# US Airline Twitter Sentiment Classification and Analysis

CS: 4440 Web Mining Project

Changze Han
Computer Science Department
University of Iowa
Iowa City, IA, USA
changze-han@uiowa.edu

Fred Morsy
Computer Science Department
University of Iowa
Iowa City, IA, USA
fred-morsy@uiowa.edu

Maaz Musa
Computer Science Department
University of Iowa
Iowa City, IA, USA
maazbin-musa@uiowa.edu

## ABSTRACT

This paper presents a semester-long team project for the CS:4440 Web Mining course. The theme of this project is US Airline Twitter Sentiment. The study methods and techniques we adapted were inspired by the great results of previous research. For example, we imitated the process of classifying sentiments using the famous Naïve Bayes Classifier which is an extension of the Bayes' theorem. Detailed procedures and workflows are introduced in the paper. We also provided various data visualizations for different statistical analyses to illustrate the underlying patterns. In addition, all source code and project data can be found within each page's footnote. The purpose of this project is to demonstrate our understanding of Web Mining.

## KEYWORDS

Twitter, Sentiment Classification, Bayes' Theorem, Data Visualization

## 1 Introduction

Sentiment analysis is a popular text analysis technique, especially for businesses to analyze their customer's reactions towards their products or services [1]. The rise of social media has enabled businesses to harvest large amounts of data through the internet. This data may be analyzed using sentiment analysis tools to reflect the real reactions of their customers [2]. Our project aims to investigate the sentiment with regards to what people are tweeting about US airlines. Using statistical methods and tools, we would like to draw meaning from data that helps us better understand the public's perception of US airlines, giving us a better understanding of how businesses can make improvements and better accommodate their customers. In this project, we used two real world data sets. The first is a dataset of airline-related tweets collected from Kaggle[1]. The data was already parsed into a nice format for analysis in which the tweets were already processed with their corresponding sentiments. This format is not only useful for parsing the metadata for a user profile, but also provides labeled data that can be used to test our classifiers. The second dataset[2] was obtained from Prof. Srinivasan. It is a CSV file containing raw tweets from Twitter in which we later parsed into its relevant fields to be appended to the original Kaggle data. We explored two tools used for sentiment analysis being VADER and Naïve Bayes Classifier. We used both tools to predict airline sentiment and compared features of each model that can explain the differences in model performance and accuracy. After reviewing some research and works related to sentiment analysis, we found that using a Naive Bayes Classifier within the Natural Language Toolkit (NLTK) was commonly used as it typically yields high accuracy for predicting sentiment. Once we have the sentiment labeled for all tweets, we used an aspect extraction and comparison technique to extract and assign negative reasons to our tweet data as well. This provides a complete view with respect to sentiments and the underlying reasons for negative sentiments. At the end, we present our data through various visualizations in order to draw meaningful conclusions.

## 2 Related Research

There are several choices when it comes to the text classification algorithms, for instance, Naïve Bayes Classifier (NBC), Logic regression, Support vector machine, Neural Networks, etc. [3]. In this paper, we decided to use NBC because of its simplicity and high accuracy [4]. In addition, we think it is relatively easy to implement than other machine learning models for beginners. We also did a brief literature review on the aspect extraction which we applied at the end of the project to extract negative reasons.

### 2.1 Naïve Bayes Classifier and Bayes' Theorem

The NBC is a technique based on Bayes' Theorem which is specially used for calculating conditional probabilities [7]. According to the definition conditional probability, "the probability of a hypothesis conditional on a given body of data is the ratio of the unconditional probability of the conjunction of the hypothesis with the data to the unconditional probability of the data alone". Thomas Bayes unveiled the significance of conditional probability in his famous work "An Essay Toward Solving a Problem in the Doctrine of Chances". The specialty of

[1] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[2] https://www.dropbox.com/sh/ahdnsx1sj7a9ulw/AAAb6vj-80R1xKFeTKpNktnia?dl=0

NBC is it assumes that the features are independent of given class [5]. And because of this feature, it can simplify the learning.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \qquad [5]$$

- P(c|x) stands for the posterior probability. It is the probability of 'c' being Ture given that 'x' is True.
- P(x|c) stands for the likelihood. It is the probability of 'x' being True given that 'c' is True.
- P(c) stands for the prior probability. It is the probability of 'c' being True.
- P(x) stands for the evidence probability. It is the probability of x being True.

In NBC, P(x) is considered as constant, so it is ignored in a way. What we actually calculate would just be:

$$P(c|x) = P(x|c)P(c)$$

In the context of tweet texts, since the chance of a tweet fall into a specific class is really low, we would have to tokenize the tweet in order to calculate the probability for each word during the training [6]. In this case, the formula we are adapting would become:

$$P(c|X) = P(x1|c)P(x2|c) \dots P(xn|c)P(c) \qquad [6]$$

- X stands for one tweet.
- x1,x2,…xn stand for each work in tweet X

This is how the NBC simplifies the learning. NBC also has its disadvantages. One of the problems is "Zero Conditional Probability Problem" which means it could wipe out all the information in other probabilities too [4]. In addition, the assumption of independence class features of NBC is very strong [4]. However, we think these flaws would not affect the result and performance in our project at all.

## 2.1 Aspect Extraction

Aspect extraction is a fairly new and non-trivial technique. Previous work has looked into aspect extraction in two ways. First, the website MonkeyLearn is famous for its text analysis. It has a tool[3] that allows you to divide a sentence into "units." Each unit is the smallest sentence that can be formed. This will contain only aspect and one sentiment. Using this tool was not feasible because it was not free. After that, we investigated the second technique[4] which mentions a generic way that can be used based on intuition, which allowed us to extract aspects from a sentence. In this we use the knowledge that nouns are usually the aspects of a sentence and word vector algebra to deduce the aspect of a sentence which is closest to the known negative aspects.

# 3 Project Procedure and Analysis

In this section, we cover the entire project workflow and source code we implemented. We divided the procedure into three general steps which are data collection, sentiment classification, and assigning aspects.

## 3.1 Data Collection

We constructed our data set using two real world datasets. First we used a Twitter US Airline Sentiment dataset from Kaggle. This data provided useful metadata for user profiles used to predict airline sentiment and perform analysis such as a user's username (name), their tweets related to US Airlines (text), the airline mentioned in their tweet (airline) and the corresponding sentiment (airline_sentiment). The sentiments for each tweet were labeled within the data so that we could try multiple classification models to try to predict the sentiment of newly scraped tweets, and to compare our models results against the original labels. The second dataset we used was one we obtained from Prof. Srinivasan. This was a CSV file from a Twitter API containing about 8,417,915 raw tweets. The datasets for both files were already formatted as CSV files so we simply needed to parse the raw tweets into their relevant fields to be appended to the original dataset. A python program[5] was written to extract relevant information from the raw tweets using regular expressions which included using the airlines mentioned in the original dataset. When parsing the raw tweets, we only keep columns from both datasets that were necessary for our analysis. In total we extracted 3,043 airline-related tweets[6] by searching keywords. Since our program also extracted tweets not relevant to airlines, it is possible that some irrelevant tweets are included in the final results because they accidentally contain the same keywords. After visually examining the results, we concluded that the percentage error was acceptable because there were approximately 3 to 4 irrelevant tweets among every 100 tweets. After extracting the raw tweets related to airlines, we formatted the data to match the header of the original Kaggle data's header. This was so that we could easily merge the two files for a complete dataset. The original Kaggle data contained 14,641 tweets and our scraped data contained 3,043 tweets giving us a combined total of 17,684 tweets[7].

## 3.2 Sentiment Classification

### 3.2.1 Vader Classification

At first, we tried out a sentiment analysis tool called VADER (Valence Aware Dictionary and sEntiment Reasoner) in Python. VADER is a comprehensive sentiment analysis tool used for analyzing sentiment on social media [8]. It is commonly used for analyzing text such as tweets on Twitter, news comments, Facebook comments and blog comments. VADER analyzes a column of text in a CSV file and assigns compound scores to each

[3] https://monkeylearn.com/blog/aspect-based-sentiment-analysis/
[4] https://medium.com/@Intellica.AI/aspect-based-sentiment-analysis-everything-you-wanted-to-know-1be41572e238

[5] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Naive%20Bayes%20Classifier/parse_new_tweets.py
[6] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Naive%20Bayes%20Classifier/parsed_tweet_0.csv
[7] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Vader%20Classifier%20and%20Data%20Visualization%20---%20Fred/finalOutput.csv

tweet. A compound score is computed by summing the valence scores of each word in the lexicon, adjusting to the rules and then normalizing them to be between -1 (extremely negative) and +1 (extremely positive). Using a threshold, these scores are then used to determine whether a sentence carries positive, negative, or neutral sentiment.

Positive >= 0.05
Neutral between -0.05 and 0.05
Negative <= -0.05

The VADER score allows us to not only classify the sentiment of a particular tweet, but also tells us the intensity of the sentiment [8]. For example, "okay" and "excellent" would be classified the same using polarity-based classification. However, the VADER score would identify "excellent" as carrying more positive sentiment than "okay" [9]. Although VADER gives a low classification accuracy, it allows us to also measure the intensity of the sentiment. Also, VADER does not require any training data since it is a lexicon and rule-based tool making it easier to use. However, the prediction accuracy for this model was about 58%. Since we didn't get a satisfying prediction accuracy, we considered using a more common classification tool that could give us better accuracy. We gained a sense of how sentiment analysis works through this short experiment using VADER.

### 3.2.1 Naïve Bayes Classification

There are available tools[8] online that can perform the Bayes classification for text analysis. However, those tools are not flexible enough and our dataset requires a customized program so that we can manipulate the variables within it. In addition, we wanted to apply some course knowledge in this section such as generating a confusion matrix to evaluate the classifier or computing precision and recall. So, we decided to start from scratch and write a python program[9] using the NLTK package to imitate the entire NBC process. With the guidance of an online tutorial[10], we divided the entire process into five steps.

1. Kaggle dataset came with classified sentiments which makes it a perfect training dataset candidate. First we separated it into two datasets: positive and negative. The positive part contains 2363 tweets[11] and the negative part contains 9178 tweets[12]. We also inspected the sentiment classification by manually reading through the corresponding tweet text and we noticed that the accuracy of Kaggle dataset was not 100%. So, we are prepared to have a mediocre accuracy on our NBC due to the errors in the training dataset.
2. After preparing positive and negative dataset separately, we use NLTK functions to clean up the tweet texts through 6 steps:
    a. Extract texts from csv file
    b. Tokenize texts
    c. Add tags for all tokens
    d. Lemmatize tokens
    e. Remove stop words and other useless stuff
    f. Convert to dictionary for training in the NBC
3. After generated clean tokens, we combined the positive and negative tokens together and did a random shuffle. Then we divided the entire token set into a training set and a testing set according to a 7:3 ratio. (7 is the training data) And we got 8079 tweets in the training set and 3462 tweets in the testing set.
4. Then we train the NBC with the training dataset. After running the program over 50 times, we scored a 92.1% accuracy as our best result. After reading through different literatures, we were under the impression that we could achieve at least 95% accuracy. We think there are 3 possible reasons:
    a. The sentiment classification in the training dataset was not accurate enough.
    b. The text in the training dataset was not clean enough. Some tweets were not related to the airline at all.
    c. We didn't clean up the texts enough. There might still be disrupting tokens.
5. The last step is deploying the trained NBC on our new dataset which was extracted from 8.4 million raw tweets. The way we did filtering is by keyword searching. We wrote a program in python and ran a regular expression search on different airline names with a prefix of '@', '#' as we thought that an individual airline name itself could appear in many different contexts which are irrelevant to airline. "United" is a great example that could be tweeted in various topics. However, "#United" or "@United" would narrow down the search a lot which increases the chance of it being United Airline. The classification result of the new tweet can be found in the footnote link and it was predicted by a 92.1% accuracy trail.

We also dived into the evaluating process of the NBC. The NLTK package came with a function which can display more informative features:

| | |
|---|---|
| amazing = True | Positi : Negati = 53.7 : 1.0 |
| :-) = True | Positi : Negati = 40.2 : 1.0 |

These data shows the likelihood ratios. For example, the first row should be interpreted as word "amazing" is 53.7 times more likely to occur in a positive tweet than in a negative tweet. Likelihood gives us one perspective on how the model performs. In addition, we also generated a confusion matrix and visualized it in python:

[8] https://monkeylearn.com/text-classification-naive-bayes/
[9] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Naive%20Bayes%20Classifier/naive_bayes_classifier.py
[10] https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk
[11] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Naive%20Bayes%20Classifier/positive_tweets.csv
[12] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Naive%20Bayes%20Classifier/negative_tweets.csv

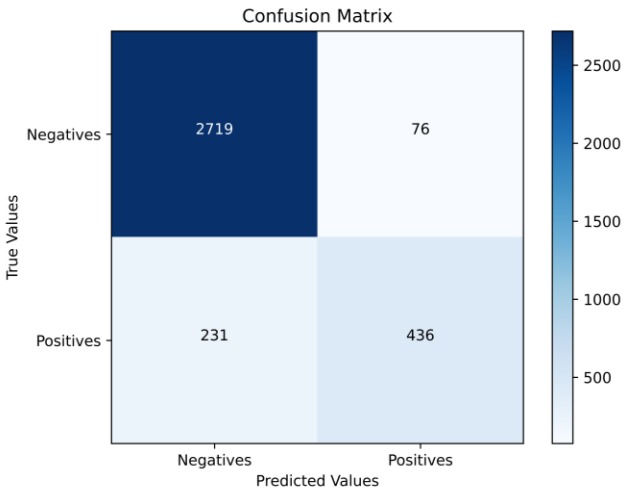


**Figure 1: Confusion matrix for a Naïve Bayes Classifier with 91.13% accuracy**

From the confusion, we can get some insights on how the model was performing. There are 2719 negative predictions when the actual result is negative. And there are 436 positive predictions when the actual result is positive. From this result, we can understand one hidden reason that makes our accuracy below 95%. The Kaggle dataset contains too few positive tweets, so the random shuffle did not work well on distributing the tweets. The NBC did not get enough learning on the positive tweets. Other than that, we also calculated the precision, recall and f score based on this confusion matrix for negative and positive separately. Take the data in Figure as an example to show the calculation:

$$Positive\ Precision = \frac{436}{436 + 76} = 0.851$$

$$Positive\ Recall = \frac{436}{436 + 231} = 0.653$$

$$Positive\ F\ measure = \frac{1}{\frac{0.5}{0.851} + \frac{0.5}{0.653}} = 0.739$$

And by the same calculation for negative, we have 0.921 for negative precision, 0.972 for negative recall and 0.946 for negative F measure. Now we can evaluate the performance of our NBC model more closely than just having the accuracy score. And we can see that the prediction on negative tweets is a lot more accurate than the prediction on positive tweets. (In addition, when we calculated the F measure, we used a default 0.5 as alpha)

## 3.3 Assigning Aspects

One of the problems we faced was the missing 'negative reason' in the Twitter dataset that we collected. Without knowing the reason for the sentiment the analysis felt incomplete. For this reason we implemented an approach to extract the reason for the sentiment as well. As described in the related work section, there are two ways we could achieve this. Given the clear advantages of affordability we choose the second option described in the related work. This technique first downloads pre trained word vectors from google which have about 100 billion phrases and words. Word vectors as we all know are numeric vector representations of words and we can perform vector algebra on them to uncover relations such as

"Man - Woman = King - Queen".

Next we made a list of negative reasons from our already labeled kaggle dataset and found word vectors for each word from our word vector list. The following is the list of negative reasons we gathered

['Bad Flight', 'Late Flight', 'Customer Service Issue', 'Booking Problems', 'Lost Luggage', 'Flight Attendant Complaints', 'Cancelled Flight', 'Damaged Luggage', 'longlines']

Our next step involved processing each tweet and assigning one aspect from our list to it. To do that we used a POS tagger from nltk to extract nouns from a sentence and looked for word vectors for those nouns in our Google word vector list. After retrieving that we took a cosine similarity between that vector and each vector of our negative reason from kaggle. The highest similarity reason is assigned to this tweet. The program can be found in footnote[13].

## 4 Data Visualizations

1. We counted the occurrence of each airline company for all tweets and reported percentages using a pie chart. Here we see that United, American, Delta, US Airways and Southwest are the most frequently tweeted about. However, Virgin America and JetBlue are not tweeted about as frequently. This could mean that Virgin America and JetBlue are less popular among the other US airlines.

```
Number of Tweets Per Airline (airline)
United           4648
American         3450
Delta            3052
US Airways       2913
Southwest        2814
Virgin America    568
JetBlue           238
Name: airline, dtype: int64
```

**Figure 2.1: Frequency of Tweets Per Airline (Data Frame)**

[13] https://github.com/MrColinHan/Twitter-US-Airline-Sentiment/blob/master/Vader%20Classifier%20and%20Data%20Visualization%20---%20Fred/airlineTweets.py

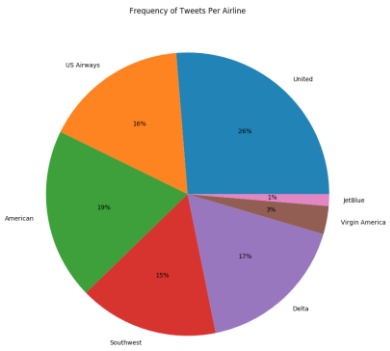


**Figure 2.2: Frequency of Tweets Per Airline (Pie Chart)**

2. We counted the total occurrence of each airline sentiment and visualized the data on a bar chart. The number of negative tweets with regards to US airlines is about twice as much as both positive and neutral tweets. This implies that the majority of tweets carry negative sentiment.

```
Total Number of Negative, Neutral and Positive Tweets
negative    11503
neutral      3099
positive     3081
Name: airline_sentiment, dtype: int64
```

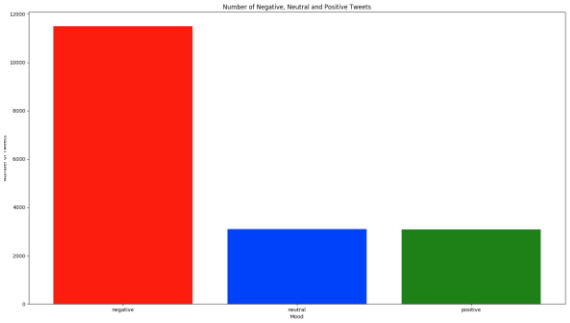


**Figure 3: Number of Negative, Neutral and Positive Tweets (Data Frame and Bar Graph)**

3. We counted the occurrence of each negative reason and then presented the top ten reasons in descending order on a bar chart. Just by looking at the bar graph we can see that flight delays are the leading reason by a long shot (excluding the Can't Decide). This makes intuitive sense as in most of our experiences 'Late Flight' is our most common problem. The reason I exclude 'Can't Decide' in my verbal analysis is because these are probably the positive tweets that we were not able to mark.

```
Top Ten Reasons for Poor Service (negativereason)
late_flight                  5830
Cant Decide                  3324
flight_attendant_complaints  2230
customer_service_issue       1902
cancel_flight                1188
bad_flight                    884
lost_luggage                  838
booking_problems              754
damage_luggage                501
longlines                     232
Name: negativereason, dtype: int64
```

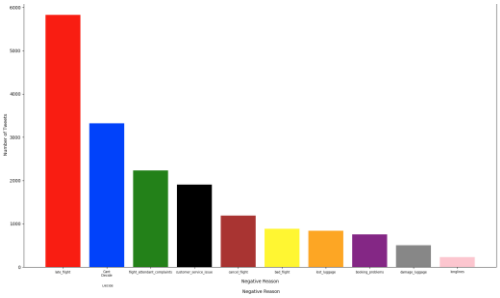


**Figure 4: Top Ten Reasons for Poor Service (Data Frame and Bar Graph)**

4. We ranked the airlines according to the amount of negative reasons and presented the rank on a pie chart. We see that United, American, US Airways, Delta and Southwest have the highest number of negative reasons associated with their tweets. This makes sense as this matches the number of tweets per airline and that most of the tweets carry negative sentiment.

```
Worst Airlines by Number of Negative Reasons (negativereason)
airline
United           3301
American         2518
US Airways       2263
Delta            1588
Southwest        1440
Virgin America    230
JetBlue           163
Name: negativereason, dtype: int64
```

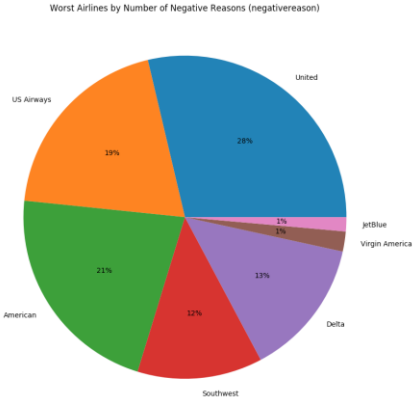


**Figure 5: Worst Airlines by Number of Negative Reasons (Data Frame and Pie Chart)**

5. We calculated the frequency distribution of sentiments within each airline and then visualized the data on a cross tabulation bar chart. Generally speaking, the majority of US airline tweets carry more negative sentiment than both positive and neutral sentiment regardless of airline. About 70% or more of all the tweets with regards to American, JetBlue, US Airways and United carry negative sentiment whereas only about 40% of tweets with regards to Virgin American carry negative sentiment. Although the majority of all the tweets carry negative sentiment for each airline we see that Virgin America has the lowest

proportion of negative tweets which may imply a higher quality of service vs the other US airlines.

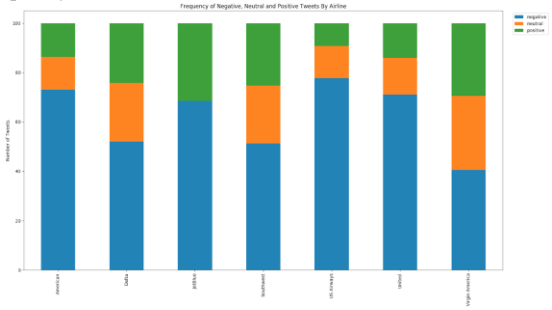

**Figure 6: Frequency of Negative, Neutral and Positive Tweets By Airline (Cross Tabulation)**

6. We calculated the frequency distribution of different negative reasons within each airline and then visualized the data on a cross tabulation bar chart. We can see that there is a good uniform distribution of negative reasons across all airlines which is to say that all of them have the same service overall. However there is one exception and that is flight attendant complaints. Delta, JetBlue, US Airways seem to have several fold more flight attendant complaints than the other 4. This can indicate that if you have an issue with flight attendant services these are the airlines to avoid.

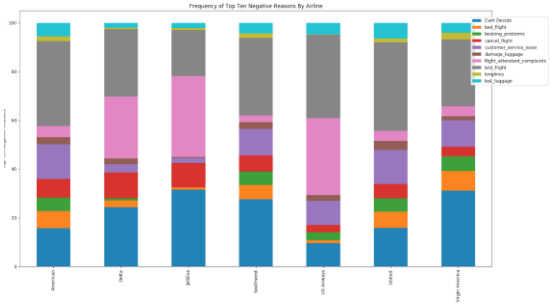

**Figure 7: Frequency of Top Ten Negative Reasons By Airline (Cross Tabulation)**

# 5 Signatures and Contributions

## 5.1 Signatures

*Changze Han, Fred Morsy, Maaz Bin Musa*

## 5.2 Contributions

1. **Changze Han: (50% Contribution)**
    a. Develop project strategy
    b. Literature Review and finding Citations
    c. Extract airline-related tweets out of 8.4 million raw tweets. (Python program)
    d. Cleaning up tweet texts using NLTK
    e. Perform Naïve Bayes Classification including training model, deploying model. (Python program)
    f. Evaluating NBC performance.
    g. Drafting Paper (60%)
    h. Format entire paper into ACM format and adding footnotes.
2. **Fred Morsy:**
    a. All data Visualizations section: overall data analysis on both Kaggle data and new data
    b. Present project data through different Visualizations. (Python program)
    c. Helped with data extraction
    d. Draw meaningful conclusions from sentiment results
    e. Build VADER Classifier and analysis. (Python program)
    f. Drafting paper
3. **Maaz Bin Musa:**
    a. Extract negative reasons from Kaggle data and convert them into vectors
    b. Extract negative reasons for each tweet in our new data
    c. Draw meaningful conclusions from top negative reasons
    d. Drafting the paper

# REFERENCES


[1] S. Chaturvedi, V. Mishra and N. Mishra, "Sentiment analysis using machine learning for business intelligence," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, 2017, pp. 2162-2166, doi: 10.1109/ICPCSI.2017.8392100.

[2] Curran, Kevin & O'Hara, Kevin & O'Brien, Sean. (2011). The Role of Twitter in the World of Business. IJBDCN. 7. 1-15. 10.4018/jbdcn.2011070101.

[3] Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. Information 2019, 10, 150.

[4] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

[5] Rish, Irina. (2001). An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell. 3.

[6] Suppala, Kavya & Rao, Narasinga. (2019). Sentiment Analysis Using Naïve Bayes Classifier. International Journal of Innovative Technology and Exploring Engineering(IJITEE). Volume-8 Issue-8 June, 2019

[7] Bayes, Thomas. (1763). An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. https://doi.org/10.1098/rstl.1763.0053

[8] Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

[9] Asghar, Muhammad Zubair et al.(2017). "Lexicon-enhanced sentiment analysis framework using rule-based classification scheme." PloS one vol. 12,2 e0171649. 23 Feb. 2017, doi:10.1371/journal.pone.0171649