**Delivery #6 - Final Presentation**

Social Media Sentiment Analysis

**Group #3**

**AKASH SG (Information science)**

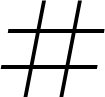
**SUDHANSHU (Information science)**

**INDRESH MJ (information science)**

**PREETHAM KM (information science)**

**SHASHANKK (information science)**

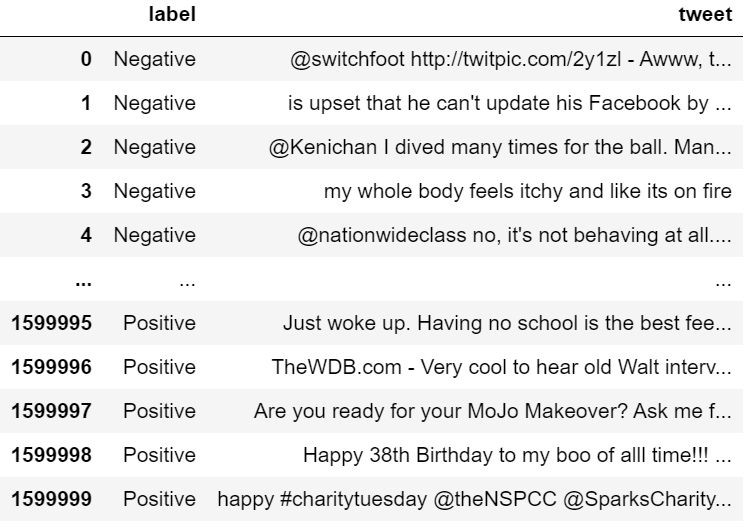
# Project Delivery 1

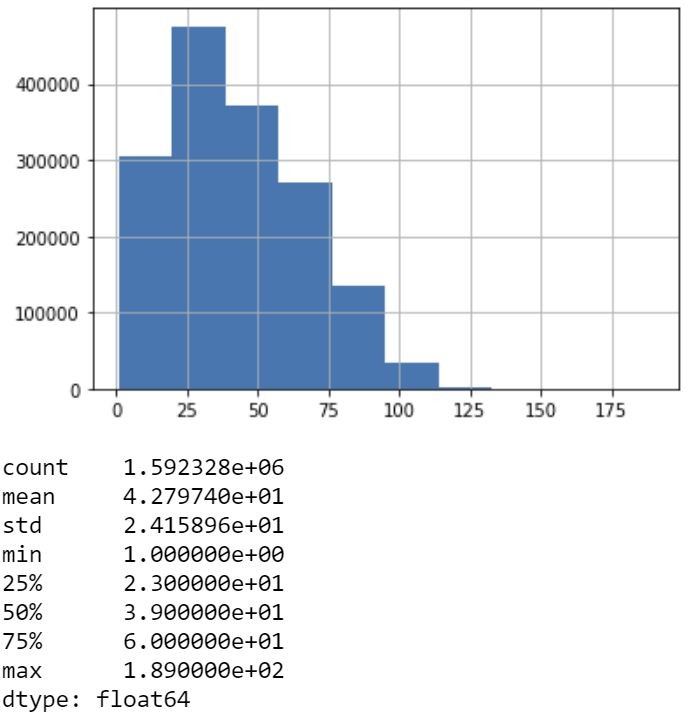


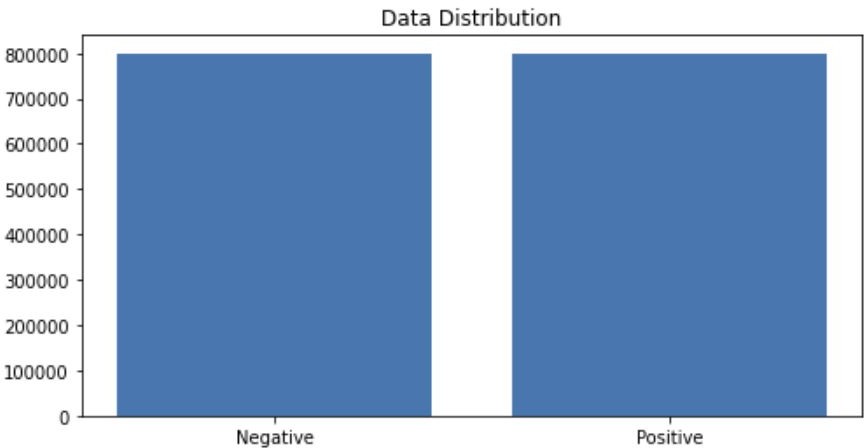
## Dataset

*Figure 1. A sample from the dataset.*

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.

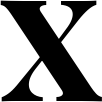
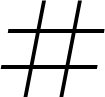
These 6 ﬁelds are shown below.

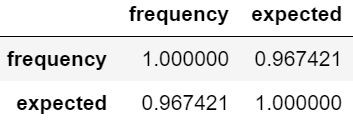
* target: The polarity of the tweet (0 - negative, 2 - neutral, 4 - positive)
* ids: The id of the tweet.
* date: The date of the tweet.
* ﬂag: The query. If there is no query, then this value is NO\_QUERY.
* user: The user that tweeted.
* text: The text of the tweet

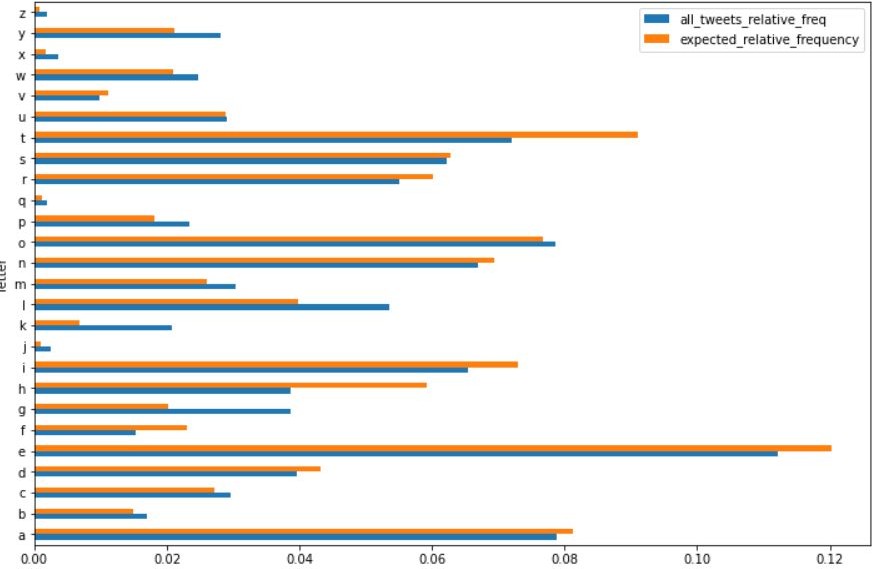
*Figure 2. Dataset after reduction.*

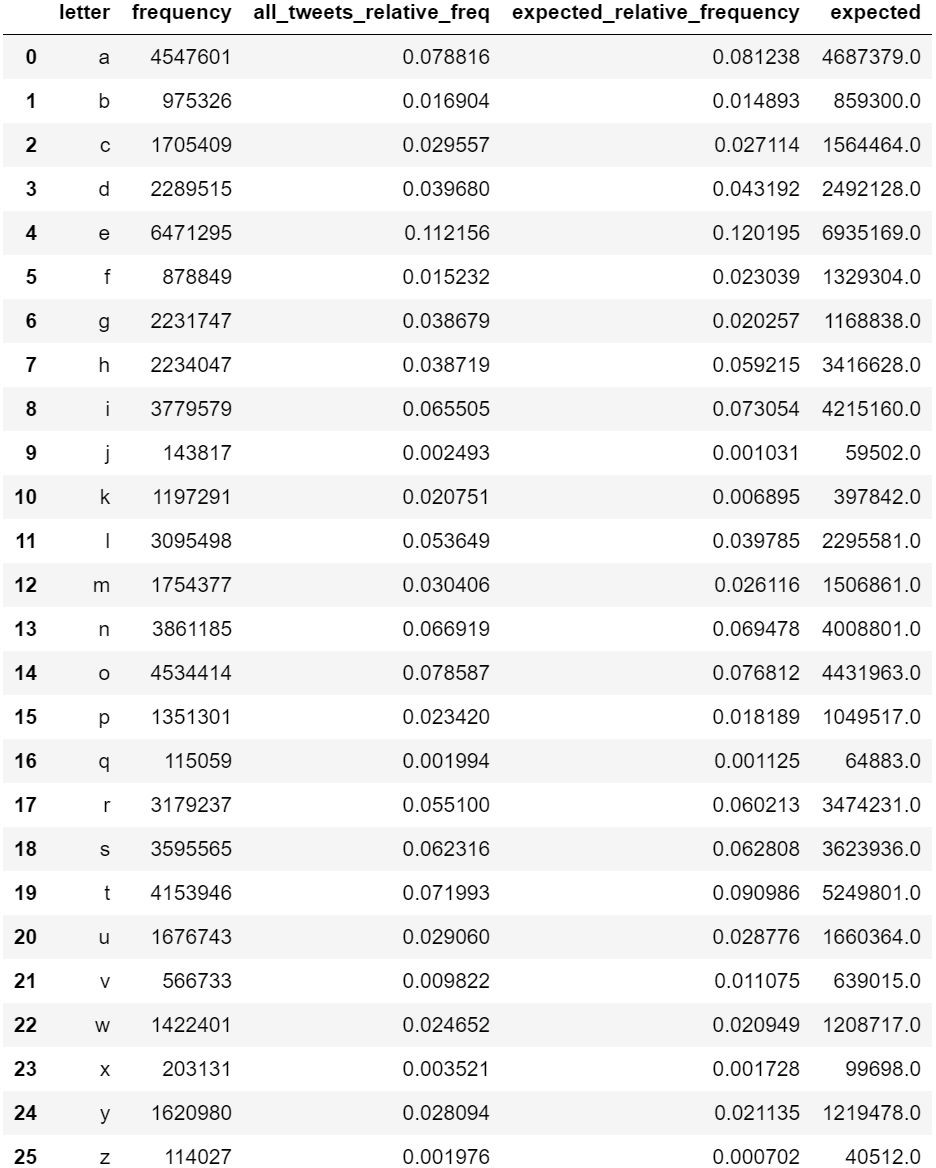
*Figure 3. Data distribution.*

# Project Delivery 2-3 E ploring Your Data





*Figure 4. Letter frequencies of each 26 characters in English Alphabet.*

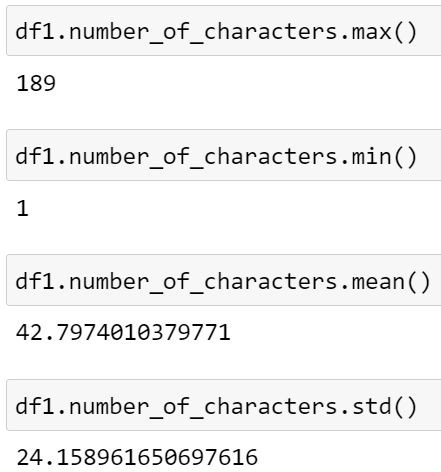
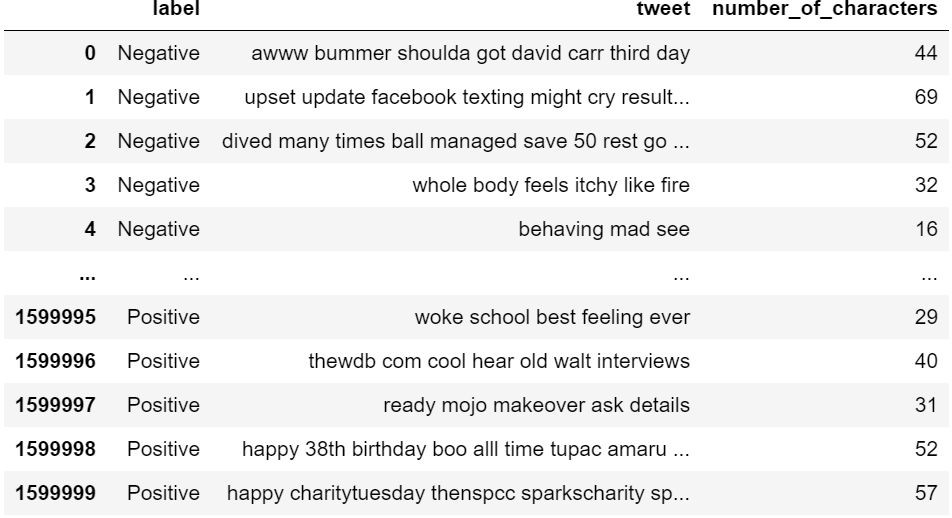
*Figure 5. Letter frequency of the dataset, relative frequencies of the dataset, expected relative frequency according to the English language and expected character length according to the English language.*

*Figure 6. Correlation*

*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%)*

*.*

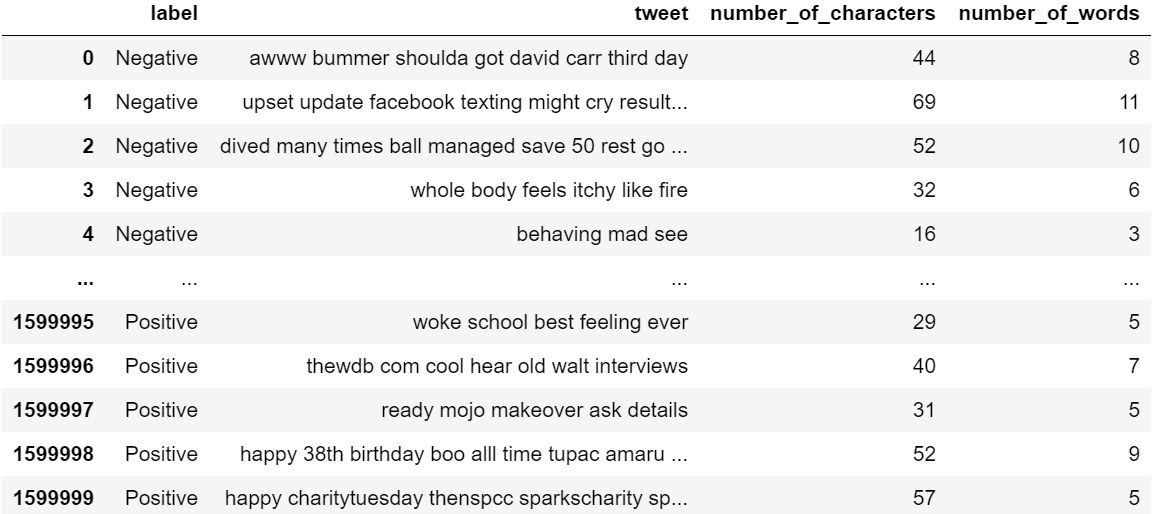
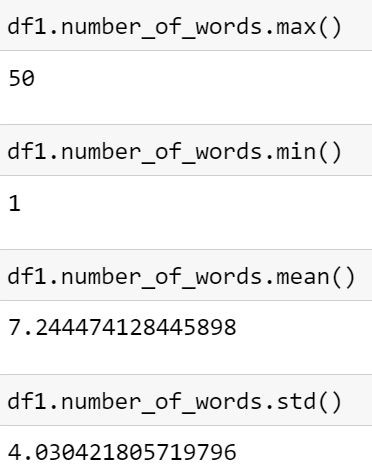
We counted the number of characters for each tweet and analyzed the data frame according to maximum number of characters, minimum number of characters, mean of the number of characters column and its standard deviation. Our longest tweet is 189 characters long, the shortest tweet is 1 character long and mean of all tweets’ character length 42.78. The standard deviation of all tweet character length is 24.16 as shown in Figure 8.



*Figure 7. Number of characters. Figure 8. Letter frequency of the dataset, relative frequencies of the dataset, expected relative frequency according to the English*

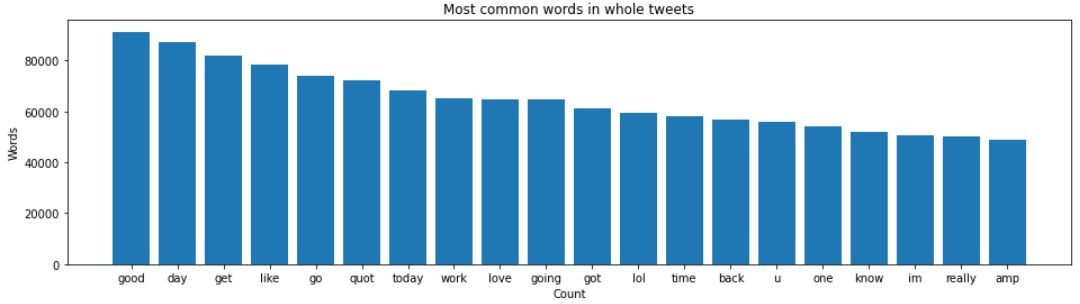
*language and expected character length according to the English language.*

We counted the number of words for each tweet 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 10.

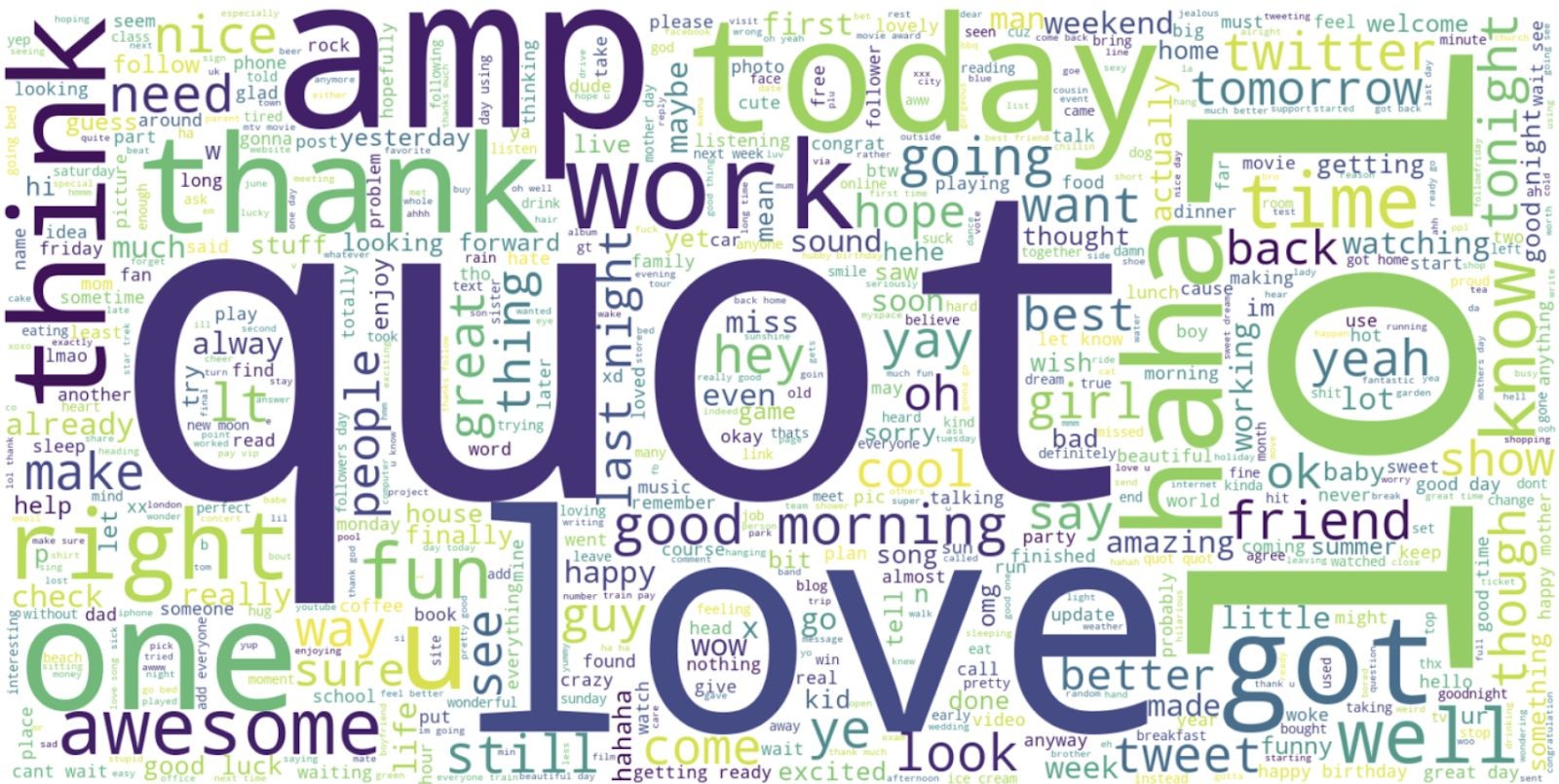
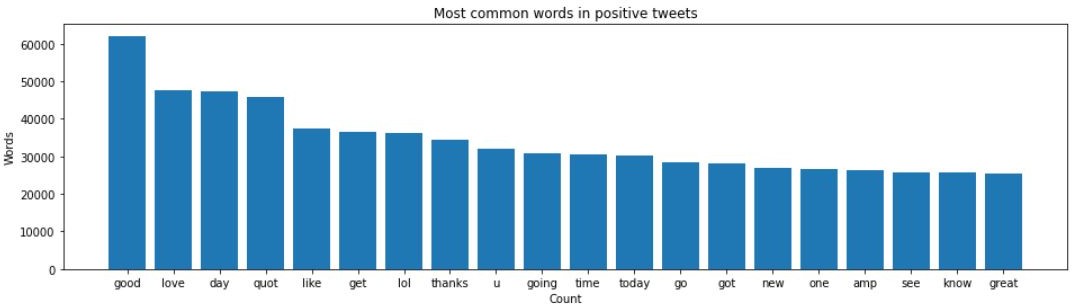
 

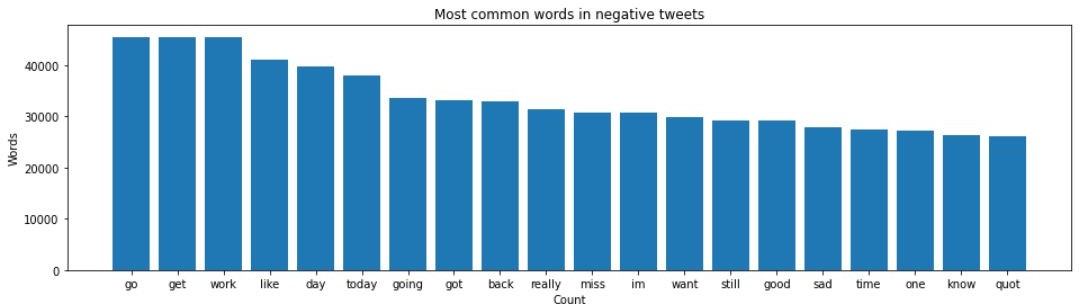
*Figure 9. Number of words of each tweet.*

*Figure 10. Max, min, mean and standard deviation of each tweet in terms of number of words.*

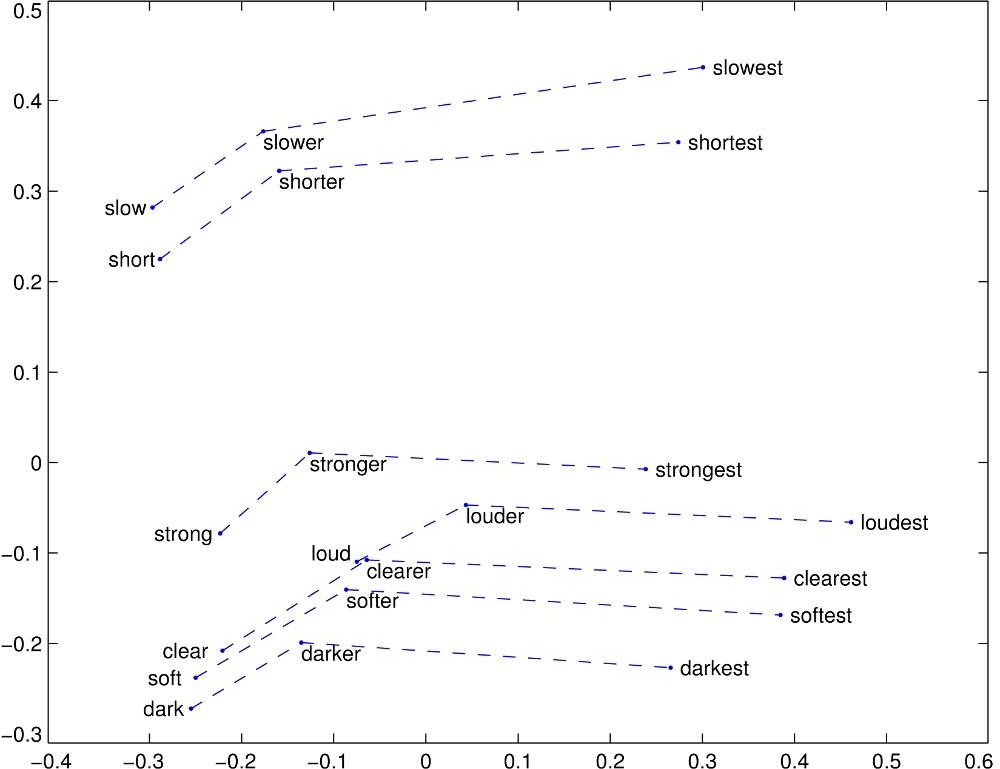


*Figure 11. Most common words in whole data.*

*Figure 12. Most common words in positive tweets.*



*Figure 13. Most common words in positive tweets.*

We can train the embedding ourselves. However, that approach can take a long time to train. So, we use transfer learning technique, and we use GloVe: Global Vectors for Word Representation.

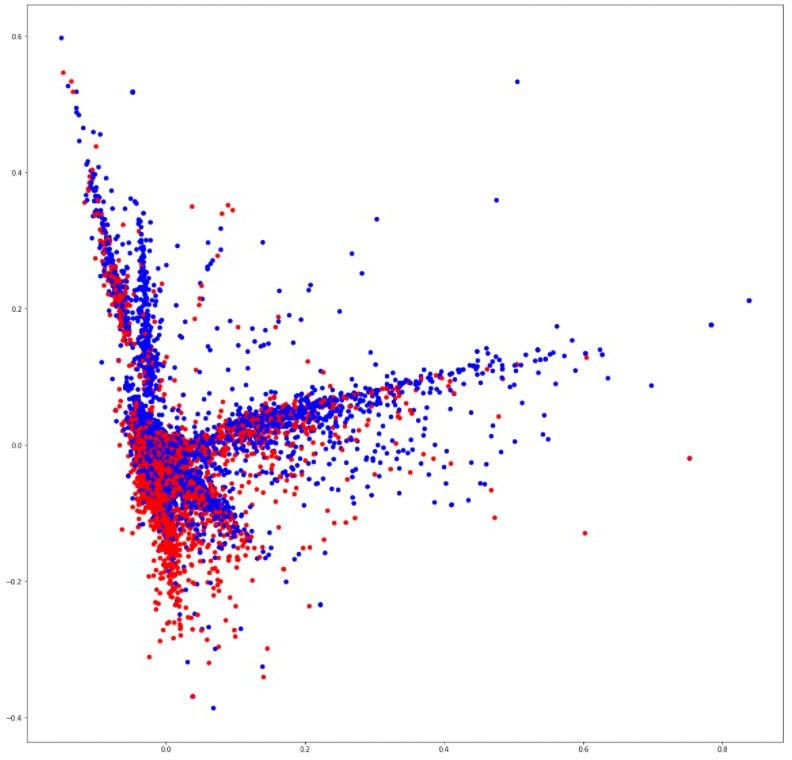
The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for eﬃciently learning word vectors, developed by Pennington, et al. at Stanford. It is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

We download the GloVe. Then, we initialize an embedding

index that has 400000 word vectors, and embedding matrix. *Figure 14. GloVe embedding example*

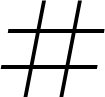
We used feature extraction methods, bag-of-words, and word embedding. Bag of words with TF-IDF is a common and simple way of feature extraction. Bag-of-Words is a representation model of text data and TF-IDF is a calculation method to score the importance of words in a document.

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



*Figure 15. Scatter plot that shows correlation of words in the corpus: red indicates negatives, blue indicates positives.*

# Project Delivery 4 Predictive Analysis



### 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.

### First entropy of dataset = 41.08269441306875

* Entropy after preprocess = 14.73368002815221

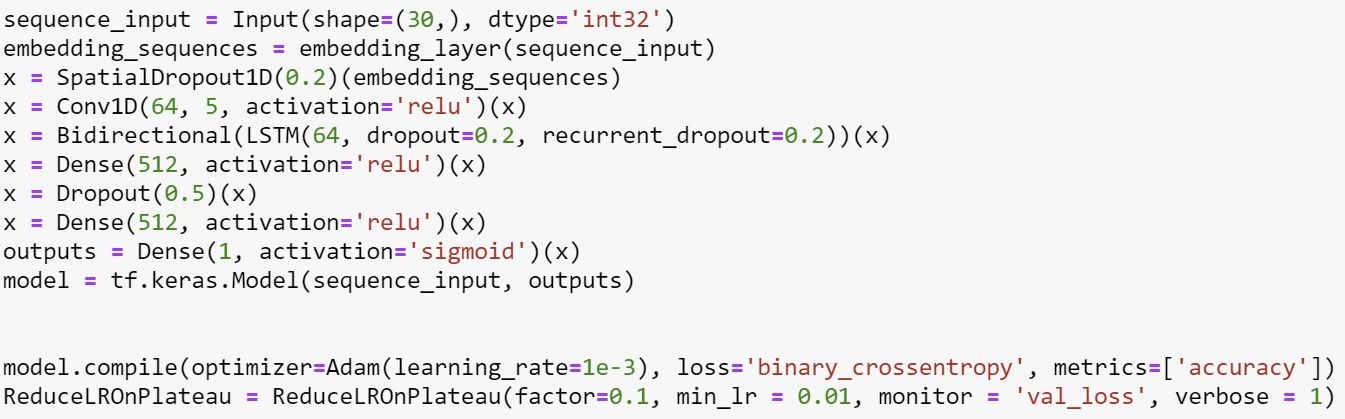
For classiﬁcation/regression experiments, the test set percentage is set to be 20%.

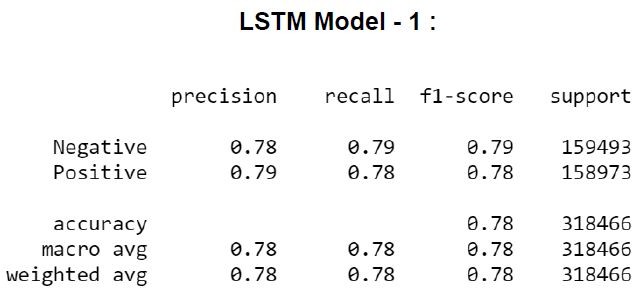
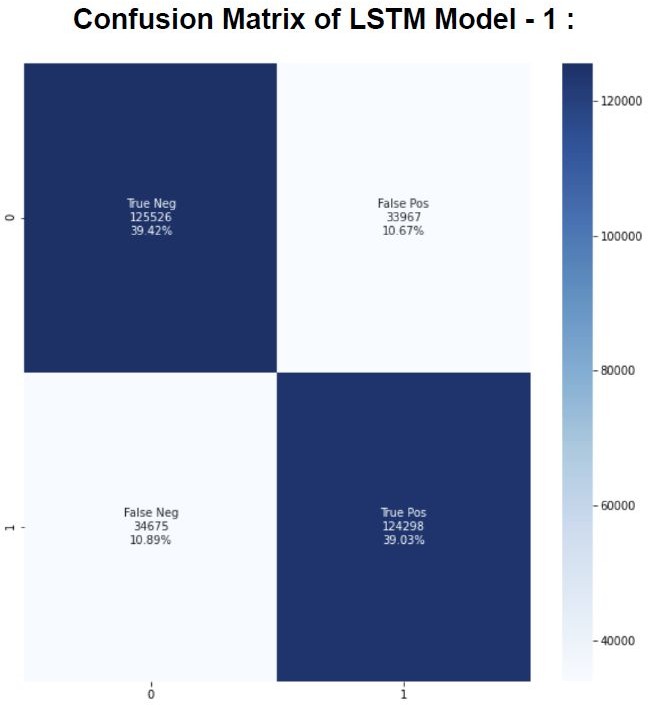
* Total Data = Train Data (80%) + Test Data (20%)

Used 3 different algorithm and 6 different model for classiﬁcation. These are :

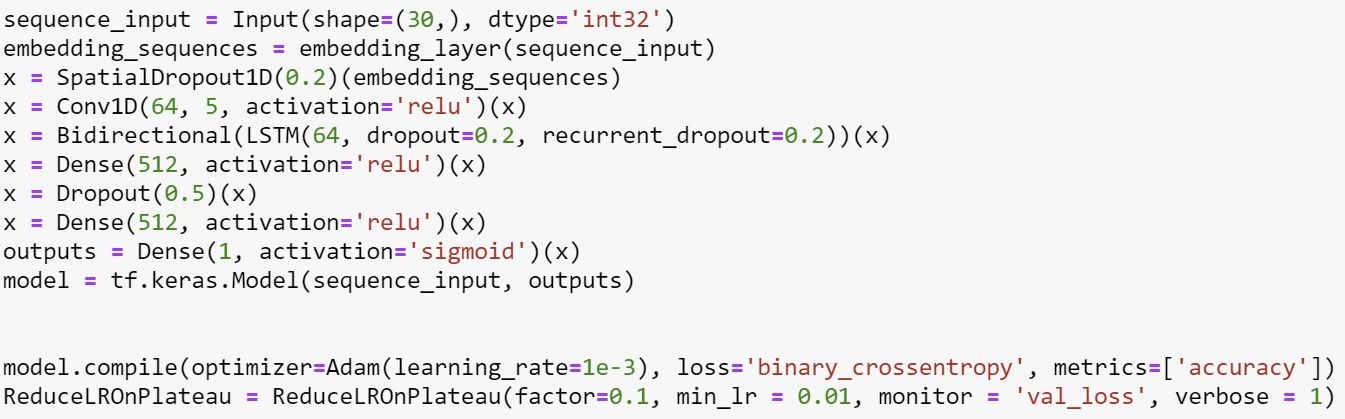
* LSTM with 1024 Batch Size
* LSTM with 512 Batch Size
* CNN with 1024 Batch Size
* CNN with 512 Batch Size
* Multinomial Naive Bayes with Count Vectorizer
* Multinomial Naive Bayes with TF-IDF Vectorizer

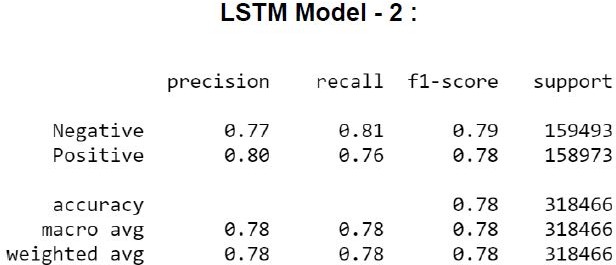
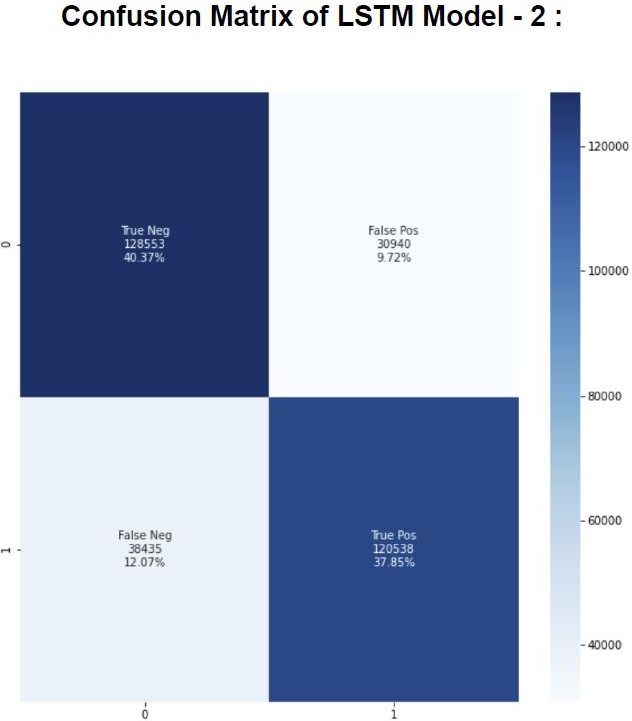
### Batch Size = 1024



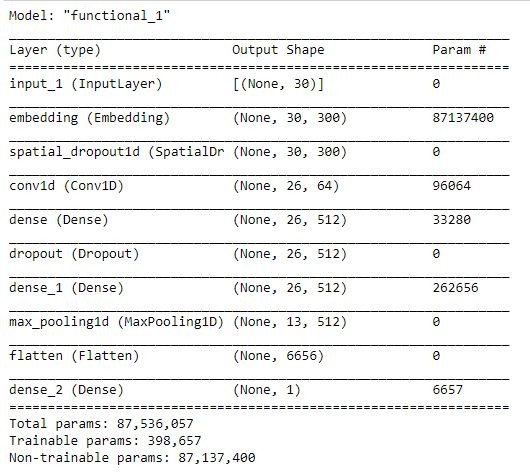
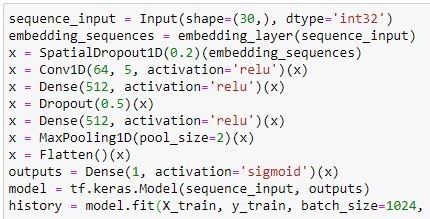
### Batch Size = 512



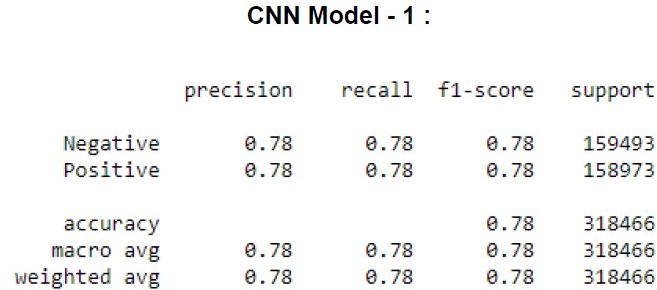
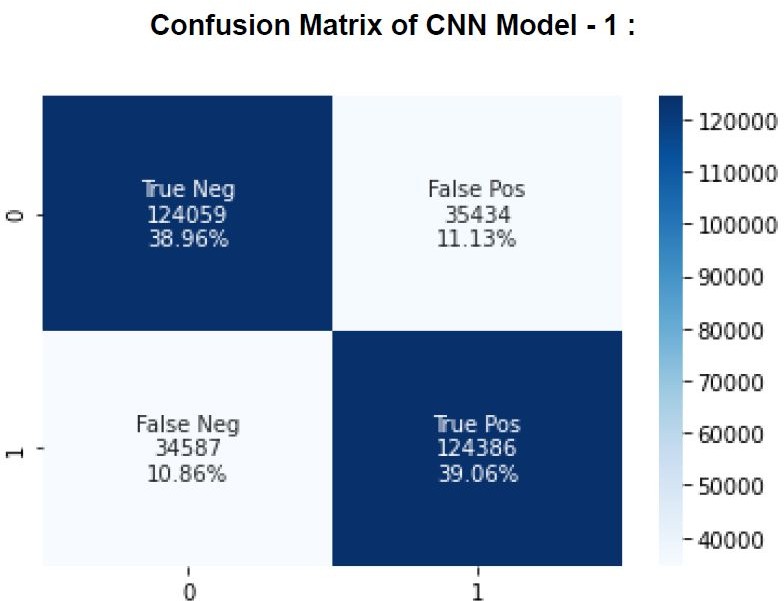
 

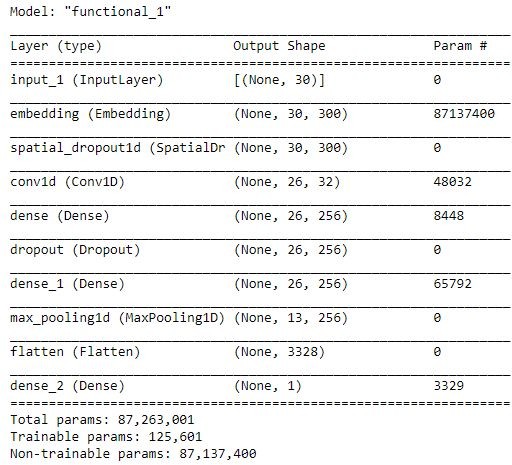
### Decreasing the batch size from 1024 to 512 did not make a signiﬁcant change in

accuracy.

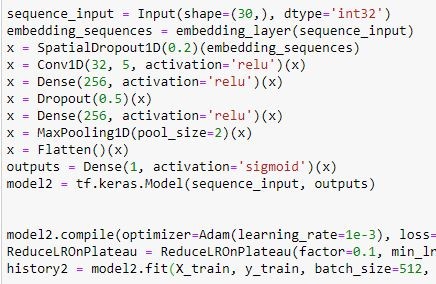


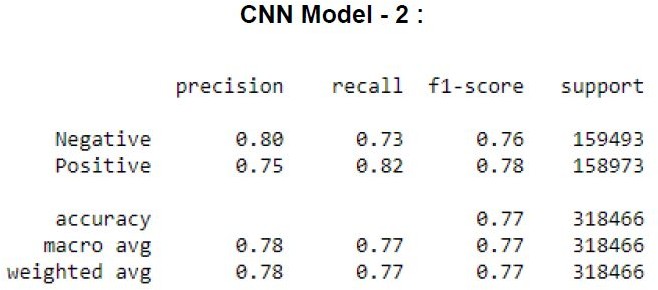
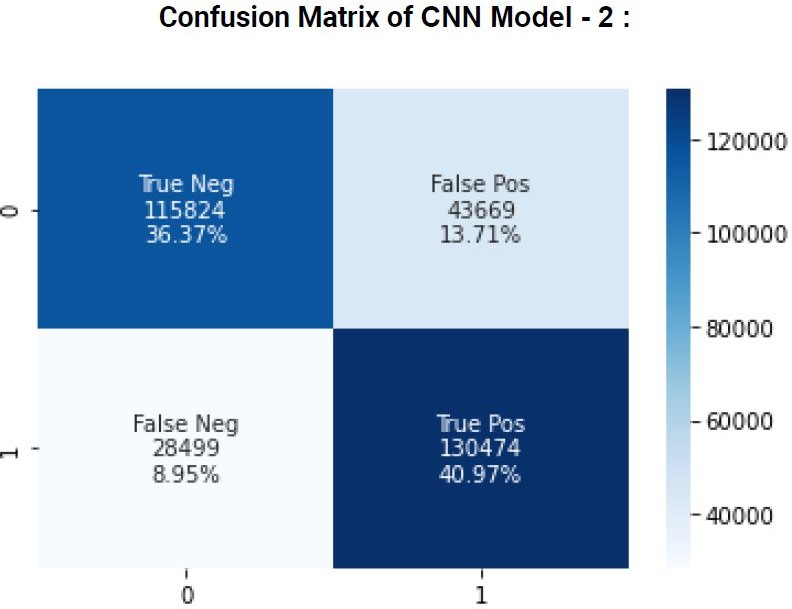
### Batch Size = 1024



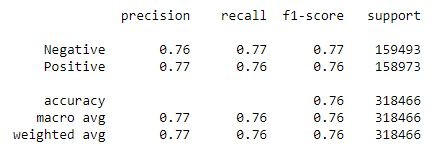
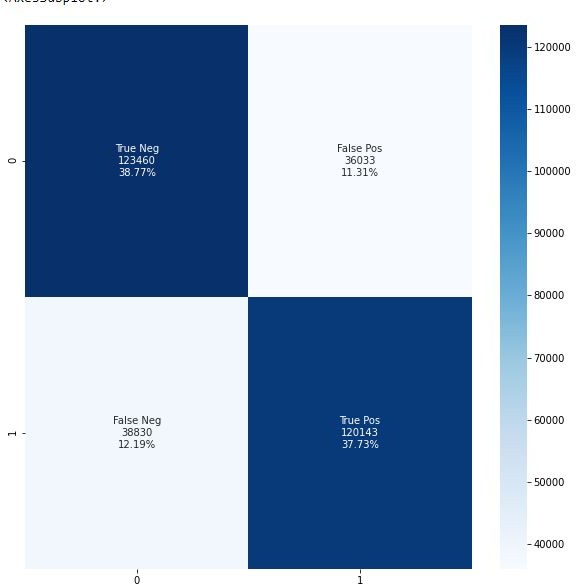
### Batch Size = 512

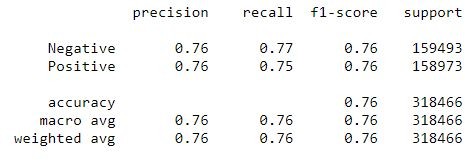
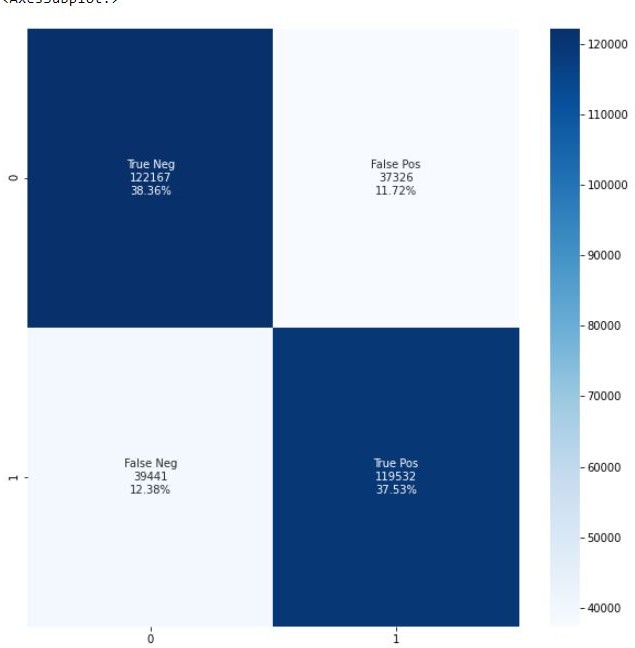


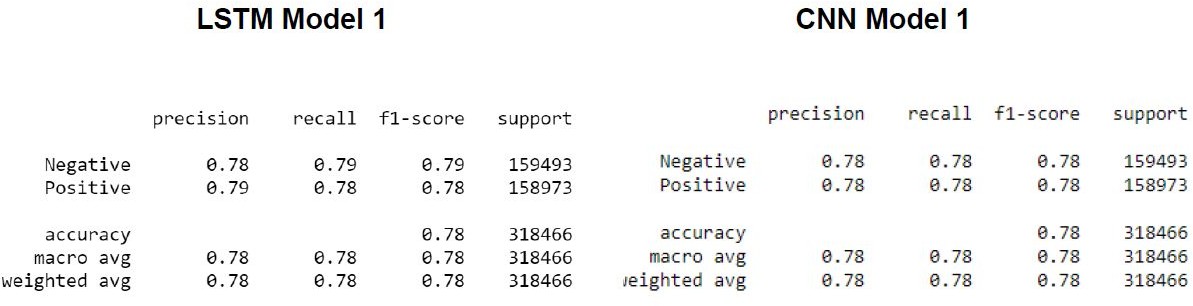
### Decreasing the batch size from 1024 to 512 did not make a signiﬁcant change in

accuracy.





* Best performing model is LSTM Model - 1 with accuracy 0.789
* Second best performing model 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.



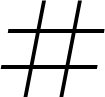
### LSTM and CNN results are very close to each other.

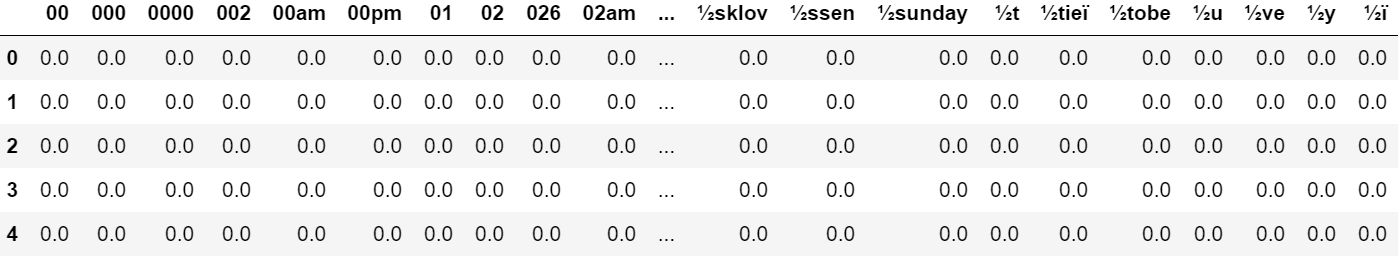
* Naive Bayes models performed slightly worse.

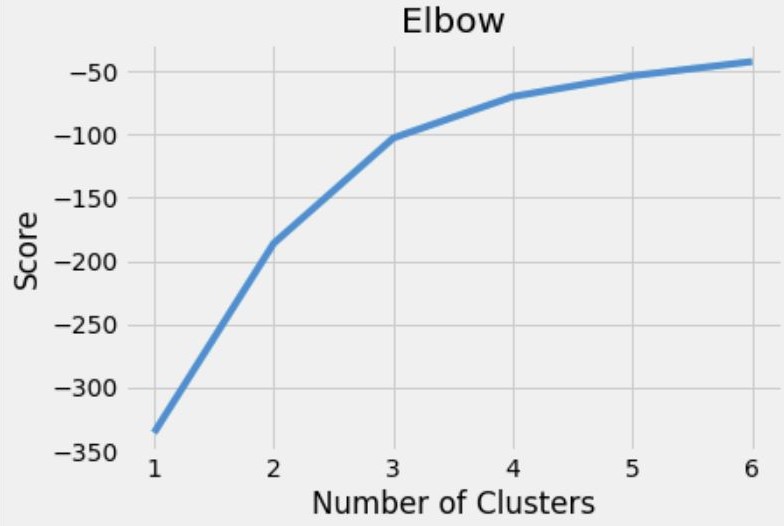
### Naive Bayes models have the best training time durations.

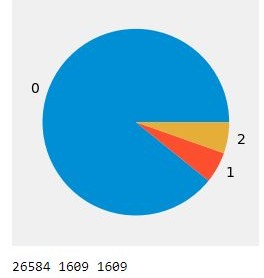
* LSTM Model-1 has 78.9% accuracy rate and LSTM model-2 has 78.6% accuracy rate. CNN model-1 has 78.2% accuracy rate and CNN model-2 has 77.2% accuracy rate.
* Both algorithms have better training times with 512 batch size, are better than their 1024 batch sized models and their accuracy rates are really 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.

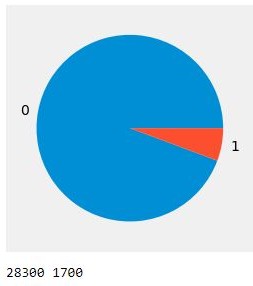
# Project Delivery 5 Descriptive Analytics



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

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



Cluster 0 Percentage = 94.3%

Cluster 1 Percentage = 5.7%

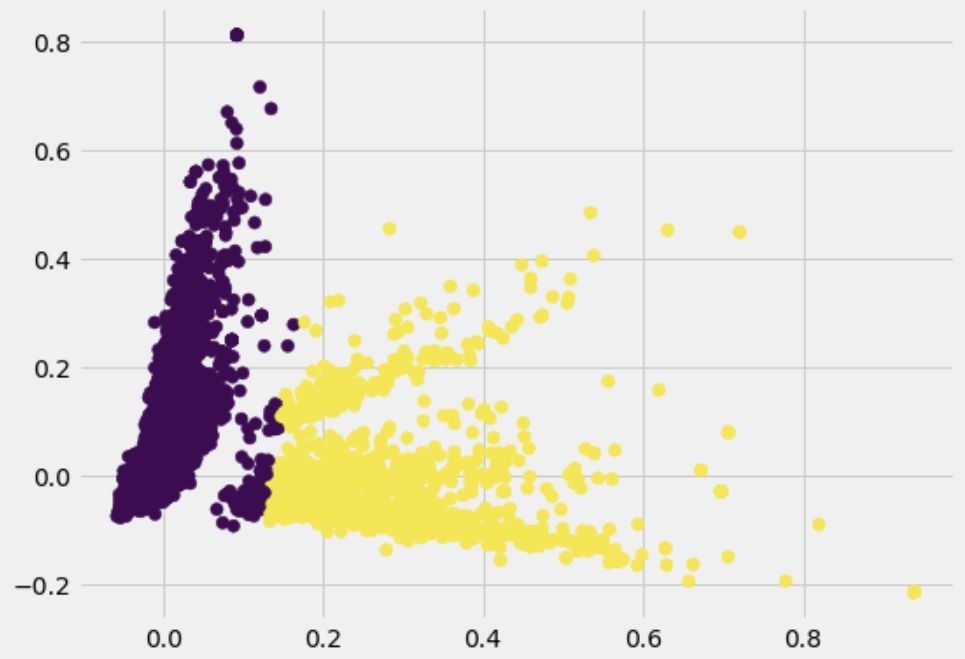
Figure 3. A pie chart showing the instance distributions for 2 clusters.

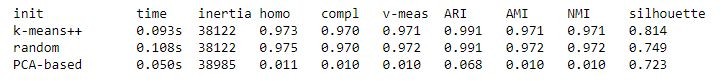
Cluster 0 Percentage = 88.6%

Cluster 1 Percentage = 5.7%

Cluster 2 Percentage = 5.7%

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



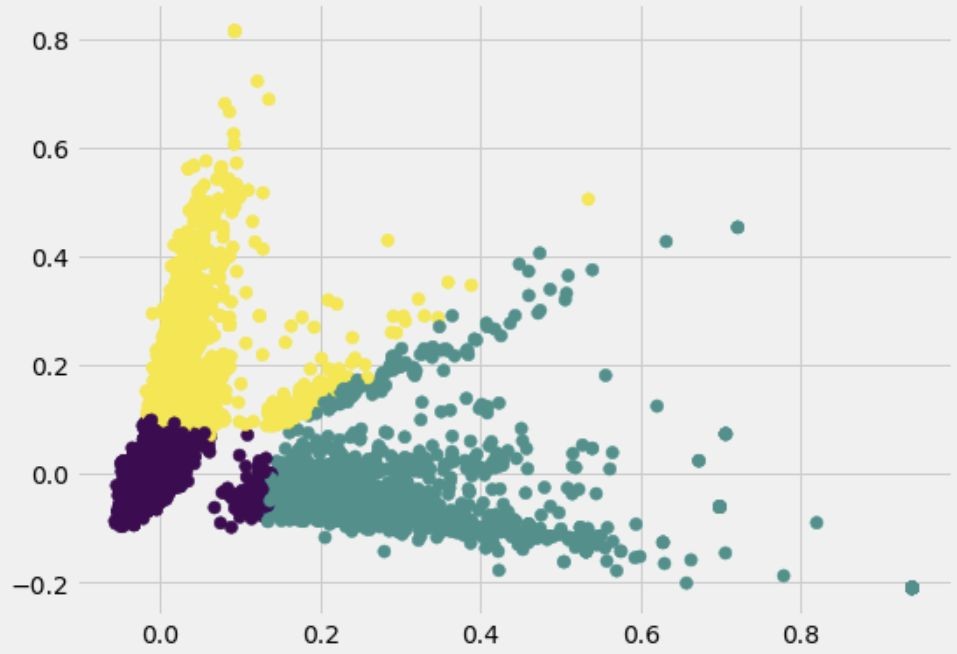


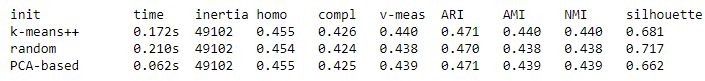
Evaluation metrics for 2 clusters.

Most important words in Cluster 0 Most important words in Cluster 1

## Evaluation of Experiments - Experiment 2 Number of Clusters = 3





Evaluation metrics for 2 clusters.

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

## 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.

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.

# Thank you for listening