R: Twitter Sentiment Analysis of Airline Companies

Eisha Patel

[Twitter is a great tool companies can use for gathering consumer data and reflecting on consumer sentiments. Often, consumers will resort to social media to state their personal experiences about a certain product/service using a direct mention. For my CKME 136 Capstone project I will be investigating the sentimentality of consumers towards the major U.S airline.]

Ryerson University
CKME136 – Data Analytics Capstone

August 8th 2018

Abstract

This research study attempts to perform a sentiment analysis on the public opinion regarding U.S airline companies using Twitter data. Furthermore, this study tries to identify an efficient method to classify tweets based on sentiment. All analysis has been performed in R Studio. An examination of popular hashtags proved that all U.S airliners were suffering from customer dissatisfaction. An examination of popular bigrams provided further justification for the customer dissatisfaction. The following 4 methods for text classification were implemented and tested; Scoring System, Decision Trees, Random Forest, and Logistic Regression. The machine-learning algorithms, Random Forest and Logistic Regression, classified tweets with highest accuracy.

1. Introduction

Traditionally, companies would invest millions of dollars to gain access to public sentiment. Whether it is public opinion regarding a new marketing campaign or the news release regarding a recent product launch, people in charge of handling a company's public image are constantly relying on such sentiment. And currently, in the age exponential advancement of technology and competition between businesses, companies need answers fast! Fortunately, social media outlets have made public opinion extremely accessible and affordable. The first motivation for this project is to identify the American airline companies, which have poor consumer reviews and further identify why. This analysis can provide the airline company vital information on how to improve their consumer experience. The second motivation of interest is to identify an effective machine-learning algorithm to automate classifying tweets by sentiment.

2. Dataset

The dataset used for analysis consists of 14,641 tweets that make specific mentions to an US Airline company. The dataset can be accessed from *https://www.figure-eight.com/data-for-everyone/* - as an excel file. Figure 2.1 below summarizes the attributes that will be considered for analysis.

Attribute	Description	Data Type	Missing Values
unit_id	Twitter ID unique to each user	Categorical	No
airline_sentiment	User opinion categorized into either positive, neutral, or negative	Categorical	No
negativereason	Justification for negative tweets	Textual	No
airline	Name of airline company	Categorical	No
text	Tweet remark	Textual	No

Figure 2.1: Airline sentiment dataset summary

Volunteers manually produced the two attributes: airline_sentiment and negativereason. Each tweet was read and classified into one of the three categories: positive, neutral, or negative. If a tweet were classified negative, the volunteer would further provide a reason as to why so. The accuracy of manual classification is higher than that of traditional machine learning algorithms because a human reader can interpret linguistic structures with deeper knowledge of actual context. Hence these two attributes will be treated as labeled data. For this project, we will be implementing and testing the accuracy of various text classification and machine learning algorithms. Although a few tweets maybe misinterpreted, we have the ability to read a larger volume of tweets in just a few seconds. It's not practical to have volunteers read tweets every time.

3. Background

To obtain a high-level understanding of the dataset, the following plots were produced. Figures 3.1 and 3.2 below summarize the raw data to show the percentage value of each sentiment category for each airline.

	Sentiment		
Airline	negative	neutral	positive
American	71.040232	16.781443	12.178325
Delta	42.979298	32.538254	24.482448
Southwest	49.008264	27.438017	23.553719
United	68.890633	18.236525	12.872841
US Airways	77.686234	13.079300	9.234466
Virgin America	35.912698	33.928571	30.158730

Figure 3.1: Airline Sentiment as a percentage value

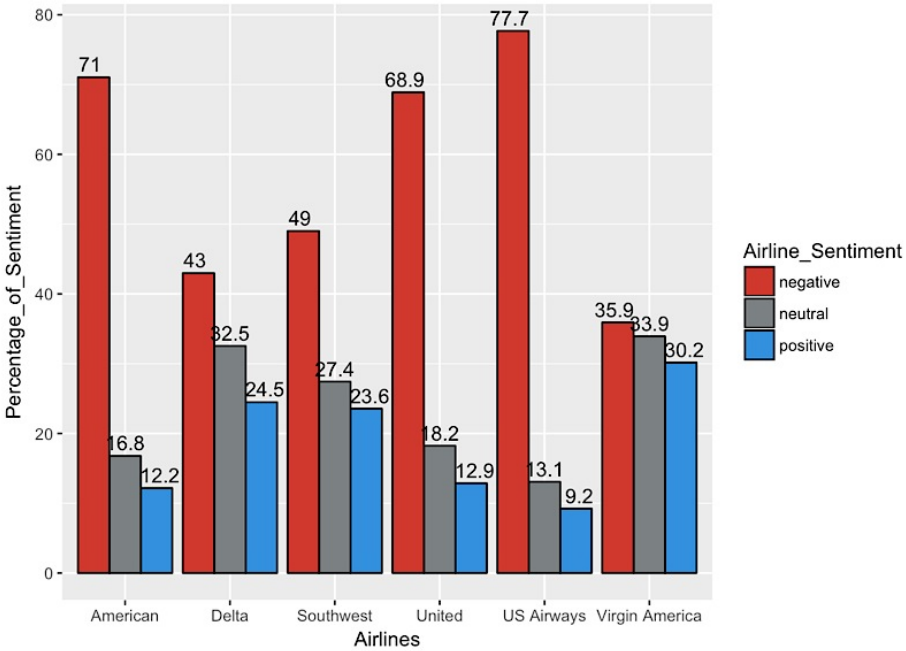


Figure 3.2: Airline Sentiment summarized as a histogram

From the histogram above, it's accurate to say that the major public opinion regarding all 6 airline companies is negative. This serves as motivation for further investigation. Plotting a histogram of the

attribute negativereason, we can rationalize all the negative sentiment. Figure 3.3 below shows the common issues leading to customer dissatisfaction.

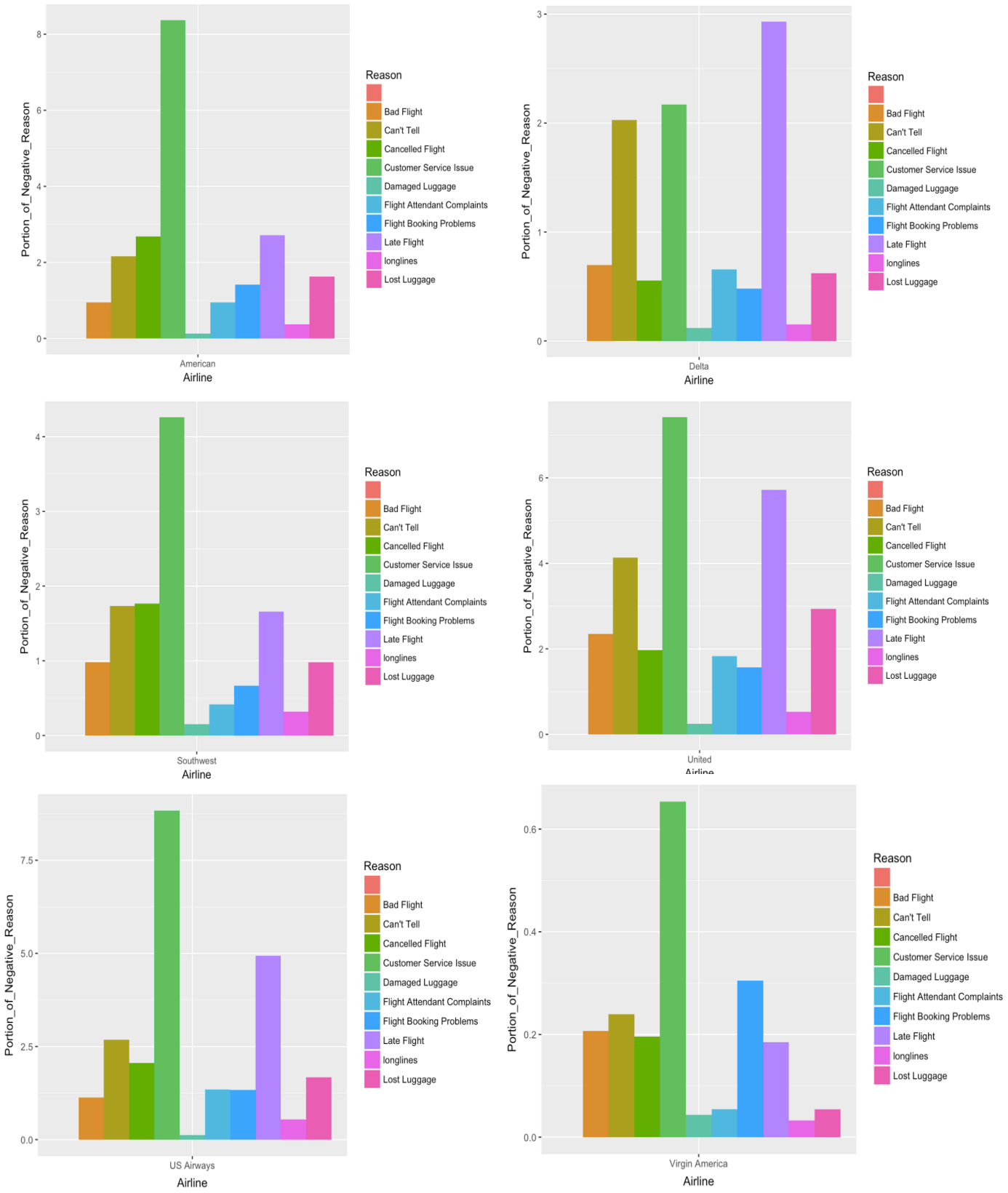


Figure 3.3: Airline Sentiment summarized as a histogram

From face value, it's easy to conclude that customer service is a major issue in all airlines. Surprisingly it is also the issue these companies have most control over. Unlike lost luggage and cancelled flights, providing decent customer service is a company's choice. According to the American Express 2017 Customer Service Barometer, more than half of Americans have scrapped a planned purchase or transaction due to bad service. In addition, 51% of consumers state that they will never do business with a company after a single negative experience. Complaints regarding customer service must be taken into attention!

4. Approach

The incentive for this project is to implement and evaluate algorithms that effectively perform text analytics. All coding and analysis will be performed in *R Studio*. The following techniques will be implemented:

1. Scoring System:

Tweets are run against 2 dictionaries containing a lengthy collection of positive and negative words. For each positive word match, a score of +1 is given and for each negative word match, a score of -1 is given. An overall positive score implies positive sentiment and an overall negative score implies negative sentiment. A score of 0 implies neutral sentiment. The accuracy of this system will tested against the labeled attribute.

2. Hashtag Analysis:

On social media sites such a Twitter, a hashtag is a word/phrase preceded by a pound sign (#) and used to identify messages on a specific topic. Extracting the most popular hashtags in the dataset will give insight on the common remarks tweeters rant or praise about.

3. N-grams:

N-grams are a contiguous sequence of n items from a given sample of text. R has a built-in package called "tm" which allows us to implement n-grams and calculate the frequency of coupled terms such as "*cancelled flight*" or "*lost luggage*".

4. Decision Tree:

A decision tree is a support tool that predicts the outcome of an event. The package "rpart" allows us to implement decision trees in R. The algorithm will run thru a test data set to identify the key terms specific to both negative and positive tweets. A higher frequency of a particular term in a particular sentiment type will compose a node of the decision tree.

5. Random Forest:

Random forest is essentially an ensemble of decision trees. The "rpart" package also contains a feature for random forest.

6. Logistic Regression:

The logistic model is usually applied to a binary dependent variable, which in this case will be positive and negative. The "rpart" package also contains a feature for logistic regression.

Before implementing any of the 6 algorithms above, its important to clean up the tweets. URL's, emojis, special characters, punctuations, numbers, spaces, etc. all need to be removed and tweets need to be lowercased. Initial clean up will prevent redundancy and simplify the analysis. The tweets further need to be vectorized such that each tweet is an array of words.

5. Procedure & Results

In this section, the procedure and results for each of the various approaches will be described in detail.

1. Scoring System:

1.1 Word Dictionaries

Upload 2 dictionaries of containing a list of positive and negative words from https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html and load any missing terms that are needed.

```
setwd("~/Desktop/Capstone")
neg = scan("negative-words.txt", what = "character", comment.char = ";")
pos = scan("positive-words.txt", what = "character", comment.char = ";")

# Add words to the dictionary that do not exist
# American slang words need to be included in the dictionaries
neg = c(neg, 'wtf', 'wonker', 'boo', 'mold', 'sucks', 'wait', 'waiting', 'waited', 'cancel', 'cancelled', 'cancel
ling', 'shit', 'rotten', 'epicfail', 'mechanical')
pos = c(pos, 'lit', 'bro')
```

1.2 Tweet Clean Up and Scoring

It's essential to reduce the tweets into simplified form such that all characters are lower-cased and purely text. A function called sentiment score has been designed to clean tweets and assign a score to all applicable words. The function was implemented on each of the tweets for each airline. Below is an example of the score computed for *United Airlines.*

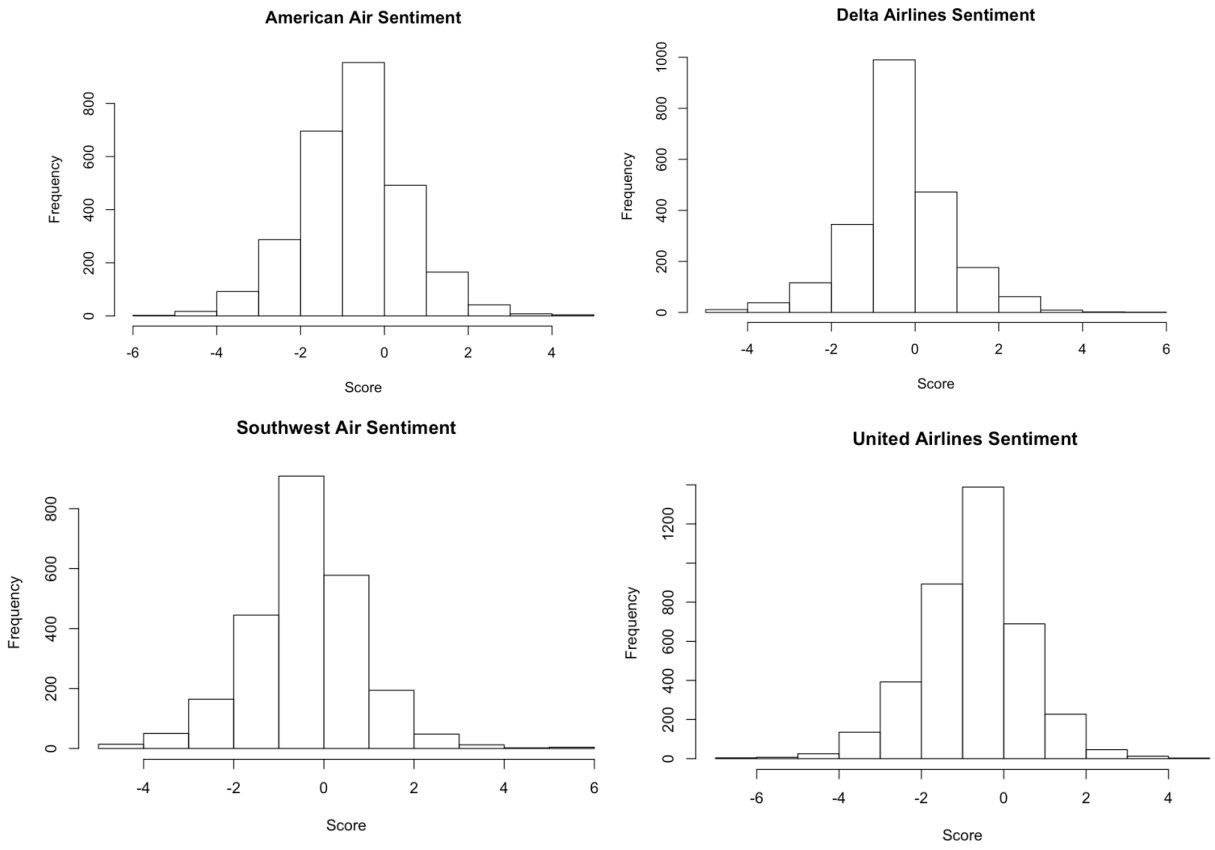
```
sentiment_score = function(tweets, pos.words, neg.words, brand)
{

  require(plyr)
  require(stringr)

  scores = laply(tweets, function(tweet, pos.words, neg.words) {

  tweet = gsub('https://','',tweet) #remove https://
  tweet = gsub('http://','',tweet) #remove http://
  tweet = gsub('[^[:graph:]]',' ', tweet) #remove emojis
  tweet = gsub('[[:punct:]]', '', tweet) # remove punctuation
  tweet = gsub('[[:cntrl:]]', '', tweet) # remove control characters
  tweet = gsub('\\d+', '', tweet) # removes numbers
  tweet = str_replace(tweet, "[^[:graph:]]", " ")
  tweet = tolower(tweet) # all lowercase letters

  word.list = str_split(tweet, '\\s+') # splits the tweets by word in a list

  words = unlist(word.list) # convert the list into a vector

  pos.matches = match(words, pos.words) # returns any matches from the positive dictionary as T/F
  neg.matches = match(words, neg.words) # returns any matched from the negative dictionary as T/F
```

```
  pos.matches = !is.na(pos.matches) # True = 1, False = 0
  neg.matches = !is.na(neg.matches)

  score = sum(pos.matches) - sum(neg.matches)

  return(score)}, pos.words, neg.words)

  scores.df = data.frame(airline = brand, score = scores, text = tweets)

  return(scores.df)

}
```

```
United_tweets = subset(Twitter_tweets$tweet, Twitter_tweets$Airlines == "United")
United_sentiment = sentiment_score(United_tweets, pos, neg, 'United')
head(United_sentiment,5)
```

	score <int>	airline <fctr>	text <fctr>
1	1	United	@united thanks
2	2	United	@united Thanks for taking care of that MR!! Happy customer.
3	-2	United	@united still no refund or word via DM. Please resolve this issue as your Cancelled Flightled flight was useless to my assistant's trip.
4	-2	United	@united Delayed due to lack of crew and now delayed again because there's a long line for deicing... Still need to improve service #united
5	2	United	@united thanks -- we filled it out. How's our luck with this? Is it common?

The table above shows the first 5 tweets scored for United Airlines. The scoring system appears to be performing well. For example tweet number 2 has a score of +2 for the words “Thanks” and “Happy” and tweet number 3 has a score of -2 for words “Cancelled” and “Useless”. Figure 5.1 below shows the histogram of the sentiment scores for each airline.



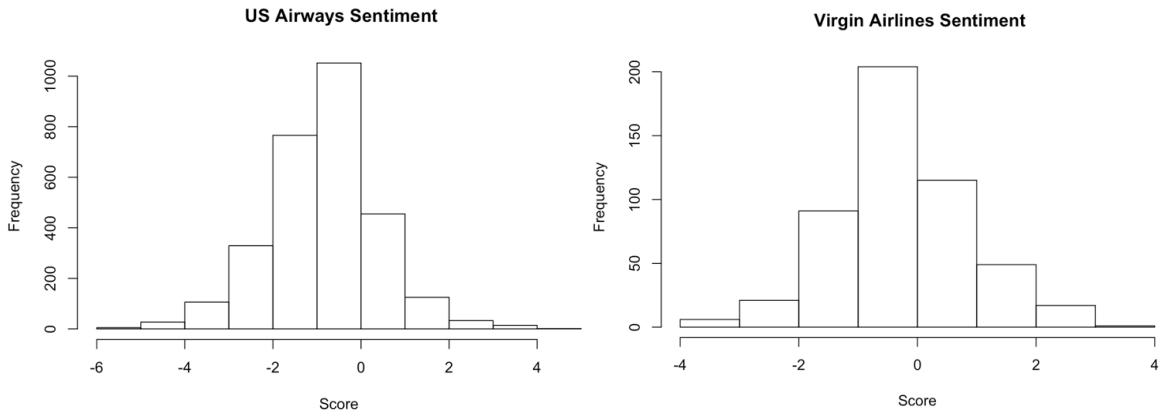


Figure 5.1: Airline Sentiment scores summarized as a histogram

From face value of these histograms, we can observe that majority of the sentiment is negative (higher frequencies to the left of zero). All negative, positive, and neutral frequencies for each airline were counted and their totals were compared against the labeled data. See figure 5.2 below.

Airline	Negative	Labeled Negative	Positive	Labeled Positive	Neutral	Labeled Neutral
American	1094	1960	711	336	954	463
Delta	510	955	722	544	990	723
Southwest	673	1186	838	570	909	664
United	1456	2633	977	492	1389	697
US Airways	1233	2263	628	269	1052	381
Virgin America	118	181	182	152	204	171

Figure 5.2: Sentiment Count Scoring System versus Labeled Data

1.3 Extreme Scores

Based on the comparison with the labeled data, the sentiment-scoring algorithm did not perform adequately for reasons that will be discussed in the conclusion. Narrow down the analysis to extreme scores such that the score is either >= + 2 or <= -2. It's likely that really positive or really negative scores are more accurate.

```
all_scores = rbind(American_sentiment, Delta_sentiment,Southwest_sentiment,
                   United_sentiment, US_sentiment, Virgin_sentiment)

all_scores$positive = as.numeric(all_scores$score >= 2) #returns 1 if condition meet
all_scores$negative = as.numeric(all_scores$score <= -2)

p = aggregate(positive ~ airline, data = all_scores, sum)
n = aggregate(negative ~ airline, data = all_scores, sum)

totals = merge(p, n, by = 'airline')
totals$total_count = p$positive + n$negative
totals$percentage_positive= round(100 * p$positive / totals$total_count)
totals$percentage_negative= round(100 * n$negative / totals$total_count)

totals
```

airline <fctr>	positive <dbl>	negative <dbl>	total_count <dbl>	percentage_positive <dbl>	percentage_negative <dbl>
American	219	398	617	35	65
Delta	250	165	415	60	40
Southwest	260	228	488	53	47
United	288	563	851	34	66
US	173	467	640	27	73
Virgin	67	27	94	71	29

2. Hashtag Analysis:

To extract the hashtags from the original tweets of each airline, a function called hash_tag was developed as below. The top 6 hashtags for each airline company are also given below.

```
hash_tag = function(x){
  hashtags = str_extract_all(x, "#\\w+")
  hashtags = unlist(hashtags)
  hashtags = tolower(hashtags) #make all tweets lowercase
  hashtag_freq = as.data.frame(table(hashtags))

  return(hashtag_freq[order(hashtag_freq$Freq, decreasing = TRUE), ])
}
```

American Airlines

hashtags	Freq
#americanairlines	25
#fail	11
#customerservice	8
#dfw	8
#filmcrew	6
#help	6

Delta Airlines

hashtags	Freq
#jetblue	46
#flyingitforward	12
#travel	10
#flyfi	9
#fail	8
#jfk	8

Southwest Airlines

hashtags	Freq
#destinationdragons	81
#disappointed	8
#fail	8
#southwest	8
#swa	7
#customerservice	6

United Airlines

hashtags	Freq
#unitedairlines	45
#united	26
#fail	23
#customerservice	13
#unfriendlyskies	10
#unitedsucks	10

US Airways

hashtags	Freq
#usairways	27
#usairwaysfail	25
#fail	18
#neveragain	9
#customerservice	8
#clt	7

Virgin Airlines

hashtags	Freq
#cheapflights	4
#farecompare	4
#help	4
#disappointed	3
#middleeast	3
#oscars	3

These hashtags were retrieved from the original dataset containing all 14641 positive, negative, and neutral tweets. The most popular hashtags are the names of the airline brands, which we can ignore because they give no context to sentiment. The hashtag “fail”, “disappointment”, “customerservice” appear commonly among all airliners. Not good news for the companies!

3. N-grams:

3.1 Corpus

A corpus is a collection of documents. Convert the tweets into a corpus for preprocessing. Eliminate stop-words using the built-in English dictionary provided by the “tm” package in R. Further, we reduce stem-words using the “snowball” package.

```
# Begin by creating a corpus: a collection of documents and clean it up
corpus = Corpus(VectorSource(reduced_data$tweet))
corpus = tm_map(corpus, content_transformer(tolower)) # lowercase all tweets
corpus = tm_map(corpus, content_transformer(stripWhitespace)) # remove spaces
removeURL = function(x) gsub(" ?(f|ht)tp(s?)://(.*)[.][a-z]+", "", x) # functions to remove URLS
corpus = tm_map(corpus, content_transformer(removeURL))
corpus = tm_map(corpus, removeWords, stopwords("english")) # remove stopwords
corpus = tm_map(corpus, content_transformer(removePunctuation)) # remove punctuation
corpus = tm_map(corpus, content_transformer(removeNumbers)) # remove numbers
# corpus = tm_map(corpus, stemDocument) # remove stemming words

# inspect the first 5 document of the corpus
corpus[[1]]$content[1]
```

3.2 Bag-of-Words

Create a document-term matrix to hold all the tweets. The rows of the matrix correspond to documents (tweets) and the columns of the matrix correspond to words (words making up the tweets). The values in the matrix correspond to the frequency of a word in each document. The N-gram of n = 2 will be taken to form a bigram (2 words that occur together). The bigram matrix will be reduced to show terms with a frequency of >=100.

```
# Create an N-gram function
NLPbigramTokenizer = function(x) {
  unlist(lapply(ngrams(words(x), 2), paste, collapse = " "), use.names = FALSE)
}

# Create a document term matrix -
# rows = documents (or tweets)
# columns = words (in the tweets)
tdm = TermDocumentMatrix(corpus, control = list(tokenize = NLPbigramTokenizer))
```

```
sparse_tdm = removeSparseTerms(tdm, 0.995) # keep terms that appear in 0.5% or more of the tweets

# Identify the bigrams which occur >= 40 times
freq_bigrams = findFreqTerms(sparse_tdm, lowfreq = 100)
freq_bigrams
```

```
##  [1] "americanair flight"    "americanair thanks"
##  [3] "booking problems"      "can get"
##  [5] "can help"              "cancelled flight"
##  [7] "cancelled flighted"    "cancelled flightled"
##  [9] "customer service"      "flight cancelled"
## [11] "flight delayed"        "late flight"
## [13] "late flightr"          "united flight"
## [15] "united thanks"         "usairways americanair"
## [17] "usairways flight"
```

These bigrams were retrieved from the original dataset containing all 14641 positive, negative, and neutral tweets. The most common bigrams in the dataset give clear insight on the issues customers face. For example, occurrences of the bigram “booking problems” give a diagnosis of an issue airliners can attempt to solve. Twitter users often make typos and incorporate slang into their language. Which could explain the various forms of “cancelled flight”.

4. Decision Tree:

4.1 Training/Testing Set

A column of Boolean data called "Negative" was added to the original dataset. For each negative sentiment, the corresponding value of column "Negative" would be true. This will allow us to clearly indicate negative tweets from the others and simplify the input data for the algorithms to come. Split the data into training set and testing set, putting 70% of the data in the training set. Set seed to reserves the same splitting for all preceding algorithms.

```
set.seed(123)
split = sample.split(tweetsSparse$Negative, SplitRatio = 0.7)

trainSparse = subset(tweetsSparse, split == TRUE)
testSparse = subset(tweetsSparse, split == FALSE)
```

4.2 rpart () Decision Tree

The rpart () function takes in the training data set and method class to formulate a classification model known as the decision tree. Terms with higher frequencies in a particular sentiment category of tweets will form a node on the decision tree.

```
tweetCART = rpart(Negative ~ . , data = trainSparse, method = "class", minsplit = 2,
                  minbucket = 1)
prp(tweetCART)
```

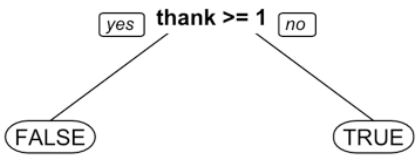


Figure 5.3: Decision Tree for Predicting Negative Sentiment

The diversity of terms making up tweets makes it difficult to find nodes. This could explain why figure 5.3 only has 1 node. The decision tree states that:

- If the term "thank" appears once or more times in a tweet → FALSE (likely a positive or neutral tweet)
- If the term "thank" doesn't appear in a tweet → TRUE (likely a negative tweet)

The logic makes sense intuitively because the term "thank" is a highly non-negative term. But keep in mind that the term "thank" and its variations (thanks, thank you, etc.), are often used to make sarcastic comments such as "Thanks for nothing!"

4.3 Confusion Matrix and Accuracy of Decision Tree

The predict () function takes in the testing data set and the decision tree model developed earlier to yield a confusion matrix. The model proves to have only 68% accuracy, which is fair considering the tree has a single node as a modeling factor.

```
predictCART = predict(tweetCART, newdata = testSparse, type = "class")

# find the confusion matrix for our predictions
cmat_CART = table(testSparse$Negative, predictCART)
cmat_CART
```

```
##        predictCART
##         FALSE TRUE
##   FALSE   366 1273
##   TRUE    120 2633
```

```
accu_CART = (cmat_CART[1,1] + cmat_CART[2,2])/sum(cmat_CART)
accu_CART
```

```
## [1] 0.6828324
```

5. Random Forest:

5.1 Training the Algorithm

R Studio has a built-in randomForest function which takes in the training dataset to yield a classification model. A call on the model showed that there were 500 trees and 18 variables at each split.

```
set.seed(123)
tweetRF = randomForest(Negative ~ . , data = trainSparse)
tweetRF
```

5.2 Confusion Matrix and Accuracy of Random Forest

The predict () function will test the accuracy of the randomForest function using the testing dataset. The model proves to have 79.5% accuracy, roughly 10% higher than that of decision tree. This makes sense because random forest is essentially a collection of decision trees and more trees imply more accuracy.

```
predictRF = predict(tweetRF, newdata = testSparse)
cmat_RF = table(testSparse$Negative, predictRF)
cmat_RF
```

```
##        predictRF
##         FALSE TRUE
##   FALSE  1123  516
##   TRUE    382 2371
```

```
accu_RF = (cmat_RF[1,1] + cmat_RF[2,2])/sum(cmat_RF)
accu_RF
```

```
## [1] 0.7955373
```

6. Logistic Regression:

6.1 Training the Algorithm

R Studio has a built-in Logistic Regression function called "glm" which takes in the training dataset to yield a classification model. The family parameter has been set to binomial because a tweet sentiment can either be negative (TRUE) or other (FALSE).

```
tweetLog = glm(Negative ~ . , data = trainSparse, family = "binomial")
```

6.2 Confusion Matrix and Accuracy of Logistic Regression

The predict () function will test the accuracy of the regression function using the testing dataset. The model proves to have 79.9% accuracy, about the same as Random Forest.

```
predictLog = predict(tweetLog, type = "response", newdata = testSparse)
cmat_Log = table(testSparse$Negative, predictLog > 0.5)
cmat_Log
```

```
##
##         FALSE TRUE
##   FALSE  1205  434
##   TRUE    445 2308
```

```
accu_Log = (cmat_Log[1,1] + cmat_Log[2,2])/sum(cmat_Log)
accu_Log
```

```
## [1] 0.7998634
```

6. Conclusion

From the in-depth analysis above, we can conclude that all 6 airline companies suffer from customer dissatisfaction. The hashtag analysis showed that the top 6 hashtags included phrase such as "#fail", "#disappointed", "#customerservice", which further support our conclusion. Bigrams in comparison provided more insight on the specific reasons leading up to customer dissatisfaction. The top 17 bigrams included "booking problems", "cancelled flight", and "late flight".

Finding a functional method for text analytics was also another initiative for this project. The scoring method proved to be ineffective at classifying tweets accurately. A potential explanation for this could be limitation of words in the dictionaries and lack of understanding of context. The human language is complex and teaching an algorithm to analyze the various grammatical nuances, cultural variations, slang and misspelling that occur on Twitter is difficult. Teaching an algorithm to understand tone is even more difficult. Consider the following tweet: "My flight's been delayed. Brilliant!" An algorithm would see the word "brilliant" and classify the tweet as positive. But to a human, that's clearly sarcasm.

The inefficiency of the scoring method further motivates us to search for an effective machine-learning algorithm. Algorithms were trained and tested using a confusion matrix to evaluate accuracy of classification. Logistic Regression and Random Forest both performed well with accuracy rates of

79.98% and 79.55% respectively. Followed by Decision Tree, which had an accuracy rate of 68.28%. Hence, we can conclude that Logistic Regression and Random Forest both out perform tweet classification in comparison to the scoring method and Decision Tree. Although roughly 20% of tweets are misread, we now have the ability to classify a large volume of airline related tweets in minimal time. It clearly pays-off manual classification!

7. Literature Review

The below papers were reviewed for my analysis.

- *An Approach to Sentiment Analysis – The Case of Airline Quality Rating* by Adeborna & Siau, discusses the sentimental analysis approach to measuring the quality rating of three major airline companies; AirTran Airways, Frontier and SkyWest Airlines. The researchers used Twitter as a source to determine customer sentiments towards the airline companies.
  Source:
  https://pdfs.semanticscholar.org/dc8e/272b9f56b935b926603bdf42eb033c6e94a3.pdf

- In the paper *Sentiment Analysis of Standford Course Review*, Walsh evaluates student reviews on Standford courses. Walsh experiments with different classification algorithms, including Naïve Bayes, to identify optimal classifier for determining the sentiment of a course review.
  Source: https://nlp.stanford.edu/courses/cs224n/2010/reports/rjwalsh.pdf

- *Learning Emotion Indicators from Tweets: Hashtags, Hashtag Patterns, and Phrases* by Qadir & Riloff presents a framework for learning emotions from Twitter hashtags. The research provides techniques to handle generic hashtags combining multiple emotions.
  Source: http://www.cs.utah.edu/~riloff/pdfs/emnlp14-hashtags.pdf

8. References (APA format)

1. Breen, J. (2011, August 15). Slides from my R tutorial on Twitter text mining #rstats. Retrieved August 3, 2018, from https://datamatters.blog/2011/07/04/twitter-text-mining-r-slides/

2. Fossati, G. (n.d.). The Analytics Edge - Unit 5 : Turning Tweets into Knowledge. Retrieved August 3, 2018, from https://rstudio-pubs-static.s3.amazonaws.com/72664_2b1ee5cd10d447e4af50a2b68ceed428.html

3. Larson, B. (2016, December 20). R: Twitter Sentiment Analysis. Retrieved August 3, 2018, from https://analytics4all.org/2016/11/25/r-twitter-sentiment-analysis/

4. Wang, C. (2016, January 10). Sentiment analysis with machine learning in R. Retrieved August 3, 2018, from https://datascienceplus.com/sentiment-analysis-with-machine-learning-in-r/