Twitter US Airline Sentiment Analysis

By William Dew

Introduction

Air travel is a huge business in the US. Since Air travel is a service industry Airlines will want to know how they can improve or what they need to do better from their customers. People are always talking about their air travel stories to each other being good or bad. With twitter people can and will tag the airline in their response to their travel. From twitter data the airlines can see what features are causing positive or negative tweets. And from the data decide what to do as a company.

This project will be focusing on building a model that will analyze the text of a tweet. There are three sentiment classes that a tweet can be marked as in this data: negative, neutral, and positive. The goal of the model is to predict, from the text, what sentiment class the tweet is. Once the unknown tweets are classified the airlines can determine how to improve or see what they are doing right.

Data

The Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service"). The data is on kaggle at https://www.kaggle.com/crowdflower/twitter-airlin.

The twitter data has 14639 rows and 15 columns. Each column is a characteristic of the tweet and each row is a unique tweet. The 15 columns can be divided into three types:

1. Qualitative:
   airline_sentiment: classification of negative, neutral, or positive
   negativereason: reason the flight was bad (late, canceled, etc.)
   airline: US airline name

airline_sentiment_gold: classification of negative, neutral, or positive
name: twitter username
Negativereason_gold: reason the flight was bad (late, canceled, etc.)
text: text of the tweet
tweet_coord: longitude and latitude
tweet_location: city or location of tweet
user_timezone: timezone of user

2. Quantitative:
   airline_sentiment_confidence: confidence number given by human
   negativereason_confidence: negative confidence number given by human
   tetweet_count: number of retweets
   tweet_created: date and time tweet was created

3. Unique- each value in column is unique, no duplicates:
   tweed_id: unique id for each tweet.

Data Wrangling

Before analyzing the text in the tweet the non-text columns will be looked at to see if they give any insight. Some columns have really low non-null object counts. Columns *airline_sentiment_gold* has 40, *negativereason_gold* has 32, and *tweet_coord* only has 1019. These columns have a lot of missing values and won't be useful in analysis. They will be dropped. The max for *retweet_count* is 44 and is too low to be useful and will also be dropped. The two other columns *airline_sentiment_confidence* and *negativereason_confidence* seem to be human confidence of rating the tweets. These columns will not be helpful because of the human bias and will be dropped.

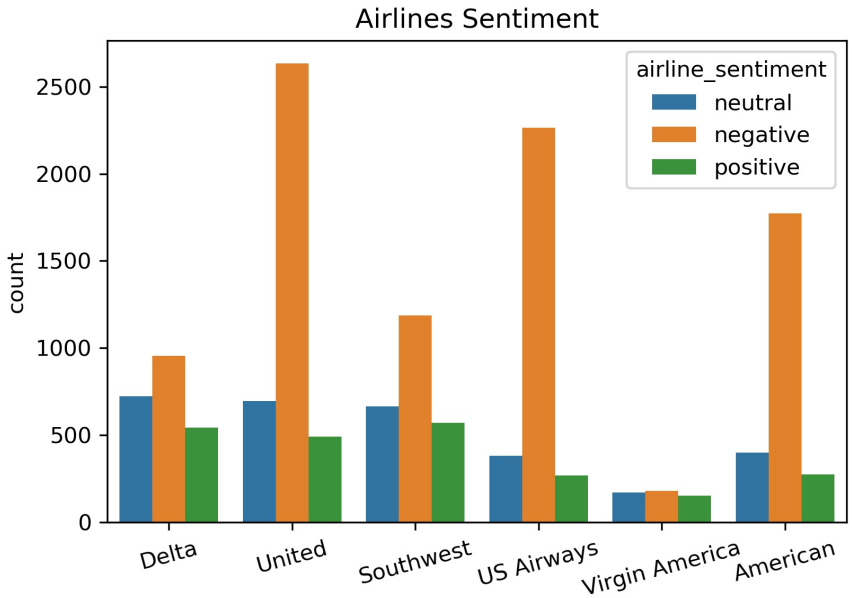

Figure 1

In the graph above we see the airline sentiments of each airline neutral, positive, and negative counts. We see that all but Virgin America have a lot more negative tweets then neutral or positive and in some cases more then both of them combined. This means the data is unbalanced with negative being the overwhelming class. This must be addressed when building the model.

The columns *name*, *tweet_created*, *tweet_location* when grouped by *airline_sentiment* have very low counts and might not be a great feature to find correlation with. Looking at *tweet_created* as a different type of date would probably bring the counts up. We could do day of week, hour, or month date.

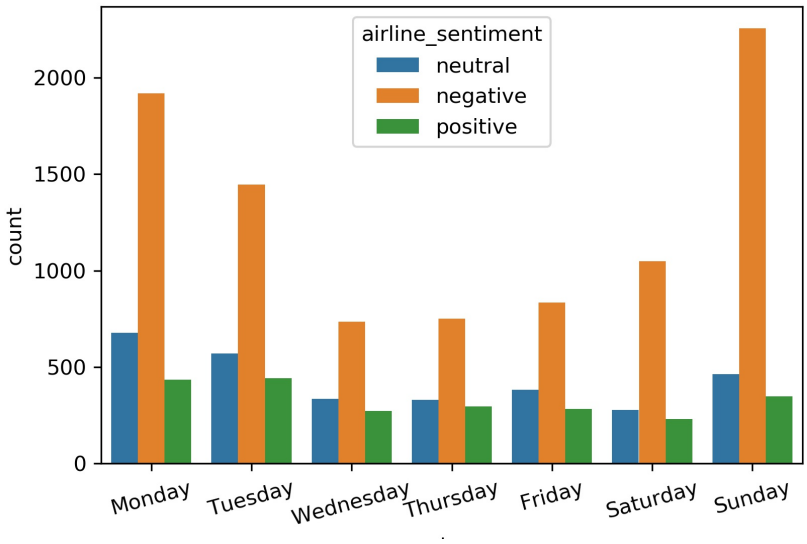

Figure 2

In figure 2 we see that most tweets were created on Monday, Tuesday, and Sunday. This is probably when people are coming home from their vacations. Tweets by hour of the day have a normalized pattern suggesting people tweet mostly during business hours with declines in the early morning and late night. Looking at the dates of collection the data spans 8 days between February 16 and 24 in 2015.

Text Cleaning

A tweet needs to be cleaned to help improve the learning efficiency of machine learning models. Tweets contain punctuation, stopwords, and combination of lower and uppercase words, which affect the model's learning capability. The following are the steps used in cleaning the tweets:

1. Convert text to lower-case: Converting tweets to lowercace because text analysis is case sensitive. This means that "Good" and "good" are considered two different words by the model.

2. Remove URLs: Remove all possible URL links since we are not going to dive deeper into the links
3. Remove usernames: Remove all possible usernames since we are not going to dive deeper into the usernames
4. Remove punctuation: Punctuation is removed from data because it does not contribute to text analysis. It impairs the model's ability to differentiate between punctuation and other characters.
5. Remove stopwords: Stopwords have no analytic value in text analysis. They need to be removed to reduce complexity of the model.

After cleaning the text we can see the the words that are most common in each sentiment in the following word balloons:

Figure 3 Negative Word Balloon

Figure 4 Positive Word Balloon

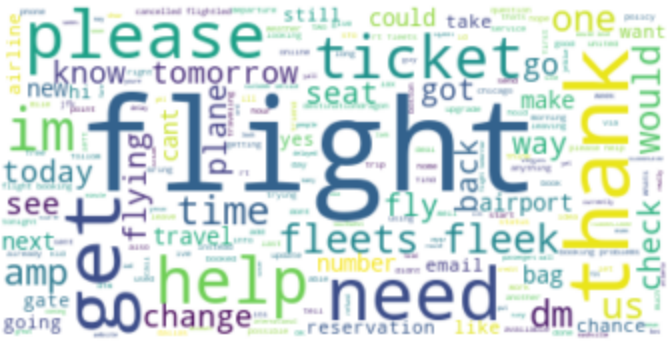

Figure 5 Neutral Word Balloon

The word balloons above show which words show up the most in each type of sentiment. The word ‘flight’ is an often used word in all three sentiments. Negative sentiments contain: customer service, help, plane, and bag. On the positive sentiment word balloon we see ‘thank’ as the most prominent word. Neutral sentiments have please, need, and get as prominent words. These word balloon graphs show a snapshot of prominent words that will likely be used in the models.

Model

Now that the data is cleaned and organized, a predictive model may be built with the sklearn algorithms. The model will be trying to predict the *airline_sentiment* using text from the tweets. Classifier machine learning algorithms will be used.

The text data will be put into vectors to build a vocabulary for analysis. The two vectorizers are Bag of Words and TFidf. Bag of Words provides a simple way to both tokenize a text and build a vocabulary of known words. It also encodes the text using the vocabulary it built. TFidf calculates word frequencies. It will tokenize, learn the vocabulary, inverse text frequency weightings and encode new text.

After the vocabulary vectorizers are built the data will be scaled. The scaler will translate each feature individually such that the maximal absolute value of each feature is the training set will

be 1.0. It does not shift/center the data, and thus does not destroy any sparisity. Scaling helps to not let features with larger units have undue influence on the classifier as would be the case if the classifier uses some sort of distance measurement as a similarity metric.

After scaling the data, classification algorithms will be used to predict the twitter sentiment. Three algorithms will be tested. Naive Bayes, SVM, and Random Forest. Each of these algorithms will also be tuned with cross validation to find the best algorithm parameters for the model.

The scoring parameter the models will use is F1. F1 score conveys a balance between precision (number of positive predictions divided by the total number of positive class values predicted) and recall (number of positive predictions divided by the number of positive class values in the test data). A higher F1 score means the model is better at prediction. Each model during the Randomized Search and Grid Search Cross Validation will give a F1 score and a mean and standard deviation will be taken of each F1 score of all models.

The text data was split into training and testing sets and kfold was used to split the training set into training and validation sets. After running all the different models with Randomized Search and Grid Search cross validation results were obtained. The models with the best mean F1 score and mean standard deviation were used to predict the test set. The following table shows the F1 scores for the training and validation sets of each model:

Vectorizer	Classifier	Training Set F1	Validation Set F1
Bag of Words	Naïve Bayes	0.969 (0.002)	0.634 (0.009)
Bag of Words	SVM	0.968 (0.001)	0.707 (0.010)
Bag of Words	Random Forest	0.991 (0.001)	0.701 (0.012)
Tfidf	Naïve Bayes	0.985 (0.001)	0.639 (0.009)
Tfidf	SVM	0.880 (0.001)	0.695 (0.005)

Tfidf	Random Forest	0.990 (0.001)	0.698 (0.011)

Table 1

From the table above we see that using the Bag of Words and SVM algorithm had the highest validation F1 scores.

Vectorizer	Classifier	Precision	Recall	F1
Bag of Words	SVM	0.731	0.699	0.713

Taking the test set through the pipeline and the best estimator from the random search we see that we get a F1 score of 0.713. This is higher than validation and training set. This shows that the model is good at predicting the sentiment from a tweet.

Conclusion

Airlines can do a lot with a model that can take tweets and determine what sentiment they are. An airline could reward users who leave positive sentiment. An airline could see what people are complaining about the most and see what fixes they could do. The airline could see what services are getting good praise. These and other information can be obtained from having a predictive model.

One area that could be explored is the use of usernames and URLs. Both were removed from the previous models but could be explored more to determine if certain URLs are mentioned in certain sentiments.