CPSC-6300-Applied Data Science Spring 2020

**Twitter Sentiment Analysis**

Abhinav

Shaurya Gupta

Srihith Alwala

horizontal line

# 

# Introduction

The idea is to extract tweets from twitter and run different Machine Learning models on it to predict the sentiment of the people regarding the upcoming election in the US. It is not a hidden fact that in the elections, social media platforms play a very important role. Social media has opened a whole new world for people around the globe. People are just a click away from getting a huge chunk of information. With information comes people’s opinion and with this comes the positive and negative outlook of people regarding a topic. Sometimes this also results in bullying and passing on hate comments about someone or something.

**Motivation**:

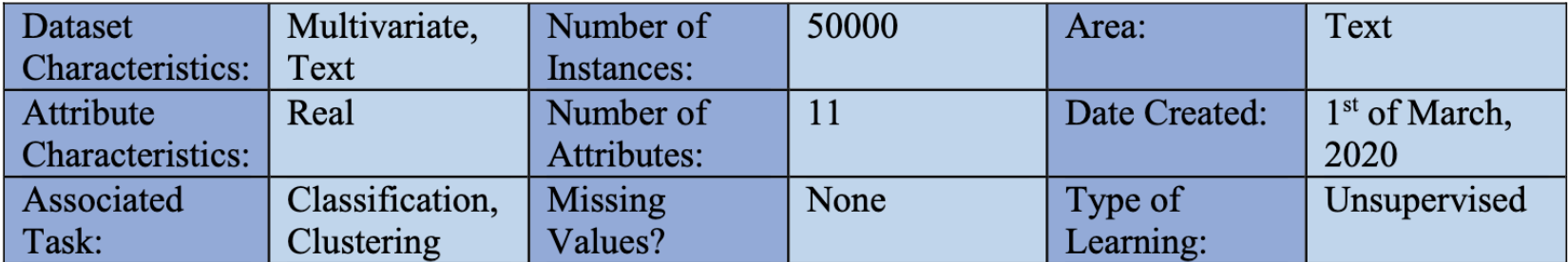
In the last US election social media platforms like Twitter and Facebook were used to understand the general inclination of the voters. To understand this further, it is important to understand how it works and what are the factors that determine which way is the majority’s inclination. The findings will also suggest what we can expect from the social media platform in the coming days. Will it help the citizens to be more aware? Or, will it help the users from differentiating between facts and rumors? The outcome of this project is to predict the sentiments (Positive, Negative and Neutral) of the people from their tweets.

**Source of data:**

The data is not available directly, we extracted data from Twitter API Tweepy. The dataset has 50000 instances and 11 attributes and more information regarding the dataset is provided in the summary of EDA.

# Summary of EDA

Tweet Dataset

****

Dataset Information: The dataset was collected using the Twitter API called Tweepy. Tweepy API provides access to the entire RESTful API methods. Methods accept various parameters and return responses. To collect the records, the tweets were streamed and stored in a json format first and then converted to the csv format. It has 11 attributes and more than 50000 records collected over three days’ period.

Attribute Information: The attributes collected here are the information about the tweets like, from where it was tweeted (location), who tweeted it (id), when this was tweeted (created\_at), how many followers does the person doing tweet have (followers\_count) etc. These attributes are considered to be parameters that will help in determining the sentiments of the tweets.

Time Period Covered: As the Tweepy API helps in extracting the tweets of the time when it is running, we have collected the recent tweets over the three days’ period, that was between 27th Feb, 2020 and 29th Feb, 2020.

Data Cleaning: There were two different stages in which data cleaning process were done. The two processes are listed below:

* *During Streaming:* This was the time when with the use of Tweepy API we listened to the tweets and started collecting the tweet’s information. Every tweet had over 25 attributes and not all of them were needed for the data analysis. We captured those features and attributes that were necessary to evaluate the sentiments of the tweets and discarded the rest of them. This is one of the dimensionality reduction techniques where unnecessary attributes were removed. To tackle redundancy, we removed the retweets from consideration and so set the retweeted status to False.
* *After Streaming:* The first thing to be taken care of, was the missing values. Sometimes there are some missing values that can result in bad analysis, so removal of those missing values was a good step. We could have replaced the value with the mean or median of the rest of the tuples but it doesn’t seem right because it is hard to judge something like sentiments by just some random numbers. The next step of cleaning was to remove all the stop words from the tweets. Stopwords are useless in such analysis and will only add to the computation and complexity of the task. This will help us further in just focusing on the words that actually add meaning to our analysis. This way we could focus more on the adjectives that were used which is considered to be an important factor in analyzing the sentiments of any tweet.

**Different Types of Plots for Visualization:**

Histogram Plot:This plot could be used to find the number of tweets that were made between different groups.

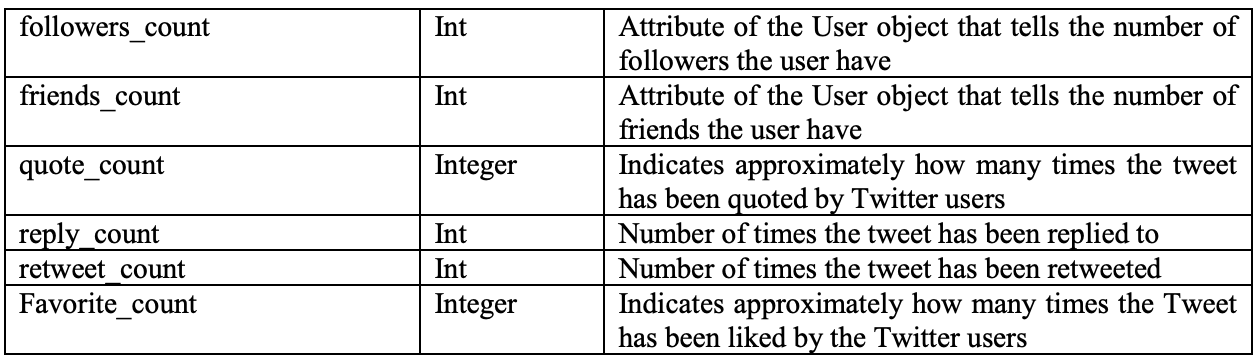
Bar Charts: It can be used to categorize the tweets based on the locations of the tweets. Like the number of tweets that came from New York.

Pie Chart: Percentage of tweets that came from a certain zone. Suppose if we categorize the locations around the US in 5 zones. Then the percentage of tweets from each zone could be projected in the pie chart.

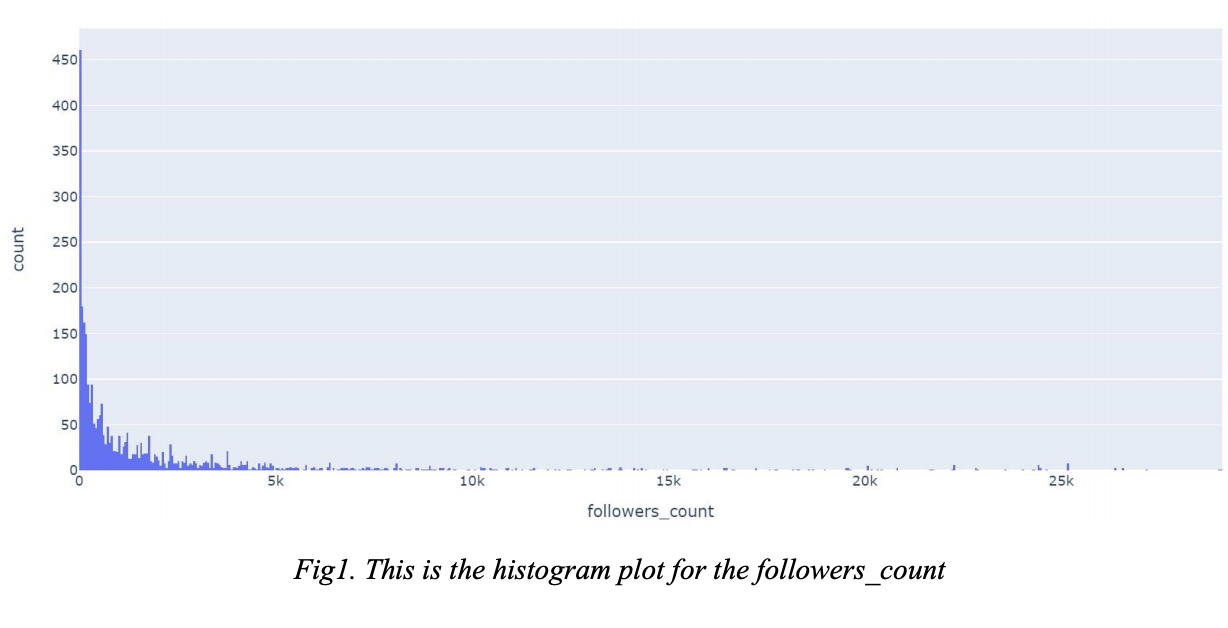
Scatter Plot: Scatter plot can be used to visualize the relation between two attributes to decide whether there is any correlation between any attribute or not in order to counter any multi-linearity problem.

Here are some of the plots that were visualized and analyzed by us with each briefly described below:

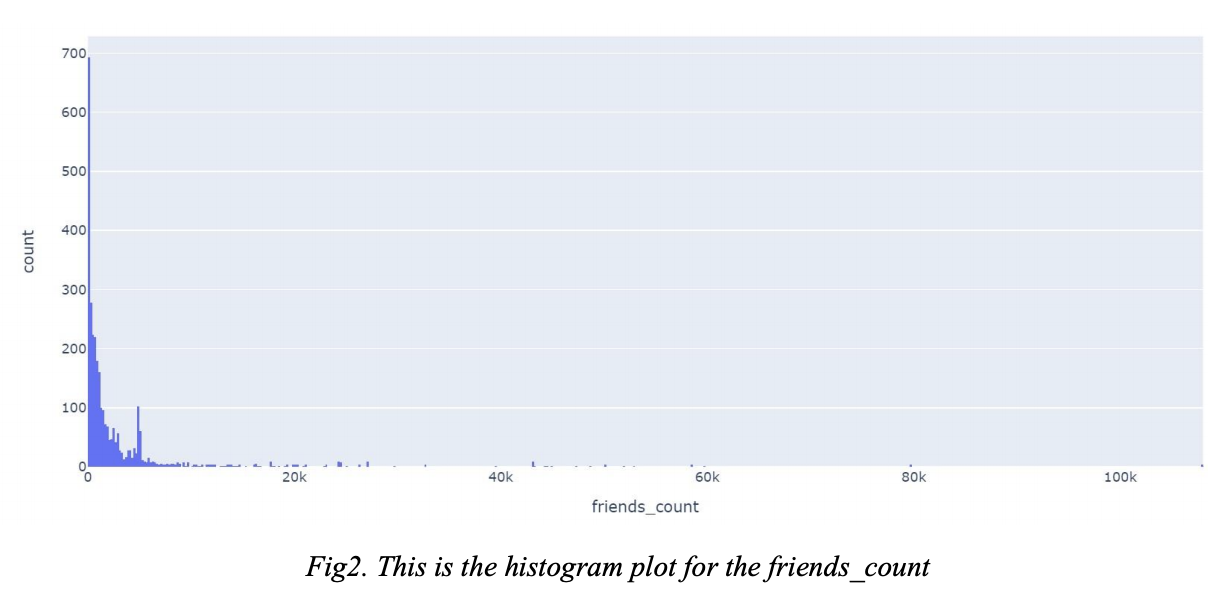
**Key Predictors:**

****

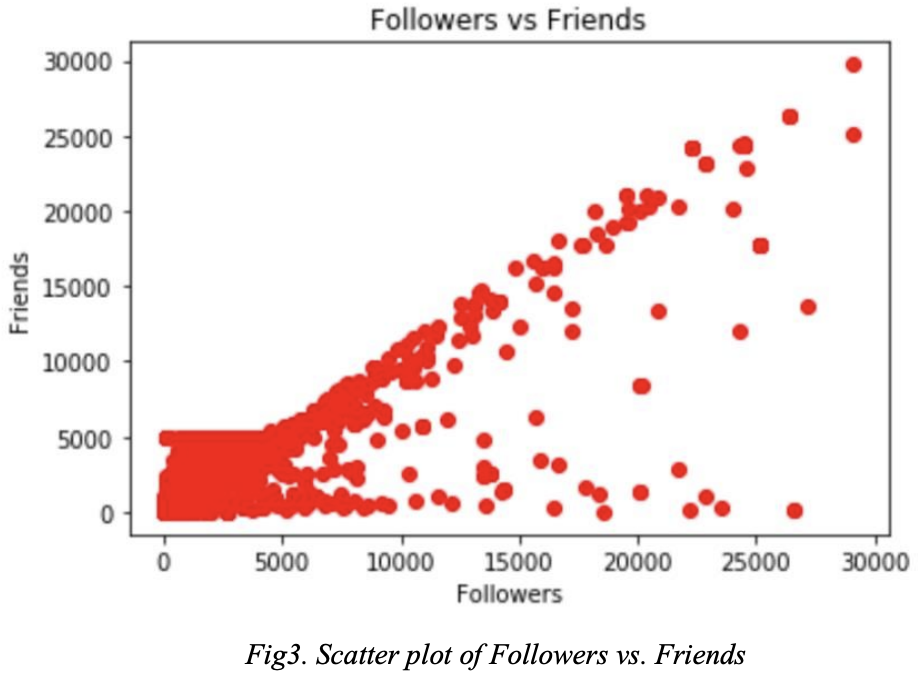
**Graphs Plotting:**



**Fig 1.** *This is the plot for the number of tweets that were tweeted by the people that falls under a certain range of followers. From this we concluded that individuals having followers count between 0-49 made the highest number of tweets which was 460. This information can be further used in determining whether there is any role of followers\_count on the sentiment analysis.*



**Fig 2.** *This is the plot for the number of tweets that were tweeted by the people that falls under a certain range of followers. From this we concluded that individuals having followers count between 0-199 made the highest number of tweets which was 692. This information can be further used in determining whether there is any role of friends\_count on the sentiment analysis.*



**Fig 3.** *This is the plot between the number of followers and number of friends of the individual tweeting. First of all, looking at it there does seems to have some relation between the two, but cannot say it for sure as looking at the lower left section we can says something else might be going and to understand that we need to do a plotting for those having followers count less than 5000, and the same goes for the few data points that are on the lower right section of the graph. This needs a further analysis.*

# Summary of Machine Learning Model

More than 30000 tweets were extracted and used for training the models. The training was however done in two steps: 1) To find the sentiments using the nltk and tweepy API to find the sentiments of every tweet, and 2) To use these sentiments to train the model and then do an analysis of the model. This is the most basic approach for now but in the future will be using some more complex models like Decision Tree and Neural Network. In the following sections we have discussed the experiment, setup and result for our prediction.

**Experiment:**

We used the Tweepy API to stream the tweets from the twitter stored it as a csv file with different features or information about the tweets such as the date created, location, retweet count, favorite count etc. Our experiment objective is to develop different models that can be used to predict the upcoming tweets with the same information used to train the model. To achieve this objective, we first use the sentiment score from the TextBlob API/NLTK API to find the sentiments of the tweets. Then used a Linear model to train the model on it, predicted the outcome and accuracy for the testing data.

**Setup:**

*TextBlob Library:*

It is a python library for processing text data. It provides consistent solutions into common Natural Language Processing (NLP) tasks, here we used it for sentiment analysis. This sentiment module uses PatternAnalyzer and NaiveBayesAnalyzer of TextBlob to determine the polarity and subjectivity of the text. If the text is long, then it averages out the polarity of the whole text. It is fast and accurate as it uses statistical approaches and regular expressions.

*NLTK Library:*

NLTK is a python library that is used to work with human language data. This library uses VADER (Valence Aware Dictionary and Sentiment Reasoner) Sentiment Analysis, which is a rule based sentiment analysis tool that is specifically attuned to sentiments expressed in a text. Calculation of these scores are based on the lexicon metrics which is further normalized between -1 and 1, -1 is considered to be extremely negative and 1 is considered to be extremely positive.

If the scores are above 0.5 then we have considered it to be positive, if the scores are below -0.5 then we have considered this to be negative and for the rest of the cases it is considered to be neutral. Using this function, we have created a new column in our dataset that stores the sentiments as ‘positive’, ‘negative, and ‘neutral’.

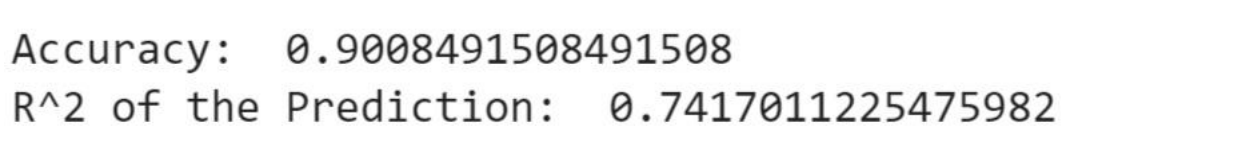
***Note:*** *We have used two different libraries to find the sentiments, we will show the comparison between the two and will be using only one for training the model.*

*Positive, negative and neutral words count:*

Using the NLTK library while determining the sentiments we also stored the number of positive, negative and neutral counts for every tweet. The purpose for all this was to create additional data that could be used further while training any model, as the direct information of ‘positive’, ‘negative’ and ‘neutral’ were not enough for determining the models.

*Model Training:*

We used Multiple Linear Regression models to train our dataset. Our target was to predict the sentiments to be ‘positive’, ’negative’ and ‘neutral’. The features used for predicting our target other than the positive, negative and neutral word counts were ‘reply\_count’, ‘retweet\_count’ and ‘favorite\_count’. The result of the model is summarized below.



The predicted result was then converted into the 1 and 0 format. This transformation was conducted by choosing the maximum value or score (predicted by the model) among the three sentiments and assigning it a score of 1 and others as 0. This transformation was important as our training model was predicting a category and which was converted as dummy values. In order to have the accuracy the predicted values should also be in the same format, but instead it was a float value.

**Results:**

Figure 1*:* *This figure shows us that the predicted sentiments by TextBlob were almost all Neutral. I have used almost as the result considering only 3 tweets to be positive and 3 tweets to be negative and the rest of the 20012 tweets to be neutral. While plotting it was very difficult to show the positive and negative as it was comparatively very small.*

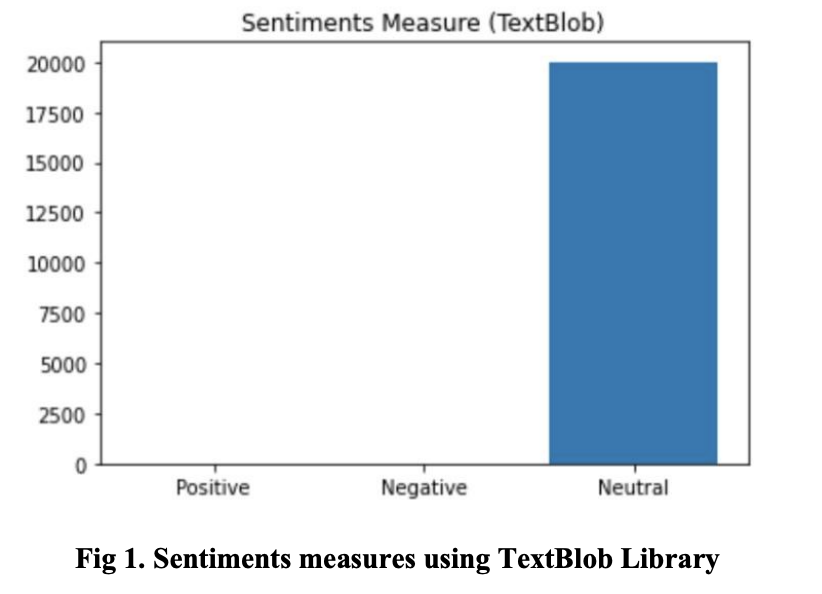


Figure 2: *This figure has in my opinion a better result than what TextBlob. The way TextBlob works it will always come up with a smaller score for any given tweets. There was no preprocessing done like removal of stopwords and tokenizing the text. NLTK on the other hand has a big repository of words and symbols and each of these words and symbols have their own scores.*

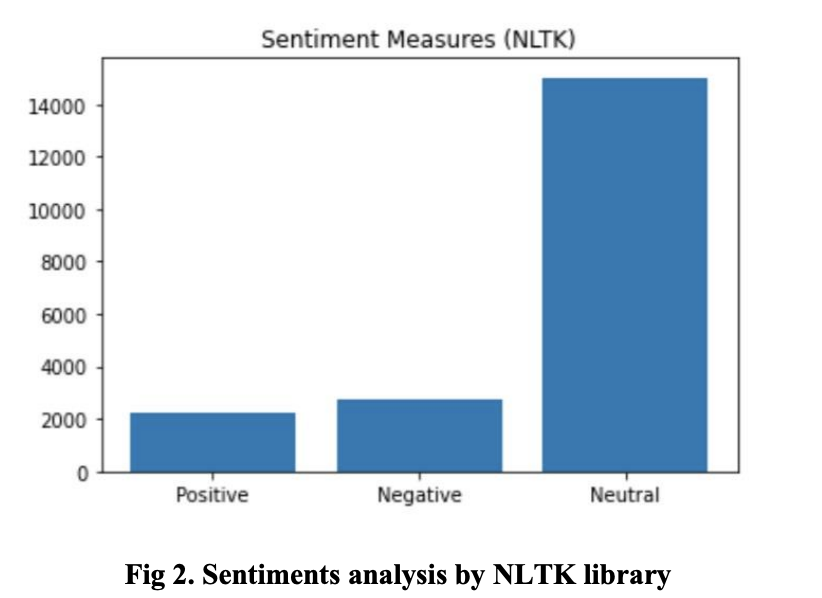
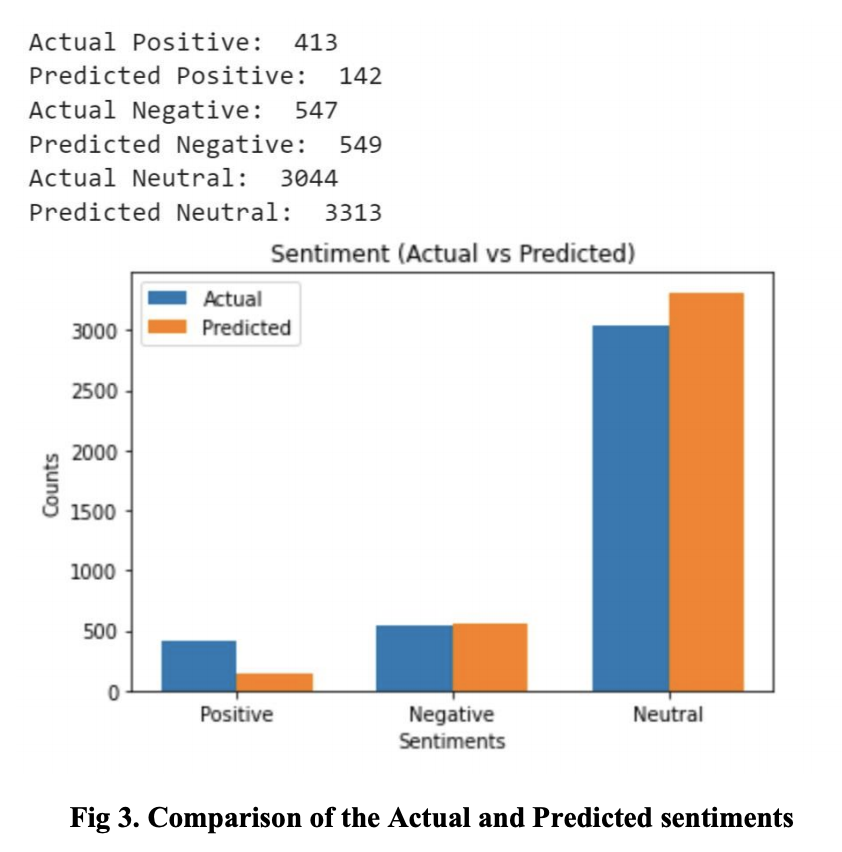


Figure 3: *This figure just shows the graph of actual and predicted sentiments of the test data. This graph shows that our model was nearly accurate to predict our data. It almost predicted negative sentiment correctly. From here we can say that the model was okay as it had the accuracy of 90 percent, and some changes like forward selection or backward selection can be done to further check the accuracy of the linear model.*



**Limitations:**

There are however still few limitations with the tweet data, these limitations were not attended in this part of the project and would try to find a way to overcome at least few of them. The limitations are as follows:

***1) Sarcasm:*** The libraries used in this part of the project have no way of checking the sarcasm. In fact, identifying sarcasm is one of the most difficult tasks till date as it depends a lot on the tone of the speaker and could only be identified if a human is reading it.

***2) Short-cuts:*** As twitter allows only 140 characters, people have a tendency to use shortcuts like instead of writing ‘I don’t know’ they will write ‘IDK’. Words and characters like this are hard to be identified by the ongoing libraries and so there is this shortcoming where a lot of information is still not identified.

***3) Emoji’s:*** We haven’t considered the emoji’s for now, but will surely do it in the future.

# Summary and Conclusion

**Going back to the question that has motivated your project, how would you answer that question given the results of your analysis?**

Considering the above results, the model was 90% successful in predicting the sentiments of the tweets. One of the questions we tried to answer is to predict the sentiment of the tweets which we were successful in. Based on this we can find out how people are inclined towards a candidate who is participating in the election.

**Think about domain experts in the field you have analyzed. What can they learn from your project? How could the results of your analysis inform their work?**

This project gives a fair outline of predicting people’s opinion on the candidates contesting for elections. The domain experts in the field of politics would have great knowledge and insight in this area. They too would be having data and would be making some predictions on who is more likely to win the elections. Our project provides an additional reference to these experts and sort of a comparison between their predictions and our machine learning model predictions. The results of our analysis would give them more insight on how people are responding in social media. Because, the people who share their opinion on social media are very large compared to the surveys conducted in person.

**Identify one way that your project could be improved if you had more time and resources to work on this project. For example, what additional data would you gather? What alternative data cleaning decisions would you make? What additional models would you estimate?**

If we had more time and resources, we could have collected even larger data over a span of seven days at least, which would increase the accuracy and also use other softwares like Apache Spark, Kafka, Elasticsearch in improving the model. We would have added functionality to handle emoticons and emoji recognition. We could have also worked in the area where identifying sarcasm is still a challenge.

# References

* <https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-inpython-f9e6ec6fc52f>
* <https://textblob.readthedocs.io/en/dev/quickstart.html>