INDIAN INSTITUTE OF TECHNOLOGY (ISM), DHANBAD

ACADEMIC PROJECT REPORT [2017-18]

ON

US Airlines Tweets sentiment analysis

UNDER GUIDANCE OF:

Dr. Rajendra Pamula
Assistant Professor
Department of CSE
IIT ISM Dhanbad

SUBMITTED BY:

Siddharth Shukla
15JE001462
B. Tech 3rd Year

Acknowledgement

I take this opportunity to express my deep sense of gratitude and respect towards my project guide, Dr. Rajendra Pamula, Assistant Professor, Department of Computer Science and Engineering, IIT(ISM) Dhanbad. I am very much indebted to him for the generosity, expertise and guidance that I have received from him while working on this project. I would also like to thank Praful Jain, for providing me valuable guidance at each and every step of completion of this project.

I wish to express profound gratitude to Prof. P.K. Jana, HOD, Computer Science and Engineering, IIT(ISM) Dhanbad for his inspiration and guidance.

Siddharth Shukla

15JE001462

B. Tech CSE 3rd Year

Certificate

This is to certify that the report entitled '**US Airlines Tweets sentiment analysis**' submitted by **Siddharth Shukla**, Department of Computer Science and Engineering, IIT(ISM) Dhanbad has successfully completed a project in 5^{th} semester of academic year 2017–2018.

Prof. P K Jana
Head of Department
Department of CSE
IIT(ISM) Dhanbad

Dr. Rajendra Pamula
Assistant Professor
Department of CSE
IIT(ISM) Dhanbad

Abstract

The ability to exploit public sentiment in social media is increasingly considered as an important tool for market understanding, customer segmentation and stock price prediction for strategic marketing planning and manoeuvring. This evolution of technology adoption is energized by the healthy growth in big data framework, which caused applications based on Sentiment Analysis in big data to become common for businesses.

In this project, an attempt is made on analyzing the sentiment of tweets by passengers of 6 different US airlines, and to find most common reasons associated with positive as well as negative sentiment of tweets. Different machine learning classifiers were used to predict to results. An additional method for sentiment analysis using a different criteria is also proposed and analyzed by the author. The dataset consists of 14,485 tweets (rows), with each having 15 parameters (columns). The dataset represents tweets of 6 different airlines.

Table of Contents

Introduction

Data Mining is an analytic process designed to explore data (usually large amounts of data, typically business or market related, also known as "big data") in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction and predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining consists of three stages:

- **Initial exploration**: This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and in case of data sets with large numbers of variables ("fields") performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods which are being considered).
- **Model building or pattern identification with validation/verification**: This stage involves considering various models and choosing the best one based on their predictive performance (i.e., explaining the variability in question and producing stable results across samples).
- **Deployment**: This final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

The concept of Data Mining is becoming increasingly popular as a business information management tool where it is expected to reveal knowledge structures that can guide decisions in conditions of limited certainty. Recently, there has been increased interest in developing new analytic techniques specifically designed to address the issues relevant to business Data Mining (e.g., Classification Trees).

Sentiment Analysis

Firstly, the tweet dataset is downloaded & is imported into Rstudio using 'Import dataset' tool. We need packages such as tm (for topic modeling), SnowballC (for sentiment analysis) & wordcloud (to generate wordclouds). After the dataset is imported, following operations are performed,

```
> library(tm); library(SnowballC)
> library(wordcloud)
```

```
> # these words appear quite frequently in tweets and in my opinion are not informative,
> # so I will remove them
```

```
> wordsToRemove = c('get', 'cant', 'can', 'now', 'just', 'will', 'dont', 'ive', 'got', 'much')
```

```
> # generate a function to analyse corpus text
> # analyse text and generate matrix of words
> # Returns a dataframe containing 1 tweet per row, one word per column
> # and the number of times the word appears per tweet
```

```
> analyseText = function(text_to_analyse){
+     CorpusTranscript = Corpus(VectorSource(text_to_analyse))
+    CorpusTranscript$text    <-    sapply(CorpusTranscript$text,function(row) iconv(row, "latin1", "ASCII", sub=""))
+     CorpusTranscript = tm_map(CorpusTranscript, removePunctuation)
+     CorpusTranscript = tm_map(CorpusTranscript, removeWords, wordsToRemove)
+      CorpusTranscript      =      tm_map(CorpusTranscript,      removeWords, stopwords("english"))
+      CorpusTranscript = DocumentTermMatrix(CorpusTranscript)
+      CorpusTranscript = removeSparseTerms(CorpusTranscript, 0.97) # keeps a matrix 97% sparse
+      CorpusTranscript = as.data.frame(as.matrix(CorpusTranscript))
```

```
+     colnames(CorpusTranscript) = make.names(colnames(CorpusTranscript))
+     return(CorpusTranscript)}
```

> # sum the number of times each word appears in total across all negative tweets.

```
> freqWords_neg = colSums(words)
> freqWords_neg = freqWords_neg[order(freqWords_neg, decreasing = T)]
> head(freqWords_neg)
   flight cancelled   service     hours      help      hold
     2901       920       742       644       610       607
```

> # analysis of positive tweets

```
> words = analyseText(positive$text)
> dim(words)
[1] 2363   19
> freqWords_pos = colSums(words)
> freqWords_pos = freqWords_pos[order(freqWords_pos, decreasing = T)]
> head(freqWords_pos)
 thanks   thank  flight   great service    love
    609     453     373     233     160     133
```

> # combine thanks and remove extra column

```
> freqWords_pos[1] = freqWords_pos[1] + freqWords_pos[2]
> freqWords_pos = freqWords_pos[-2]
> head(freqWords_pos)
  thanks   flight    great  service     love customer
    1062      373      233      160      133      113
```

> # word clouds

```
> par(mfrow = c(1,2))
> wordcloud(freq = as.vector(freqWords_neg), words = names(freqWords_neg), random.order = FALSE,
```

```
+  random.color = FALSE, colors = brewer.pal(9, 'Reds')[4:9])
> wordcloud(freq = as.vector(freqWords_pos),words=names(freqWords_pos),
random.order = FALSE, random.color = FALSE, colors = brewer.pal(9, 'BuPu')
[4:9])
```

> # generate a function to analyse corpus text and return a document term matrix instead of dataframe

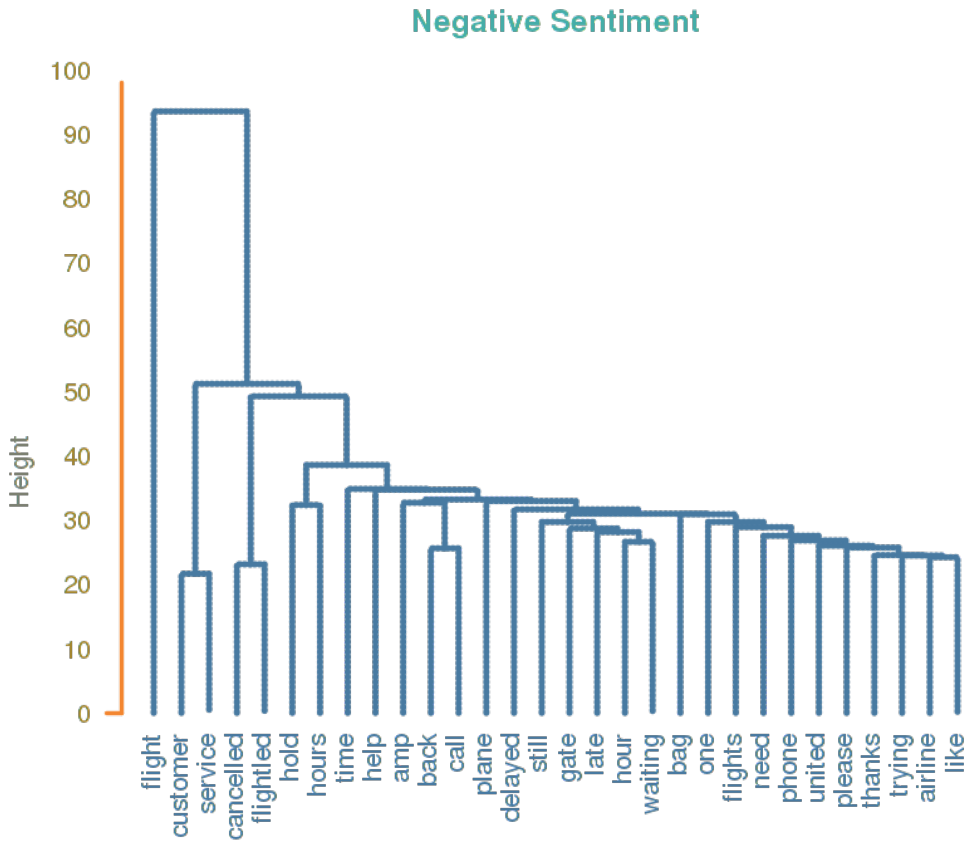
> # we can perform further analysis on document term matrices

```
> analyseText2 = function(text_to_analyse){
+     # analyse text and generate matrix of words
+     # Returns a document term matrix containing 1 tweet per row,
one word per column
+     # and the number of times the word appears per tweet
+     CorpusTranscript = Corpus(VectorSource(text_to_analyse))
+     CorpusTranscript$text   <-   sapply(CorpusTranscript$text,function(row)
iconv(row, "latin1", "ASCII", sub=""))
+     CorpusTranscript = tm_map(CorpusTranscript, removePunctuation)
+    CorpusTranscript = tm_map(CorpusTranscript, removeWords, wordsToRemove)
+CorpusTranscript=tm_map(CorpusTranscript,removeWords, stopwords("english"))
+     CorpusTranscript = DocumentTermMatrix(CorpusTranscript)
+     CorpusTranscript = removeSparseTerms(CorpusTranscript, 0.97)
```

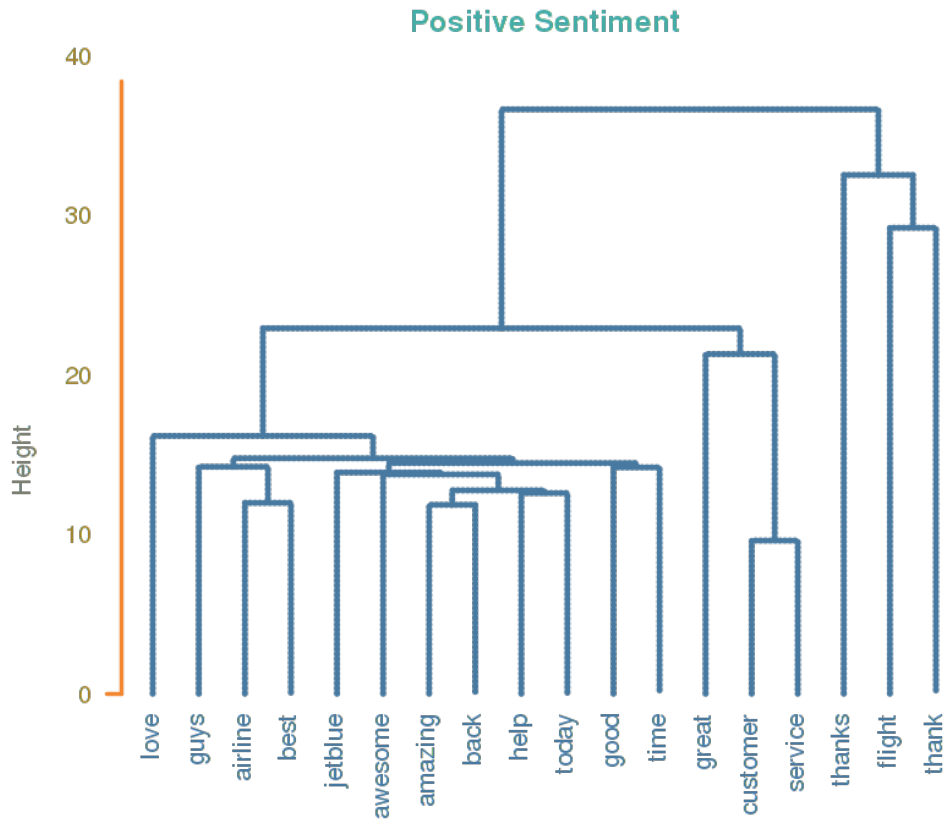
```
# keeps a matrix 97% sparse
+     return(CorpusTranscript)
+ }

> words_neg = analyseText2(negative$text)
> # find words correlated with the ones mentioned below (correlation at 70%)
> findAssocs(words_neg, c("flight", 'customer', 'gate', 'phone'), .07)
$flight
cancelled      late flightled   delayed
     0.36      0.25      0.23      0.16

$customer
service
   0.65

$gate
waiting   plane
   0.09    0.08

$phone
help
0.07
> words_pos = analyseText2(positive$text)
> findAssocs(words_pos, c("flight", 'awesome', 'amazing', 'service'), .07)
$flight
great
 0.13

$awesome
guys
0.07

$amazing
```

```
you customer
0.12     0.08
$service
customer    great    today
    0.66     0.16     0.08
```

```
> # hierarchical clustering
> d = dist(t(as.matrix(words_neg)), method = 'euclidean')
> fit = hclust(d = d, method = 'ward.D')
> op = par(bg = "#DDE3CA")
> plot(fit, col = "#487AA1", col.main = "#45ADA8", col.lab = "#7C8071", main
= 'Negative Sentiment', xlab = '',
+ col.axis = "#F38630", lwd = 3, lty = 3, sub = "", hang = -1, axes = FALSE)
> # add axis
> axis(side = 2,at=seq(0, 400, 100),col="#F38630",labels = FALSE,+lwd = 2)
> # add text in margin
>mtext(seq(0,100,10),side=2,at=seq(0,100,10),line=1,col="#A38630", las = 2)
```



```
# positive sentiment tweets

d = dist(t(as.matrix(words_pos)), method = 'euclidean')

fit = hclust(d = d, method = 'ward.D')

op = par(bg = "#DDE3CA")

plot(fit, col = "#487AA1", col.main = "#45ADA8", col.lab = "#7C8071", main =
'Positive Sentiment', xlab = '',col.axis = "#F38630", lwd = 3, lty = 3, sub
= "", hang = -1, axes = FALSE)

# add axis

axis(side = 2, at = seq(0, 400, 100), col = "#F38630", labels = FALSE,lwd=2)

# add text in margin

mtext(seq(0, 100, 10), side = 2, at = seq(0,100,10), line =1,col="#A38630",
las = 2)
```



Alternative approach to sentiment analysis [proposed]

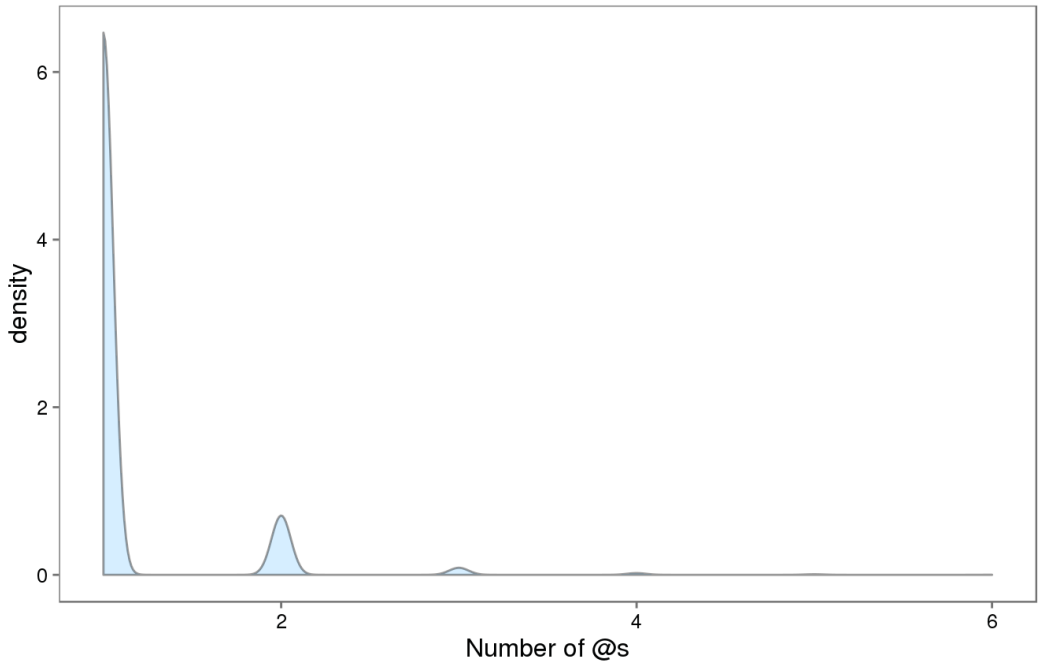
In this script, I aim to test a couple of hypotheses. The first is that the more addressees in a tweet, the harsher its words. I do this by first counting the number of @ symbols in the text of each tweet. I then show a few visualizations to investigate my theory further. My second hypothesis is that longer tweets are also less likely to contain favorable language.

Just like in previous script, firstly the dataset is imported into RStudio. Then,

```
> library('readr')    # to read files
> library('ggplot2')  # for visualization
> library('ggthemes') # for visualization
> library('dplyr')    # for data manipulation
> str(Tweets)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':    14640 obs. of  4 variables:
 $ airline_sentiment: chr  "neutral" "positive" "neutral" "negative" ...
 $ negativereason   : chr  NA NA NA "Bad Flight" ...
 $ airline : chr  "Virgin America" "Virgin America" "Virgin America" "Virgin America" ...
 $ text : chr  "What @dhepburn said." "plus you've added commercials to the experience... tacky." "I didn't today... Must mean I need to take another trip!" "it's really aggressive to blast obnoxious \"entertainment\" in your guests' faces &amp;amp; they have little recourse" ...
 - attr(*, "spec")=List of 2
  ..$ cols   :List of 15
  .. ..$ tweet_id : list()
  .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
  .. ..$ airline_sentiment: list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ airline_sentiment_confidence: list()
  .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
```

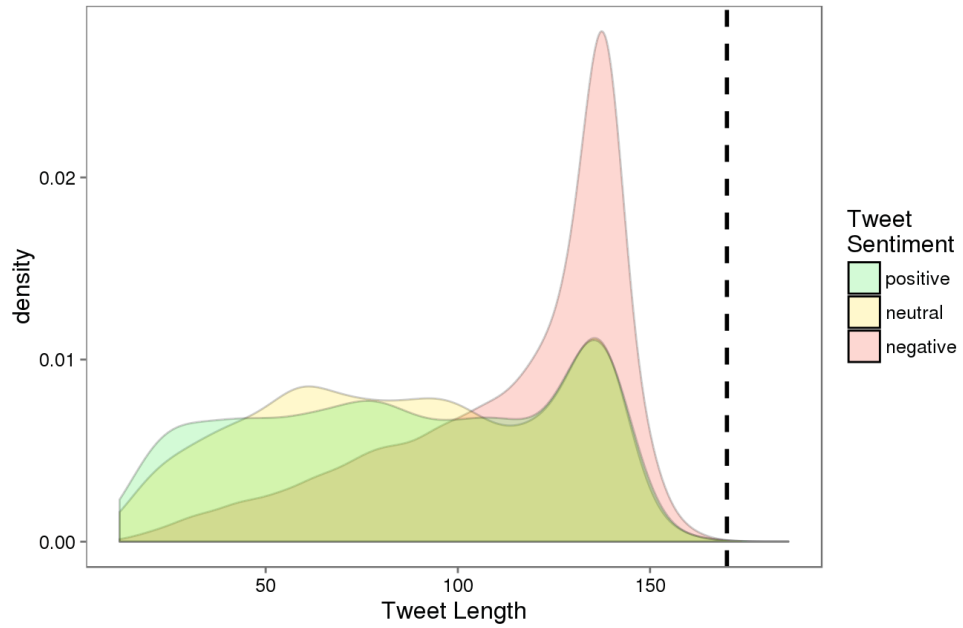
```
  .. ..$ negativereason : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ negativereason_confidence : list()
  .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
  .. ..$ airline : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ airline_sentiment_gold : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ name : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ negativereason_gold         : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ retweet_count : list()
  .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
  .. ..$ text : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ tweet_coord : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ tweet_created : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ tweet_location : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  .. ..$ user_timezone : list()
  .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
  ..$ default : list()
  .. ..- attr(*, "class")= chr  "collector_guess" "collector"
  ..- attr(*, "class")= chr "col_spec"
> Tweets$at_count <- sapply(Tweets$text, function(x) str_count(x, '@'))
> maxAt <- max(Tweets$at_count)
> Tweets$at_countD[Tweets$at_count == 1] <- '1'
> Tweets$at_countD[Tweets$at_count == 2] <- '2'
> Tweets$at_countD[Tweets$at_count %in% c(3:maxAt)] <- '3+'
> Tweets$at_countD <- factor(Tweets$at_countD)
> Tweets$text_length <- sapply(Tweets$text, function(x) nchar(x))
```

```
# Getting my sentiment colors & breaks ready.
> sentPlt     <- c('#f93822','#fedd00','#27e833')
> sentBreaks  <- c('positive','neutral','negative')

# Visualize distribution of `@` counts
> ggplot(Tweets, aes(x = at_count)) +
+     geom_density(fill = '#99d6ff', alpha=0.4) +
+     labs(x = 'Number of @s') +
+     theme_few() +
+     theme(text = element_text(size=12))
```

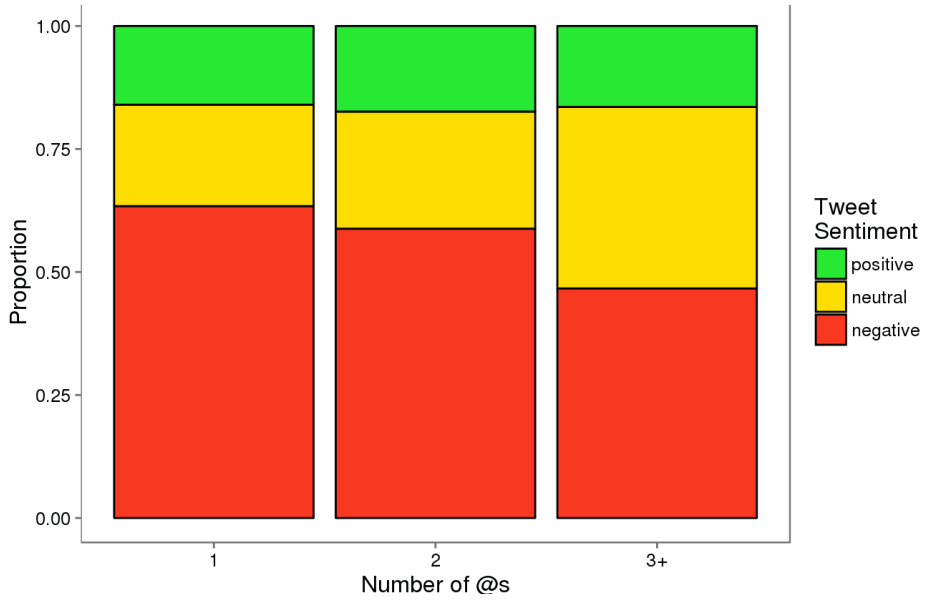



```
# Show counts
> table(Tweets$at_countD)
   1    2   3+
1640  281   63

# Visualize distribution of tweet length by sentiment
> ggplot(Tweets, aes(x = text_length, fill = airline_sentiment)) +
+geom_density(alpha = 0.2) +
+  scale_fill_manual(name   = 'Tweet\nSentiment',
+  values = sentPlt,
+  breaks = sentBreaks) +
```

```
+    geom_vline(xintercept = 170,lwd=1, lty = 'dashed'),labs(x = 'Tweet
Length'),theme_few(),theme(text = element_text(size=12))
```

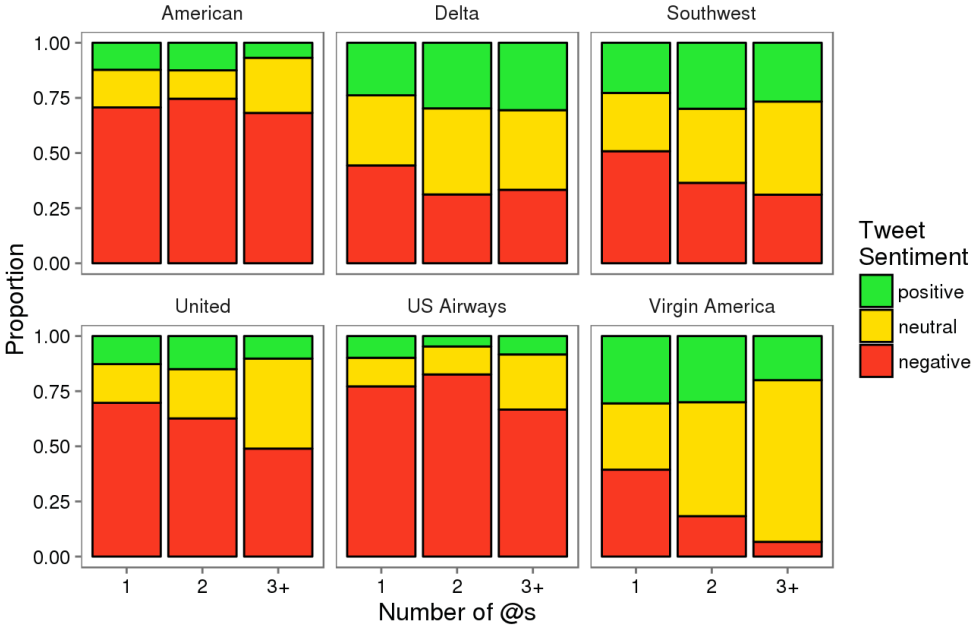



```
# Visualize proportions of positive, neutral, and negative
# sentiment tweets by number of @ symbols used
>     ggplot(Tweets, aesx = at_countD, fill = airline_sentiment)) +
+     geom_bar(position = 'fill', colour = 'black') +
+     scale_fill_manual(name   = 'Tweet\nSentiment',values = sentPlt,
+     breaks = sentBreaks),labs(x = 'Number of @s', y = 'Proportion') +
+     theme_few(),theme(text = element_text(size=12))
```

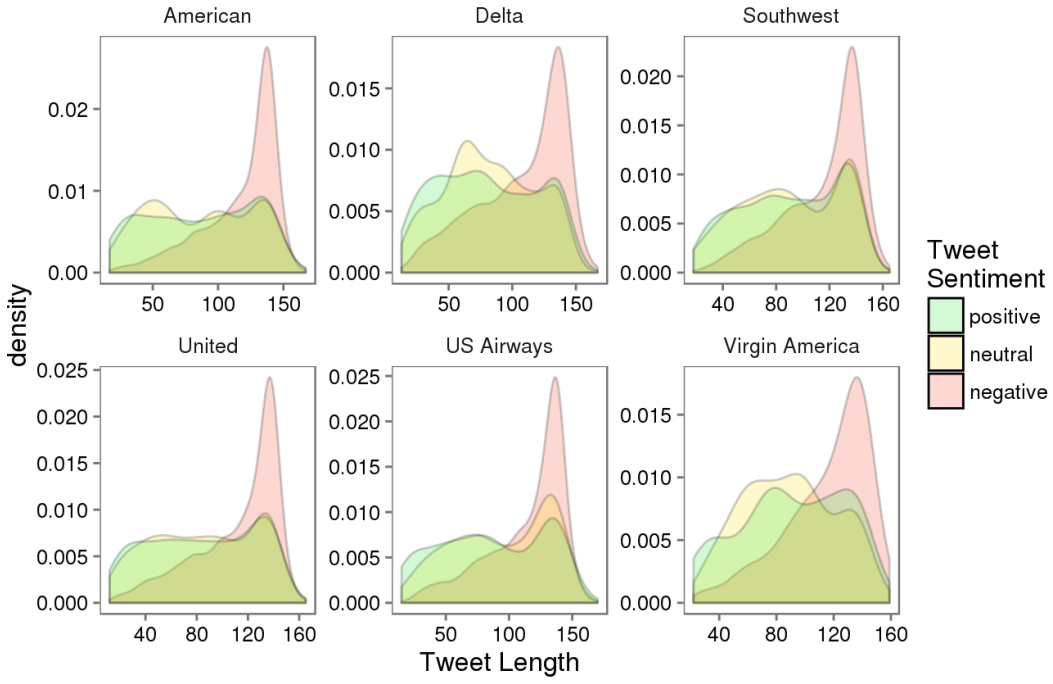


 # Visualize the same plot as before but add airline

```
> ggplot(Tweets, aes(x = at_countD, fill = airline_sentiment)) +
+     geom_bar(position = 'fill', colour = 'black') +
+     facet_wrap(~airline),scale_fill_manual(name   = 'Tweet\nSentiment',
+      values = sentPlt,breaks = sentBreaks),labs(x = 'Number of @s', y =
'Proportion'),theme_few(),theme(text = element_text(size=12))
```




```
>ggplot(Tweets,aes(x=text_length,fill=airline_sentiment)),geom_density(alpha
=0.2),facet_wrap(~airline,scale='free'),scale_fill_manual(name='Tweet\nSenti
ment',values=sentPlt,breaks=sentBreaks),labs(x='TweetLength'),theme_few(),he
me(text = element_text(size=12))
```



Analysis

Every tweet begins with a @airline tag, which indicates the airline towards which the message is directed. To analyze the content of the tweet, this part is not relevant, so it will be removed from the tweet texts. Tweets classified as negative or positive will be analyzed separately.

The cloud of words provide a nice visual representation of the word frequency for each type of sentiment (negative: left or positive: right). The size of the word correlates with its frequency across all tweets. We can get an idea of what people are talking about. For example, for negative sentiment, people seem to complain about cancelled or delayed flights, and hours waiting. However, for positive sentiment, people are mostly thankful and they talk about great service/flight.

We see that in negative tweets, the appearance of the word 'flight' correlates with the appearance of the words 'cancelled', 'late' and 'delayed', indicating that people are complaining about delayed flights. The word 'customer' is associated with the word 'service', which is expected, as customer service was a recurrent issue in negative tweets. Interestingly, the word 'gate' is associated with the words 'waiting' and 'plane', which probably means that people were left waiting at the gate for some time before departure. So from this study, and without having read any tweet, we understand that people are generally complaining about.

For positive sentiment tweets, we observe that the word 'flight' is associated with 'great', suggesting that people have had great flight experiences. The word 'amazing' is associated with the word 'customer', which is in turn associated with the word 'service', indicating that people experienced an amazing customer service in many opportunities. Similarly, without having actually read any tweet, with this analysis we get an idea of what people are saying about the airlines.

For the proposed script :

I use the **stringr** package to count the number of @ symbols in the tweet. Of course if there is only one, then it is the airline. I also use the same package to count the number of characters used in the tweet where the maximum length should be 170.

On taking a closer look at testing the hypotheses, angrier tweets have more @ symbols meaning discontent Twitter users want an audience for their displeasure. While we're at it, we see if the airline makes a difference. It looks like my theory is not quite holding. While tweets containing 1, 2, and 3+ @ symbols have roughly the same proportion of positive tweets, the negativity goes down and neutrality goes up.

I'm ready to move on to looking at tweet length as our variable of interest. As derived before, it appeared that the distribution of negative sentiment tweets was shifted rightward towards longer tweets in comparison to neutral and positive distributions. We see that negative tweets tend to be considerably longer than positive or neutral ones. In fact, it's interesting to see that ceiling effect of the 170 character limit among tweets directed at Virgin America.

Conclusion

Studies in sentiment analysis approaches have existed for more than a decade and now are exploited by enterprises as an important tool for strategic marketing planning and manoeuvring. This move is also due to the advancement in data storage, access and analytics enabled through big data frameworks. However, the big data frameworks regard sentiment analysis as just another possible application that can benefit through its advanced data management. Although several literatures are available that study the challenges of sentiment analysis in the big data frameworks, such as through the volume, velocity and variety issue, the value, veracity and volatility have not been explored as much, though in fact taming the data is key for big data analytics. This paper discusses sentiment analysis approaches and their suitability for the big data framework. The ratio of standard sentiment analysis approaches to the sentiment analysis approaches in big data platform is still huge. Implementation and evaluation of the effectiveness of close monitoring of social customer relationship management is also still scarce although big data technologies adoption is healthy. Gaps in the existing approaches and possible future works are suggested according to each of the big data issues. It is predicted that studies and skills development on sentiment analysis on big data platform for brand monitoring and customer relation management are going to get increasing attention and its growth will be energised by the high demands and a promise of higher revenues for companies. This prediction is supported by analysing the current marketing reports, surveys and summits on SA–based big data analytics for application in customer behaviour understanding and social network comments analysis for consumer sentiments. Furthermore, brand management approaches through sentiment analysis are expanding and creating a marketing tsunami in many organisations, which has got companies to shift focus towards personalisation and a consumer–centric engagement.