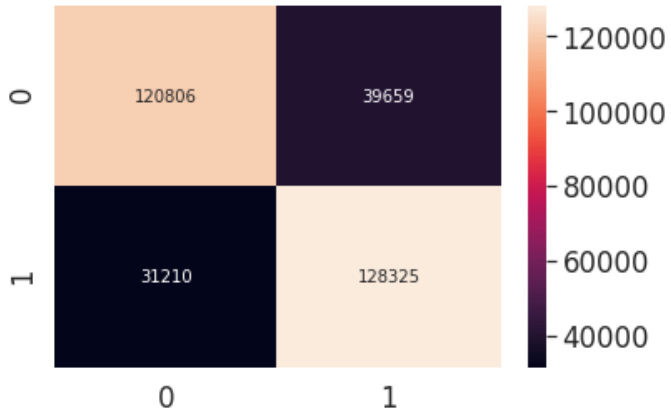
Naive Bayes : AUC = 0.76

	precision	recall	f1-score	support
0	0.74	0.79	0.77	160465
4	0.77	0.73	0.75	159535
accuracy			0.76	320000
macro avg	0.76	0.76	0.76	320000
weighted avg	0.76	0.76	0.76	320000



Logistic Regression : AUC = 0.78

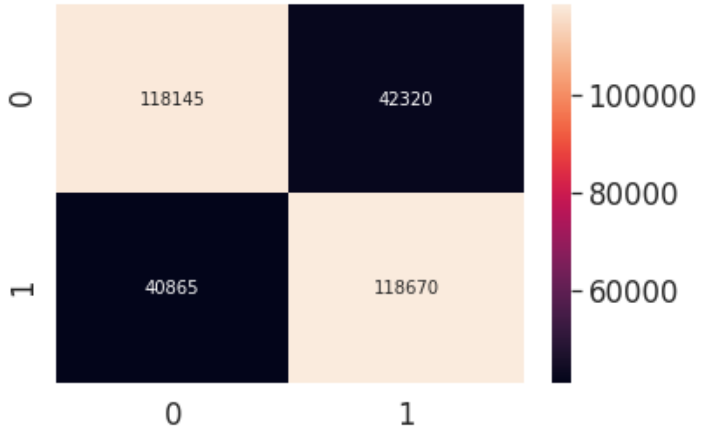
	precision	recall	f1-score	support
0	0.79	0.75	0.77	160465
4	0.76	0.80	0.78	159535
accuracy			0.78	320000
macro avg	0.78	0.78	0.78	320000
weighted avg	0.78	0.78	0.78	320000



<u>Evaluation of Model through Bi-Gram</u> :

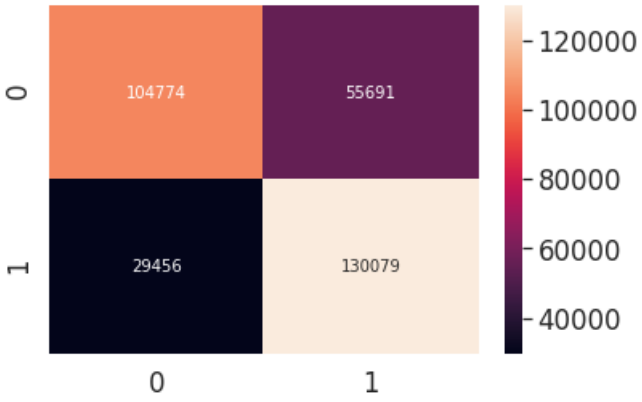
Naive Bayes : AUC = 0.74

	precision	recall	f1-score	support
0	0.74	0.74	0.74	160465
4	0.74	0.74	0.74	159535
accuracy			0.74	320000
macro avg	0.74	0.74	0.74	320000
weighted avg	0.74	0.74	0.74	320000



Logistic Regression : AUC = 0.73

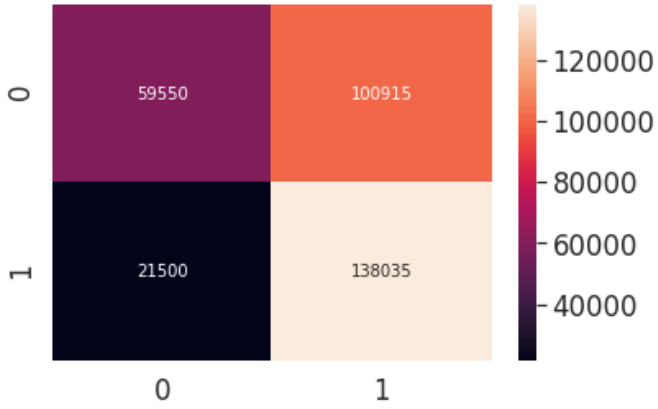
	precision	recall	f1-score	support
0	0.78	0.65	0.71	160465
4	0.70	0.82	0.75	159535
accuracy			0.73	320000
macro avg	0.74	0.73	0.73	320000
weighted avg	0.74	0.73	0.73	320000



Evaluation of Model through Tri-Gram :

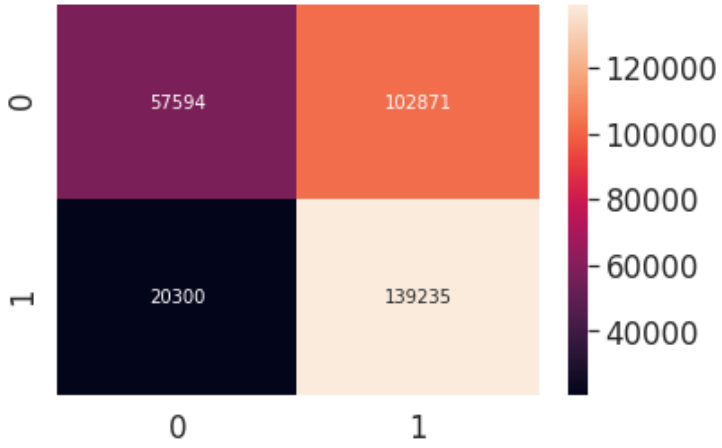
Naive Bayes : AUC = 0.62

	precision	recall	f1-score	support
0	0.73	0.37	0.49	160465
4	0.58	0.87	0.69	159535
accuracy			0.62	320000
macro avg	0.66	0.62	0.59	320000
weighted avg	0.66	0.62	0.59	320000



Logistic Regression : AUC = 0.62

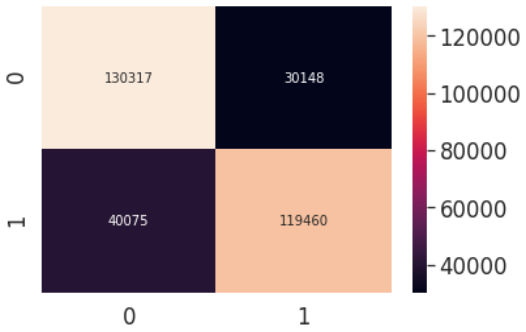
	precision	recall	f1-score	support
0	0.74	0.36	0.48	160465
4	0.58	0.87	0.69	159535
accuracy			0.62	320000
macro avg	0.66	0.62	0.59	320000
weighted avg	0.66	0.62	0.59	320000



Evaluation of Model through (1,2) i.e combination of unigram and bi-gram :

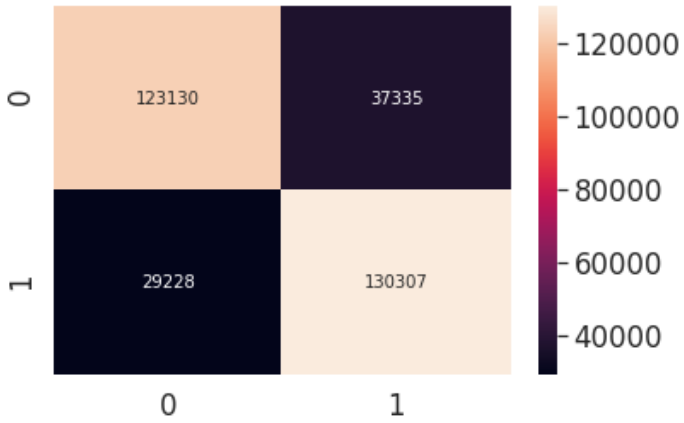
Naive Bayes : AUC = 0.78

	precision	recall	f1-score	support
0	0.76	0.81	0.79	160465
4	0.80	0.75	0.77	159535
accuracy			0.78	320000
macro avg	0.78	0.78	0.78	320000
weighted avg	0.78	0.78	0.78	320000



Logistic Regression : AUC = 0.79

	precision	recall	f1-score	support
0	0.81	0.77	0.79	160465
4	0.78	0.82	0.80	159535
accuracy			0.79	320000
macro avg	0.79	0.79	0.79	320000
weighted avg	0.79	0.79	0.79	320000



Findings :

In our sentiment analysis, we used two feature extractors as, TF-IDF ,Bi-gram, Tri-Gram and combination of uni and bi gram . Further, we found that ngram accuracy decreases as ngram length increases and the reason behind this is that the matrix become more and more sparse as the number of combination of words are not frequency in tweet rather than single word . So, we further mixed different ngram length and found the increased accuracy , also, few of the expert advises was found on internet to use the mixed length ngram to enhance our accuracy so we used (1,2) mixed ngram to enhance our accuracy. Word Cloud is nice way the visualize the words that are occurred more frequently in tweets.

Conclusion :

In this paper, we have thus applied 2 different algorithms of classification on the tweeter dataset considering features extractions with TF-IDF and N-Grams. Also, LDA as a feature extractor can't be used here because it did not apply on sparse matrix. Thus, after doing sentiment analysis of these tweets, we founded that, TF-IDF features are giving better results (3-4%) as compared to Bi-Gram and Tri-Gram features.But combination of uni and bi-gram which is (1,2) grams results in even better accuracy than considering TD-IDF using uni gram only.But the cost of training combination of uni gram and bi gram in one model is costlier than TD-IDF. Thus ,we have to decide appropriately as per our need. But if we want good accuracy as well as low cost then we can use TD-IDF for machine learning algorithm for text classification .. By overall comparison of machine learning algorithms, we found out that logistic regression gave best predictions of sentiments by

giving maximum output for all four comparison parameters namely – accuracy, recall, precision, and f-score and for both feature extraction methods namely – N-Gram and word-level TF-IDF.

Model	Training Accuracy	Test Accuracy	Precision		Recall		f1-score		AUC
			0	4	0	4	0	4	
Naive Bayes model	83.88%	75.54%	0.74	0.77	0.79	0.73	0.77	0.75	0.76
Logistics Regression model	81.38%	77.75%	0.79	0.76	0.75	0.8	0.77	0.78	0.78
Logistics Regression model (2-gram)	90.26%	72.98%	0.78	0.7	0.65	0.82	0.71	0.75	0.73
Naive Bayes model (2-gram)	93.99%	73.54%	0.74	0.74	0.74	0.74	0.74	0.74	0.74
Logistics Regression model (3-gram)	95.27%	60.95%	0.74	0.58	0.36	0.87	0.48	0.69	0.62
Naive Bayes model (3-gram)	95.92%	61.14%	0.73	0.58	0.37	0.87	0.49	0.69	0.62
Logistics Regression model (1-2 gram)	84.78%	78.94%	0.81	0.78	0.77	0.82	0.79	0.8	0.79
Naive Bayes model (1-2 gram)	92.28%	77.77%	0.76	0.8	0.81	0.75	0.79	0.77	0.78

Future Work :

Machine learning techniques perform well for classifying sentiment in tweets. We believe that the accuracy could still be improved. Below is a list of ideas we think could help in this direction.

1) Semantics : Our algorithms classify the overall sentiment of a tweet. The polarity of a tweet may depend on the perspective you are interpreting the tweet from. For example, in the tweet Federer beats Nadal :), the sentiment is positive for Federer and negative for Nadal. In this case, semantics may help. Using a semantic role labeler may indicate which noun is mainly associated with the verb and the classification would take place accordingly. This may allow Nadal beats Federer :) to be classified differently from Federer beats Nadal :).

2)Domain-specific tweets : Our best classifier has an accuracy of 83.0% for tweets across all domains. This is a very large vocabulary. If limited to particular domains (such as movies) we feel our classifiers may perform better.

3)Handling neutral tweets : In real world applications, neutral tweets cannot be ignored. Proper attention needs to be paid to neutral sentiment.

4)Internationalization : We focus only on English sentences, but Twitter has many international users. It should be possible to use our approach to classify sentiment in other languages.

5)Utilizing emoticon data in the test set : Emoticons are stripped from our training data. This means that if our test data contains an emoticon feature, this does not influence the classifier towards a class. This should be addressed because the emoticon features are very valuable.

Contribution of Each Member :

1) **Aniket Wani (B19EE009) :** Build the code for preprocessing and visualization and error correction in code and training different classifier. Build report
2) **Aanchal (B19EE001) :** Build the report and code for n grams and preprocessing , error correction in code
3) **Harpal Sahil Santosh (B19CSE107) :** Build readme file,preprocessing of code and code for TD-IDF vectorizer and error correction in code.