# **Sentiment Text Mining**

**By Sam Vuong, Raymond (shanhua) Huang, Kyle Murphy, Carmon Ho**

**Abstract:** The goal of this project is to be able to interpret the feelings of travellers on Twitter in February 2015. The success of this project will be determined by how well we can analyze the sentiments that were submitted to Twitter by various travellers in that period. This model can be used by airline companies who want to know the satisfaction of their customers' travel experiences.

Our application on ShinyApps can be accessed at: <https://skvuong.shinyapps.io/project/>

Our application code can be found in GitHub at: <https://github.com/skvuong/sentimentTextMining>

# **1. Introduction and Discussion**

Social media is becoming trends for many people to express their feelings and opinions. It can get conversations starting to hear different points of view from people amongst various countries and race. Twitter is the most popular social media platform for people expressing their emotions and feelings. For our project we will be analyzing the sentiment feelings of travellers amongst US Airlines in February 2015.

We will employ text mining techniques to analyze and process tweet text messages and use Random Forest classification algorithm to model and predict the sentiments tweet text messages.

## **1.1 Dataset**

The dataset that we use is the “Twitter US Airline Sentiment” dataset from the Kaggle website. The dataset can be found at: <https://www.kaggle.com/crowdflower/twitter-airline-sentiment>.

The dataset has14,640 rows and 15 columns.

The text messages in the dataset had been classified as follows:

* 9,178 (63%) negative sentiments
* 3,099 (21%) neutral sentiments
* 2,363 (16%) positive sentiments

The column names are:

1. tweet\_id
2. airline\_sentiment
3. airline\_sentiment\_confidence
4. negativereason
5. negativereason\_confidence
6. airline
7. airline\_sentiment\_gold
8. name
9. negativereason\_gold
10. retweet\_count
11. text
12. tweet\_coord
13. tweet\_created
14. tweet\_location
15. user\_timezone

## **1.2 Ethical ML Framework**

In order to conduct our research in an ethically responsible manner we have considered a number of important factors. We have removed all variables that can be linked to personal information of the participants in the dataset. This will prevent the results from discriminating against people based on factors like gender, religion, and region.

The dataset that we are using includes the name of the user and the Tweet\_ID which can be used to trace back to the Tweeter identification/account. We have decided not to have any code to view or print out these information from our dataset in order to protect the privacy of the participants.

## **1.3 Assumptions**

For this dataset we would assume travellers would live in North America because their location would have easier access to US airline companies. Contributors to this dataset would be at least teenagers who have access to a smartphone device or laptop with internet service as Twitter is an online application to be able to post their tweets.

# **2. Data Preparation**

## **2.1 Loading R Libraries**

* library(caTools) #data wrangling
* library(dplyr) #data manipulation
* library(forcats) #ggplot frequency
* library(ggplot2) #visualizations
* library(ggthemes) #visualizations
* library(lubridate)
* library(mgsub)
* library(RColorBrewer) #color palettes
* library(randomForest) #randomforest
* library(scales)
* library(stringr)
* library(syuzhet)
* library(SnowballC) #text stemming
* library(stopwords) #stop-words removal
* library(tm) #text mining
* library(wordcloud) #word-cloud generator
* library(wordcloud2) #word-cloud generator

## **2.2 Loading Data**

We read input data files and perform basic checks on data.

Read input CSV data files

1. Read movies file and look at first few lines:

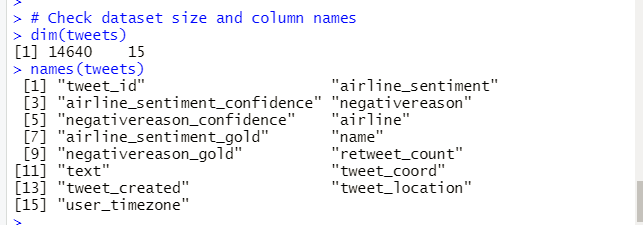


# 

# **3. Data Exploration Analysis**

1. Check the dataset size and column names:

The output below shows that the set has 15 columns and 12,640 rows.



2. Check how many unique tweets are in the dataset:



3. Checking how many users have tweeted:



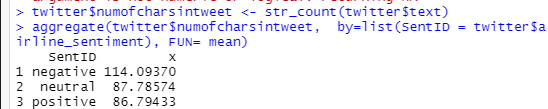
4. Checking average characters per tweet:



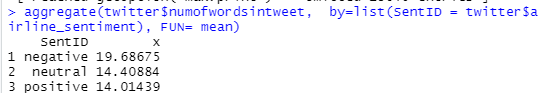
5. Checking average words per tweet:



6. Checking average tweet characters per sentiment:



7. Checking average tweet words per sentiment:

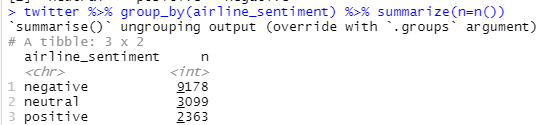


The summary outputs below show the longer the tweet length, the more negative the tweet. This implies that longer tweet length results in negative sentiment.

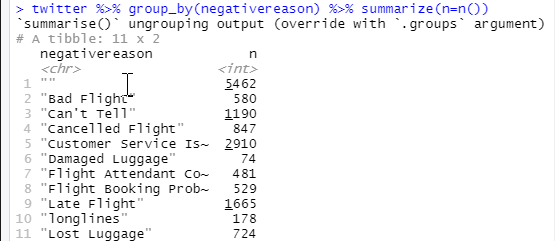
8. Checking what are the types of sentiments:



9. Checking the count of sentiments:



10. Checking the count of Negative Reasons.



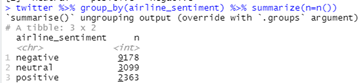
# 

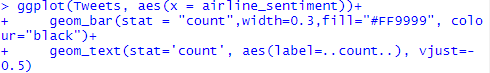
# **4. Data Visualization**

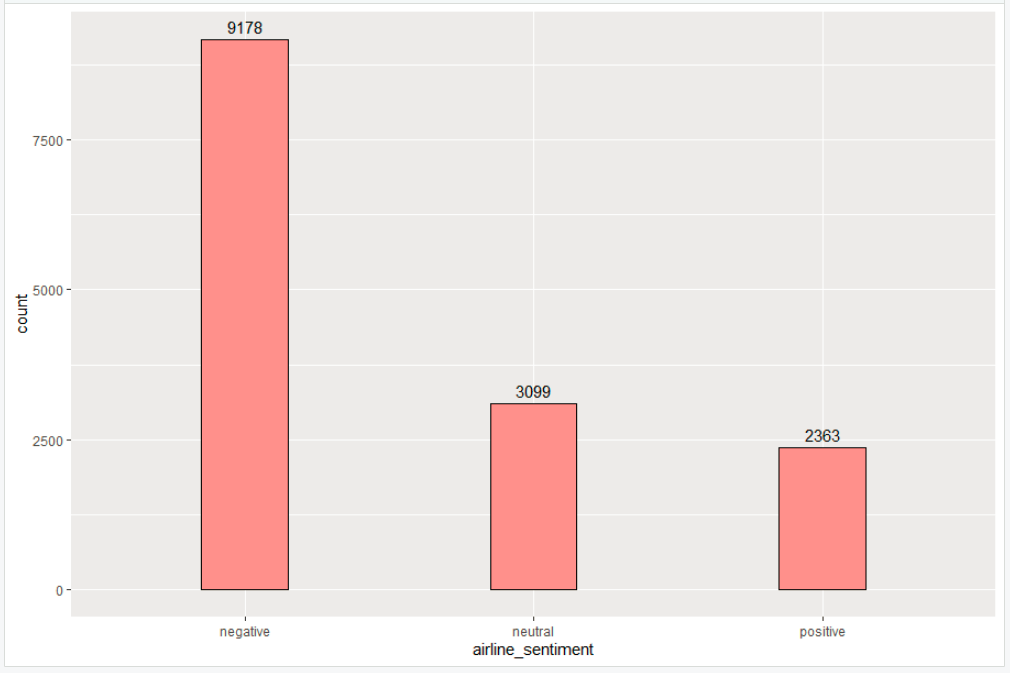
## 

## **4.1 Tweet Sentiment Distribution**

The Tweet sentiments distribution chart presented below shows that **negative** tweets are more than **neutral** tweets and **positive** tweets combined. The counts show that there are 63% negative tweets, 21% neutral tweets and 16% positive tweets.





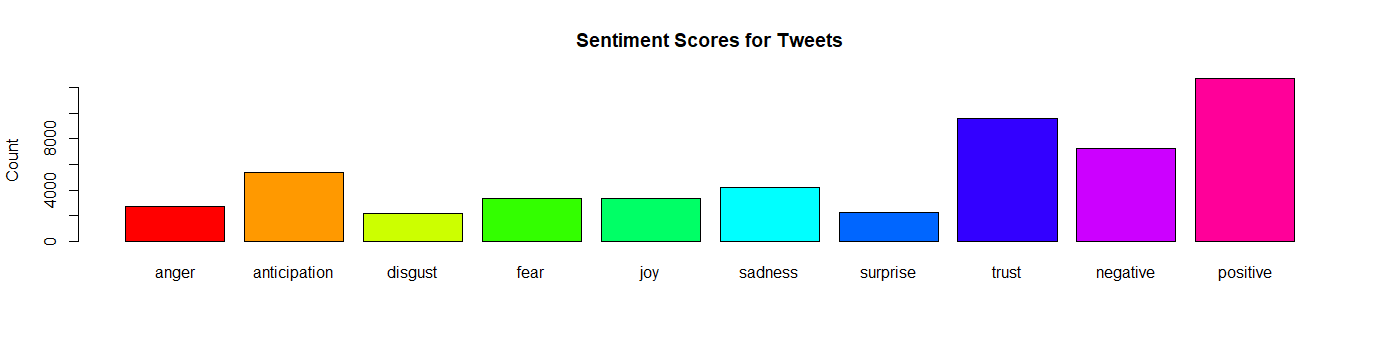


## 

## **4.2 Tweet Text Sentiment Scores**

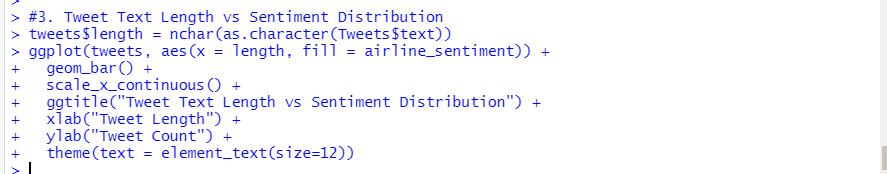
If the package syuzhet was used to perform analysis that digs into the text, the sentiment scores will be presented as below. From the below graph, it seems that the positive sentiment score was the highest. The score in trust was second and negative follow behind. This result looks the opposite to the 4.1 chart. It is because the negative review was further broken down to anger, sadness, fear, disgust and surprise. This also helps to explain what kind of negative feeling hides behind. Based on those reviews and sentiment breakdown, the Tweets presented a positive trend though there were some negative amounts of feeling coming from anger, sadness, fear, disgust and surprise. After all tweets presented some anticipations as the anticipation score ranks no.4 in the graph.

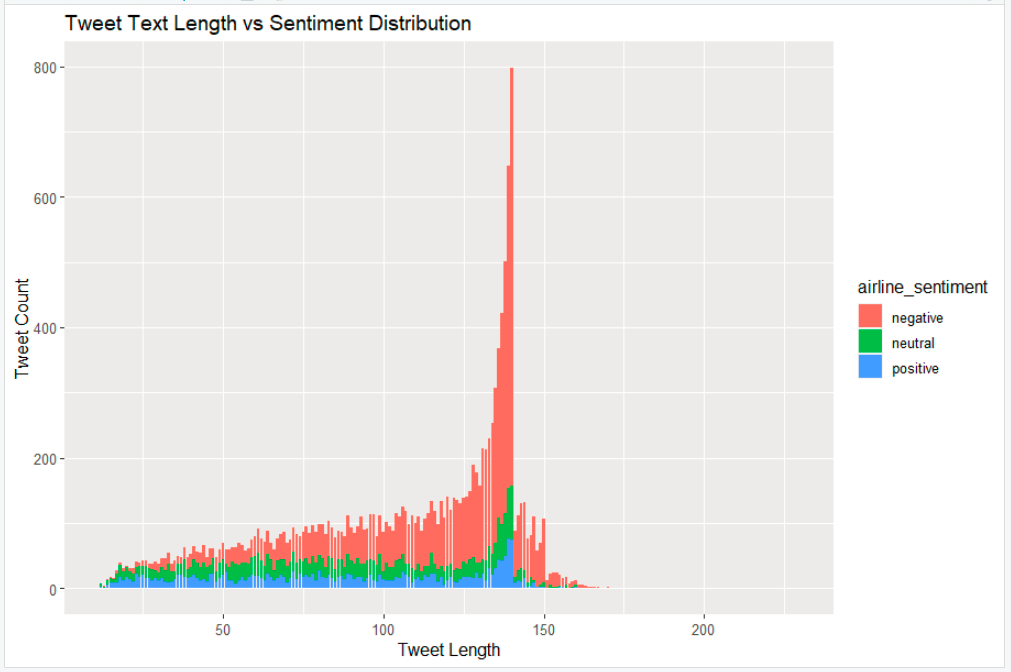




### 4.3 Tweet Text Length vs Sentiment Distribution

Below is the chart for tweet text length vs sentiment. The chart shows that there is a high number of tweets with length between 130 and 140.





## **4.4 Word-Cloud chart for Negative Sentiment Tweets (Top 100)**

The word cloud was used to investigate reasons why more negative Tweets were presented. The results show that customer service and late flight were the reason causing negative Tweets. Here the services stand for customer services issues as it may not present in word cloud due to length.

negative <- twitter %>% filter(airline\_sentiment == "negative")

negativetext <-negative$text

negativetext <- mgsub(negativetext, c("https\\S\*", "@\\S\*", "amp", "[\r\n]","[[:punct:]]","@united", "@VirginAmerica","@usairways","@americanair","@united","@jetblue"), c("","","","","","","","","","",""))

docsnegative <- Corpus(VectorSource(negativetext))

dtmnegative <- TermDocumentMatrix(docsnegative)

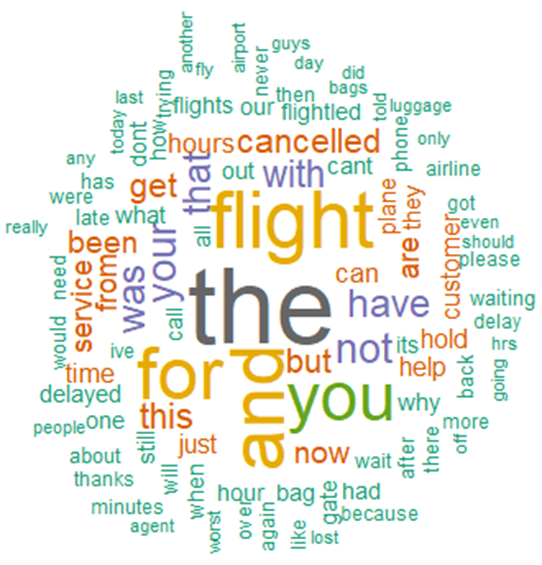
matrixnegative <- as.matrix(dtmnegative)

wordsnegative <- sort(rowSums(matrixnegative),decreasing=TRUE)

dfnegative <- data.frame(word = names(wordsnegative),freq=wordsnegative)

set.seed(1234) # for reproducibility

wordcloud(words = dfnegative$word, freq = dfnegative$freq, min.freq = 1, max.words=100, random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))



## 

## 

## **4.5 Word-Cloud chart for Neutral Sentiment Tweets (Top 100)**

The word cloud was used to investigate reasons why more neutral Tweets were presented. The results do not show any obvious reasons causing neutral tweets. This is expected given that neutral tweets would not contain any negative or positive cogitations.

neutral <- twitter %>% filter(airline\_sentiment == "neutral")

neutraltext <-neutral$text

neutraltext <- mgsub(neutraltext, c("https\\S\*", "@\\S\*", "amp", "[\r\n]","[[:punct:]]","@united", "@VirginAmerica","@usairways","@americanair","@united","@jetblue"), c("","","","","","","","","","",""))

docsneutral <- Corpus(VectorSource(neutraltext))

dtmneutral <- TermDocumentMatrix(docsneutral)

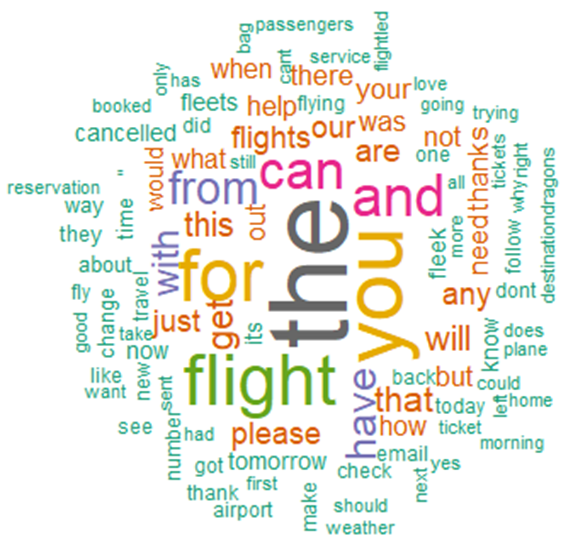
matrixneutral <- as.matrix(dtmneutral)

wordsneutral <- sort(rowSums(matrixneutral),decreasing=TRUE)

dfneutral <- data.frame(word = names(wordsneutral),freq=wordsneutral)

set.seed(1234) # for reproducibility

wordcloud(words = dfneutral$word, freq = dfneutral$freq, min.freq = 1, max.words=100, random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))



## 

## 

## **4.6 Word-Cloud chart for Positive Sentiment Tweets (Top 100)**

The word cloud was used to investigate reasons why more positive Tweets were presented. The results show that customer gratitude was the reason causing positive Tweets. Words such as ‘thank you’, ‘thanks’, ‘great’, ‘love’, ‘appreciate’, and ‘amazing’ can be found in the top 100.

positive <- twitter %>% filter(airline\_sentiment == "positive")

positivetext <-positive$text

positivetext <- mgsub(positivetext, c("https\\S\*", "@\\S\*", "amp", "[\r\n]","[[:punct:]]","@united", "@VirginAmerica","@usairways","@americanair","@united","@jetblue"), c("","","","","","","","","","",""))

docspositive <- Corpus(VectorSource(positivetext))

dtmpositive <- TermDocumentMatrix(docspositive)

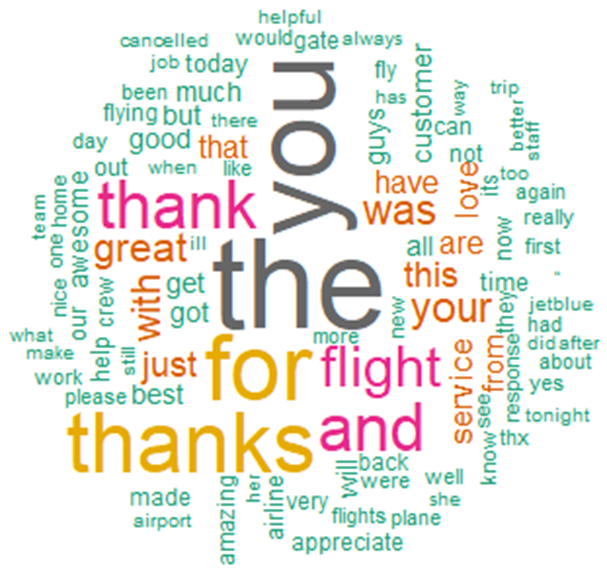
matrixpositive <- as.matrix(dtmpositive)

wordspositive <- sort(rowSums(matrixpositive),decreasing=TRUE)

dfpositive <- data.frame(word = names(wordspositive),freq=wordspositive)

set.seed(1234) # for reproducibility

wordcloud(words = dfpositive$word, freq = dfpositive$freq, min.freq = 1, max.words=100, random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))

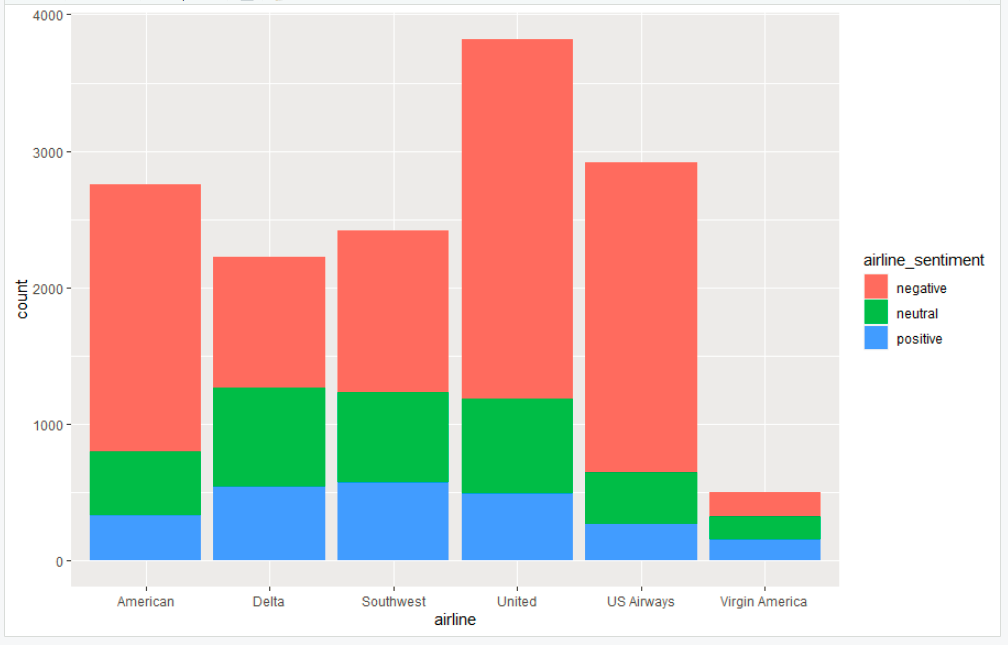


## 

## **4.7 Airlines Sentiment Distribution**

If airline company as well as sentiment was chosen as dimension and count was used as measure, United airline has the greatest negative Tweets among all. Delta airline favored more positive Tweets percentages.





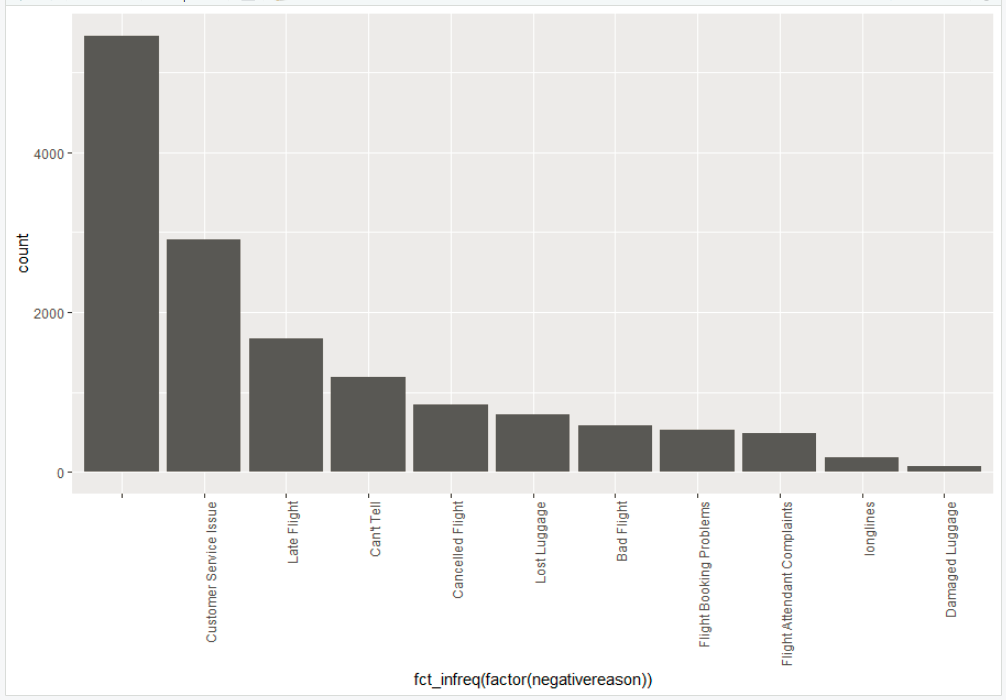
### 4.8 Negative Tweet Reasons Distribution

Based on the bar graph below, ‘Customer Service Issue’ appears to be the most common negative tweet reason and ‘Damaged Luggage’ appears to be the least common.

**ggplot(twitter, aes(x = fct\_infreq(factor(negativereason))))+**

**+ geom\_bar(stat = "count")+**

**+ theme(axis.text.x = element\_text(angle = 90, hjust = 1))**



## 

## **4.9 Airlines Negative Reason Confident Distribution**

Based on the three ‘Count of Negative Confidence Distribution’ charts below, the data suggests that negative tweets tend to have a higher confidence level with American Airlines being the greatest.

**American Airline - Negative Reason Confident Distribution**

American <- twitter %>% filter(airline == "American")

American %>%

ggplot(aes(x= negativereason\_confidence)) +

geom\_histogram(color = "white") +

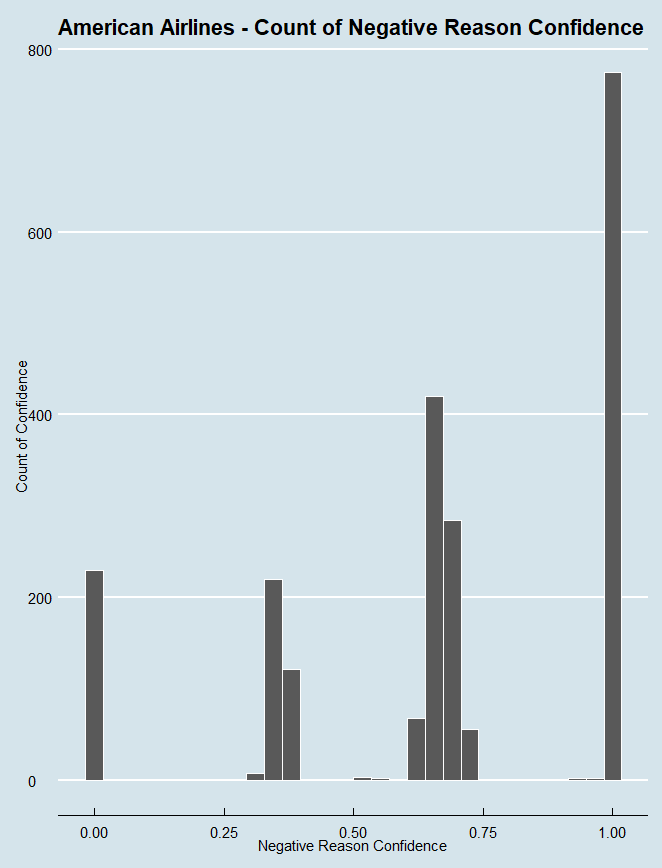
ggtitle("American Airlines - Count of Negative Reason Confidence") +

xlab("Negative Reason Confidence") +

ylab("Count of Confidence") +

scale\_y\_continuous(labels = comma) +

theme\_economist()

****

**United Airline - Negative Reason Confident Distribution**

United<- twitter %>% filter(airline == "United")

United%>%

ggplot(aes(x= negativereason\_confidence)) +

geom\_histogram(color = "white") +

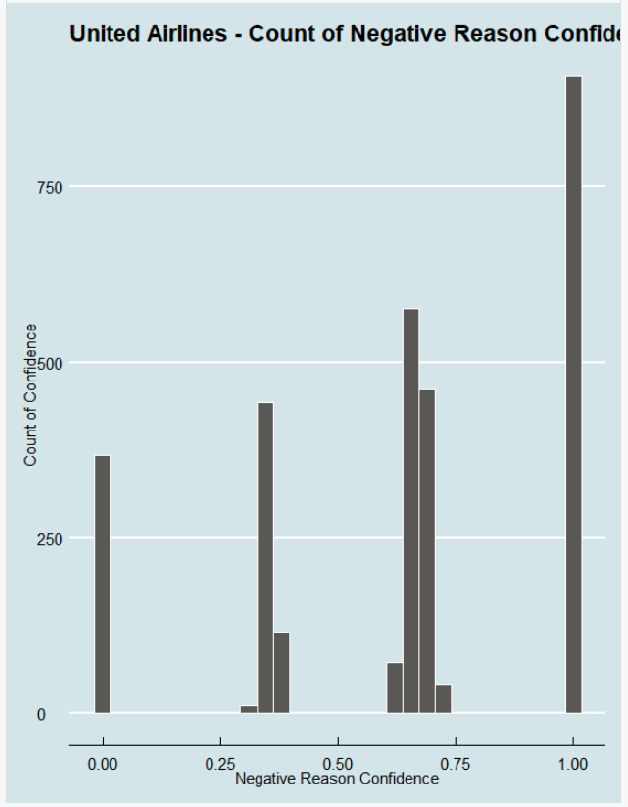
ggtitle("United Airlines - Count of Negative Reason Confidence") +

xlab("Negative Reason Confidence") +

ylab("Count of Confidence") +

scale\_y\_continuous(labels = comma) +

theme\_economist()



**US Airways - Negative Reason confident Distribution**

USAirways <- twitter %>% filter(airline == "US Airways")

USAirways %>%

ggplot(aes(x= negativereason\_confidence)) +

geom\_histogram(color = "white") +

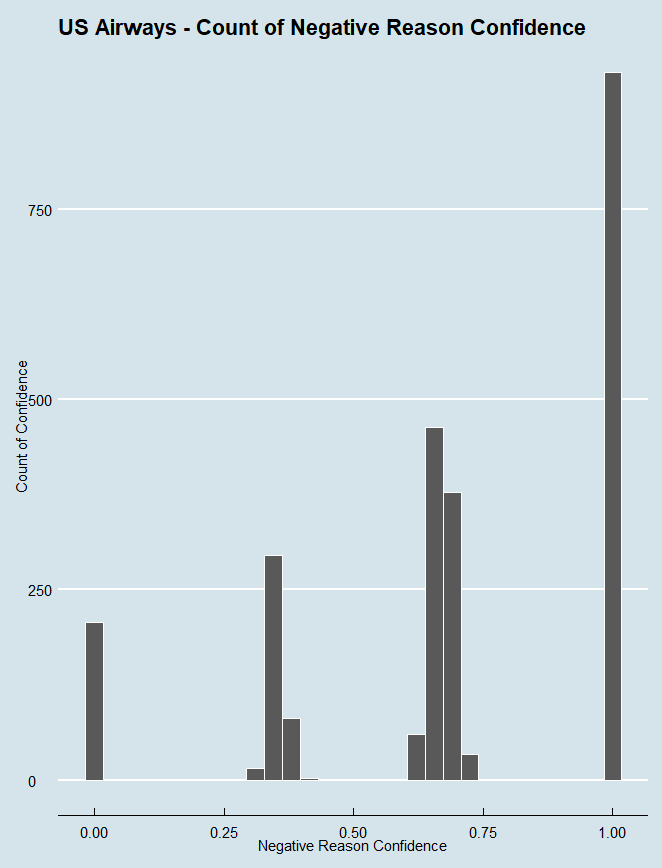
ggtitle("US Airways - Count of Negative Reason Confidence") +

xlab("Negative Reason Confidence") +

ylab("Count of Confidence") +

scale\_y\_continuous(labels = comma) +

theme\_economist()

****

**Delta Airline - Negative Reason confident Distribution**

Delta<- twitter %>% filter(airline == "Delta")

Delta %>%

ggplot(aes(x= negativereason\_confidence)) +

geom\_histogram(color = "white") +

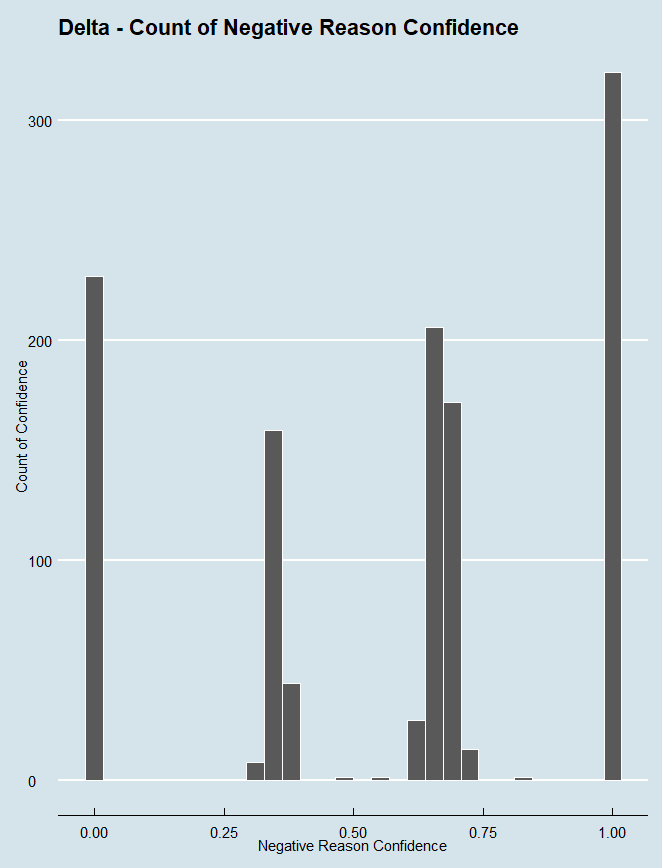
ggtitle("Delta - Count of Negative Reason Confidence") +

xlab("Negative Reason Confidence") +

ylab("Count of Confidence") +

scale\_y\_continuous(labels = comma) +

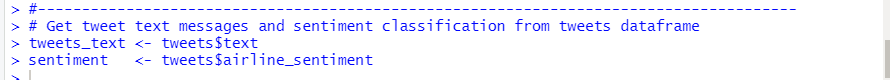
theme\_economist()

****

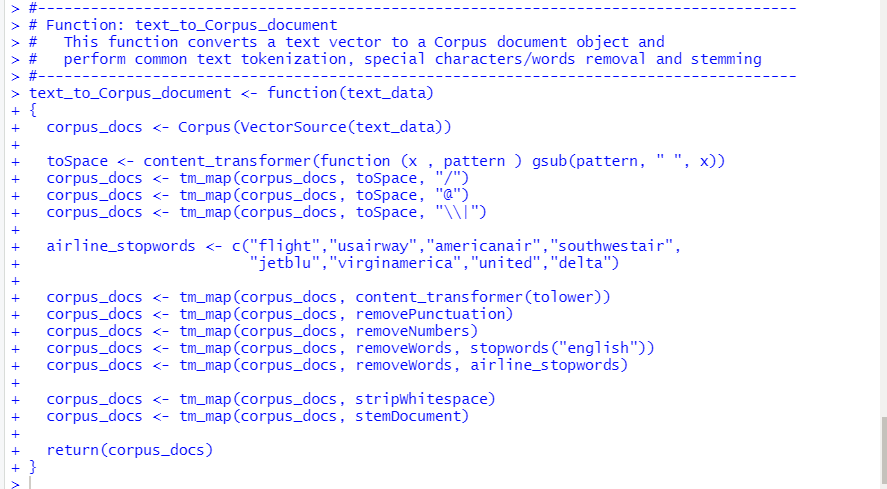
# **5. Modeling**

## **5.1 Data Preparation for Modeling**

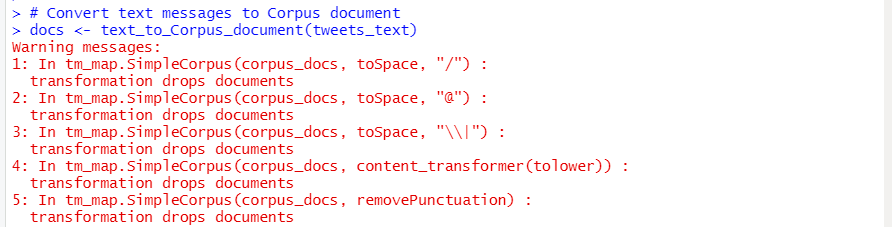
First we get the tweet text messages and the sentiment classification from tweets dataframe.



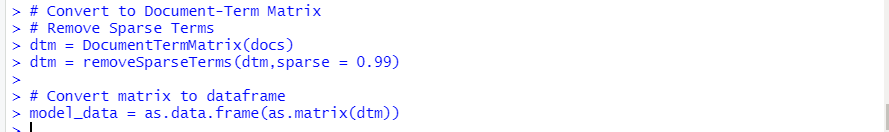
We create a function text\_to\_Corpus\_document() to convert a text vector to a Corpus document object and perform common text tokenization, special characters/words removal and stemming.



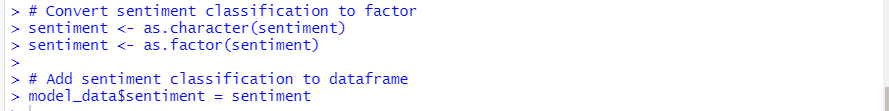
We then convert text messages to Corpus documents. Note that there are few warning messages. However, these are just warnings.



We convert Corpus documents to Document-Term matrix, remove sparse terms and convert the matrix to Dataframe.



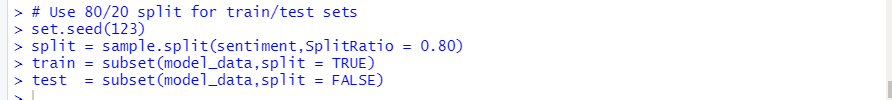
We convert the sentiment classification to a factor and add it to the dataframe for modeling.



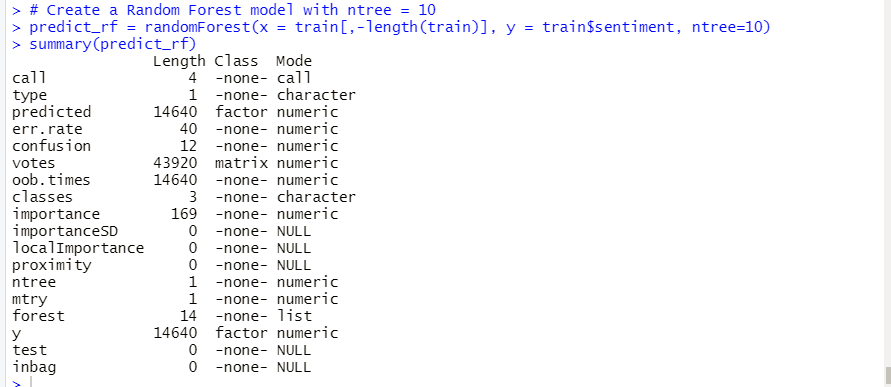
## 

## **5.2 Modeling**

We use 80/20 split train/test sets for our modeling:



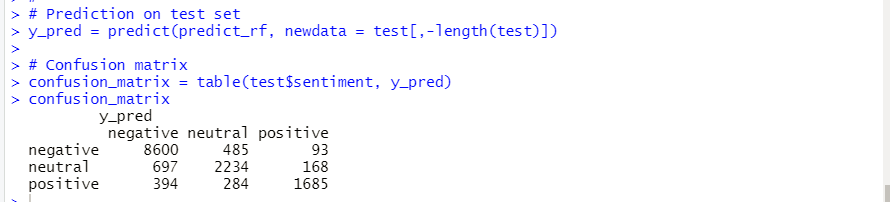
We create a Random Forest model with Number of trees to grow (ntree) = 10.



# 

# **6. Model Evaluation**

We use the model built about to predict the outcomes for test set and print out the Confusion matrix values:



We calculate the accuracy of the model prediction.

The result shows that the model has an accuracy of 85%.

# 

# 

# **7. Deployment**

We used the ShinyApps website to deploy our application.

Our application on ShinyApps can be accessed at: <https://skvuong.shinyapps.io/project/>

Our application code can be found in GitHub at: <https://github.com/skvuong/sentimentTextMining>

**7.1 Code for ShinyApps**

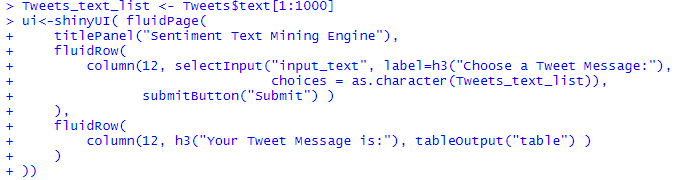
The code for ShinyApps includes 2 parts: **user interface** code and **server code**.

**1. Code for User Interface:**

The user interface code does the following:

* Asks the user to choose a tweet text message from the list of messages available.
* Calls the server code to predict sentiment for the input text message. (This is done by the ShinyApps framework.)
* Displays to predicted sentiment on the screen

Below is the code for the user interface:

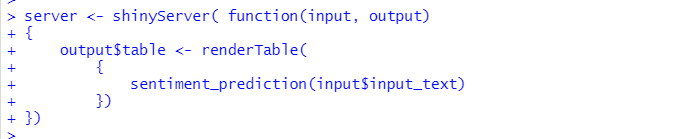


**2. Code for Server:**

The **server code** does the following:

* Gets the input text message from the UI code.
* Calls the sentiment\_prediction() function to predict sentiment for the input text message.
* Returns the Predict\_Result to the UI code.

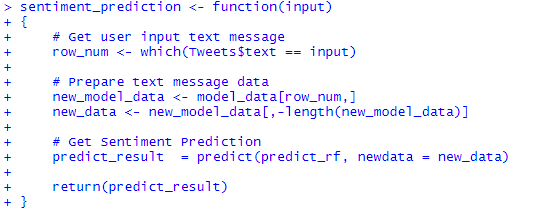
Below is the server code:



The **sentiment\_prediction() function** does the following:

* Locate the input text message in the dataset.
* Prepare model data for the input test message.
* Calls the model code to predict sentiment for the input text message. (The model code is described in section 5. Modeling and section 6. Evaluation above.)
* Returns the Predict\_Result.

Below is the code for the sentiment\_prediction() function:



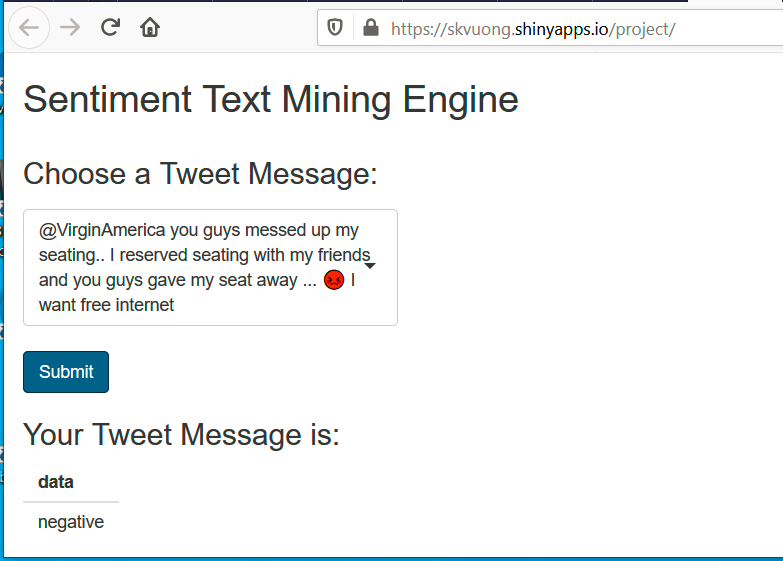
**7.2 Test The Application on ShinyApps**

There are 3 possible outcomes for sentiment from tweet text messages:

* negative
* neutral
* positive

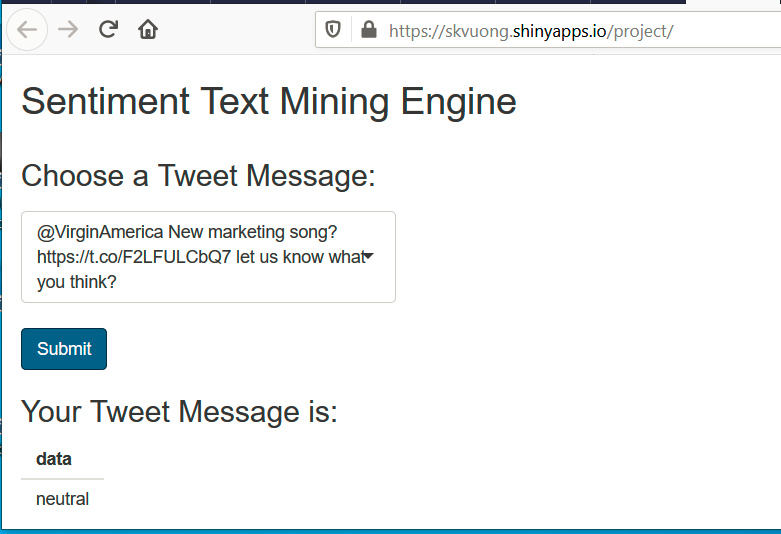
**1. Negative Sentiment tweet:**

Below is a tweet message that is negative sentiment.



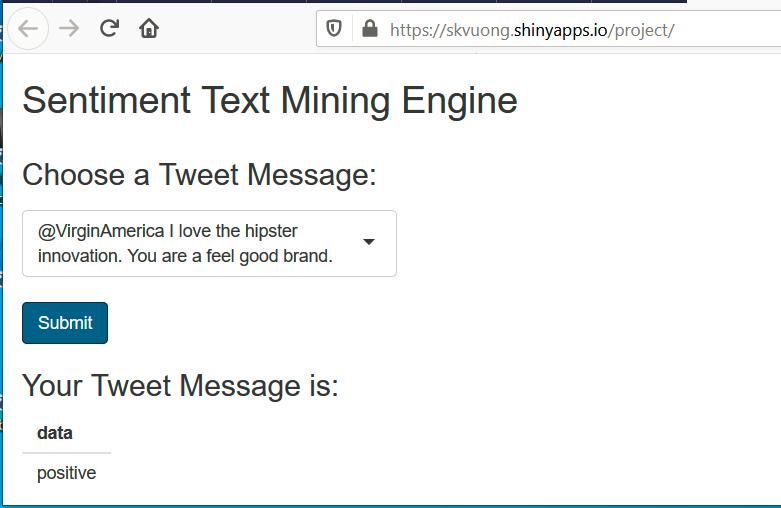
**2. Neutral Sentiment tweet:**

Below is a tweet message that is neutral sentiment.



**3. Positive Sentiment tweet:**

Below is a tweet message that is positive sentiment.

****

# 

# **8. Discussion**

In this paper, the Tweet dataset was explored. The first part includes data exploration and visualization. Then modeling and evaluation follow behind. Lastly, a web application which identifies one of the three sentiments, namely negative, positive and neutral from Tweets text in the drop down list, was created.

From the sentimental analysis in section 4, customers comments presented hopes that customers could be satisfied by improvement in customer services. In section 4.2, it confirmed that customers presented anticipations and things could be better off.

For the modeling part, the accuracy is 85% using the Random Forest algorithm. This was exciting but more data needed to be fed into the model and further increase the scope of the model.

Due to the tight timeline, further work on how to improve user interface will be of great interest. To further improve the user interface, the dropdown list can be replaced by typing words from users. On the other hand, more data is needed to be fed into the model to further improve the scope of the model.

# **9.** **Reference / Citation**

Eight, Figure. “Twitter US Airline Sentiment.” *Kaggle*, Crowdflower’s Data for Everyone, 16 Oct. 2019, [www.kaggle.com/crowdflower/twitter-airline-sentiment](http://www.kaggle.com/crowdflower/twitter-airline-sentiment).