University of Oklahoma

Data Science and Analytics

DSA 5103 – Intelligent Data Analytics

SENTIMENT ANALYSIS OF AIRLINE TWEETS

Joe Trivisonno

Fall 2016

Contents

1. Executive Summary

In a competitive industry such as airline travel, companies are looking for any way to gain an advantage over their competition. One way to do this is to address common complaints that customers may have and improve on them, and Twitter can be a valuable source for discovering what these complaints are. While no company can keep every customer happy, monitoring sentiment across social media such as Twitter can help a company identify concerns and improve their customer experience. For this analysis, tweets about major US airlines from February 2015 were examined, and three models were built to classify tweets as either "negative," "neutral," or "positive." These models were a Naïve Bayes classifier, a Random Forest classifier, and a Support Vector Machine (SVM) classifier.

The dataset consisted of 14,640 rows with 15 variables. The sentiment, negative reason, airline, and text of the tweet were looked at in this analysis. Approximately 62% of the tweets in the data set were classified as negative, while just over 16% of the tweets were classified as positive. Exploratory analysis revealed that United was the airline that was most tweeted about in the data set, however, American and US Airways have since merged, and their combined total number of tweets was greater than the number of tweets about United. It was noted that airlines that are tweeted about less often also have a lower proportion of negative tweets. The reason for negative tweets was also examined and it was determined that customer service issues and late flights were the most common complaints.

A Naïve Bayes, Random Forest, and Support Vector Machine classifier were trained and their performances were compared. The Naïve Bayes model had only a 55.57% accuracy, making it worse than the no information rate of classifying every tweet as negative. The major issue with the Naïve Bayes classifier was that it significantly overclassified tweets as positive. Since correctly identifying negative tweets is more important than identifying positive tweets, this is extremely problematic. One positive of this was that very few tweets that were classified as negative were classified incorrectly. Both the Random Forest and SVM models achieved greater than 70% accuracy, and both models had comparable performance measure across "negative," "neutral," and "positive" tweets. Both models correctly identified over 85% of negative tweets while keeping tweets incorrectly classified as negative to an acceptable level.

Based on these findings, it was determined that both Random Forest and SVM classifiers are well suited to sentiment classification of tweets, however no conclusion could be made as to which model was superior. Future work could include further tuning of the model parameters as well as the sparsity that is acceptable in the document term matrix. Other models, such as Neural Networks could also be examined. An alternate approach to this analysis could be to classify tweets as either "negative" or "not negative" in order simplify the model to a binary classification.

2. Problem Description

Twitter can be a valuable source of information for companies to determine what they are doing well, and where they can improve in the eyes of their customers. While no company can keep every customer happy, tracking sentiment across social media such as Twitter can help a company adapt and improve their customer experience. This is especially important in the airline industry where the competition is fierce, and customers expect a high quality experience due to the cost of traveling by air. It is essential for airlines to identify common complaints quickly, and sentiment analysis of tweets is one way to do that. Sentiment analysis has been used in many other industries, to great effect. One example is the use of internet search data to recommend products by advertisers [3]. For this analysis, I examined tweets about major US airlines from February 2015. The data was provided by the user Crowdflower on Kaggle [2].

The objective of this analysis was to build a model to predict whether a tweet is positive, negative, or neutral. To do this, I built three different classification models and compared their performance. These models were Naïve Bayes, Random Forest, and Support Vector Machine. Such models are valuable to airlines as they allow them to easily identify negative (or positive) tweets, which can aid in quickly identifying and resolving any issues.

3. Exploratory Data Analysis

The dataset contains 14,640 observations with 15 variables, including sentiment, reason (if negative sentiment), airline, the text of the tweet, date, and location. For the purpose of this analysis, we will look at sentiment, negative reason, airline, and the text of the tweet. The data set contains 9,178 negative tweets, 3,099 neutral tweets, and 2,363 positive tweets. Figure 1 shows a breakdown of the number of tweets and the sentiment by airline.

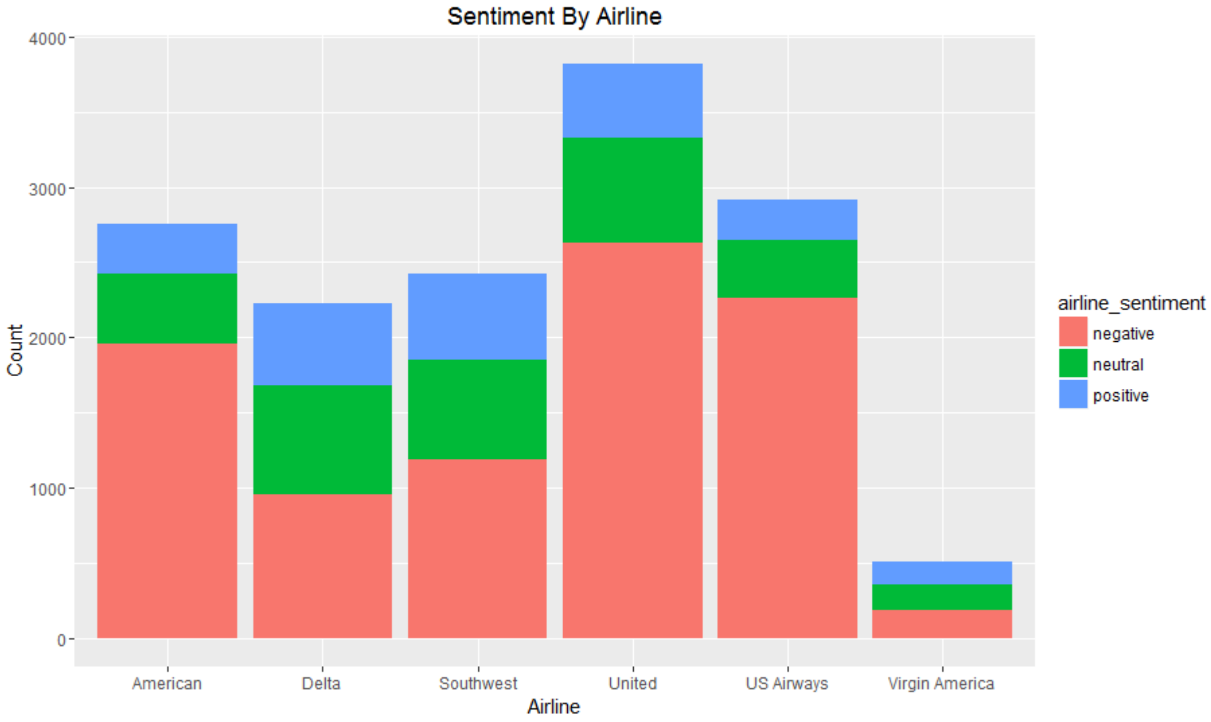


Figure 1: Number and sentiment of tweets by airline

The airline tweeted about most often in the data set is United Airlines. However, the second and third most commonly tweeted airlines, American and US Airways, merged in 2015 under the American Airlines brand [4]. The combined number of tweets about these two airlines is far greater than the number of tweets about United. Virgin America is the airline tweeted about least often by a significant margin, which is expected since it is by far the smallest airline in the data set. It is interesting that the airlines that are tweeted about least often, Delta, Southwest, and Virgin America, also have the lowest proportion of negative tweets.

Insight can also be gained from looking at the reason for negative tweets. It is expected that these words will be common in negative tweets, making them good indicators of a negative tweet for our machine learning algorithms. Figure 2 shows the distribution of reasons for a negative tweet for the entire data set. The distribution of reasons for a negative tweet for each airline can be found in Appendix A.

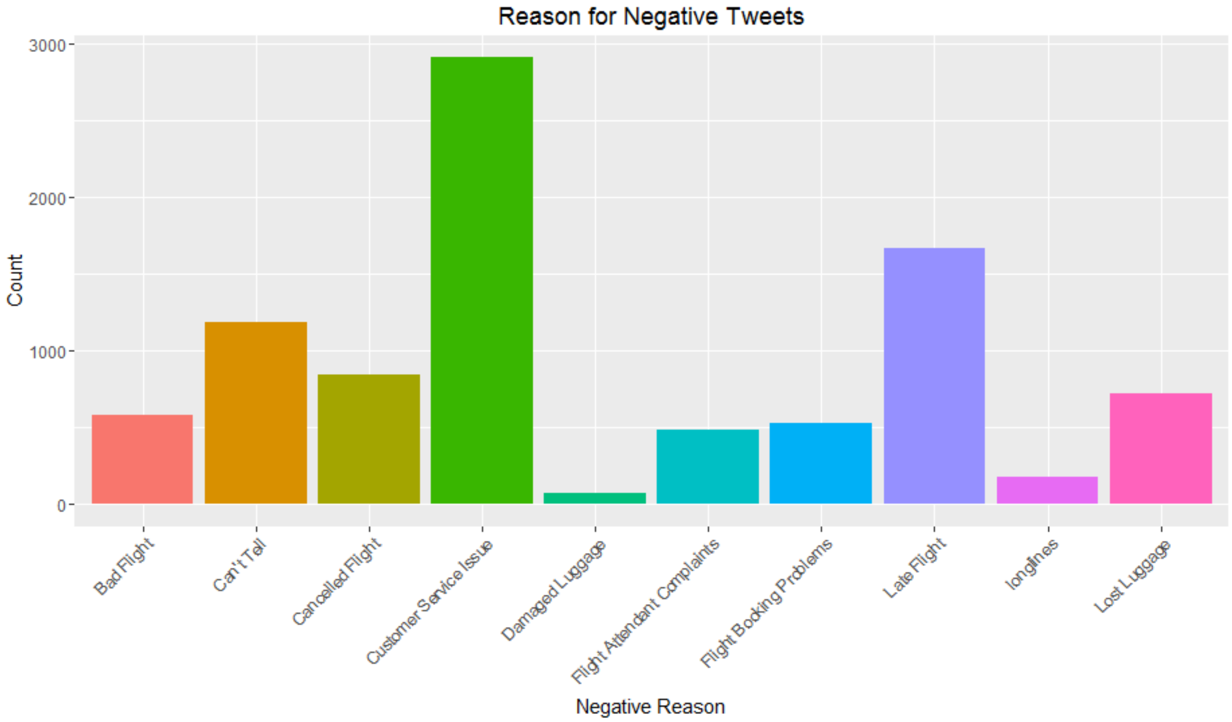


Figure 2: Distribution of the reason for negative tweets in the dataset

From Figure 2, we see that customer service issues are by far the leading reason for negative tweets, with late flights also being very popular. Other common issues are lost luggage and cancelled flights. While late or cancelled flights are often not preventable, customer service issues and lost luggage are. These are two areas that airlines could focus on to improve customer experience. Damaged luggage and long lines are the least common reasons for negative tweets. With the exception of Delta, all of the airlines individually had the same trends as the combined data. The most common complaint about Delta was late flights, with customer service issues being the next most common.

Next, words that appear frequently in both positive and negative tweets were examined. This will aid in visualizing words that will be useful in determining the sentiment of a tweet. Before doing this analysis however, stop words and other common words that do provide any insight such as “the,” “you,” and “for” were removed from the tweets. Twitter handles were also removed as we have already seen which airlines are tweeted about frequently, and in what context. Table 1 shows the most common words in both positive and negative tweets.

Table 1: Most common words in positive and negative tweets

Frequent Words in Positive Tweets					
thanks	thank	flight	great	service	love
608	453	371	233	159	132
Frequent Words in Negative Tweets					
flight	cancelled	service	hours	help	hold
2900	920	740	644	610	607

We can see that “flight” and “service” unsurprisingly appear frequently in both positive and negative tweets, so they are not likely to be useful in determining the sentiment of a tweet. Other positive words include “thank,” “great,” and “love,” while negative words include “cancelled,” “hours,” and “hold.” These common negative words are consistent with the reasons for negative tweets discussed earlier.

In order to get a better understanding of words that appear in positive and negative tweets, two word clouds were created. One word cloud contains words found in positive tweets, while the other contains words found in negative tweets. The two word clouds are shown in Figure 3.

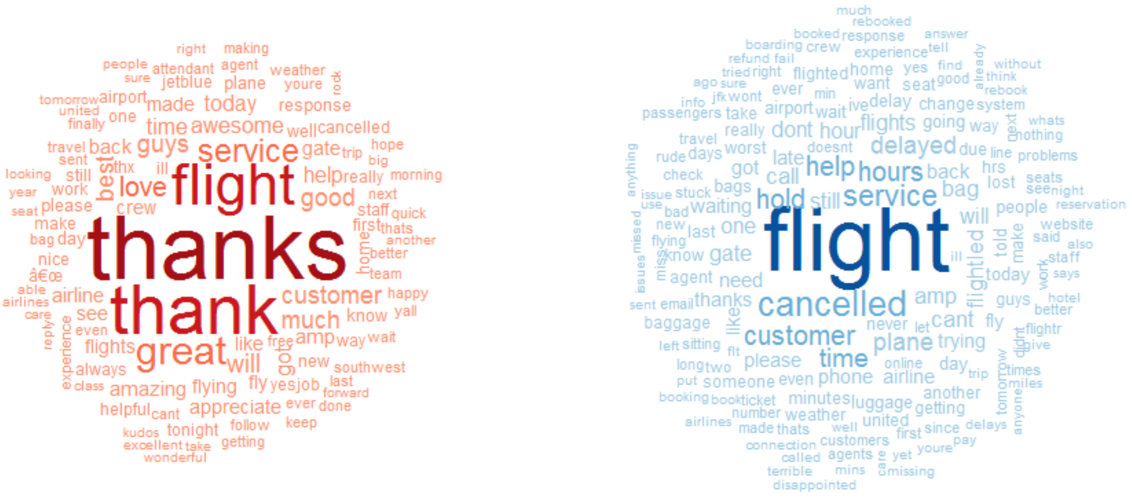


Figure 3: Word clouds showing words found in positive tweets (red) and negative tweets (blue)

From these word clouds, we can see that other words associated with positive tweets include “appreciate,” “amazing,” and “awesome,” while words associated with negative tweets include “delayed,” “waiting,” and “time.” The presence of these words will likely be a good indicator of the sentiment of a tweet.

4. Analysis Plan

4.1 Model Selection

For this analysis, I elected to build three classification models: Naïve Bayes, Random Forest, and Support Vector Machine (SVM). Naïve Bayes was chosen due to its simplicity. It will be used primarily as a baseline with which to judge the two more complex models. Random Forest and SVM models were chosen because of their performance in high dimensional space. This makes them ideally suited for text classification. Random Forests also have the benefit of being very easy to tune, making them very popular. On the other hand, SVM models can be extremely complicated to tune, so while they have the potential to be very powerful, they are hard to get right. Both Random Forests and SVM also take significantly longer to train than simpler models such as Naïve Bayes. [1]

4.2 Data Transformations

In order to classify text data, it first needs to be put through a number of transformations. First twitter handles were removed. Since they occur in every tweet, regardless of sentiment, they would not be helpful in determining sentiment. The data was then input into a function that was created for this analysis and transformed into a document term matrix. This function created a corpus from the text of the tweets. It then removed all punctuation and transformed all text to lower case. This was done to ensure different capitalizations of words were not viewed as different words in the corpus. The function then filtered stop words, which are common words with little significance, from the corpus. These words will not help in determining sentiment, so they do not need to be included in the document term matrix. Finally, a document term matrix was created from the corpus, and sparse terms were removed. By removing sparse terms from the matrix, the number of factors was reduced from over 16,000 to 153. This document term matrix, along with the sentiment for each tweet, was used to build the models.

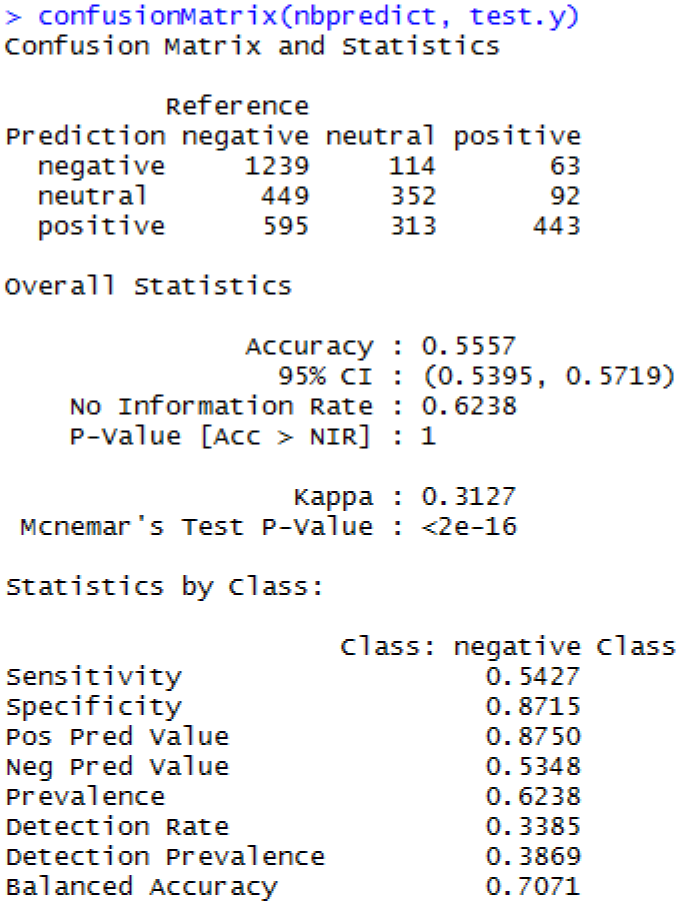
4.3 Validation Plan

To validate the model, the document term matrix was broken into a training and a test set. The training set contained 10,980 tweets, 75% of the data set. The test set contained the remaining tweets. The Naïve Bayes, Random Forest, and SVM models were trained on the training set, and 5-fold cross validation was performed on the Random Forest and SVM models. Since the Naïve Bayes model is simply being as a baseline for the other two models, cross validation was not run on it.

The trained classifiers were then run on the test set and their performances were evaluated. Evaluation measures for each model were accuracy, sensitivity, specificity, kappa statistic, and the positive predictive value. Heavy emphasis will be placed on these values for the negative class, since identifying negative tweets will be more useful for airlines. The models were also compared to the no information rate. Since the classifiers were non-binary, the area under the ROC curve was not used to evaluate performance.

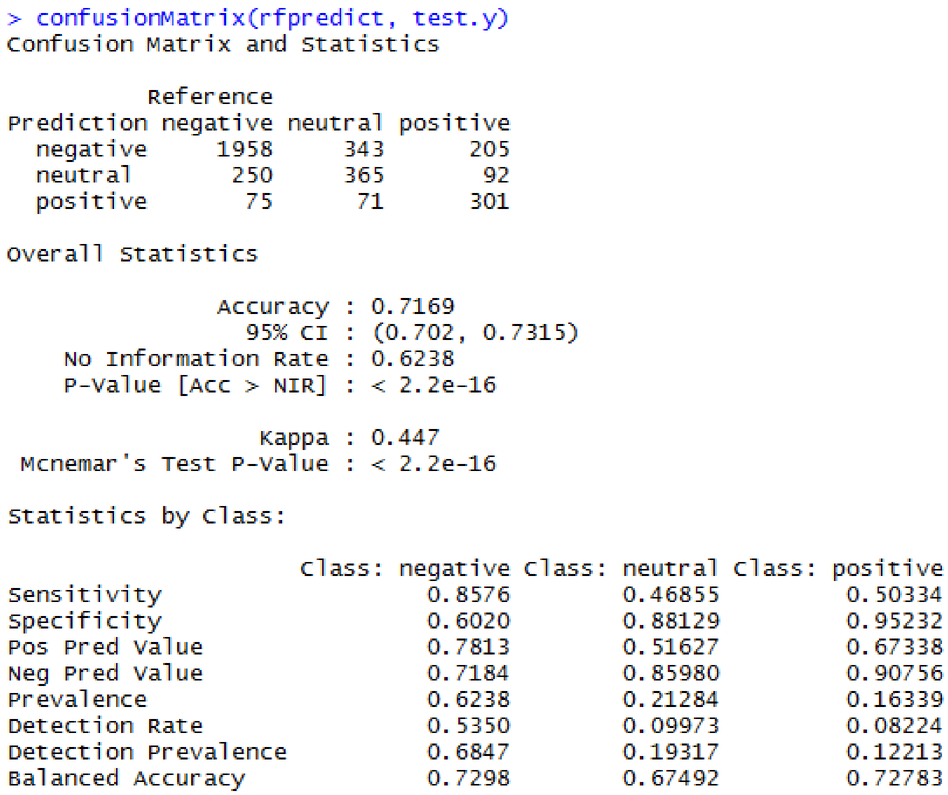
5. Results and Validation

The Naïve Bayes, Random Forest, and SVM classifiers were trained and the results were compared. A 5-Fold cross validation was then performed on the Random Forest and SVM classifiers. The Random Forest classifier had a mean accuracy of 0.7957, and the SVM classifier had a mean accuracy of 0.7284. Both of these values are well above the no information rate of 0.6238, indicating that they perform better than random. The Random Forest had significantly better accuracy during this phase, indicating that it may be a better classifier for this data than the SVM. Before making any conclusions however, we must look at the model performances on the test data. The confusion matrix and test statistics for the Naïve Bayes, Random Forest, and SVM classifiers performance on the test data can be seen in Figures 4, 5 and 6, respectively.



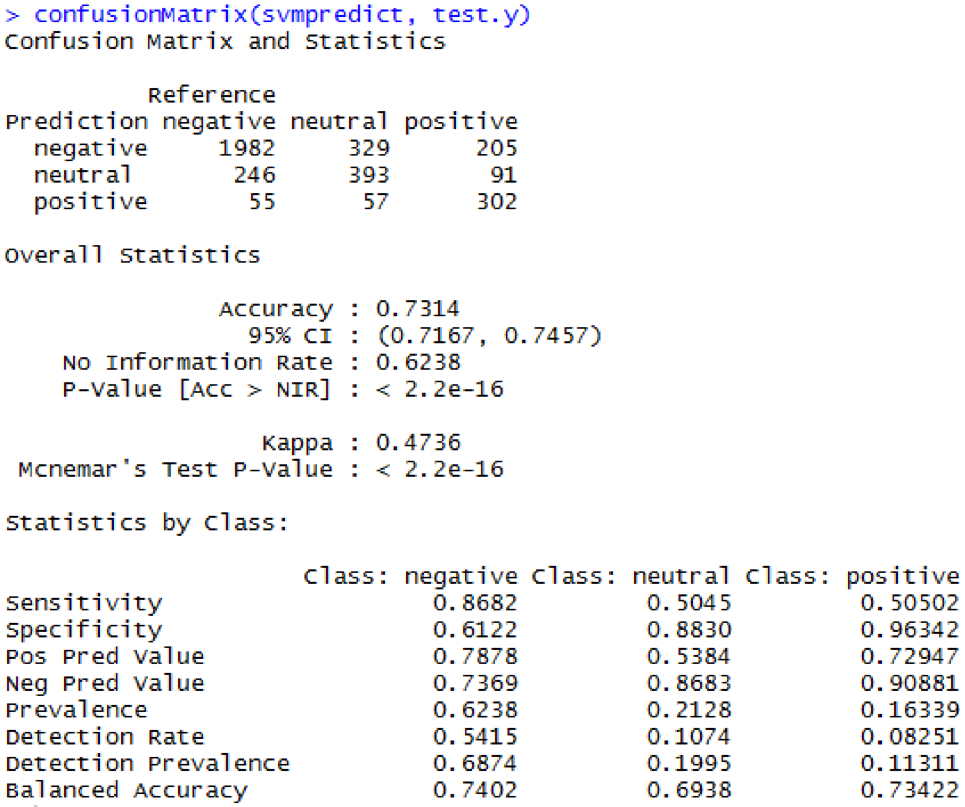
```
> confusionMatrix(nbpredict, test.y)
Confusion Matrix and Statistics

          Reference
Prediction negative neutral positive
  negative     1239     114       63
  neutral       449     352       92
  positive      595     313      443

Overall Statistics

               Accuracy : 0.5557
                 95% CI : (0.5395, 0.5719)
    No Information Rate : 0.6238
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3127
 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: negative Class: neutral Class: positive
Sensitivity                   0.5427        0.45186          0.7408
Specificity                   0.8715        0.81222          0.7035
Pos Pred Value                0.8750        0.39418          0.3279
Neg Pred Value                0.5348        0.84568          0.9329
Prevalence                    0.6238        0.21284          0.1634
Detection Rate                0.3385        0.09617          0.1210
Detection Prevalence          0.3869        0.24399          0.3691
Balanced Accuracy             0.7071        0.63204          0.7221
```

Figure 4: Confusion Matrix and Test Statistics for the Naïve Bayes Classifier on the test data



```
> confusionMatrix(rfpredict, test.y)
Confusion Matrix and Statistics

          Reference
Prediction negative neutral positive
  negative     1958     343      205
  neutral       250     365       92
  positive       75      71      301

Overall Statistics

               Accuracy : 0.7169
                 95% CI : (0.702, 0.7315)
    No Information Rate : 0.6238
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.447
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: negative Class: neutral Class: positive
Sensitivity                   0.8576        0.46855         0.50334
Specificity                   0.6020        0.88129         0.95232
Pos Pred Value                0.7813        0.51627         0.67338
Neg Pred Value                0.7184        0.85980         0.90756
Prevalence                    0.6238        0.21284         0.16339
Detection Rate                0.5350        0.09973         0.08224
Detection Prevalence          0.6847        0.19317         0.12213
Balanced Accuracy             0.7298        0.67492         0.72783
```

Figure 5: Confusion Matrix and Test Statistics for the Random Forest Classifier on the test data



```
> confusionMatrix(svmpredict, test.y)
Confusion Matrix and Statistics

          Reference
Prediction negative neutral positive
  negative     1982     329      205
  neutral       246     393       91
  positive       55      57      302

Overall Statistics

               Accuracy : 0.7314
                 95% CI : (0.7167, 0.7457)
    No Information Rate : 0.6238
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4736
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: negative Class: neutral Class: positive
Sensitivity                   0.8682         0.5045         0.50502
Specificity                   0.6122         0.8830         0.96342
Pos Pred Value                0.7878         0.5384         0.72947
Neg Pred Value                0.7369         0.8683         0.90881
Prevalence                    0.6238         0.2128         0.16339
Detection Rate                0.5415         0.1074         0.08251
Detection Prevalence          0.6874         0.1995         0.11311
Balanced Accuracy             0.7402         0.6938         0.73422
```

Figure 6: Confusion Matrix and Test Statistics for the Support Vector Machine Classifier on the test data

The Naïve Bayes classifier had an overall accuracy of just 0.5557. This is less than the no information rate of 0.6238, meaning this is a lower accuracy than could be achieved by classifying every tweet as negative. By looking at the confusion matrix, we see that the Naïve Bayes classifier heavily overclassified tweets as positive. This is particularly problematic, since it is likely that most airlines would desire a high negative sensitivity. This classifier also had a very low Kappa statistic of 0.3127. One benefit to the Naïve Bayes classifier is that it does have a high positive predictive value for the negative class of 0.8750, meaning there are very few false positives among tweets classified as negative. Now let us compare these results to the performance metrics for the Random Forest and SVM classifiers.

Both the Random Forest and the SVM classifiers performed significantly better than the Naïve Bayes classifier, particularly at negative classification. The Random Forest classifier had an overall accuracy of 0.7169, while the SVM classifier had an accuracy of 0.7314. It is noteworthy, that the Random Forest classifier performed significantly worse on the test set than during any of the folds during cross validation, while the SVM performance was similar to its accuracy during cross validation of the training set. Both the Random Forest and the SVM classifiers were significantly better at classifying negative tweets than the Naïve Bayes classifier. The Random Forest classifier had a sensitivity of 0.8576 and a positive predictive value of 0.7813 for negative tweets, while the SVM classifier had a sensitivity of 0.8682 and a positive predictive value of 0.7878. This shows that both classifiers perform well at identifying negative tweets, while limiting false positives. Neither performs well at identifying neutral or positive tweets. The Random Forest classifier has a sensitivity of 0.46855 for neutral tweets and 0.50334 for positive tweets. The SVM classifier has a sensitivity of 0.5045 for neutral tweets and 0.50502 for positive tweets. In addition to correctly classifying only about half of neutral and positive tweets, both classifiers have extremely high rates of false positives for neutral tweets. The Random Forest classifier has a positive predictive value of just 0.51627 for neutral tweets, and the SVM classifier has positive predictive value of 0.5384.

6. Conclusion

Both the Random Forest and SVM classification models performed significantly better than the Naïve Bayes model as well as the no information rate. While the SVM classifier had a slightly higher accuracy than the Random Forest model on the test, there was not enough of a difference to conclusively state that it is a better model than the Random Forest classifier. The difference in accuracy and other performance metrics were close enough that any differences could easily be caused by the distribution of the training and test set. This is further supported by the large difference in performance of the Random Forest model between the 5-Fold cross validation of the training set and the test set. Based on this analysis, we conclude that both Random Forest and Support Vector Machine classifiers have merit in sentiment classification of tweets, and there is not a significant difference between their performances.

7. Future Work

In the future, I would like to explore further tuning of the Support Vector Machine model. Unlike the Random Forest Classifier, SVM models require significant tuning of their hyper parameters to build the best model. I suspect that the performance of the SVM classifier can be improved with further tuning of these hyper parameters. I would also like to explore how varying the amount of sparsity that is acceptable in the document term matrix would affect the performance of these models. In addition, I would like to explore other classification models, such as Neural Networks, to see if they provide any improvement over Random Forests and Support Vector Machines.

Since most airlines are likely to be most concerned about negative tweets, an alternative approach to this analysis would be to classify the tweets as either “negative” or “not negative” rather than “negative,” “neutral,” or “positive.” This would make this a binary classification problem, which could simplify the problem and make it easier to tune and evaluate the models.

8. Appendices

Appendix A: Additional Figures

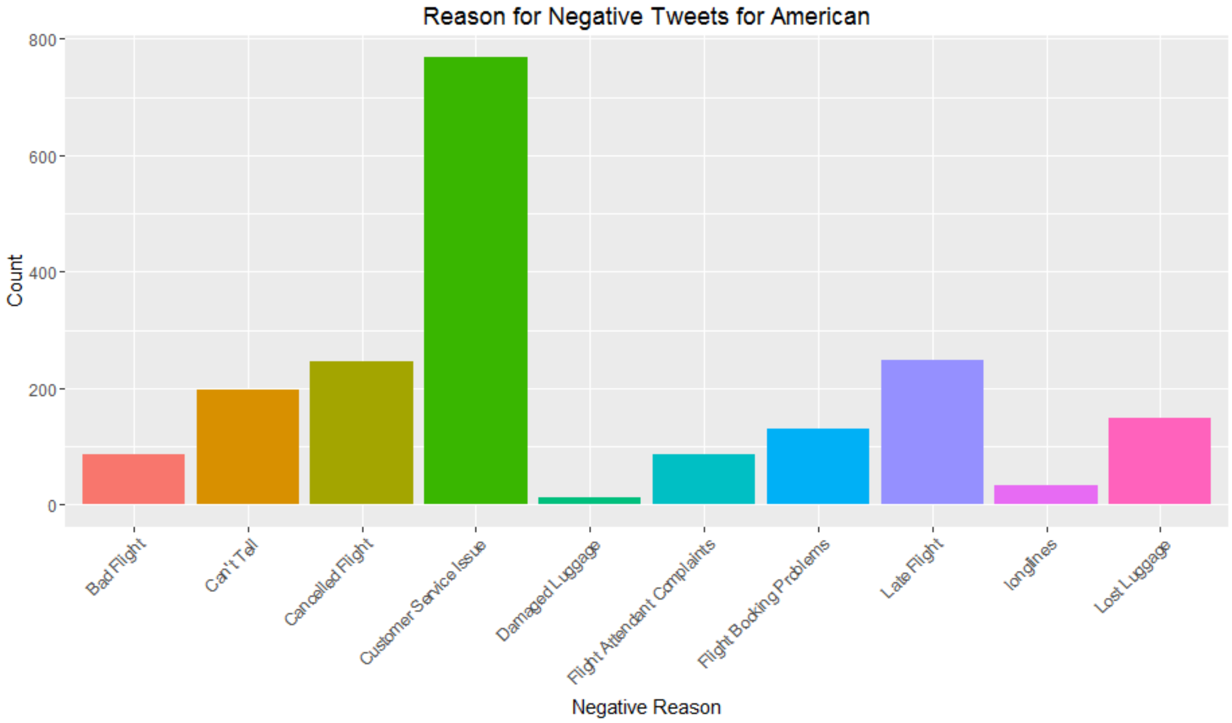


Figure A.1: Reasons for negative tweets about American

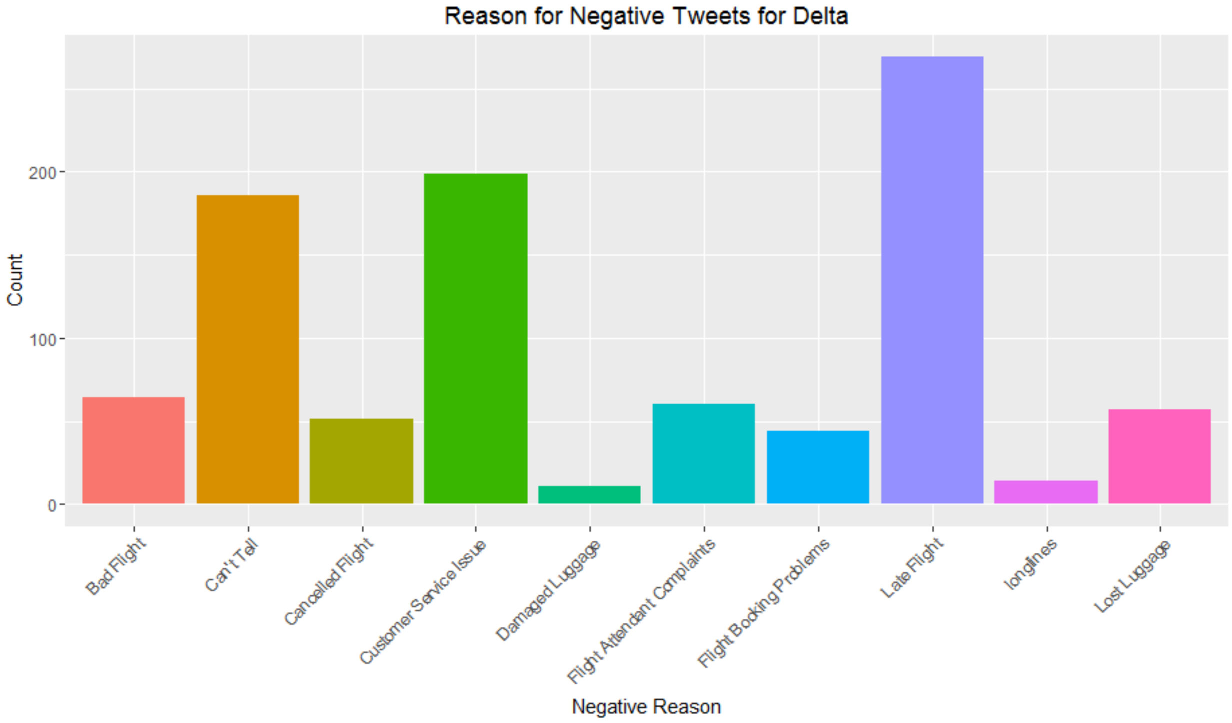


Figure A.2: Reasons for negative tweets about Delta

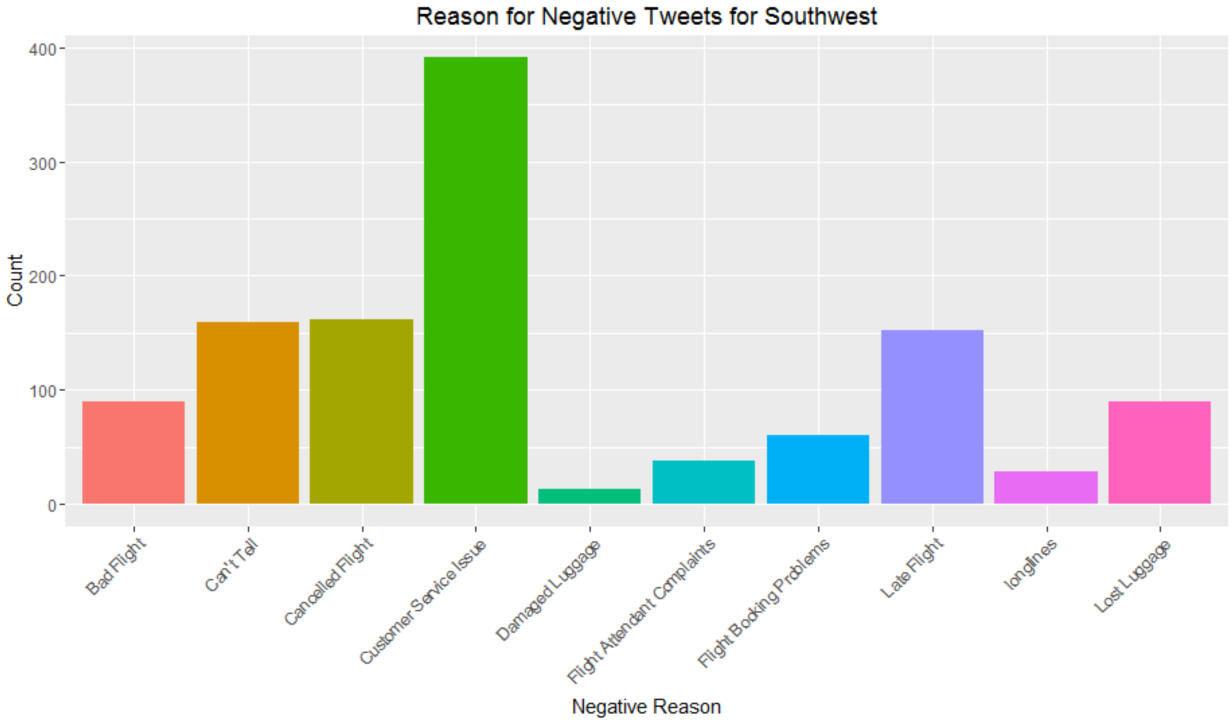


Figure A.3: Reasons for negative tweets about Southwest

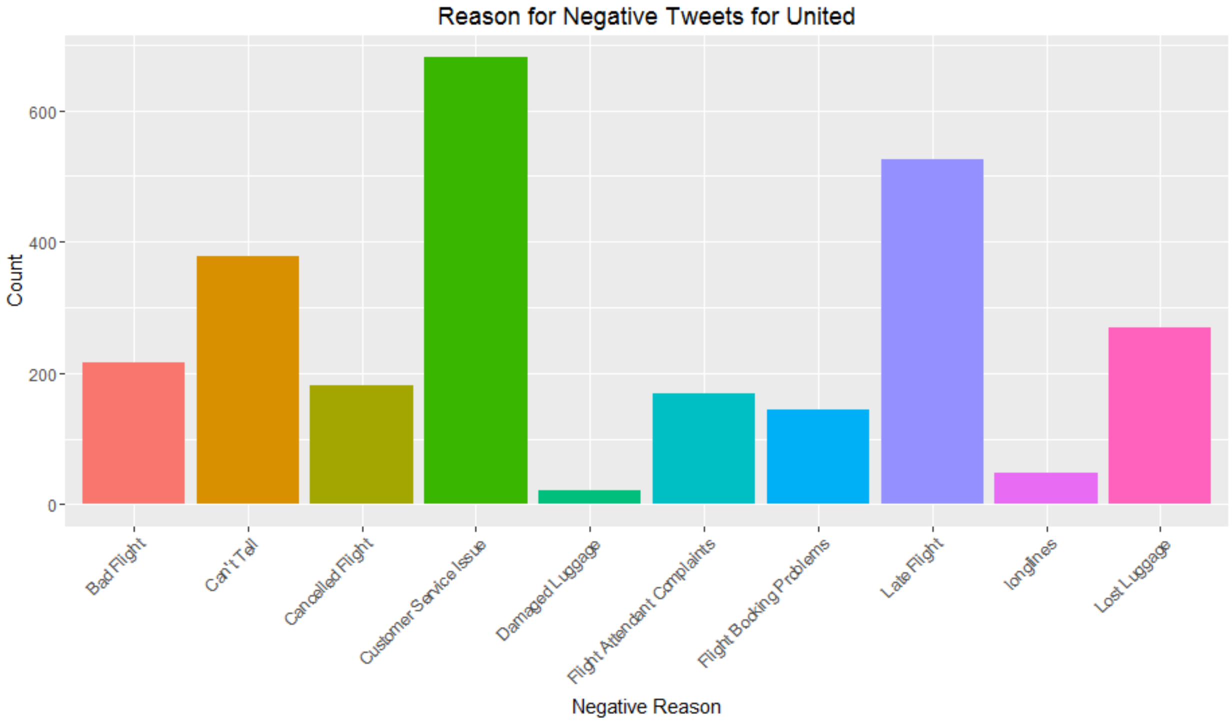


Figure A.4: Reasons for negative tweets about United

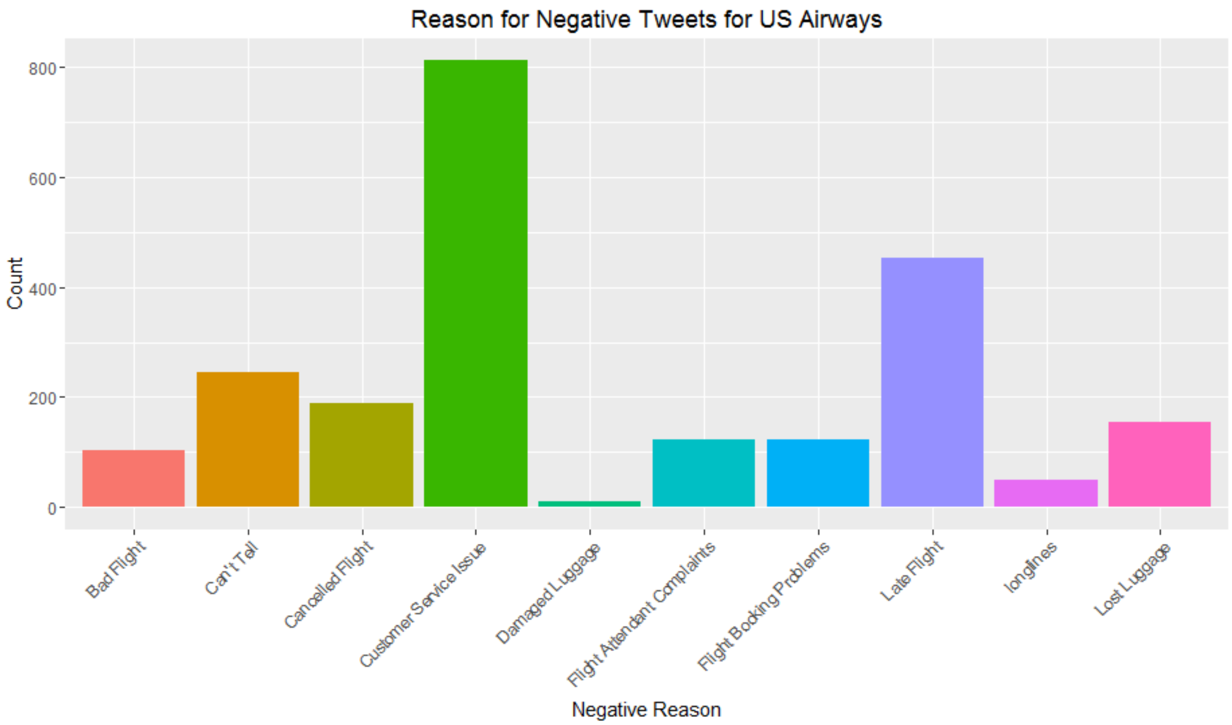


Figure A.5: Reasons for negative tweets about US Airways

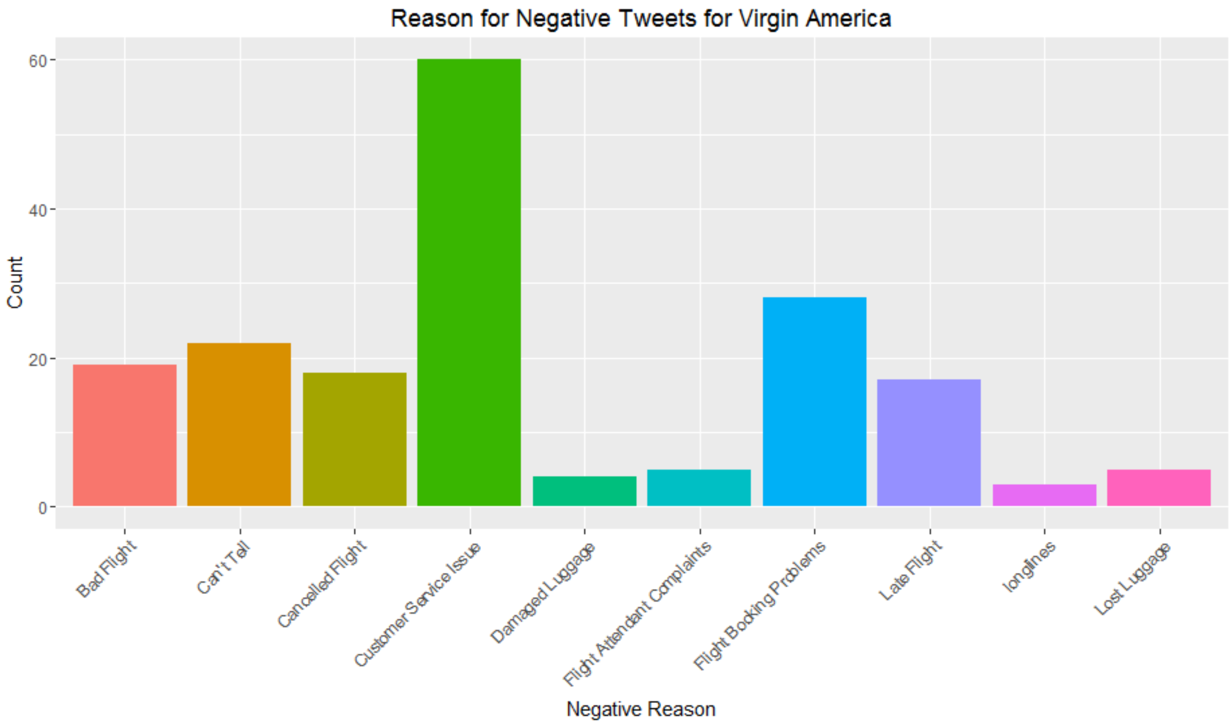


Figure A.6: Reasons for negative tweets about Virgin Atlantic

Appendix B: R Source Code

```
library(ggplot2)
library(RTextTools)
library(tm)
library(e1071)
library(wordcloud)
library(caret)
library(randomForest)
library(maxent)
library(ROCR)

tweets <- read.csv("Tweets.csv")#read csv
table(tweets$airline_sentiment)#breakdown of sentiment

#bar plot of sentiment by airline
ggplot(data = tweets) + geom_bar(aes(x = airline, fill = airline_sentiment),
stat = "count") +
  labs(x = "Airline", y = "Count", title = "Sentiment By Airline")

neg_tweets <- tweets[tweets$airline_sentiment == "negative",]#negative tweets
pos_tweets <- tweets[tweets$airline_sentiment == "positive",]#positive tweets
neut_tweets <- tweets[tweets$airline_sentiment == "neutral",]#neutral tweets

#bar plot of negative reasons across all airlines
ggplot(data = neg_tweets) + geom_bar(aes(x = negativereason, fill =
negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

#bar plot of negative reasons for American
ggplot(data = subset(neg_tweets, neg_tweets$airline == "American")) +
geom_bar(aes(x = negativereason, fill = negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets for American") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

#bar plot of negative reasons for Delta
ggplot(data = subset(neg_tweets, neg_tweets$airline == "Delta")) +
geom_bar(aes(x = negativereason, fill = negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets for Delta") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

#bar plot of negative reasons for Southwest
ggplot(data = subset(neg_tweets, neg_tweets$airline == "Southwest")) +
geom_bar(aes(x = negativereason, fill = negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets for Southwest") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

#bar plot of negative reasons for United
```

```
ggplot(data = subset(neg_tweets, neg_tweets$airline == "United")) +
geom_bar(aes(x = negativereason, fill = negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets for United") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

#bar plot of negative reasons for US Airways
ggplot(data = subset(neg_tweets, neg_tweets$airline == "US Airways")) +
geom_bar(aes(x = negativereason, fill = negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets for US Airways") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

#bar plot of negative reasons for Virgin America
ggplot(data = subset(neg_tweets, neg_tweets$airline == "Virgin America")) +
geom_bar(aes(x = negativereason, fill = negativereason), stat = "count") +
  labs(x = "Negative Reason", y = "Count", title = "Reason for Negative
Tweets for Virgin America") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
guides(fill=FALSE)

tweets_only <- subset(tweets, select = c(airline_sentiment, text))#data frame
of only sentiment and tweet text

tweets_only2 <- tweets_only
tweets_only2$text <- gsub("@\\w+", "", tweets_only$text)#remove twitter
handles from tweet text
rmWords <- c("the", "you", "for", "and", "get", "can", "now", "just")#list of
words to remove

#function to create document term matrix
dtm = function(rawTweets){
  #rawTweets: list of tweets to be transformed
  createCorpus <- Corpus(VectorSource(rawTweets))#create corpus out of
rawTweets
  createCorpus <- tm_map(createCorpus, content_transformer(tolower))
#transform all text to lowercase
  createCorpus <- tm_map(createCorpus, removePunctuation) #remove puntcuation
from tweets
  createCorpus <- tm_map(createCorpus, removeWords,
stopwords("english"))#remove stopwords from corpus
  createCorpus <- tm_map(createCorpus, removeWords, rmWords)#remove
additional words from corpus
  createCorpus <- DocumentTermMatrix(createCorpus)#create document term
matrix
  createCorpus <- removeSparseTerms(createCorpus, 0.99)#remove sparce terms
from matrix

  return(createCorpus)#return document term matrix
}

par(mfrow=c(1,2))
pos_matrix <- dtm(tweets_only2$text[tweets_only2$airline_sentiment ==
"positive"])#document term matrix of positive tweets
```

```
pos_matrix <- as.data.frame(as.matrix(pos_matrix))#transform matrix to data
frame

pos_freq_words <- colSums(pos_matrix)#count of word frequency

pos_freq_words <- pos_freq_words[order(pos_freq_words, decreasing = T)]#order
words by frequency
head(pos_freq_words)#6 most frequent words
#create word cloud of words from positive tweets
wordcloud(freq = as.vector(pos_freq_words), words =
names(pos_freq_words),random.order = FALSE,
          random.color = FALSE, colors = brewer.pal(9, 'Reds')[4:8])

neg_matrix <- dtm(tweets_only2$text[tweets_only2$airline_sentiment ==
"negative"])#document term matrix of negative tweets
neg_matrix <- as.data.frame(as.matrix(neg_matrix))#transform matrix to data
frame

neg_freq_words <- colSums(neg_matrix)#count of word frequency

neg_freq_words <- neg_freq_words[order(neg_freq_words, decreasing = T)]#order
words by frequency
head(neg_freq_words)#6 most frequent words
#create word cloud of words from negative tweets
wordcloud(freq = as.vector(neg_freq_words), words =
names(neg_freq_words),random.order = FALSE,
          random.color = FALSE, colors = brewer.pal(9, 'Blues')[4:8])


par(mfrow=c(1,1))
#function to create document term matrix without the words in rmWords removed
dtmFull = function(rawTweets){
  #rawTweets: list of tweets to be transformed
  createCorpus <- Corpus(VectorSource(rawTweets))#create corpus out of
rawTweets
  createCorpus <- tm_map(createCorpus, content_transformer(tolower))
#transform all text to lowercase
  createCorpus <- tm_map(createCorpus, removePunctuation) #remove puntcuation
from tweets
  createCorpus <- tm_map(createCorpus, removeWords, stopwords("english"))
  createCorpus <- DocumentTermMatrix(createCorpus)#create document term
matrix
  createCorpus <- removeSparseTerms(createCorpus, 0.99)#remove sparce terms
from matrix

  return(createCorpus)#return document term matrix
}

matrix <- dtmFull(tweets_only2$text)#create document term matrix for all
tweets

mat <- as.matrix(matrix)#transform to matrix

set.seed(123)#make the train set selection repeatable
train_index <- sample(seq_len(nrow(tweets_only2)), size = floor(0.75 *
nrow(tweets_only2)))#generate random indexes
train_set <- mat[train_index,]#create train set
```

```
train.y <- tweets_only2$airline_sentiment[train_index]#response variable for
train set
test_set <- mat[-train_index,]#create test set
test.y <- tweets_only2$airline_sentiment[-train_index]#response variable for
test set

nbClassifier = naiveBayes(train_set, train.y)#naive bayes classifier

container <- create_container(train_set, train.y, trainSize =
1:length(train_set[,1]), virgin = F)#create container
rfClassifier <- randomForest(train_set, train.y, ntree = 500)#random forest
classifier
rfcv <- cross_validate(container, nfold = 5, "RF")#5- fold cross validation
of random forest classifier


#svm
svmClassifier <- svm(train_set, train.y, cross = 5)#svm classifier with 5-
fold cross validation
svmcv <- svmClassifier$accuracies#cross validation results


nbpredict <- predict(nbClassifier, test_set)#predict test set classification
using naive bayes classifier
rfpredict <- predict(rfClassifier, test_set)#predict test set classification
using random forest classifier
svmpredict <- predict(svmClassifier, test_set)#predict test set
classification using svm classifier

confusionMatrix(nbpredict, test.y)#confusion matrix for naive bayes
classifier
confusionMatrix(rfpredict, test.y)#confusion matrix for random forest
classifier
confusionMatrix(svmpredict, test.y)#confusion matrix for svm classifier
```

9. References

[1] Amatriain, Xavier. (2015). What are the advantages of different classification algorithms? [Msg 1]. Message posted to https://www.quora.com/What-are-the-advantages-of-different-classification-algorithms

[2] Kaggle. (2016). Twitter US Airline Sentiment [Data file]. Retrieved from https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[3] Galitsky, Boris, and Eugene William McKenna. (2009). *Sentiment Extraction from Consumer Reviews for Providing Product Recommendations. US Patent 20090282019 A1*. Retrieved from https://www.google.com/patents/US20090282019

[4] US Airways. *Wikipedia*, 2016. Retrieved from https://en.wikipedia.org/wiki/US_Airways

[5] Wang, Cheng-Jun. (2016, 10 January). *Sentiment Analysis with Machine Learning in R*. Retrieved from https://www.r-bloggers.com/sentiment-analysis-with-machine-learning-in-r/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+RBloggers+%28R+bloggers%29