**US AIRLINE**

**SENTIMENT ANALYSIS**

**CSC 6850 Final Project Report**

**BY**

Jesse Annan

July 25, 2023

**PROBLEM DESCRIPTION**

Sentiment analysis is the computational examination of people's opinions, attitudes, and sentiments about a topic. One of the key topics among travelers has been the topic of airline service satisfaction. A company naturally strives to completely meet the expectations of its customers, but this isn't always the case and can occasionally put the passengers in challenging situations, such being stranded. If an airline firm can accurately evaluate, foresee, and respond to the complaints of its consumers, will popularity and revenue increase? This project aims to provide an answer to that question as well as a tool to help travelers make wise choices before making a reservation with an airline.

**Dataset**

The dataset used in this project was sourced from [Kaggle.com](https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment). Below is a list of all the features in the dataset, as well as a brief description of each feature:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Tweet\_id | A unique identification number for each tweet |
| Airline\_sentiment | Feedback sentiments {-1: negative, 0: neutral, 1: positive} |
| Airline\_sentiment\_confidence | Confidence of annotation for each tweet |
| Negativereason | Main negative reason why feedback was given |
| Negativereason\_confidence | Confidence of negative reason |
| Airline | Name of all (6) airlines in the dataset |
| Airline\_sentiment\_gold | Annotation of a tweet {negative, neutral, positive} |
| Name | Tweeter account (username) providing the feedback |
| Negativereason\_gold | Main negative reason why feedback was given |
| Retweet\_count | Number of times feedback was retweeted |
| Text | Main content of feedback |
| Tweet\_coord | Coordinate (location) of account providing the feedback |
| Tweet\_created | Full date time feedback was created or posted |
| Tweet\_location | Location of the feedback provider |
| User\_timezone | User time zone |

**Proposed Analytics Solution**

The goal of this project is to develop a machine learning model that will help airline companies identify services that need to be improved to satisfy their customers and better understand the needs and complaints of their customers through analysis of their feedback. By incorporating these insights, airline firms will have the resources they need to enhance customer satisfaction, increase brand recognition, and ultimately increase revenues.

**DATA PREPROCESSING AND EXPLORATION**

The dataset has one target feature and 14 features overall, as can be seen above. However, the majority of the characteristics are deemed unnecessary for my project, thus before starting the data exploration process, I took into consideration features with fewer than 11 unique entries. As a result, our features—excluding the target feature—dropped from 14 to 4. I initially take a look at the data quality report for the four chosen features before I start my preprocessing and exploratory analysis.

*A screenshot of a computer

Description automatically generated*

***Figure 1:*** *A Data Quality Report on Selected Features.*

The data set comprises 14,640 instances, according to the data quality assessment; the majority of the feedback was *negative*, with a frequency of 9,178, or almost 63% of our dataset. Additionally, *customer service issues* were the most frequently reported or tweeted unfavorable issues for most people, as well as the most well-known airline, *United*, with 3,822 flights (or feedback), or around 26%.

The *Airline\_sentiment\_gold* feature was also dropped before proceeding because it had *99.73%* of its data missing and it seemed to be copy of *Airline\_sentiment* feature so I am assuming there’s no information lost dropping this very sparse feature*.*

A screenshot of a computer

Description automatically generated

***Figure 2 (a):*** *Current Dataset After Dropping Irrelevant Features.*

Since this is a tweeter dataset, our main independent feature, *text*, contains symbols, emoji, regular expressions, extra spaces, and most crucially, worthless Twitter tags from other users. *Text* will be utilized to train our models. Thus, removing these characters was the first step in the data cleaning procedure. The next stage was to deal with words or expressions referred to as *"stopwords"* that are frequently employed in (English) sentence construction but have no meaningful bearing when identifying opinions as negative, neutral, or positive. *Sample stopwords: I, it’s, didn’t, you’ve, a, the, is, are, an, so, what, my, how,* etc.

A screenshot of a computer

Description automatically generated

***Figure 2 (b):*** *Dataset after Text Preprocessing*

*A graph of negative and neutral positive

Description automatically generated*

***Figure 3:*** *Frequency of Sentiments.*

The frequency of sentiment expressed in the dataset is depicted in this graph. With around *63%* of the feedback expressing a complaint, the dataset is dominated by *negative* feedback. The least amount of tweets came from the *positive* category (2,363 tweets, or around *16%* of the responses).

The frequency of sentiment for each airline in the sample is examined in **Figure 4**. *United* appears to be the most well-liked airline in our dataset with sentiments totaling 3,822. It does, however, have some negatives, though, as it receives the most unfavorable reviews out of the six carriers. Also, since *Virgin America* only accounts for *4%* of our dataset's total flights, having the fewest bad reviews (181) does not necessarily suggest it is the most reliable airline. *US Airways* and *American* Airlines, which account up *20%* and *19%* of the dataset respectively, are not far behind in terms of popularity.

A graph of different colored bars

Description automatically generated

***Figure 4:*** *Frequency of Airline Sentiments.*

I represented the frequency of the *negative* explanations in **Figure 5.** This demonstrates that the majority of comments expressed worry about or dissatisfaction with the *customer service* received prior to or following their flights. The second unsettling problem for customers was *delayed flights*, which received 1,665 comments and accounts for *18%* of the *negative* feedback. Additionally, it is obvious that all airlines are doing a fantastic job of securing and delivering luggage because this reason receives the fewest bad comments (only 74), or around *1%* of all negative comments. **Figure 6** depicts the information shown in **Figure 5** for each airline. All airlines are doing their utmost to transport luggage, as is evident. It is also important to note that many *American Airlines'* critics expressed their dissatisfaction with *delayed flights*, with poor *customer service* a close second.

A graph of numbers and a number of different colored bars

Description automatically generated

***Figure 5:*** *Frequency of Negative Feedback Reasons.*

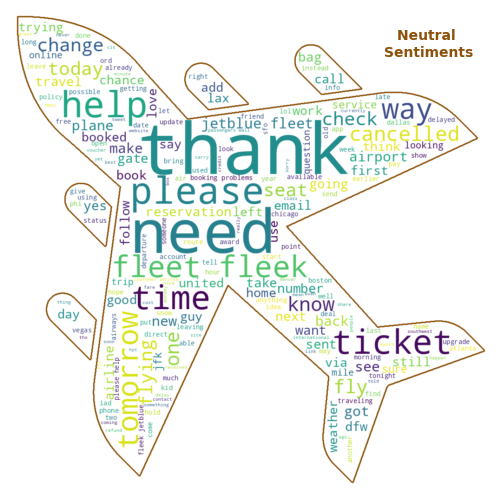
***A group of different colored bars

Description automatically generated***

***Figure 6:*** *Frequency of Negative Feedback Reasons for each Airline.*

Next, I represented in **Figure 7** the words that are frequently used to categorize tweets or feedback as *positive, neutral, or negative*. As this is a flight data set, which very probably contains the word *"flight(s)"* in at least 50% of the data set, it is crucial to highlight that all occurrences of the word *"flight(s)"* were deleted before making this graphic. This omission made it simple to identify the words that best captured the emotions of the airline passengers.

A plane with words in the shape of a plane

Description automatically generated 

A plane made out of words

Description automatically generated

***Figure 7:*** *Word Cloud of sentiments.*

**MODEL SELECTION AND EVALUATION**

**A screenshot of a phone

Description automatically generated**

***Figure 8:*** *Dataset state before modelling*

This project is a classification problem – predicting the class (*sentiment)* of a given tweet/feedback. The classes: *-1: negative, 0: neutral, 1: positive.*

**Evaluation Metrics**

The *accuracy score*, the *f1-score, the precision, recall*, and the *macro and weighted averages* generated by a classification report were the evaluation metrics employed for the classification models. The f1-score is used to determine the prediction skill of a model by focusing on its class-wise performance rather than an overall performance, as is done by the accuracy score. The accuracy score is equivalent to R2 for a classification issue.

Where: TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative

**Models**

I chose three traditional classification models—Logistic Regression, Support Vector Machine, and Naïve Bayes classification—and put them into practice.

**Sampling and Evaluation Settings**

The models chosen above had the best assessment results when the train-test-split was 80% for the train data set and 20% for the test data set.

**Hyper-Parameter Optimization**

I discovered that all models had improved performance on both the train and test sets after implementing the lemmatization technique to change some words to their stem or basic forms (for example, walk and walking were both transformed to walk) and using unigram and bigram bag of words in an attempt to extract context from tweets.

**RESULT, CONCLUSION, AND RECOMMENDATION**

***Model 1: Logistic Regression***Train Accuracy: **92.18%** Test Accuracy: **78.89%**

A yellow and purple squares with numbers

Description automatically generated A screenshot of a computer screen

Description automatically generated

*Model 2: Support Vector Machine**Train Accuracy: 91.69%**Test Accuracy: 78.62%*

*A yellow and purple squares with numbers

Description automatically generated A screenshot of a computer screen

Description automatically generated*

*Model 3: Naïve Bayes**Train Accuracy: 83.64% Test Accuracy: 75.61%*

A yellow and purple squares with yellow numbers

Description automatically generated A screenshot of a computer screen

Description automatically generated

Tunned Model: Lemmatization + (1,2) Ngram BOW

***Model 1: Logistic Regression***Train Accuracy: **98.39%** Test Accuracy: **80.40%**

A chart of different colors

Description automatically generatedA screenshot of a computer screen

Description automatically generated

*Model 2: Support Vector Machine**Train Accuracy: 93.16%**Test Accuracy: 77.60%*

*A chart of different colors

Description automatically generated* *A screenshot of a computer screen

Description automatically generated*

*Model 3: Naïve Bayes**Train Accuracy: 91.44% Test Accuracy: 72.75%* *A chart of different colors

Description automatically generated* *A screenshot of a graph

Description automatically generated*

We conclude that *Logistic Regression is the best* *model* based on the evaluation metrics discussed above, and we advise airline businesses to employ the model to ascertain the tone of fresh comments. It is crucial to notice that while the Naïve Bayes has the best recall value for the negative class, its accuracy performance is comparably weak, which lowers its f1-score.

Additionally, by examining the *"support"* values in the classification report, we can see that there is a class imbalance that was already recognized *(page 4),* and as a result, we can see a small difference between the "macro avg" and the "weighted avg."

Furthermore, we saw an improvement in all three models after applying lemmatization and (1,2) Ngram BOW, which increased the dimensionality of our data set from 8,556 to 61,100. But I believe that all models of the train set have an overfitting problem.

In light of the fact that, sentiments containing words that are frequently found in both the negative and positive classes led to poor classification of the neutral classes, I advise one should consider developing models based solely on the positive and negative feedback. In addition, further research should explore sampling methods to address the problem of class imbalance.

Finally, because features like *Negativereason* were not considered by my model, future research should experiment with concatenating the *Negativereason* feature and the *text* feature. The feature has distinct reasons why something is unfavorable; thus, the model might do a better job of capturing those reasons than it now does.

**CREDITS AND REFERENCES**

1. [**How to create custom colormaps in python with matplotlib**](https://www.youtube.com/watch?v=qk0n-YaKIkY)
2. [**Machine Learning Techniques for Text Representation in NLP**](https://www.analyticsvidhya.com/blog/2022/02/machine-learning-techniques-for-text-representation-in-nlp/)
3. [**Text Representations**](https://www.kaggle.com/code/nkitgupta/text-representations)
4. [**Generating WordClouds in Python Tutorial**](https://www.datacamp.com/tutorial/wordcloud-python)
5. [**Introduction to NLTK: Tokenization, Stemming, Lemmatization, POS Tagging**](https://www.geeksforgeeks.org/introduction-to-nltk-tokenization-stemming-lemmatization-pos-tagging/)