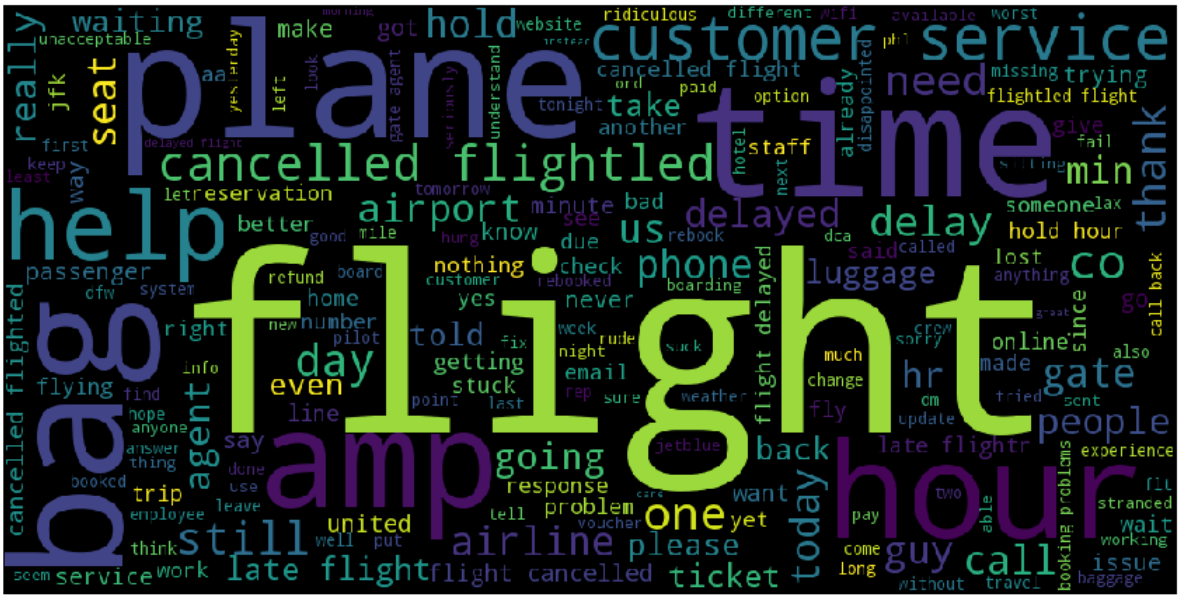
Sentiment Analysis of Tweets for the US Airline industry



Prepared by:

Donald Troy Lane
500820403

Contents

Introduction

It is important for airlines to know how they are perceived to their customer base. Passenger counts are growing but it is very important to ensure repeat business through positive client satisfaction; it is a competitive market.

One avenue of learning gaining passenger feeling is through social media channels. The study will focus on major US airlines and their popularity through the mining of Twitter 'tweets'. The primary objective of this study will be focused on the selection and assessment of various supervised classification approaches to determine the optimum classifier for future prediction of sentiment of 'tweets' focused on the US airline industry.

Furthermore, data exploration will examine comparisons of the sentiment classifications per airline, reason for tweeting, including time of day relationships.

All programming code used to generate the charts and the results of this classification project are contained in the author's public github repository. This can be accessed here:

https://github.com/troylane/capstoneDL/tree/master

It also includes the datasets used in the project

Primary objective

Supervised learning techniques primarily classification algorithms such as multinomial Naïve Bayes and Random Forest have been employed to determine the most optimum model to predict the sentiment of Tweets extracted from Twitter. The classification of the sentiment into a negative, neutral, or positive category could help the major stakeholders, namely the airlines, understand their customer's and the general public's perceptions of their business. Performing further text mining could allow these organizations to extract key pain points for improvement and therefore, potential customer retention, and.

The primary objective of this study is to assess and determine the most optimum classification method to predict sentiment of future tweets.

Literature Review

A literature review was performed. A list of resources follows:

1. Xiaotong Duan, Tianshu Ji, Wanyi Qian. Twitter US Airlines Recommendation Prediction

In this paper, the authors reviewed various supervised learning techniques to classify or label positive or negative sentiment of tweets. Four different classification algorithms were evaluated to extract sentiment; these included Naïve Bayes, Support Vector Machine, neural network, and recurrent neural network. Upon classifying sentiment, they further used classification to predict most negative reason

for each tweet. They used tweets that had previously been labeled with a negative reason to train and test the algorithms, then applied the resulting model to the 20 percent of tweets that did not have a negative reason tweet. They did not mention tools used, but the process will be useful for this capstone project.

2. Quan, Vicky. Step-by-step sentiment analysis: Visualizing United Airlines' PR Crisis. April 26, 2107.

http://ipullrank.com/step-step-twitter-sentiment-analysis-visualizing-united-airlines-pr-crisis/
Looks at similar problem statement as this capstone project. Outlines the steps, at a high level, to derive results (sentiment) using MonkeyLearn and Python. Originally important to this project was it also provides a step by step instruction on how to scrape data from Twitter using the Twitter API in Python. Also speaks about unigrams (adding individual words to the feature vector) which can be used for categorization of reason for negative sentiment.

3. Tutorial: Sentiment Analysis of Airlines Using the syuzhet Package and Twitter

https://colinpriest.com/2017/04/30/tutorial-sentiment-analysis-of-airlines-using-the-syuzhet-package-and-twitter/
An individual's website/blog where he provides a step by step guide which includes code for snippets to complete a sentiment analysis of airlines (different data set). He used a library called syuzhet to perform the sentiment analysis in R. He details in a section the sentiment scoring using syuzhet which Is a different approach than other libraries. For this capstone, the hope is to create a scoring algorithm, even if a rudimentary one, so that a scoring can be applied based on weighting of negative and positive words and therefore severity of negative tweets can be categorized.

4. Justin Martineau, and Tim Finin Delta TFIDF: An Improved Feature Space for Sentiment Analysis

http://ebiquity.umbc.edu/_file_directory_/papers/446.pdf
Highlights a new variation of TF-IDF technique, referred to as Delta TF-IDF which used for sentiment analysis. The paper illustrates a case study where several different algorithms and approaches were used and shows that Delta TF-IDF proved to be more accurate a performing binary sentiment analysis (negative-positive). This approach will be used/tested for this capstone project.

5. R: Twitter Sentiment Analysis: https://analytics4all.org/2016/11/25/r-twitter-sentiment-analysis/

A blog that provides an overview, with code snippets of performing sentiment analysis in R. They used plyr and stringr libraries to perform the analysis., This is a useful link as it provides code snippets in the order one should perform a sentiment analysis project, including scraping twitter data.

6. A tutorial on Sentiment analysis https://web.stanford.edu/class/cs124/lec/sentiment.pdf.

Provides a detailed overview of sentiment analysis and illustrates various algorithms to use, how to baseline and how to approach finding the attribute / category or reason for the sentiment.

Other web sources reviewed with no specific paper or direct content included Quora and Stack Overflow.

The Data

The primary dataset that will be utilized for this study was obtained on Kaggle.com. It is the US Airline Sentiment dataset. It was provided in two formats
The dataset is available in two forms:

1. CSV format (comma-separated values)
2. SQLite database format.

The Kaggle Twitter US Airline Sentiment dataset can be accessed here:
https://www.kaggle.com/crowdflower/twitter-airline-sentiment/data

Dataset Description

Table 1 Primary Dataset description

Attribute	Definition	To be used in project
Tweet_id	Unique id of tweet	No
airline_sentiment	Sentiment towards the airline (positive, negative, neutral)	Yes
airline_sentiment_confidence	Calculated sentiment confidence or score.	No
negativereason	Reason for the tweet	Yes
negativereason_confidence	Confidence level/score of extraction of negative reason for the tweet	No
airline	Name of airline	Yes
airline_sentiment_gold	Unknown (very few entries)	No
name	Name of Twitter user	No
negativereason_gold	Unknown (very few entries)	No
retweet_count	Number of retweets	No
text	Tweet text	Yes
tweet_coord	Latitude, longitude of tweet origin	Yes
tweet_created	Date and time the tweet was posted	Yes
tweet_location	User specified tweet location	No
user_timezone	Time zone of tweet origin	No

A second dataset was generated which mimicked Tweets that would be posted to Twitter. 3rd party individuals wrote 10-20 tweets each and provided a sentiment for each tweet. The purpose of this dataset will be to perform classification on a fresh new 'test' dataset, further validating the selected classifier based on the larger dataset used for training.

The format of this data set is as follows:

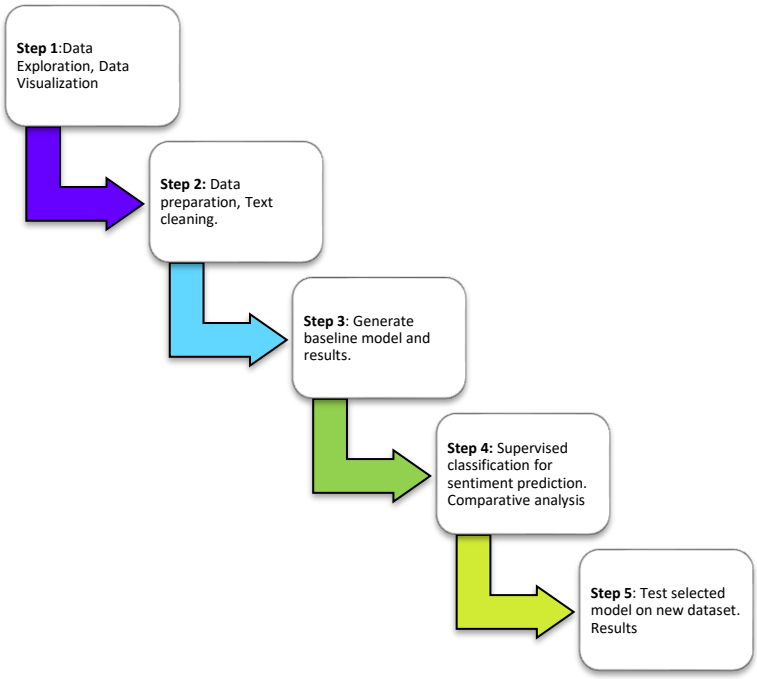
Table 2 Secondary Test Dataset used to further test the selected classifier

Attribute	Definition	To be used in project
ID	Unique id of tweet	No
Tweet	Tweet test	Yes
Sentiment	Sentiment of the tweet (-1 = negative, 1 = positive	Yes

Approach

Create a block diagram for the steps of your approach to clearly provide an overview. For example, if you first scrapped twitter, second applied NLP techniques to extract keywords, third labelled the tweets as positive and negative using a set of keywords, and fourth build a classifier, then you should create a box for each of the steps with arrows connecting one step to the next one. A sample block diagram is shown below.

Figure 1 Project Approach



Step 1 - Data Exploration. Data visualization to understand dataset. Organize data for classification.

Step 2 – Text cleaning. Tokenize each tweet. Remove special characters, mentions, hashtags, stop words, etc.

Step 3 – Generate and assess baseline model (using Multinomial Naïve Bayes classifier with unigrams-based feature set).

Step 4 – Supervised classification for determining optimum model for prediction of tweet sentiment. Generate and assess other supervised classification approaches including Multinomial Naïve Bayes classifier with bi-grams, TF-IDF and Random Forest classification algorithm.

Step 5 – Select optimum algorithm/approach and test on new clean test dataset. Generate and assess results.

Step 6 – Conclude results and findings.

Step 1: Data Exploration, Visualization and organization

Exploratory analysis of the dataset was performed. Due to the nature of this project, and dealing with text, most of the analysis performed focused on examining the content of the text in the tweets, as well as understanding how the airlines represented in the dataset are represented through tweets. The following tables and charts help characterize the original data set.

Table 3: Total number of records and number of variables.

```
'data.frame':  14640 obs. of  15 variables:
```

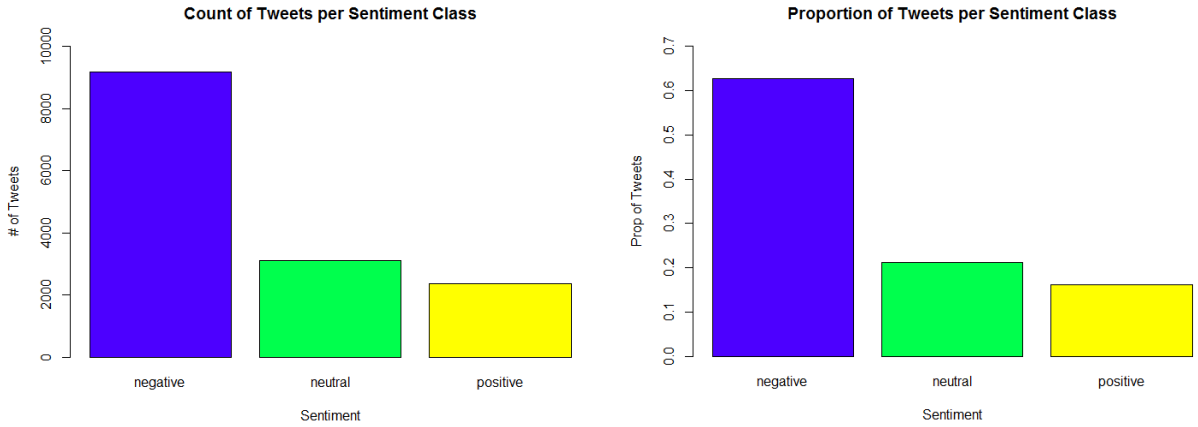
The dataset has a total of 14640 tweets. It also has 15 attributes, including a classified sentiment as well as airline and the text of the tweet. Other attributes will be presented below so that we understand more about the dataset prior to preparing the dataset for classification.

As illustrated in Table 4 and Figure 2, the dataset is unbalanced.

Table 4: Frequency of each Class

	Negative	Neutral	Positive
Count	9178	3099	2363
Proportion	0.63	0.21	0.16

Figure 2 Breakdown of Tweets per sentiment class in the complete data set



Most tweets have been classified as a negative sentiment. It can be seen that there are twice as many negative tweets as there are neutral and positive combined. This may be an important consideration when determining what metrics to weigh more when determining the optimum classifier.

Figure 3 illustrates the number of overall tweets per airline. This has been determined by extracting the mentions (@USER) in the original data. As illustrated, United airlines has the most amount of Tweet mentions in this dataset, followed by US Airlines and American. Virgin America has the lowest amount of tweet mentions.

Figure 3: Total number of Tweets per Airline

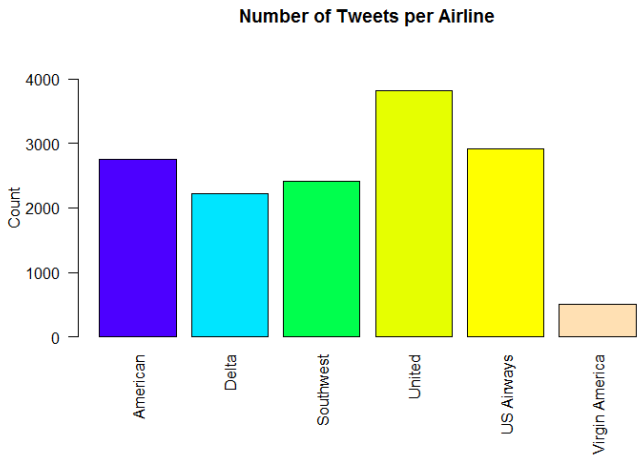
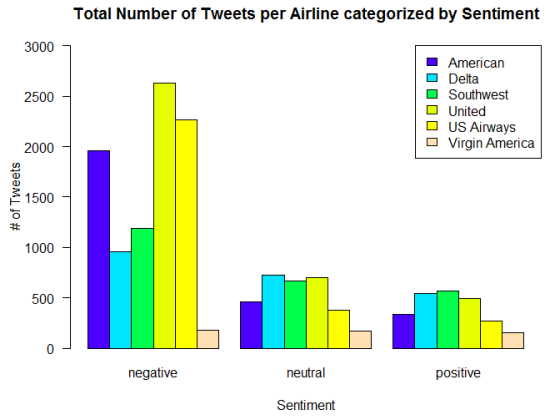


Figure 4 show the distribution of tweet per airline and further breaks down the class (sentiment) per airline. As expected the negative class has the most tweets. Furthermore, as identified above, while United has the most tweets, it also has the most negative tweets out of any airline. However, it does not have as many neutral or positive tweets as other airlines, including Delta, and Southwest.

Figure 5 illustrate the percentage or proportion of tweets for each class for each airline. It highlights the distribution of each class, totaling 100 percent. For example, there is considerable variation in the proportion of which airline received negative sentiment.

Figure 4: Breakdown of sentiment of tweets per airline

Figure 5: Total percentage of Tweets per Airline for each Sentiment class (Negative, Neutral, Positive)



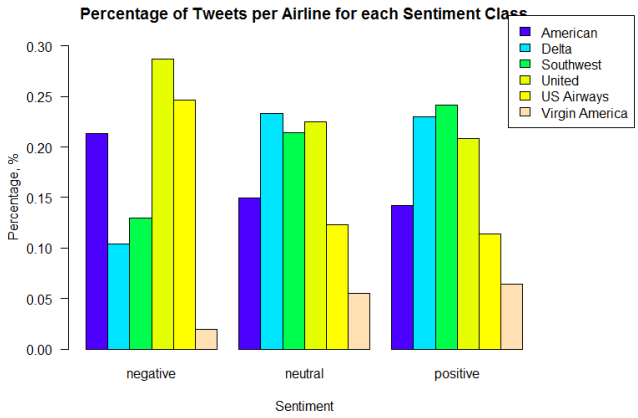
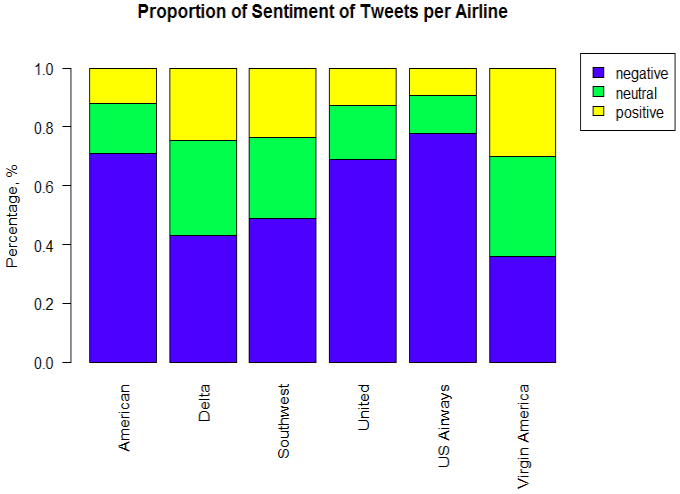


Figure 6 shows the distribution of sentiment for each airline in the form of a stack bar chart. For example, it can be quickly determined that Virgin America has the most even distribution of negative, neutral and positive tweets in the dataset. Conversely, US Airways has a very bias distribution (towards negative sentiment).

Tweets per time of day and sentiment has been plotted in Figure 7. From this we can quickly see that a high number of tweets are tweeted in the morning time, in this case between the hours of 4:00 am and 11:00 am.

Figure 6: Proportion of Sentiment of Tweet per Airline

Figure 7: Tweets per Time of Day and Sentiment



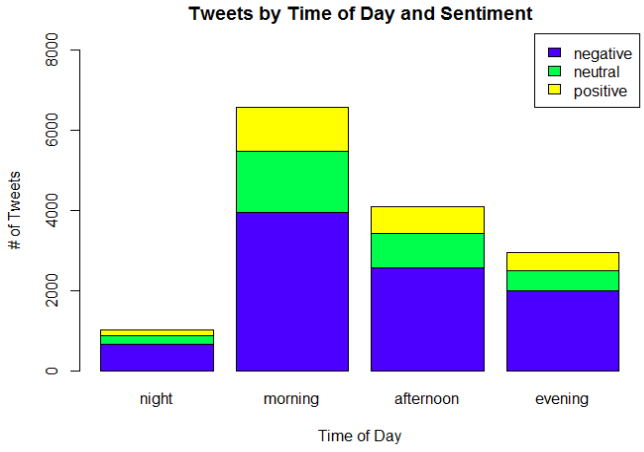
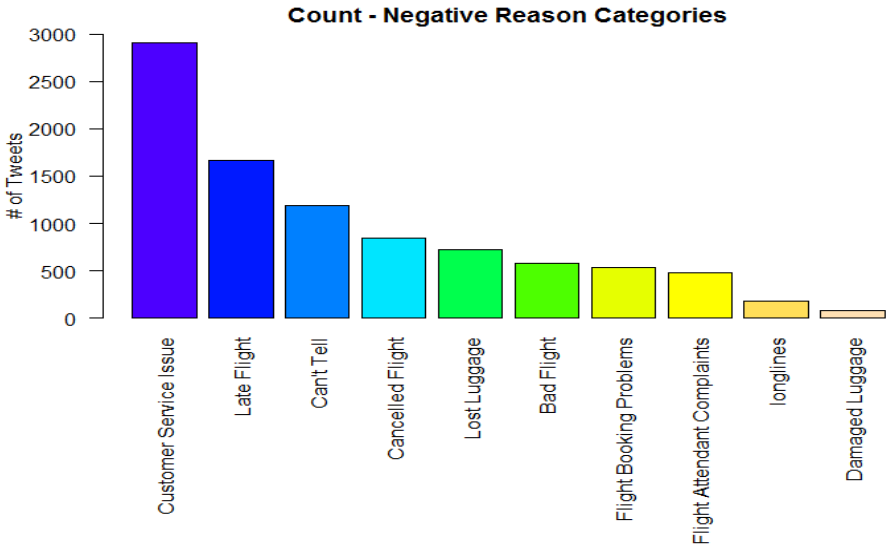


Figure 7 is an ordered bar chart that illustrates the top 10 reasons for tweeting a negative sentiment towards any airline. Customer service and late flights are the main reasons why people are tweeting about the airlines for this period.

Figure 8: Count - Negative Reason Categories



Step 2: Text Cleaning and Data Preparation

To prepare the Tweet text for classification, a series of text processing and text cleaning steps were performed. This included:

1. **Text processing**
   - Tokenize
   - Transform to lowercase

2. **Cleansing**
   - Removal of stop words. To perform this function, the stop-words library found in NLTK library was used.
   - Removal of punctuation
   - Removal of special characters
   - Removal of usernames (mentions, @)
   - Removal of hashtags (#)
   - Removal of single letters
   - Removal of numbers
   - Removal of 2 or less characters

3. **Split the dataset into training and test data sets.**

In addition to the processed dataset with all tweets and classes included (14640 tweets of 3 classes; negative, neutral and positive) a second dataset was generated to filter out neutral class tweets and therefore only include negative and positive classes.

Each of these datasets were then split into 2 subsets: a training set and a test set. The training set for each will be used to build the classification model and 10-fold cross validation will be performed to measure the results. The test sets will be used, acting as “fresh and new” data sets to grade the performance of the classifier. A 75:25 split was performed resulting in the following info for each dataset:

Table 5 results on the full data set, containing 3 classes (negative, neutral, positive)

Dataset	Percentage	Number of Records
Full (Neg, Neu, Pos)	100	14640
Training	75	10980
Test	25	3660

Table 6 Split results on the filtered data set containing 2 class (negative, positive)

Dataset	Percentage	Number of Records
Filtered (Neg, Pos)	79	11541
Training	75	8656
Test	25	2885

Furthermore, prior to generating the baseline model and running each classification algorithm, the dataset was further prepared for classification by transforming and fitting both the training set and the test set.

Step 3 Selecting measurement criteria and Generating the baseline model

Before any classification task, it is important to determine how each classification model will be measured. Unfortunately, there is no one single measurement value or score that determines the overall performance of a classification process. One must consider many metrics as well as the overall problem statement and what is trying to be achieved/predicted to determine the best metrics. For example, in this study, one could assume that the airline industry is mostly interested in negative sentiment towards their service, so they can generate solutions to mitigate or eliminate the pain points. Therefore, should the predictability accuracy of negative sentiment from Tweets take precedence?

That being said, for this study, the following approach and metrics will be utilized and measured: respectively

For each model, a **10-fold cross validation** was performed on the training dataset. In 10-fold cross-validation, the dataset is randomly partitioned into 10 equal size subsamples, and a single subsample is retained as the validation dataset for testing the model and the remaining 9 are used to train the model.

To measure performance of the baseline model, as well as the additional classification models, a set of scores were calculated. These are described as follows:

Accuracy: Also, referred to as the classification accuracy. It represents the percentage of the classifier's correct predictions.

Confusion Matrix: A table that identifies the classification model's ability to identify the existing classes in a dataset. Many other metrics (also used in this study) are derived from info in the confusion matrix.

Recall: A ratio of the number of true positives to the number of true positive and the number false negatives (TP / (TP + FN)). Sometime referred to as 'sensitivity'.

Precision: A ratio of the number of true positives to the number of true positive and number of false positives (TP / (TP + FP)).

F-Score: The F Score, also referred to as the F-beta score, is a weighted mean between Recall and Precision. Recall is weighted more than precision, therefore attempting to create equality between the two measurements.

Support: The number of class occurrences in the dataset.

Null Accuracy: The null accuracy represents the accuracy that can be achieved by always predicting the most frequent class.

ROC-AUC Score: Receiver Operating Characteristic – Area Under the Curve. A single number summary of classifier performance. AUC is useful even when there is high class imbalance (unlike classification accuracy). It is important to note that an ROC-AUC score can only be generated for a 2-class, or binary dataset. Therefore, for this report, ROC-AUC will not be reported for the multiclass classification model results.

ROC-AUC Plot – A chart that plots True Positive Rate (Sensitivity) against False Positive Rate (Specificity)

Naïve Bayes is a probabilistic learning method and is extremely fast, making it an ideal candidate for text classification. Its overall premise is based on Bayes theorem of probability to predict the classes of an unknown data set. It is a robust, proven and fast method for text classification.

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c) and P(x|c). The Naïve Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is referred to as class conditional independence.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

where: P(c|x) is the posterior probability of class (target) given predictor (attribute)
P(c) is the prior probability of the class
P(x|c) is the likelihood which is the probability of the predictor given class
P(x) is the prior probability of the predictor.

The above description was extracted from youtube video “How Naive Bayes Classifier Works 1/2.. Understanding Naive Bayes and Example” and can be found here
https://www.youtube.com/watch?v=XcwH9JGfZOU

In this study, multinomial Naïve Bayes was used. In this implementation, the features are based on frequencies in the dataset, or corpus. It should also be noted that to avoid issues when the frequency is zero, a Laplace smoothing is performed on the feature vectors.

A Multinomial Naïve Bayes classifier, selected from the NLTK library, with uni-gram counts was used as the baseline model. A maximum number of features totaling 3000 was applied, meaning that the top frequent 3000 words were utilized in the classification.

For the baseline model, a Multinomial Naïve Bayes algorithm, using unigrams with a set of 3000 features was selected. Furthermore, it was decided to perform a multiclass classification, maintaining 3 classes: negative, neutral and positive.

A 10-fold cross validation was performed on the training dataset. Results of the 10-fold cross validation step is listed in Table 7.

Table 7: 10-fold cross-validation, accuracy for each set

0.74636364	0.7488626	0.75614195	0.7431694	0.74043716
0.72586521	0.73199635	0.76116682	0.7739289	0.75296263

Furthermore, the test data set produced in the training-test split phase at the beginning was further used to validate the model. The accuracy of the test dataset was computed:

Test data Accuracy = 0.76530054644808743

.

Comparing the final accuracies from the cross-validation results to the test data results indicates that overfitting did not occur. Overfitting refers to a condition where the data fits the model too well. It typically results in a low bias, yet a high variance situation. Conversely, there may be some evidence of underfitting meaning that the classification method or algorithm cannot determine any underlying trend in the data. Underfitting is often represented by low variance but a high bias, and is often a derivative of a overly simple model.

The confusion matrix computed, as well as classification measurement scores for the baseline model are as follows:

Table 8 Metrics for baseline model

MN NB, unigram	**Predicted Sentiment**					
Actual Sentiment	Negative (Predicted)	Neutral (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2048	148	91	0.90	0.85	2287
Neutral, 0	375	351	60	0.45	0.52	786
Positive, 1	132	53	402	0.68	0.71	587
Precision	0.80	0.64	0.73			

Table 9 Null accuracy for the dataset - multiclass

Class	Null Accuracy
-1	0.624863
0	0.214754
1	0.160383

Step 4: Finding the optimum classifier

To select the optimum classifier for this dataset and use case, further classification approaches were conducted using Multinomial Naïve Bayes, using 500 features, Multinomial Naïve Bayes, using TFIDF , and Random Forest with TF-IDF.

The results are presented below

Multinomial Naïve Bayes, unigrams, using 500 features – multiclass

10-fold cross validation accuracy results are as follows:

Table 10 10-fold Cross-Validation Accuracy results

0.71181818	0.7133758	0.73066424	0.70673953	0.71129326
0.69763206	0.68003646	0.7183227	0.73564266	0.71194166

the test data set produced in the training-test split phase at the beginning was further used to validate the model. The accuracy of the test dataset was computed:

Accuracy = 0.72185792349726774

The confusion matrix computed, as well as classification measurement scores for the baseline model are presented in Table 11:

Table 11 Condusion matrix and measurement scores for MN NB, 500 feartures

MN NB, unigram	**Predicted Sentiment**					
Actual Sentiment	Negative (Predicted)	Neutral (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	1968	215	104	0.86	0.81	2287
Neutral, 0	432	307	47	0.39	0.45	786
Positive, 1	164	56	367	0.63	0.66	587
Precision	0.77	0.53	0.71			

A Receiver Operating Characteristic – Area Under the Curve cannot be computed for a multiclass classification dataset.

Multinomial Naïve Bayes, using TF-IDF - multiclass.

Again, a multinomial Naïve Bayes algorithm was used for assessing prediction-ability of the tweets. This set differs in how the feature set (words) were selected for building the classification model. In this

scenario, Term Frequency-Inverse Document Frequency (TF-IDF) was used to select the feature set used in the Naïve Bayes classifier.

TF-IDF is an acronym for Term Frequency – Inverse Document Frequency. Is a statistic that reflects the importance of a word to a document in a series of documents. In this study, the tweet is a document and the series or collection is the full data set of tweets. TF-IDF is often used in text mining and information retrieval.

There are two parts to TF-IDF

1. Term Frequency - measures how frequently a term occurs in a document. It is computed as follows:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

2. Inverse Document Frequency – measures the importance of a term.

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

TF-IDF is computed using the following:

$$\text{TF-IDF}(w,d)=tf(w,d)*log(N/f(w))$$

When performing the TF-IDF vectorization of the data, a a couple settings were used to further determine final TF-IDF feature set.

```
vect2 = TfidfVectorizer(use_idf=True, min_df=5, max_df = 0.8)
#use_idf – weight factor must use inverse document frequency
#min_df – remove the words from the vocabulary which have occurred in less
than 'min_df' number of tweets.
#max_df – remove the words from the vocabulary which have occurred in more
than 'max_df' * total number of tweets in dataset
```

Table 12 Results for 10-fold Cross Validation, MN NB, TF-IDF, Multi-class

0.72818182	0.74613285	0.74067334	0.73132969	0.74225865
0.72131148	0.72926162	0.74658159	0.73928897	0.74384686

The overall classification accuracy for MN NB, TD-DF was computed as:

Accuracy = 0.7368852459016394

This indicates a model that is not overfitted as if falls very close to the training data set results computed using the 10-fold cross validation method shown above.

The confusion matrix and final measurement scores / metrics computed for the test set follows:

Table 13 Confusion Matrix and mettics for Multinomial Naïve Bayes classifer, using TF-IDF, Multiclass

MN NB, TF-IDF	Predicted Sentiment					
Actual Sentiment	Negative (Predicted)	Neutral (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2233	37	17	0.98	0.83	2287
Neutral, 0	569	190	27	0.24	0.36	786
Positive, 1	274	39	274	0.47	0.60	587
Precision	0.73	0.71	0.86			

Top features in the TF-IDF vector

```
['zurich', 'data', 'desperately', 'reserv', 'design', 'responsive', 'ripped',
'degree', 'rough', 'decided', 'dealt', 'dang', 'reported', 'custserv',
'scale', 'scavenger', 'curious', 'scheduling', 'crash', 'cranky', 'selfie',
'coupon', 'requesting', 'died', 'costing']
```

At this stage in the project, it was decided to gauge examples tweets that were being misclassified. The following section outlines some of the results. This information can be used for further insight and postentially could be used to help create a classification algorithm that will perform at a much higher accuracy than the models presented in this project.

```
# first 10 false positives (negative incorrectly classified as positive
sentiment)
X1_test[y1_test < y1_pred_class].head(10)
```

Row No.	Normalized Tweet
1249	comment question thanks copy paste response
2280	social listening capabilities awful reply cont...
2856	thanks ground crews surprised flights arrive b...
3009	thanks reply mine missing added
3013	haha clean plane held overnight hangar sounds ...
4479	heart southwest commercials aimed satisfy nano...
5181	breaking heart
5541	easy pleasant difficult stressful learned lesson
5650	fares time service great manufactured quirkine...
6703	kudos crew routing alleviate sale graded

```
# first 10 false negatives (positive incorrectly classified as negative)
X1_test[y1_test > y1_pred_class].head(10)
```

Row No.	Normalized Tweet
8	well didn't
37	done done best airline around hands

81	applied member inflight crew team interested f...
119	love team running gate tonight waited delayed ...
203	cool picture another virginamerica plane wing ...
227	like
229	best flight attendant ever
248	love music blasting gate boston waiting flight...
264	thanks gate checking baggage full flight givin...
307	mean probably inappropriate board

Random Forest Classification, TD-IDF, Multiclass, using TD-IDF

Simply stated, Random Forest is a bunch of decision trees, each holding a random subset of the data. A decision tree first compares the features and calculates which one has the most information gain (TechWorld Australia, 2016). From this calculation the root node and is this process is continued, splitting the nodes until a leaf, or decision node is reached (where entropy = 0, and no further splitting is required).

The model is trained in parallel and on random samples therefore generating a Random Forest. The final prediction is taken from voting on the results of the individual trees or going with decisions that appear most of the times in the trees. In Random Forest, most of class predictions will dictate the final class prediction. (Techworld Australia, 2016).

This makes it a viable approach for text classification. That said, it should be noted that when training the data using cross-validation, it took 30-50 times longer for computer processing than using Naïve Bayes (with a larger feature set)

The results of the Radom Forest classification on the multiclass dataset is as follows:

Table 14 10-fold cross validation results for Random Forest, TF-IDF, multiclass

0.73272727	0.7388535	0.75432211	0.73497268	0.73679417
0.73132969	0.73837739	0.73381951	0.74749316	0.7538742

The accuracy of the test dataset was recorded at

Accuracy = 0.73360655737704916

This indicates a model that is not overfitted as if falls very close to the cross validation results computed using the 10-fold method shown above.

The confusion matrix computed, as well as classification measurement scores for the test set are presented next:

Table 15 Confusion Matrix and mettics for Random Forest classifer, using TF-IDF, Multiclass

Random Forest, TF-IDF	Predicted Sentiment					
Actual Sentiment	Negative (Predicted)	Neutral (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2002	185	100	0.88	0.83	2287
Neutral, 0	404	320	62	0.41	0.47	786
Positive, 1	151	73	363	0.62	0.65	587
Precision	0.78	0.55	0.69			

Using 2 classes only (negative and positive)

Based on the poor performance of the above classification approaches to predict "neutral" sentiment, and in some cases, poorer performance on predicting positive sentiment, it was decided to perform classification on a filtered dataset to determine if the prediction of negative and positive sentiment was improved; therefore, neutral sentiment class tweets were removed. Given the nature of the study, to determine pain points for passengers of US airlines, the main goal is to predict negative sentiment with the highest "accuracy". Essentially, by performing this, the feature sets are altered and potentially can have a positive effect on the overall results.

For each algorithm, and approach listed above using the full, multiclass dataset, the same was performed on the filtered bi-class dataset. Results are presented below.

Multinomial Naïve Bayes, using unigrams, with 2 classes; negative and positive – 3000 features

This approach duplicated the baseline model, however with only 2 classes; negative and positives.

The results are presented below:

Table 16 10 fold cross validation results for multinomial Naïve bayes, 3000 feature, bi-class

0.89850058	0.91234141	0.89734717	0.8867052	0.89364162
0.89248555	0.89942197	0.91213873	0.9017341	0.9017341

The accuracy of the test dataset was recorded at **0.91022530329289431**

Table 17 Confusion Matrix and mettics forMN Naïve Bayes, unigrams, 3000 words, bi-class

MN NB, unigrams	Predicted Sentiment				
Actual Sentiment	Negative (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2230	95	0.96	0.95	2325

Positive, 1	164	396	0.71	0.75	560
Precision	0.93	0.81			

The null accuracies were calculated at:

Table 18

Class	Null Accuracy
-1	0.805893
1	0.194107

The ROC-AUC score for the classification was **0.9459358678955454**

Multinomial Naïve Bayes, using unigrams, with 2 classes; negative and positive – 500 features

The following are the results for the Multinomial Naïve Bayes classifier, using 500 features on the filtered bi-class dataset.

Table 19 Confusion Matrix and mettics forMN Naïve Bayes, unigrams, 500 words, bi-class

0.87889273	0.88119954	0.89042676	0.87283237	0.89248555
0.88439306	0.87283237	0.89942197	0.87745665	0.88554913

The accuracy of the test dataset was recorded at **0.8876949740034662**

The following table highlights the confusion matrix as well as key measurement results for the test dataset:

Table 20 Confusion Matrix and mettics forMN Naïve Bayes, unigrams, 500 words, bi-class

MN NB, Unigrams, 500 words	**Predicted Sentiment**				
Actual Sentiment	Negative (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2194	131	0.94	0.93	2325
Positive, 1	193	367	0.66	0.69	560
Precision	0.92	0.74			

The null accuracies were calculated at:

Table 21

Class	Null Accuracy
-1	0.805893
1	0.194107

The ROC-AUC score for the classifier on the test dataset is **0.92571390168970813**

Multinomial Naïve Bayes, with 2 classes; negative and positive – TF-IDF

The following are the results for the Multinomial Naïve Bayes classifier, using features derived from TF-IDF on the filtered bi-class dataset.

Table 22 Confusion Matrix and mettics forMN Naïve Bayes, TF-IDF, bi-class

0.89504037	0.87773933	0.88811995	0.89017341	0.8867052
0.89595376	0.88092486	0.88901734	0.88092486	0.88439306

The accuracy of the test dataset was recorded at **0.89428076256499134**

The following table highlights the confusion matrix as well as key measurement results for the test dataset:

Table 23 Confusion Matrix and mettics forMN Naïve Bayes, TF-IDF, bi-class

MN NB, TF-IDF	Predicted Sentiment				
Actual Sentiment	Negative (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2304	21	0.99	0.94	2325
Positive, 1	284	276	0.49	0.64	560
Precision	0.9	0.89			

Table 24

Class	Null Accuracy
-1	0.805893
1	0.194107

The ROC-AUC score for the classifier on the test dataset is **0.94375268817204294**

Random Forest Classification, TD-IDF, bi-class, suing TD-IDF

A Random Forest classification model using 2 class filtered data set was generated. Results are below:

Table 25 Confusion Matrix and mettics Random ForestTF-IDF, bi-class

0.88119954	0.89504037	0.89388697	0.88554913	0.90057803
0.89479769	0.89364162	0.87745665	0.86358382	0.89479769

The accuracy of the test dataset was computed:

Accuracy = 0.89601386481802425

The following table highlights the confusion matrix as well as key measurement results for the test dataset:

Table 26 Confusion Matrix and mettics for Random Forest, TF-ID, bi-class

Random Forest, TF-IDF	**Predicted Sentiment**				
Actual Sentiment	Negative (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	2233	92	0.96	0.94	2325
Positive, 1	208	352	0.63	0.70	560
Precision	0.91	0.79			

The ROC-AUC score for the classifier on the test dataset is **0.90534447004608298**

Step 5: Model selection and Test with new dataset with results

When considering the multi-class dataset, there is evidence to support all methods. Using Multinomial Naïve Bayes with a smaller feature set of 500 words returned the worst prediction metrics. In this case, it did not perform as well as the baseline model which utilized a feature set of 3000 words.

Overall, if the primary goal of this study is taken into consideration; to learn of pain points within the commercial airline industry, one would want to have the best results include the highest number of true negatives. If this were the only requirement, the MN NB, using TF-IDF based feature set, would be the optimum classifier based on this dataset, having a count of 2304 True Negatives, with a precision of 0.90 (Negative) and a recall of 0.99 (Negative). The downside to this model was that it performed marginally on the prediction of positive sentiment in Tweets.

For further validation on the selected model, a dataset was generated. This dataset, while small, served as a valid dataset consisting of text that closely resemble tweets in the original dataset. Third party individuals generated the text, and classified each as negative or positive. The dataset was cleaned and processed following the same method used on the original data set.

Results are as follows:

The accuracy of the test dataset was recorded at:

Accuracy = 0.66923076923076918

The following table highlights the confusion matrix as well as key measurement results for the test dataset:

Table 27 Confusion Matrix and mettics for MN Naïve Bayes, TF-IDF, bi-class

MN NB, TF-IDF	**Predicted Sentiment**				
Actual Sentiment	Negative (Predicted)	Positive (Predicted)	Recall	F-Score	Support
Negative, -1	69	2	0.97	0.76	71
Positive, 1	41	18	0.31	0.46	59
Precision	0.63	0.90			

ROC-AUC Score was computed at **0.79374552399140608**

Conclusions

For this study, measurement scores are generally "good". If the confusion matrix values, accuracy, precision and recall are all considered, the most optimum classifier is the Naïve Bayes using TF-IDF. A very close second would be the Naïve Bayes, using 300 word or features. This holds true for both the multi-class data set as well as the 2 class filtered data set.

A clean and fresh test dataset further substantiate this claim that Naïve Bayes classier is a good algorithm to predict sentiment from text, namely tweets.

For both data sets used, we saw a strong prediction-ability for the negative sentiment tweets using all classification methods.

In the second test data set, we saw that the classifier was able to accurately classify True Negatives (classify that the tweet was of negative sentiment) all but two tweets. When comparing this dataset using the Multinomial Naïve Bayes, TF-IDF based feature set, to the full data cross validation results, there may be some evidence of overfitting. The original bi-class test data set had much better overall numbers with a much higher ROC-AUC. Given the consistency of the model's results, it is believed that the dip in accuracy and metrics of the new test dataset is likely due to a poor model for prediction of positive sentiment and small data size.

Given the overall results, there is high confidence that the classifier derived in this study could be used to predict sentiment of US airline passenger using tweets as their means of communication.

References

Evaluating a classification Model
http://www.ritchieng.com/machine-learning-evaluate-classification-model/

Toward optimal feature selection in Naïve Bayes for Text Classification.
https://arxiv.org/pdf/1602.02850.pdf

What does TF-IDF mean?
http://www.tfidf.com/

Appendices

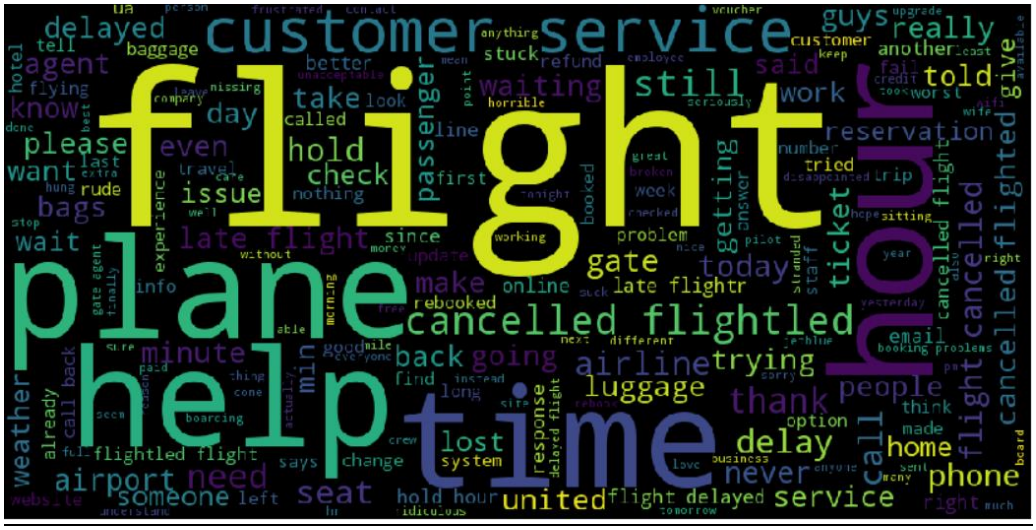
Appendix A

Word Clouds

Word cloud have been generated using python. Word cloud represent a quick glimpse of words that have high frequency in a corpus. Negative, Neutral and Positive word clouds are depicted below. Note that the dataset was cleaned prior to generating the clouds.

Negative Sentiment Word Cloud

Neutral Sentiment Word Cloud



Positive Sentiment Word Cloud