**Review-1 Report**

**Twitter Sentiment Analysis: Insights from 1.6 million Tweets**

Guide name: Dr.Eliazer M.

Designation: Assistant Professor  
Department: C.Tech.

Batch ID:

Student 1 Reg. No: RA2011003010370

Student 1 Name: Ali Hassan

Student 2 Reg. No: RA2011003010395

Student 2 Name: Nitin Shukla

## 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.

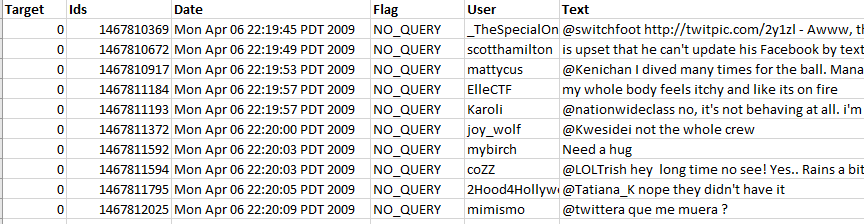
We searched for some tools for our project and found some popular and powerful open-source NLP frameworks in Python. We will probably use Natural Language Toolkit (NLTK) [3]. It comes with all the pieces you need to get started on sentiment analysis.

## Data

**Dataset :** Sentiment140

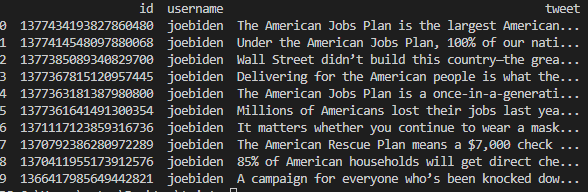
This dataset contains 1,600,000 tweets extracted using the twitter api. The tweets have been classified from 0 (negative) to 4 (positive). The dataset contains 6 fields which are target as integer, ids as integer, date as date, flag as string, user as string and text as string.These 6 fields are shown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



*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.

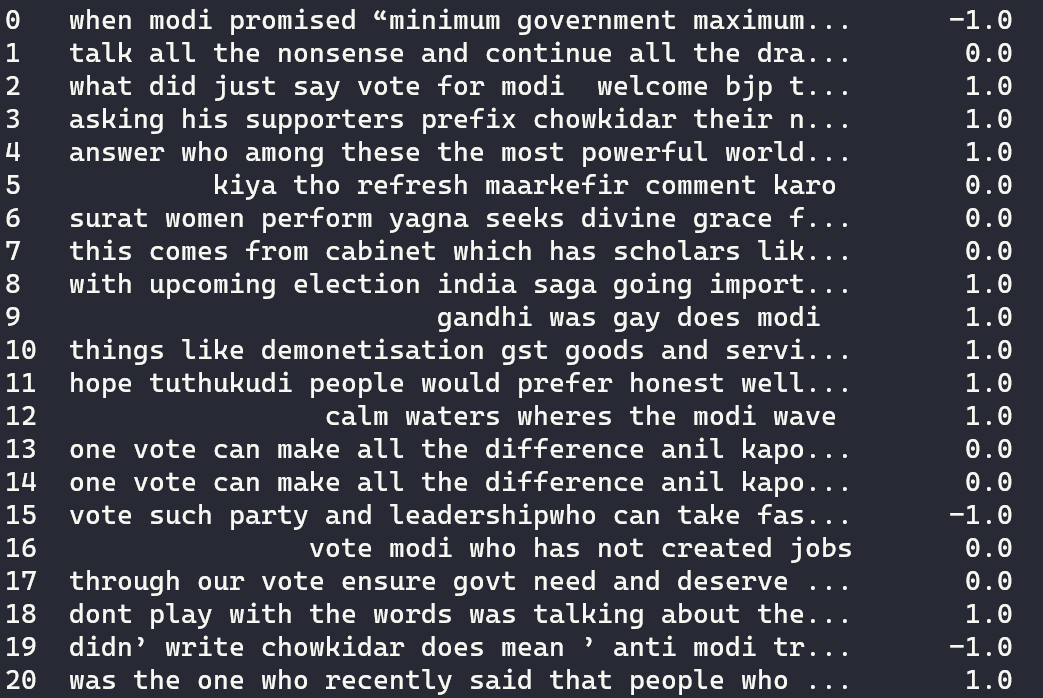


*Figure 2. Joe Biden’s tweets collected using TWINT.*

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.

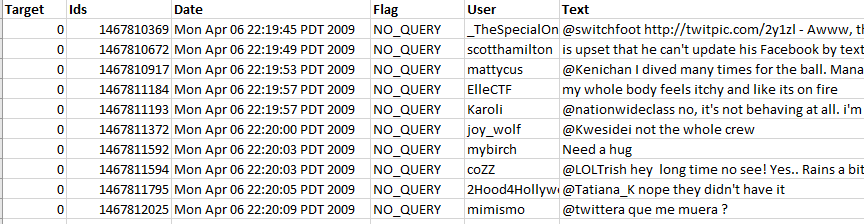


*Figure 3. Labels indicate that 1 positive, 0 neutral and -1 negative comment.*

# Dataset

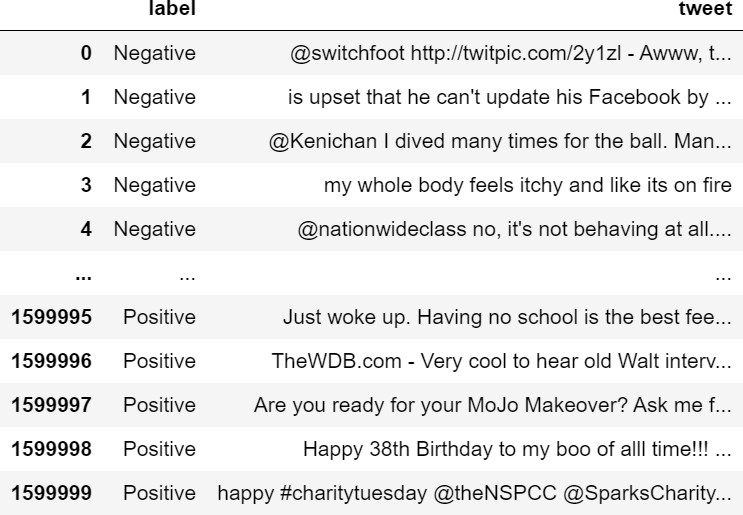
The dataset [1] contains 1,600,000 tweets extracted using the twitter api. The tweets have been classiﬁed from 0 (negative) to 4 (positive). The dataset contains 6 ﬁelds which are target as integer, ids as integer, date as date, ﬂag as string, user as string and text as string.These 6 ﬁelds are shown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

The dataset has a dimension of 1600000×2 after necessary data reduction is applied(It can be seen in Figure 2).



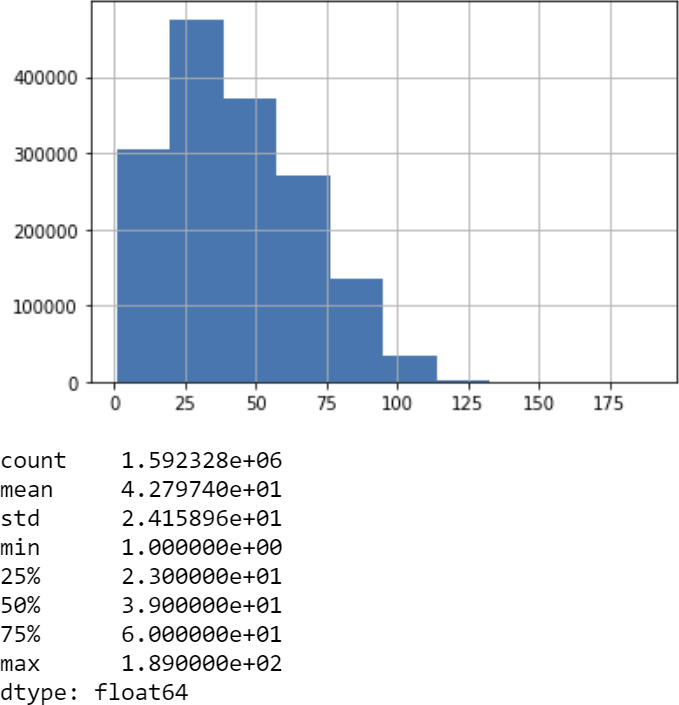
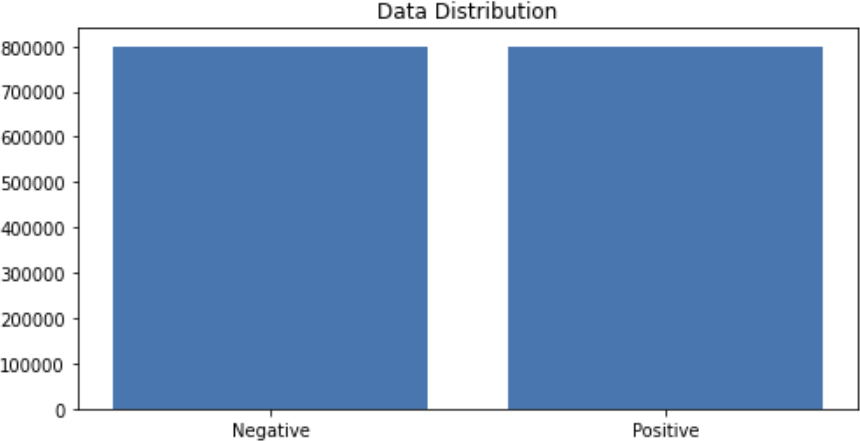
*Figure 2. Dataset*

The features/attributes of the dataset is as follows after data reduction is applied:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Feature Name | Description | Type | # of values | Missing Values % |
| 1 | label | Negative or Positive | nominal | 1600000 | %0.4795 |
| 2 | tweet | Tweets | text | 1600000 | %0.4795 |

*Figure 3. Dataset features/attributes.*

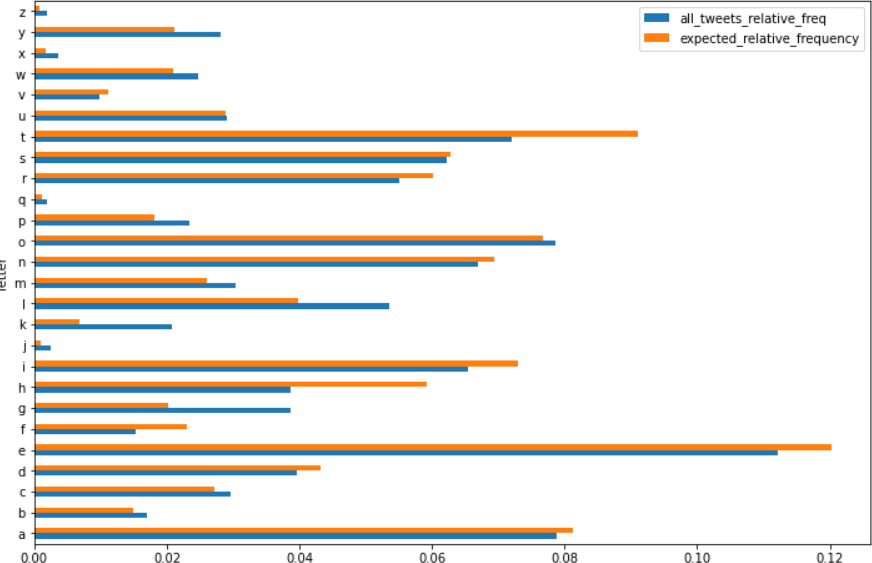
We remove tweets that have a length of 0. After this process, the dataset has a dimension of 1592328×2

Positive and negative samples are equal. The dataset distribution has not any skewness as shown in Figure 4.

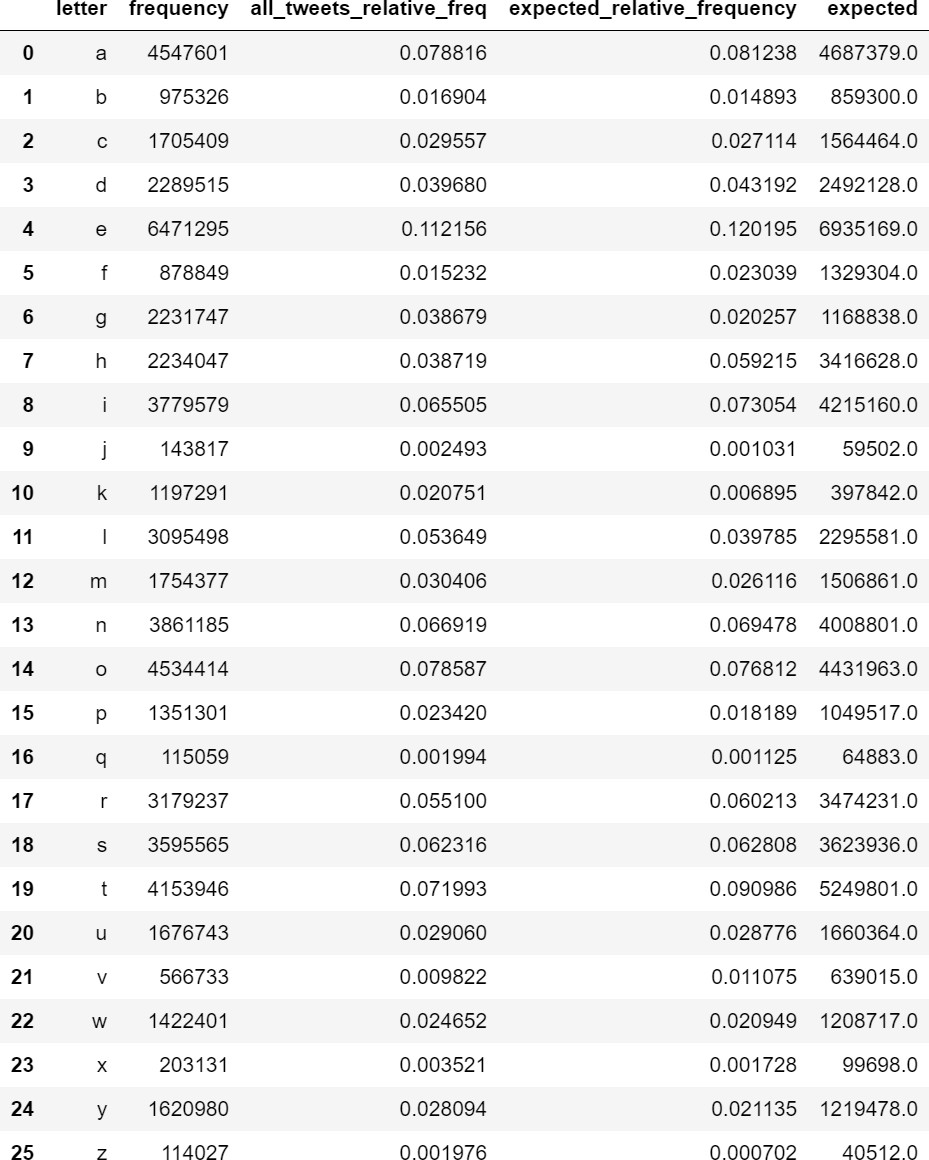
# Number of Letters

*Figure 4. Dataset distribution*

We provide the frequency and the relative frequency of the letters of the whole tweets. Finally, we will apply a chi-square test to test if the distribution of the letters in tweets is the same with what we see in English texts.

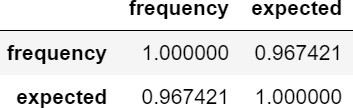


*Figure 5. Letter frequencies of each 26 characters in English Alphabet.*



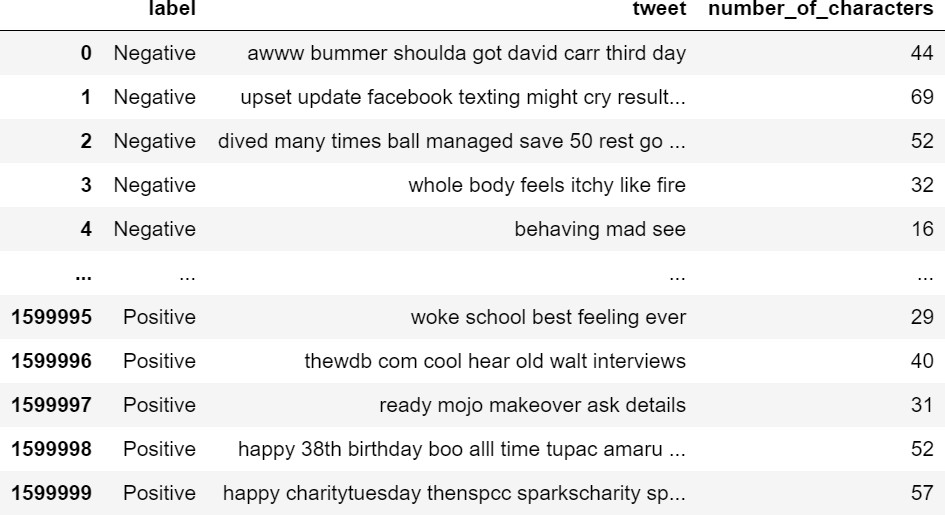
*Figure 6. Letter frequency of the dataset, relative frequencies of the dataset, expected relative frequency according to the English language and expected character length according to the English language.*

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

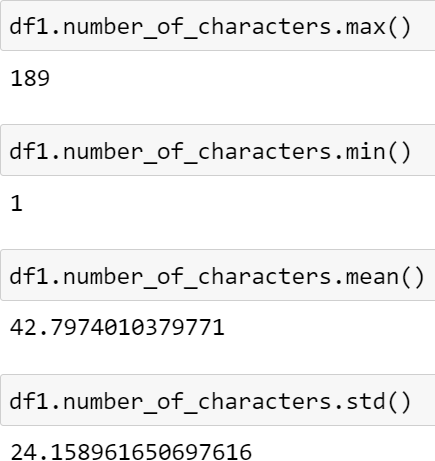


*Figure 7. Correlation.*

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 9.



*Figure 8. Number of characters.*

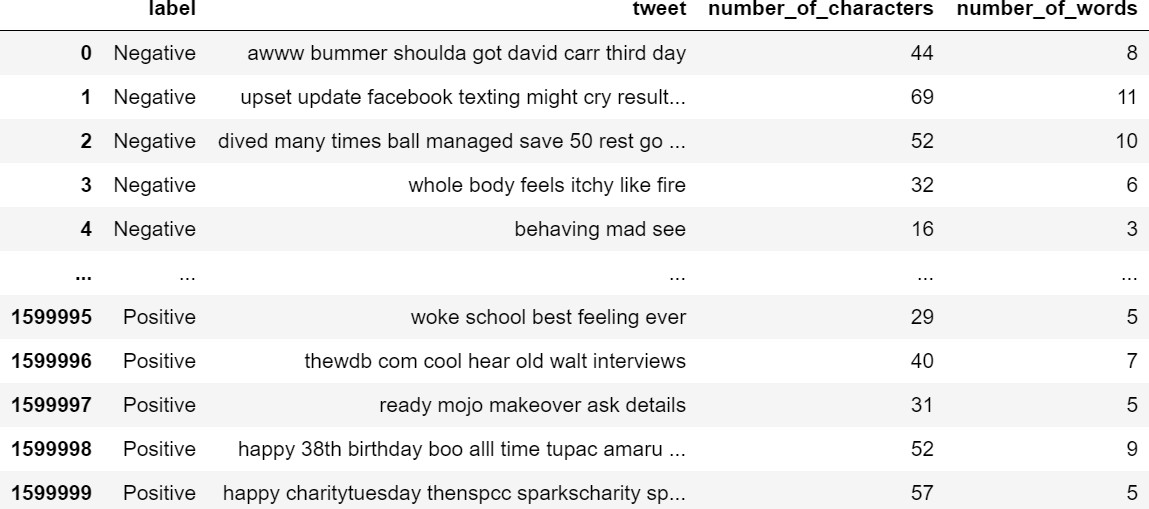


*Figure 9. Max, min, mean and standard deviation of each tweet in terms of character length.*

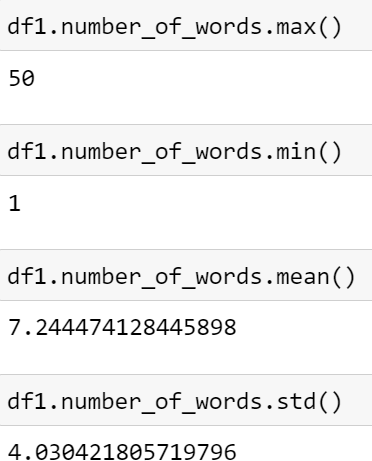
# Number of Words

We counted the number of words for each tweet 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

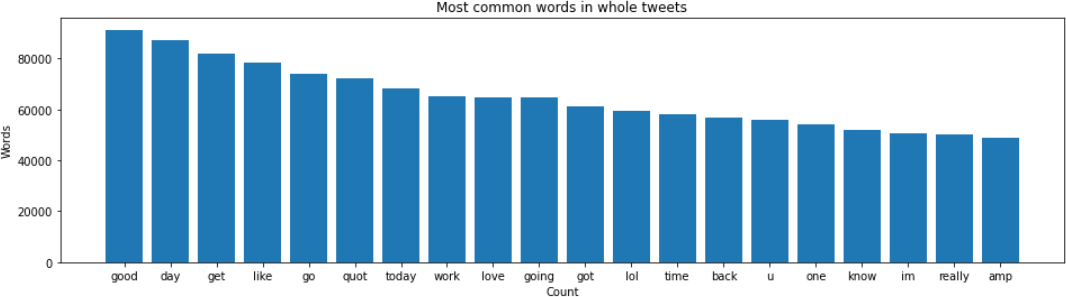
7.24. The standard deviation of all tweet character length is 4.03 as shown in Figure 11.



*Figure 10. Number of words of each tweet.*

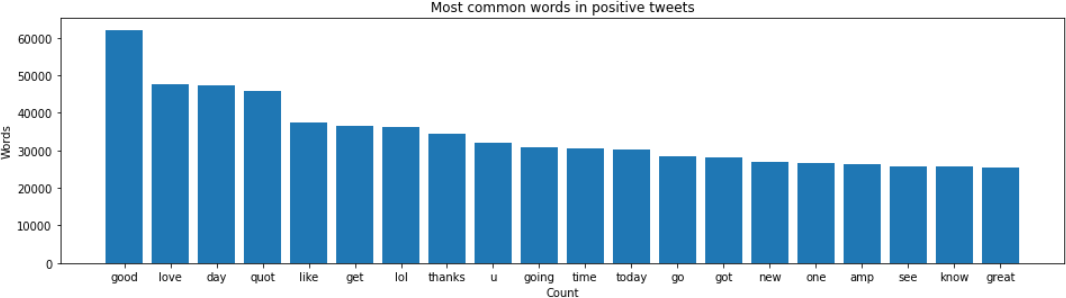


*Figure 11. Max, min, mean and standard deviation of each tweet in terms of number of words.*



*Figure 11. Most common words in our dataset.*

# Positive Tweets

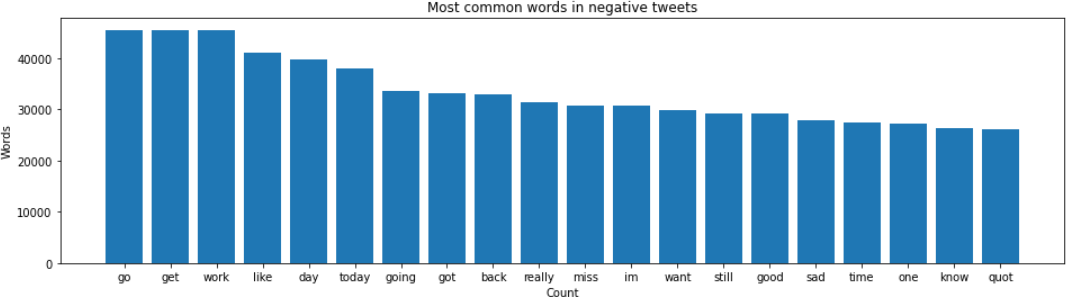


*Figure 12. Most common words in positive tweets in our dataset.*



*Figure 13. Word cloud of positive tweets.*

# Negative Tweets



*Figure 14. Most common words in negative tweets in our dataset.*



*Figure 15. Word cloud of positive tweets.*

# GloVe: Global Vectors for Word Representation [2]

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.

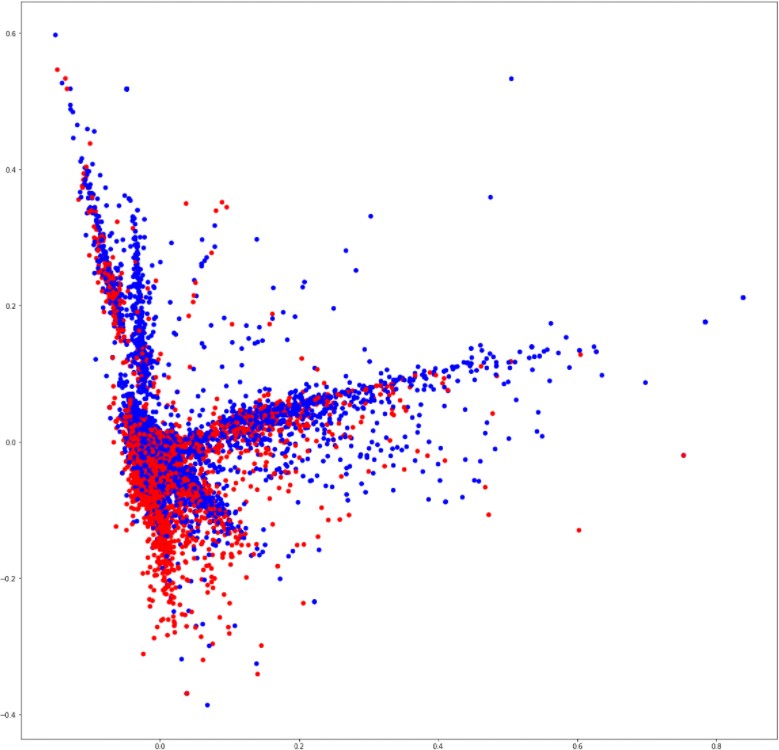
The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for eﬃciently learning word vectors, developed by Pennington, et al. at Stanford. It is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated 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.

# 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.*

# Results

In this review, 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.