Project

Sentimental Analysis

1. Project

Sentiment analysis, a sub-field of Natural Language Processing, is one of the most popular topics and research fields in data science. We will be working on social media sentiment analysis. We aim to be able to classify tweets, reviews and comments from social media as positive, negative or neutral.

The most important point of our project is data mining to collect a large amount of data from several sources. For this purpose, we found open source datasets such as Sentiment140 [1] and many others. Besides, we would like to collect our own data from social media and expand our dataset. We also found some tools and APIs to retrieve new data. TWINT - Twitter Intelligence Tool [2] is an advanced Twitter scraping tool written in Python that allows for scraping tweets. One of the most important features of TWINT is that there is no need to use Twitter Developer API. This feature allows collecting data with no rate limitations.

Most of the open-source datasets that we found on the internet are properly labeled and structured. Data collected by ourselves need to be properly labeled. Then, we will go through the cleaning, preprocessing and separation of test and training data steps.

We searched for some tools for our project and found some popular and powerful open-source NLP frameworks in Python. We will probably use Natural Language Toolkit (NLTK) [3]. It comes with all the pieces you need to get started on sentiment analysis.

2. Data

Dataset : Sentiment140

This dataset contains 1,600,000 tweets extracted using the twitter api. The tweets have been classified from 0 (negative) to 4 (positive). The dataset contains 6 fields which are target as integer, ids as integer, date as date, flag as string, user as string and text as string.These 6 fields are shown below.

- target: The polarity of the tweet (0 - negative, 2 - neutral, 4 - positive)
- ids: The id of the tweet.
- date: The date of the tweet.
- flag: The query. If there is no query, then this value is NO_QUERY.
- user: The user that tweeted.
- text: The text of the tweet

Target	Ids	Date	Flag	User	Text
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOn	@switchfoot http://twitpic.com/2y1zl - Awww, tl
0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by text
0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Mana
0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all. i'm
0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
0	1467811592	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	mybirch	Need a hug
0	1467811594	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	coZZ	@LOLTrish hey long time no see! Yes.. Rains a bit
0	1467811795	Mon Apr 06 22:20:05 PDT 2009	NO_QUERY	2Hood4Hollyw	@Tatiana_K nope they didn't have it
0	1467812025	Mon Apr 06 22:20:09 PDT 2009	NO_QUERY	mimismo	@twittera que me muera ?

Figure 1. A sample from the dataset

Dataset : The dataset consists of tweets or comments and sentimental labels. We will be using Twitter Developer API or a web scraping library called Twint - sample output shown in Figure 2 - to create our own dataset from twitter. This sample has 3 fields which are id as integer, username as string and tweet as string.

	id	username	tweet
0	1377434193827860480	joebiden	The American Jobs Plan is the largest American...
1	1377414548097880068	joebiden	Under the American Jobs Plan, 100% of our nati...
2	1377385089340829700	joebiden	Wall Street didn't build this country—the grea...
3	1377367815120957445	joebiden	Delivering for the American people is what the...
4	1377363181387980800	joebiden	The American Jobs Plan is a once-in-a-generati...
5	1377361641491300354	joebiden	Millions of Americans lost their jobs last yea...
6	1371117123859316736	joebiden	It matters whether you continue to wear a mask...
7	1370792386280972289	joebiden	The American Rescue Plan means a $7,000 check ...
8	1370411955173912576	joebiden	85% of American households will get direct che...
9	1366417985649442821	joebiden	A campaign for everyone who's been knocked dow...

Figure 2. Joe Biden's tweets collected using TWINT.

If we use Twitter Developer API or Twint to create our data set, we need to label the tweets to prepare such a supervised dataset.

Dataset : Twitter.csv

This dataset is a supervised dataset which includes tweets. Twitter.csv Dataset has around 163,000 tweets along with sentiment labels samples shown in Figure 3. This dataset has 3 fields which are id as integer, tweet as string and label as integer. Reddit.csv dataset has around 37,000 comments along with its sentimental label.

0	when modi promised “minimum government maximum...	-1.0
1	talk all the nonsense and continue all the dra...	0.0
2	what did just say vote for modi welcome bjp t...	1.0
3	asking his supporters prefix chowkidar their n...	1.0
4	answer who among these the most powerful world...	1.0
5	kiya tho refresh maarkefir comment karo	0.0
6	surat women perform yagna seeks divine grace f...	0.0
7	this comes from cabinet which has scholars lik...	0.0
8	with upcoming election india saga going import...	1.0
9	gandhi was gay does modi	1.0
10	things like demonetisation gst goods and servi...	1.0
11	hope tuthukudi people would prefer honest well...	1.0
12	calm waters wheres the modi wave	1.0
13	one vote can make all the difference anil kapo...	0.0
14	one vote can make all the difference anil kapo...	0.0
15	vote such party and leadershipwho can take fas...	-1.0
16	vote modi who has not created jobs	0.0
17	through our vote ensure govt need and deserve ...	0.0
18	dont play with the words was talking about the...	1.0
19	didn’ write chowkidar does mean ’ anti modi tr...	-1.0
20	was the one who recently said that people who ...	1.0

Figure 3. Labels indicate that 1 positive, 0 neutral and -1 negative comment

Chart/Figures of Attribute

Number of Letters

Other than the label feature, there is an attribute named “tweet”. Related charts/figures are given below. By counting the letters of the tweets in the dataset, we created the chart in Figure 1 that shows the frequency and the relative frequency of the letters of the whole tweets.Then, we applied a chi-square test to see whether the distribution of the letters in tweets is the same with what we expect from English texts.

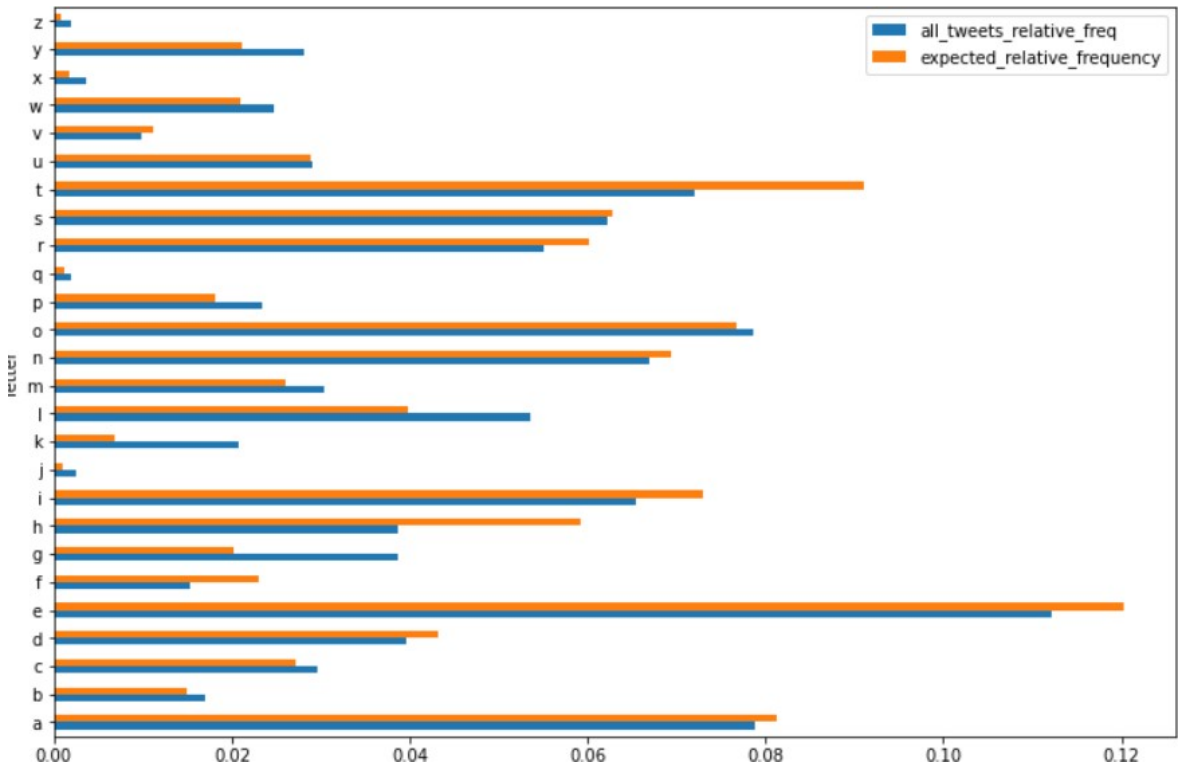


Figure 1. Letter frequencies of each 26 characters in English Alphabet.

	letter	frequency	all_tweets_relative_freq	expected_relative_frequency	expected
0	a	4547601	0.078816	0.081238	4687379.0
1	b	975326	0.016904	0.014893	859300.0
2	c	1705409	0.029557	0.027114	1564464.0
3	d	2289515	0.039680	0.043192	2492128.0
4	e	6471295	0.112156	0.120195	6935169.0
5	f	878849	0.015232	0.023039	1329304.0
6	g	2231747	0.038679	0.020257	1168838.0
7	h	2234047	0.038719	0.059215	3416628.0
8	i	3779579	0.065505	0.073054	4215160.0
9	j	143817	0.002493	0.001031	59502.0
10	k	1197291	0.020751	0.006895	397842.0
11	l	3095498	0.053649	0.039785	2295581.0
12	m	1754377	0.030406	0.026116	1506861.0
13	n	3861185	0.066919	0.069478	4008801.0
14	o	4534414	0.078587	0.076812	4431963.0
15	p	1351301	0.023420	0.018189	1049517.0
16	q	115059	0.001994	0.001125	64883.0
17	r	3179237	0.055100	0.060213	3474231.0
18	s	3595565	0.062316	0.062808	3623936.0
19	t	4153946	0.071993	0.090986	5249801.0
20	u	1676743	0.029060	0.028776	1660364.0
21	v	566733	0.009822	0.011075	639015.0
22	w	1422401	0.024652	0.020949	1208717.0
23	x	203131	0.003521	0.001728	99698.0
24	y	1620980	0.028094	0.021135	1219478.0
25	z	114027	0.001976	0.000702	40512.0

Figure 2. Letter frequency of the dataset, relative frequencies of the dataset, expected relative frequency according to the English language and expected character length according to the English language.

We got the p-value (p) as 0 which implies that the letter frequency does not follow the same distribution with what we see in English tests, although the Pearson correlation is too high (~96.7%) as shown in Figure 3.

	frequency	expected
frequency	1.000000	0.967421
expected	0.967421	1.000000

Figure 3. Correlation.

We counted the number of characters for each tweet (Figure 4) and analyzed the data frame according to maximum number of characters, minimum number of characters, mean of the number of characters column and its standard deviation. Our longest tweet is 189 characters long, the shortest tweet is 1 character long and mean of all tweets' character length 42.78. The standard deviation of all tweet character length is 24.16 as shown in Figure 5.

	label	tweet	number_of_characters
0	Negative	awww bummer shoulda got david carr third day	44
1	Negative	upset update facebook texting might cry result...	69
2	Negative	dived many times ball managed save 50 rest go ...	52
3	Negative	whole body feels itchy like fire	32
4	Negative	behaving mad see	16
...	...	...	...
1599995	Positive	woke school best feeling ever	29
1599996	Positive	thewdb com cool hear old walt interviews	40
1599997	Positive	ready mojo makeover ask details	31
1599998	Positive	happy 38th birthday boo alll time tupac amaru ...	52
1599999	Positive	happy charitytuesday thenspcc sparkscharity sp...	57

Figure 4. Number of characters.

```
df1.number_of_characters.max()
```

```
189
```

```
df1.number_of_characters.min()
```

```
1
```

```
df1.number_of_characters.mean()
```

```
42.7974010379771
```

```
df1.number_of_characters.std()
```

```
24.158961650697616
```

Figure 5. Max, min, mean and standard deviation of each tweet in terms of character length.

Number of Words

We counted the number of words for each tweet (Figure 6) and analyzed the data frame according to maximum number of words, minimum number of words, mean of the number of words column and its standard deviation. Our longest tweet is 50 words long, the shortest tweet is 1 word long and the mean of all tweets' word length is 7.24. The standard deviation of all tweet character length is 4.03 as shown in Figure 7.

	label	tweet	number_of_characters	number_of_words
0	Negative	awww bummer shoulda got david carr third day	44	8
1	Negative	upset update facebook texting might cry result...	69	11
2	Negative	dived many times ball managed save 50 rest go ...	52	10
3	Negative	whole body feels itchy like fire	32	6
4	Negative	behaving mad see	16	3
...	...	...	...	...
1599995	Positive	woke school best feeling ever	29	5
1599996	Positive	thewdb com cool hear old walt interviews	40	7
1599997	Positive	ready mojo makeover ask details	31	5
1599998	Positive	happy 38th birthday boo alll time tupac amaru ...	52	9
1599999	Positive	happy charitytuesday thenspcc sparkscharity sp...	57	5

Figure 6. Number of words of each tweet.

```
df1.number_of_words.max()
```

```
50
```

```
df1.number_of_words.min()
```

```
1
```

```
df1.number_of_words.mean()
```

```
7.244474128445898
```

```
df1.number_of_words.std()
```

```
4.030421805719796
```

Figure 7. Max, min, mean and standard deviation of each tweet in terms of number of words.

Most common words in the whole dataset and positive/negative tweets of the dataset are given in the following figures.

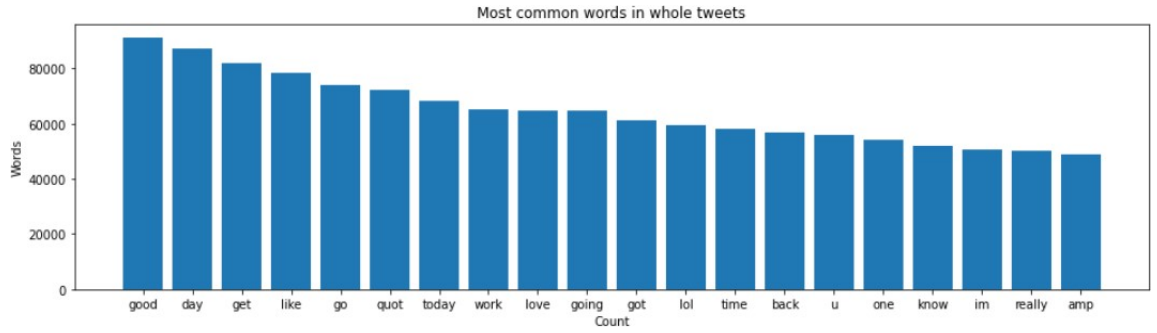


Figure 8. Most common words in our dataset.

Positive Tweets:

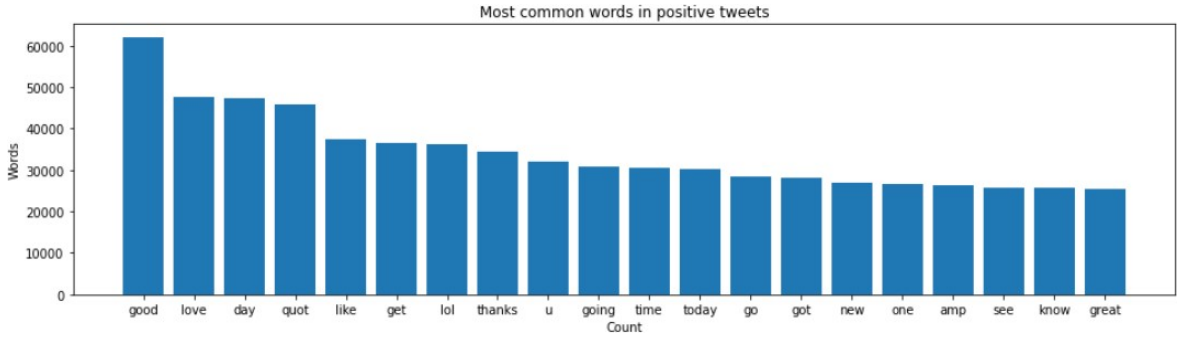


Figure 9. Most common words in positive tweets in our dataset.

Figure 10. Word cloud of positive tweets.

Negative Tweets:

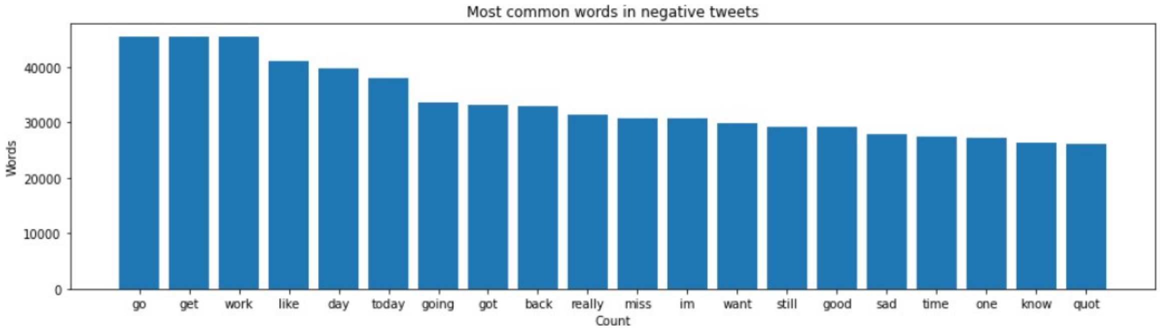


Figure 11. Most common words in negative tweets in our dataset.

Figure 12. Word cloud of positive tweets.

Positive and negative samples are equal. The dataset distribution has not any skewness as shown in Figure 13.

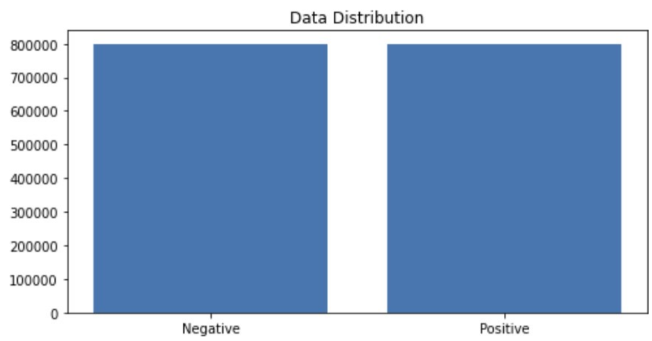


Figure 13. Positive and negative tweets distribution.

1. Scatter Plot

We used feature extraction methods, bag-of-words, and word embedding. Bag of words with TF-IDF is a common and simple way of feature extraction. Bag-of-Words is a representation model of text data and TF-IDF is a calculation method to score the importance of words in a document.

After applying bag-of-words with TF-IDF, we create the scatter plot according to these results.

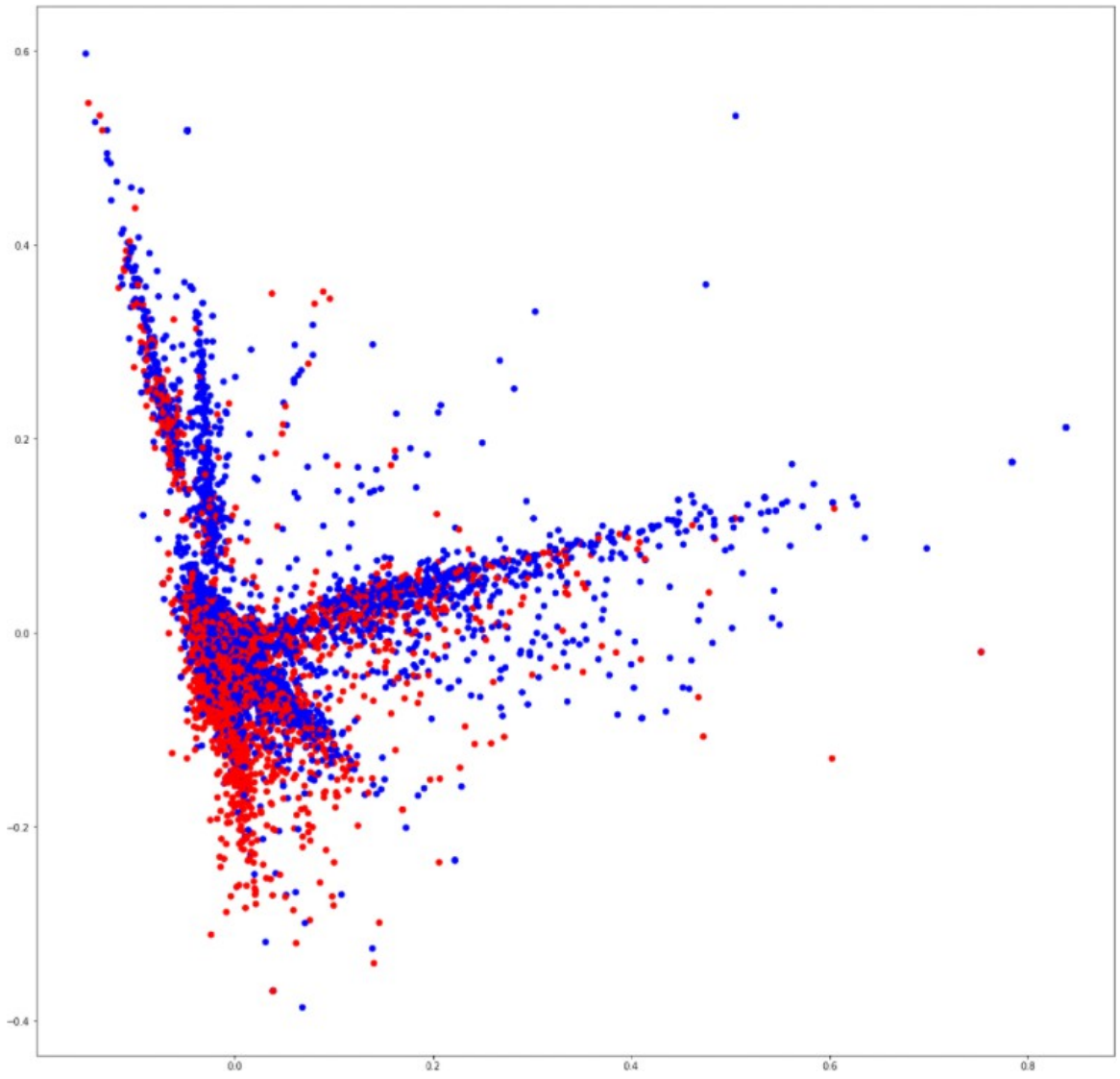


Figure 14. Scatter plot that shows correlation of words in the corpus: red indicates negatives, blue indicates positives.

2. Results

In this delivery, we explored our dataset by applying some analyses to the attributes and created related charts. There are 2 attributes in our dataset including label attribute. We applied these analyses on them.

We explored the tweets by looking at the letters and words in them. First of all, we counted the letters of all tweets and calculated the letter frequencies. Then we compared the letter frequency of our data with the expected frequency of the letters of the alphabet of English. In Figure 1, this comparison is shown. As seen in the graph, even though there are some exceptions, for most of the letter, the frequencies of our data is really close to the expected ones.

The number of characters and words are also counted and analysed. Minimum number of characters of all tweets is 1 whereas the maximum number is 189. Since the mean is around 42 and standard deviation is around 24, it can be said that a small number of tweets has a high number of characters. The similar result can be seen in word analysis . When the number of words counted, it is seen that the maximum number of words in tweets is 50 whereas the minimum number is 1. Mean is around 7 and standard deviation is around 4 which gives a similar result with the number of characters. Very small number of tweets has a high number of words. According to these results, it can be interpreted that both the number of characters and number of words graphs are skewed graphs.

After counting the number of words used in tweets, word usages are analysed. Since the stop words are usually the most used words in texts and they may prevent us from getting the right results, they are calculated by filtering the stopwords. The results are shown in Figure 8. Also, most common words for positive and negative labels are separated and shown in Figures 9, 10, 11 and 12.

Then, as mentioned in Part 2, by using some feature extraction methods, a scatter plot is obtained. The plot (Figure 14) shows the correlation between the words.

1.Features

The dataset contains 1,600,000 tweets extracted using the twitter api. The tweets have been classified from 0 (negative) to 4 (positive). The dataset contains 6 fields which are target as integer, ids as integer, date as date, flag as string, user as string and text as string.These 6 fields are shown below.

- target: The polarity of the tweet (0 - negative, 2 - neutral, 4 - positive)
- ids: The id of the tweet.
- date: The date of the tweet.
- flag: The query. If there is no query, then this value is NO_QUERY.
- user: The user that tweeted.
- text: The text of the tweet

Target	Ids	Date	Flag	User	Text
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOn	@switchfoot http://twitpic.com/2y1zl - Awww, t
0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by text
0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Mana
0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all. i'm
0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
0	1467811592	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	mybirch	Need a hug
0	1467811594	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	coZZ	@LOLTrish hey long time no see! Yes.. Rains a bit
0	1467811795	Mon Apr 06 22:20:05 PDT 2009	NO_QUERY	2Hood4Hollyw	@Tatiana_K nope they didn't have it
0	1467812025	Mon Apr 06 22:20:09 PDT 2009	NO_QUERY	mimismo	@twittera que me muera ?

Figure 1. A sample from the dataset

At the beginning, our dataset had 6 features which were target, id, date, query, user and text. We chose two of them for our purpose which are target and text. We can see that the entropy decreases significantly after this transformation.

Information gain

First entropy of dataset = 41.08269441306875

Entropy after preprocess = 14.73368002815221

2. Classification/Regression

For classification/regression experiments, the test set percentage is set to be 20%. 6 different models that are applied are CNN Model-1, CNN Model-2, LSTM Model-1, LSTM Model-2, Naive Bayes Model-1 and Naive Bayes Model-2. Below, precision, recall, f1 score and accuracy of the models are shown.

CNN Model - 1 :
Conv1D = 64
Dense = 512
Dense = 512
1024 batch size

CNN Model - 2 :
Conv1D = 31
Dense = 256
Dense = 256
512 batch size

LSTM Model - 1 : 1024 Batch size
LSTM Model - 2 : 512 Batch size

Naive Bayes Model - 1 : Multinomial, count vectorizer
Naive Bayes Model - 2 : Multinomial, Use TF-IDF

Naive Bayes Model - 1 (CountVectorizer) :

	precision	recall	f1-score	support
Negative	0.76	0.77	0.77	159493
Positive	0.77	0.76	0.76	158973
accuracy			0.76	318466
macro avg	0.77	0.76	0.76	318466
weighted avg	0.77	0.76	0.76	318466

Naive Bayes Model - 2 (tf-idf):

	precision	recall	f1-score	support
Negative	0.76	0.77	0.76	159493
Positive	0.76	0.75	0.76	158973
accuracy			0.76	318466
macro avg	0.76	0.76	0.76	318466
weighted avg	0.76	0.76	0.76	318466

LSTM Model - 1 :

	precision	recall	f1-score	support
Negative	0.78	0.79	0.79	159493
Positive	0.79	0.78	0.78	158973
accuracy			0.78	318466
macro avg	0.78	0.78	0.78	318466
weighted avg	0.78	0.78	0.78	318466

LSTM Model - 2 :

	precision	recall	f1-score	support
Negative	0.77	0.81	0.79	159493
Positive	0.80	0.76	0.78	158973
accuracy			0.78	318466
macro avg	0.78	0.78	0.78	318466
weighted avg	0.78	0.78	0.78	318466

CNN Model - 1 :

	precision	recall	f1-score	support
Negative	0.78	0.78	0.78	159493
Positive	0.78	0.78	0.78	158973
accuracy			0.78	318466
macro avg	0.78	0.78	0.78	318466
weighted avg	0.78	0.78	0.78	318466

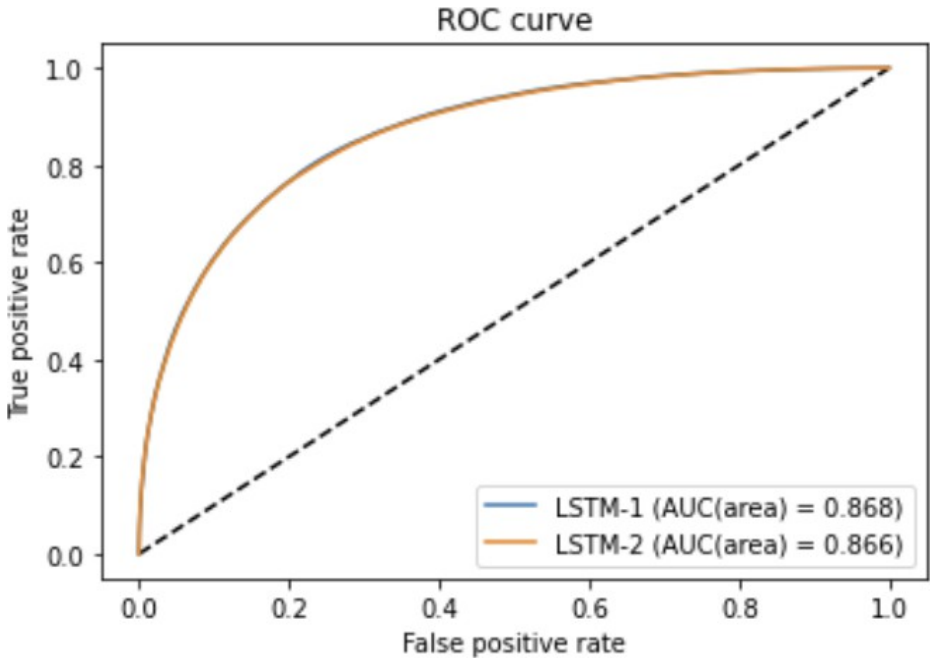
CNN Model - 2 :

	precision	recall	f1-score	support
Negative	0.80	0.73	0.76	159493
Positive	0.75	0.82	0.78	158973
accuracy			0.77	318466
macro avg	0.78	0.77	0.77	318466
weighted avg	0.78	0.77	0.77	318466

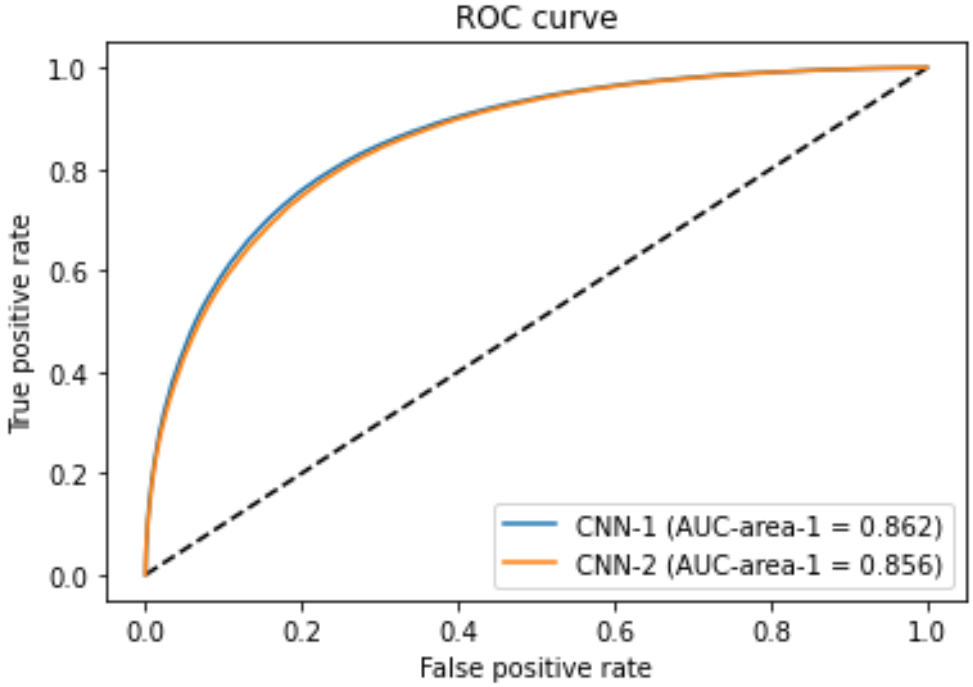
3.ROC Curve

After determining the evaluation metrics, ROC curves of the models are formed. Also AUC values are calculated and shown at the bottom of each graph.

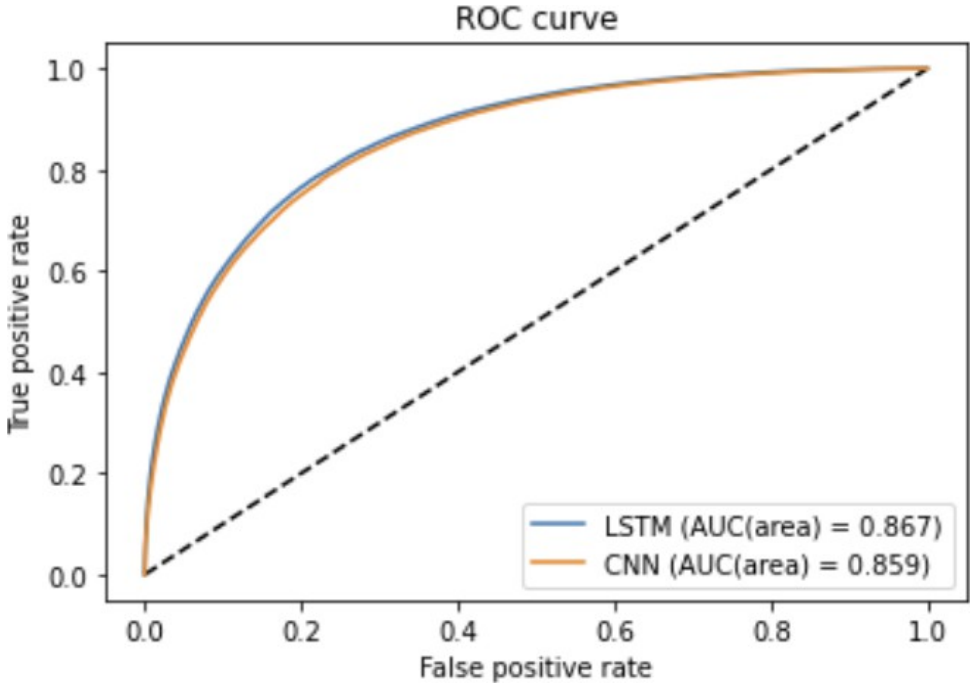
ROC Curve of CNN Model-1 and CNN Model-2 :



ROC Curve of LSTM Model-1 and LSTM Model-2 :



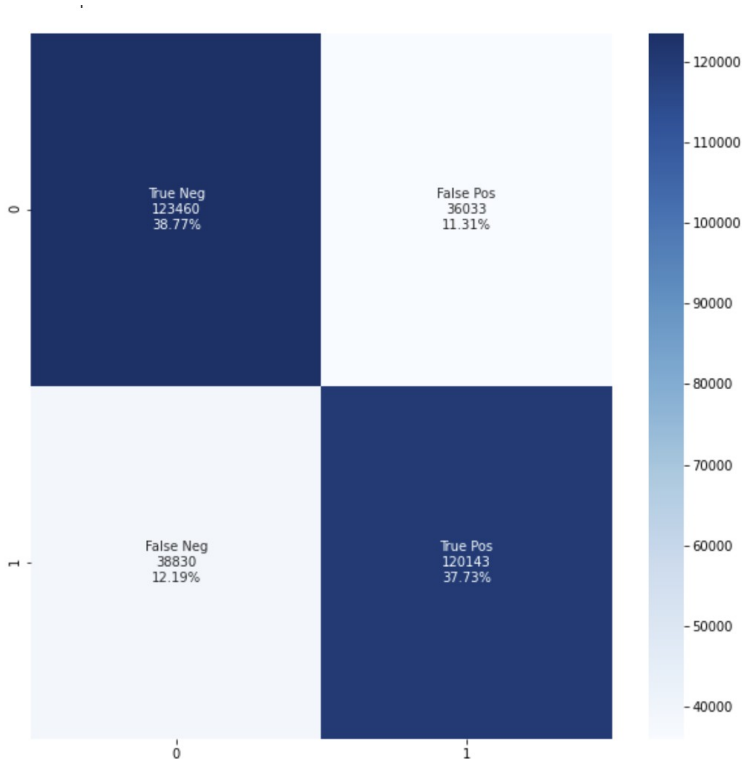
ROC Curve of best LSTM model and best CNN model :



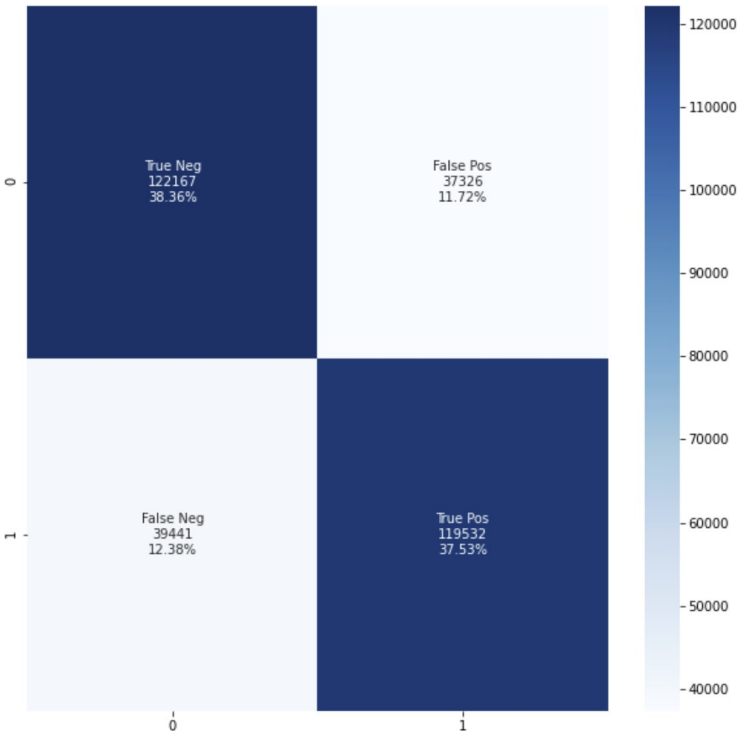
4. Confusion Matrix

Confusion matrices of the 6 model used to train the data, including the best performing model LSTM-1, are as follows:

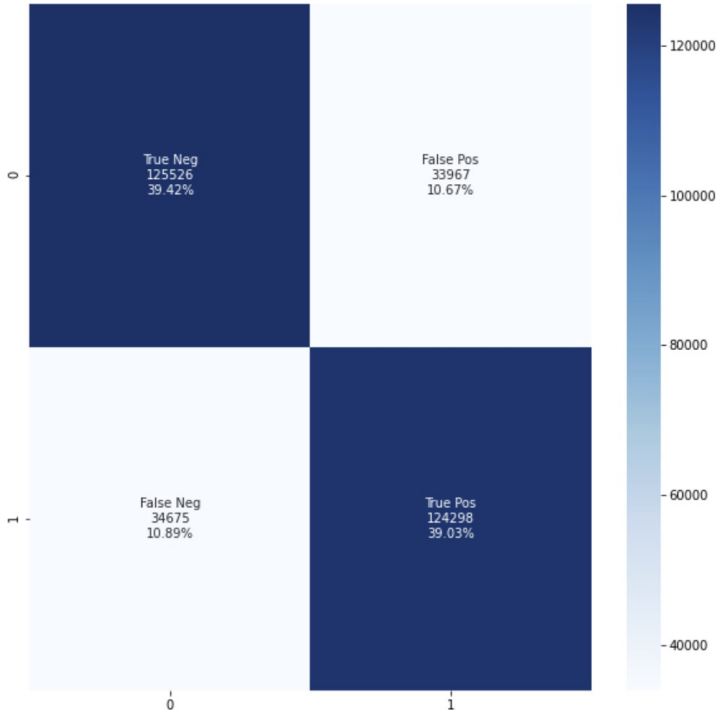
Confusion Matrix of Naive Bayes with Countvectorizer :



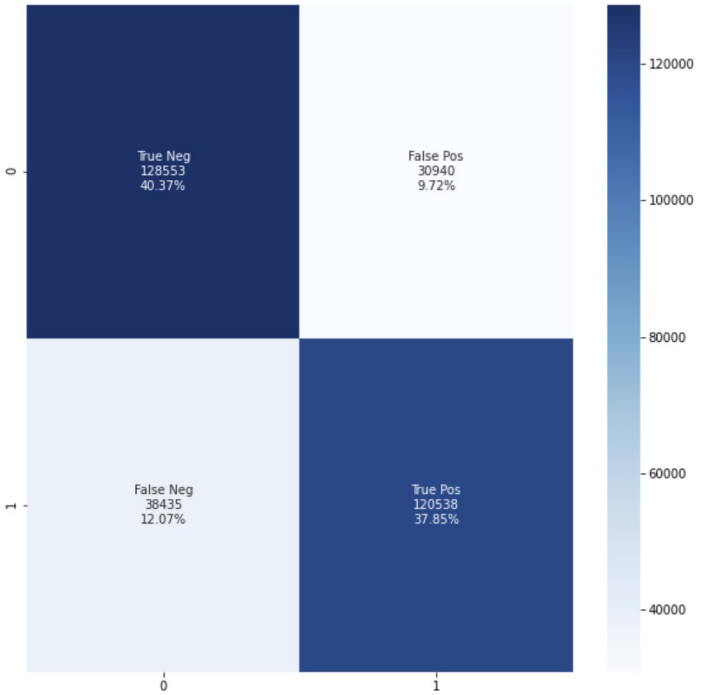
Confusion Matrix of Naive Bayes with TF-IDF :



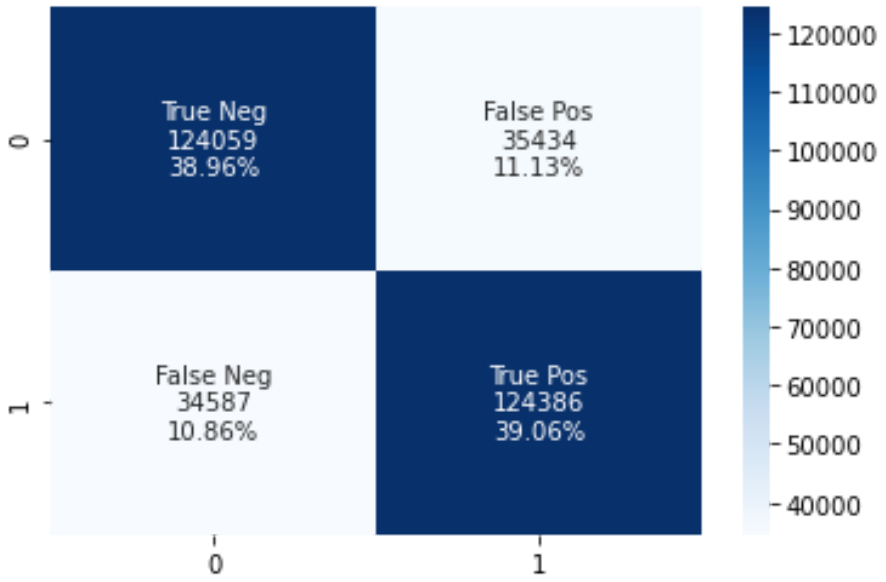
Confusion Matrix of LSTM Model - 1 :



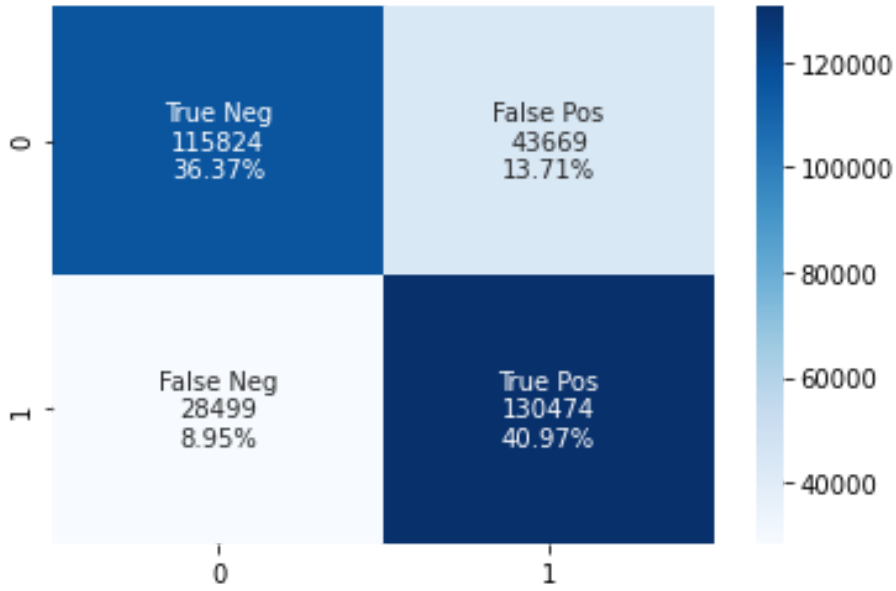
Confusion Matrix of LSTM Model - 2 :



Confusion Matrix of CNN Model - 1 :



Confusion Matrix of **CNN Model - 2 :**



5. Statistical Significance Analysis

According to Accuracy, P, R, F1, AUC, our best performing model is LSTM model 1 with 1024 batch size and 0.789 accuracy and the closest competitor to LSTM model 1 is CNN model 1 with accuracy 0.781. Multinomial Naive Bayes with tf-idf is the worst performing algorithm among them with accuracy 0.758.

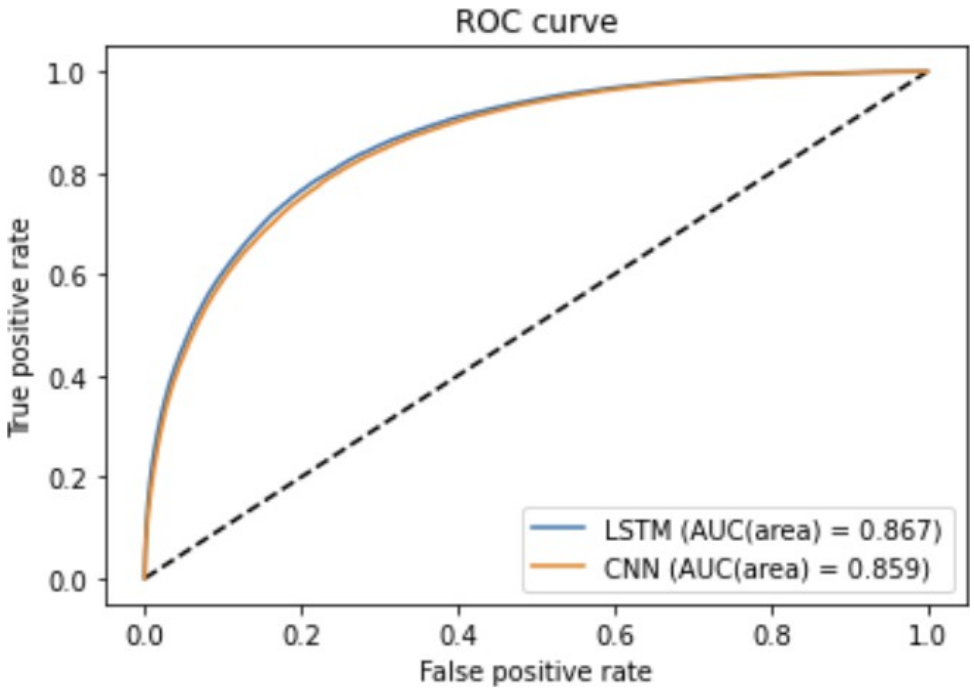
LSTM Model 1

	precision	recall	f1-score	support
Negative	0.78	0.79	0.79	159493
Positive	0.79	0.78	0.78	158973
accuracy			0.78	318466
macro avg	0.78	0.78	0.78	318466
weighted avg	0.78	0.78	0.78	318466

CNN Model 1

	precision	recall	f1-score	support
Negative	0.78	0.78	0.78	159493
Positive	0.78	0.78	0.78	158973
accuracy			0.78	318466
macro avg	0.78	0.78	0.78	318466
weighted avg	0.78	0.78	0.78	318466

ROC - AUC Analysis of Best Performing Models



6. Results

Our raw dataset has unnecessary features for our purpose. Its first entropy value was 41.08. Then we dropped the unnecessary columns, deleted the empty valued rows, and we have obtained an entropy value of 14.73. After this preprocess, we can easily see that there is an important change in entropy values.

After all six experiments, we can see that different LSTM and CNN give us very close accuracy ratios after training. Although there are really low differences, LSTM Model-1 has the best result and Naive Bayes models performed slightly worse.

Naive Bayes models have the best training time durations. It has very good speed compared to LSTM and CNN models. LSTM model-1, LSTM model-2 and CNN model-1 have close training times as each epoch takes 10 to 13 minutes for these models. Although changing the batch size in LSTM did not give an effective result difference, CNN model-2 has a better training time like 7 to 8 minutes for each epoch. Also, its accuracy is really close to the others.

LSTM model-1 has 78.9% accuracy rate with 1024 batch size and LSTM model-2 has 78.6% accuracy rate with 512 batch size. CNN model-1 has 78.2% accuracy rate with 1024 batch size and CNN model-2 has 77.2% accuracy rate with 512 batch size. Both algorithms have better training times with 512 batch size, are better than their 1024 batch sized models and their accuracy rates are really close. As a result of these, we can say that LSTM and CNN models with 1024 batch size are better for accuracy rate. But, models with 512 batch size have close accuracy rates within better training times.

For accuracy rates of Naive Bayes models there is a small difference like 1.5%. As a result of that, we can say that Naive Bayes with the CountVectorizer method gives better results than Naive Bayes with the TF-IDF method.

Features

We use Term Frequency-Inverse Document Frequency (TF-IDF) to transform the text data. You can obtain the tf-idf array from Figure 1.

	00	000	0000	002	00am	00pm	01	02	026	02am	...	½sklov	½ssen	½sunday	½t	½tieï	½tobe	½u	½ve	½y	½ï
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 1. tf-idf array.

We used the Elbow method to make sure we choose the optimal number of clusters. We decided to make experiments 2 and 3 number of clusters.

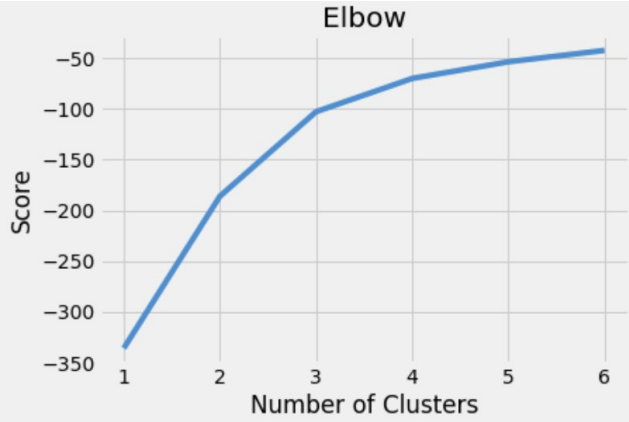
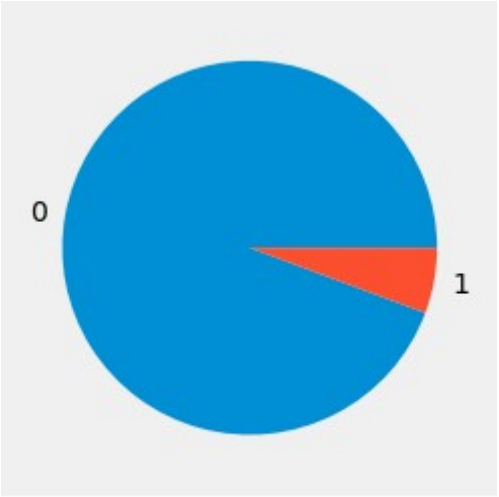


Figure 2. Elbow method to get optimal number of clusters.

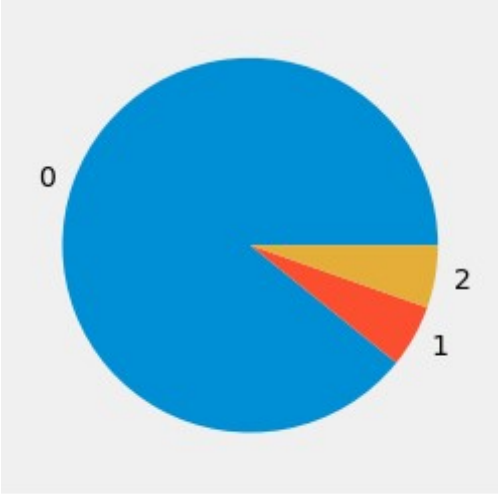
Instance Distributions Pie Chart



28300 1700

Cluster 0 Percentage = 94.3%
Cluster 1 Percentage = 5.7%

Figure 3. A pie chart showing the instance distributions for 2 clusters.



26584 1609 1609

Cluster 0 Percentage = 88.6%
Cluster 1 Percentage = 5.7%
Cluster 2 Percentage = 5.7%

Figure 4. A pie chart showing the instance distributions for 3 clusters.

Evaluation of Clustering Experiments

- **Experiment 1 - Number of clusters = 2**

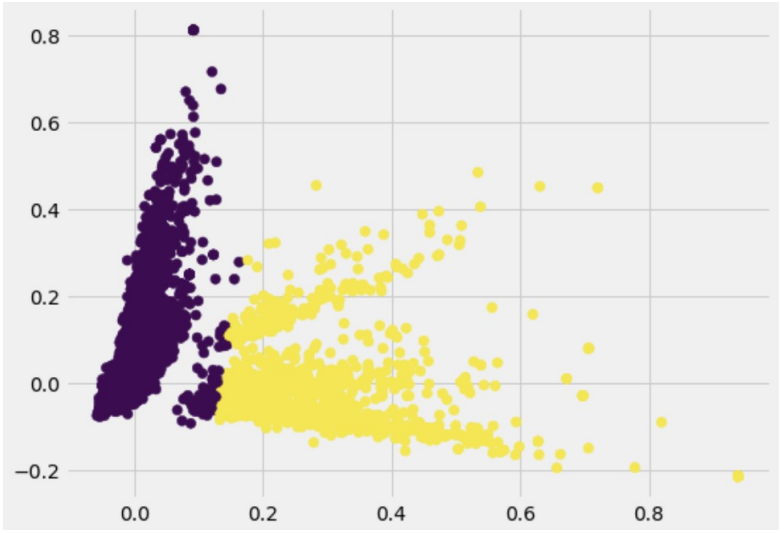


Figure 5. 2 Clusters

init	time	inertia	homo	compl	v-meas	ARI	AMI	NMI	silhouette
k-means++	0.093s	38122	0.973	0.970	0.971	0.991	0.971	0.971	0.814
random	0.108s	38122	0.975	0.970	0.972	0.991	0.972	0.972	0.749
PCA-based	0.050s	38985	0.011	0.010	0.010	0.068	0.010	0.010	0.723

Figure 6. Evaluation metrics for 2 clusters.

Most important words in Cluster 0:

	word	score
0	just	0.015132
1	day	0.012346
2	today	0.011476
3	like	0.010374
4	want	0.010060
5	going	0.010016
6	don	0.009887
7	really	0.009350
8	got	0.009332
9	sad	0.008994
10	good	0.008851
11	miss	0.008415
12	time	0.008402
13	know	0.008327
14	im	0.008257
15	wish	0.008104
16	home	0.008088
17	sorry	0.007745
18	sleep	0.007660
19	night	0.007330

Most important words in Cluster 1:

	word	score
0	work	0.302107
1	tomorrow	0.028365
2	day	0.027841
3	today	0.027249
4	going	0.023979
5	ready	0.017308
6	time	0.015752
7	home	0.015085
8	got	0.014318
9	want	0.014249
10	morning	0.014045
11	bed	0.013746
12	getting	0.013460
13	don	0.012802
14	just	0.012368
15	tired	0.011848
16	night	0.011170
17	sleep	0.010848
18	gotta	0.010505
19	hours	0.010329

- **Experiment 2 - Number of clusters = 3**

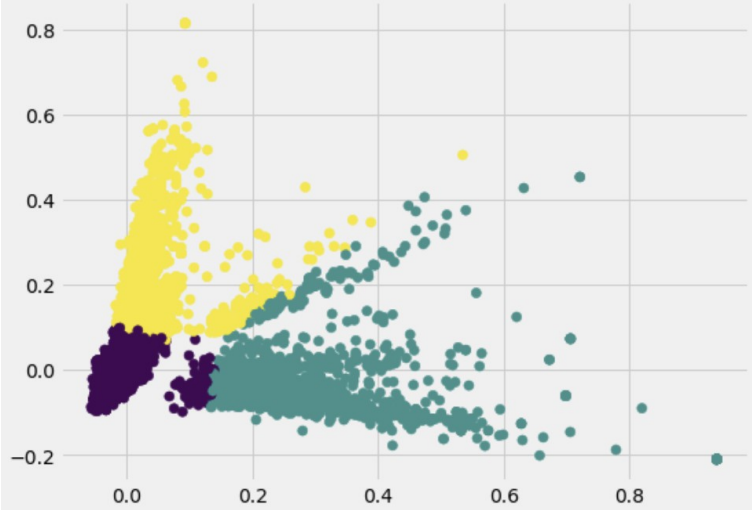


Figure 7. 3 Clusters

init	time	inertia	homo	compl	v-meas	ARI	AMI	NMI	silhouette
k-means++	0.172s	49102	0.455	0.426	0.440	0.471	0.440	0.440	0.681
random	0.210s	49102	0.454	0.424	0.438	0.470	0.438	0.438	0.717
PCA-based	0.062s	49102	0.455	0.425	0.439	0.471	0.439	0.439	0.662

Figure 8. Evaluation metrics for 2 clusters.

Most important words in Cluster 0: Most important words in Cluster 1: Most important words in Cluster 2:

	word	score
0	just	0.015528
1	like	0.010524
2	want	0.009992
3	don	0.009947
4	got	0.009460
5	really	0.009307
6	sad	0.008864
7	going	0.008797
8	miss	0.008738
9	know	0.008549
10	im	0.008349
11	good	0.008327
12	time	0.008284
13	wish	0.008086
14	sorry	0.008076
15	today	0.007920
16	home	0.007797
17	sleep	0.007639
18	need	0.007373
19	night	0.007292

	word	score
0	work	0.308336
1	tomorrow	0.027660
2	today	0.025722
3	going	0.024375
4	day	0.018545
5	ready	0.017846
6	time	0.016309
7	home	0.015314
8	got	0.014329
9	want	0.014295
10	morning	0.014260
11	getting	0.014222
12	bed	0.014176
13	don	0.013260
14	just	0.012587
15	tired	0.012140
16	sleep	0.011213
17	night	0.010812
18	hours	0.010564
19	need	0.010554

	word	score
0	day	0.199634
1	today	0.065946
2	school	0.059420
3	tomorrow	0.057504
4	going	0.028303
5	good	0.017077
6	long	0.015431
7	beautiful	0.013727
8	break	0.013477
9	bad	0.012683
10	home	0.012522
11	bed	0.012221
12	want	0.011225
13	morning	0.010797
14	sad	0.010686
15	feeling	0.010684
16	work	0.010277
17	spring	0.010034
18	time	0.010008
19	really	0.009986

Result

K-means is a very simple and powerful algorithm to cluster a dataset. However, one of the problems is that clusters are spherical. Therefore, it can not be reliable for all situations.

We are using text data for our project. So, we need to represent the data as the model understands. For this reason, firstly, we vectorize our data with tf-idf vectorizer. Then, we use the elbow method to make sure we choose the optimal number of clusters. We decided to make experiments with 2 and 3 numbers of clusters.

Therefore, we have two different experiments with 2 and 3 clusters, we have 2 different instance distributions pie charts. For two clusters, we can see that Cluster-0 has a really huge ratio, with 94.3%, in 30,000 instances. When we applied the experiment with 3 clusters, we can see that Cluster-1 did not lose any instances, but Cluster-0 is split into two. Cluster-2 has an equal ratio with Cluster-1.

We compare three approaches kmeans++, random initialization and initialization based on PCA projection for 2 and 3 numbers of clusters. Evaluation metrics for each 2 experiments as shown in Figure 6 and Figure 8.

We score the words in each cluster in order of importance. In Experiment 1 (number of cluster is 2), we obtain most important words of Cluster-0 are like, good, wish, sorry. They can be said mostly positive words. On the other hand, Cluster-1's most important words are not distinguishable.

In Experiment 2 (number of cluster is 3), Cluster-0 has mostly positive words such as good, like, miss, sorry, wish. Cluster-1 seems like neutral and Cluster-2 has some negative words such as bad and sad.

The K-means is clustering words according to some semblance of meaning in our experiments, but experiments can be developed with even more accurate parameters.

REFERENCES

Sentiment140, http://help.sentiment140.com/home

Natural Language Toolkit, https://www.nltk.org/

Go, A., Bhayani, R. & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. Processing, 1--6.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (n.d.). Combining lexicon-based and learning-based methods for twitter sentiment analysis. Retrieved June 20, 2021, from Hpl.hp.com website: https://www.hpl.hp.com/techreports/2011/HPL-2011-89.pdf

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011).
Target-dependent Twitter sentiment classification. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, 151–160. USA: Association for Computational Linguistics.

Abid, F., Li, C., & Alam, M. (2020). Multi-source social media data sentiment analysis using bidirectional recurrent convolutional neural networks. Computer Communications, 157, 102–115.

Venkatesh, Hegde, S. U., A, Z., & Nagaraju. (2021). Hybrid CNN-LSTM model with GloVe word vector for sentiment analysis on football specific tweets. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 1–8. IEEE.

Swathi Lakshmi, B., Sini Raj, P., & Raj Vikram, R. (n.d.). Sentiment analysis using deep learning technique CNN with KMeans. Retrieved June 20, 2021, from Acadpubl.eu website: https://acadpubl.eu/jsi/2017-114-7-ICPCIT-2017/articles/11/6.pd f

-

Truong, Q.-T., & Lauw, H. W. (2017). Visual sentiment analysis for review images with item-oriented and user-oriented CNN. Proceedings of the 2017 ACM on Multimedia Conference - MM '17. New York, New York, USA: ACM Press.