**CAPSTONE PROJECT Data Description\_And\_Approach**

Prepared for:

Syed Shariyar Murtaza

CKME 136 - Data Analytics: Capstone Course

Prepared by:

A. B. M. Naimul Huq

Ryerson ID: 500672338

May 25, 2015

Airline Twitter Sentiment Analysis

* **Introduction**

Anyone who travels regularly recognizes that airlines struggle to deliver a consistent, positive customer experience. Through extensive interview and survey work, the American Customer Satisfaction Index (<http://theacsi.org/>) quantifies this impression. As a group, airlines fall close to the bottom of their industry rankings, just above the Postal Service, Health Insurance, Television and Internet Service companies.



**1.1 Brief context about the problem**

A number of airlines from different operators are operating in different routes and assisting people to travel their preferred destination. Due to a great demand of air travel and strong competition between airlines, travellers are getting an increased choice of airline, airport, price and service. However, the standard of services varies. On the contrary, it is important for airline operators to remain competitive in terms of price and being preferred by travellers based on the service quality. Therefore, it is important for airline operators to understand traveller’s emotions and identify factors related to their services which might affect their brand preference.

**1.2 Statement of the Problem**

Air travellers have stated issues related to their travel experience through tweets which needs to be analyzed so that air line operators can identify, understand and fix those issues to improve their overall service quality.

**1.3 Proposal**

The purpose of this report is to perform a **“Sentiment Analysis”** job about problems of each major U.S. airlines regarding their services. We will classify tweets based on emotion (e.g. anger, joy etc.) and polarity. In contrast to the classification of emotions, the classification of polarity function will allow us to classify some text as positive or negative. We will separate key words from the texts based on polarity and visualize with a comparison cloud after removing stop words and applying stemming. This word cloud model will assist airline operators to understand traveller’s sentiment / opinion regarding their services.

* Perform sentiment analysis
* Perform exploratory data analysis
* Prepare a model with tokenized ngrams
* **Dataset**

We will be using the **“Airline Twitter Sentiment”** dataset from **“CrowdFlower”** website.

A sentiment analysis job about the problems of each major U.S. airlines was performed based on this dataset. Twitter data was captured from February, 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service").

From this airline dataset I will primarily use the following attributes for my analysis.

|  |  |
| --- | --- |
| **Attributes** | **Data Characteristics** |
| \_unit\_id | Int |
| \_id | Int |
| \_country | Factor |
| \_city | Factor |
| airline\_sentiment | Factor |
| Airline | Factor |
| Text | Factor |
| tweet\_coord | Factor |
| tweet\_id | Num |
| tweet\_location | Factor |
| user\_timezone | Factor |

However, the dataset has other 16 attributes which we will not considered for this analysis. These attributes are: created\_at, golden, missed, started\_at, tained, channel, trust, worker\_id, region, ip, negativereason, airline\_sentiment\_gold, name, negativereason\_gold, retweet\_count, tweet\_created.

Source: <http://www.crowdflower.com/data-for-everyone> (added: February 12, 2015 by CrowdFlower)

* **Approach**

An overview of the steps that I have considered for this sentiment analysis is stated below.

**Step-1: Download dataset from CrowdFlower**

Download the raw dataset of **“Airline Twitter Sentiment”** from **“CrowdFlower”** website. This dataset contains 27 attributes. However, I have selected 11 attributes with 55783 observations for the analysis and saved the new dataset **“Airline”** as csv file.

**Step-2: Import dataset into R**

Import the **“Airline”** dataset in R to perform analysis. After import the dataset, also check data characteristics to understand the dataset and remove duplicate data to avoid data duplication.

**Step-3: Install libraries**

I will install below libraries to perform the analysis.

**3.1 Library (twitteR):**

Provides an interface to the Twitter web API (e.g. registerTwitterOAuth, searchTwitter)

**3.2 Library (sentiment)**

Sentiment is a R package with tools for sentiment analysis including Bayesian classifiers for positivity/negativity and emotion classification.

**3.3 Library (tm)**

This is a text mining package. This package will be used to perform below tasks.

#Clean unstructured twitter texts to perform sentiment analysis (e.g. remove numbers, remove punctuation, strip whitespace, remove words, stop words, stem document)

#Constructs or coerces to a term-document matrix or a document-term matrix after separating words from twitter texts (e.g. Term Document Matrix, Document Term Matrix)

#Remove Sparse Terms from a Term-Document Matrix (e.g. Remove Sparse Terms)

#Find associations in a document term or term document matrix (e.g. find assocs, find frequent terms)

**3.4 Library(plyr)**

Tools for splitting, applying and combining data (e.g. laply)

**3.5 Library(Rstem)**

Rstem package provides an interface to C code that performs stemming on words.

**3.6 Library (SnowballC)**

An R interface to the C libstemmer library that implements Porter's word stemming algorithm for collapsing words to a common root to aid comparison of vocabulary.

**3.7 Library (RWeka)**

Weka is a collection of machine learning algorithms for data mining tasks written in Java, containing tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

**3.8 Library (ggplot2)**

This is a data visualization package in R.

**3.9 Library(wordcloud)**

Plot a cloud comparing the frequencies of words across documents. (e.g. comparison cloud)

**3.10 Library(RColorBrewer)**

Provides color schemes for maps and other graphics.

**3.11 Library(NLP)**

Basic classes and methods for Natural Language Processing (e.g. tokenize texts).

**Step-4: Prepare text for sentiment analysis**

Clean the data (twitter texts) and prepare for sentiment analysis. This task will require to perform a series of tasks which are stated below.

# remove retweet entities

air\_txt = gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", air\_txt)

# remove at people

air\_txt = gsub("@\\w+", "", air\_txt)

# remove punctuation

air\_txt = gsub("[[:punct:]]", "", air\_txt)

# remove numbers

air\_txt = gsub("[[:digit:]]", "", air\_txt)

# remove html links

air\_txt = gsub("http\\w+", "", air\_txt)

# remove unnecessary spaces

air\_txt = gsub("[ \t]{2,}", "", air\_txt)

air\_txt = gsub("^\\s+|\\s+$", "", air\_txt)

# define "tolower error handling" function

try.error = function(x)

{

# create missing value

y = NA

# tryCatch error

try\_error = tryCatch(tolower(x), error=function(e) e)

# if not an error

if (!inherits(try\_error, "error"))

y = tolower(x)

# result

return(y)

}

# lower case using try.error with sapply

air\_txt = sapply(air\_txt, try.error)

# remove NAs in air\_txt

air\_txt = air\_txt[!is.na(air\_txt)]

names(air\_txt) = NULL

**Step-5: Labeled tweets by polarity**

Tweets can be labeled either by emotion or polarity. In this analysis I will consider polarity option.

**5.1 Sentiment Analysis (Polarity)**

I will use **“Voter”** algorithm to classify polarity. We can perform the same task through **“Bayes”** algorithm. The twitter texts will be labeled against – positive, negative and neutral.

# classify polarity

class\_pol = classify\_polarity(air\_txt, algorithm="voter") ## instead of "bayes" can use "voter"

# get polarity best fit

polarity = class\_pol[,4]

**5.2 Create data frame with the results and obtain some general statistics**

After classification of polarity I will create a data frame with 2 observations – text and polarity. So, this new data frame contains twitter texts which are labeled by polarity.

# data frame with results

air\_sentiment <- data.frame(text=air\_txt,

polarity=polarity, stringsAsFactors=FALSE)

# sort data frame

air\_sentiment = within(air\_sentiment,

polarity <- factor(polarity, levels=names(sort(table(polarity), decreasing=TRUE))))

**5.3 Data Visualization**

I will use ggplot library to illustrate the distribution of polarity.

**Step-6: Labeled key words by polarity**

In this step, I will separate the texts by polarity, remove stop words, apply stemming and create a term document matrix with key words labeled by polarity. Later on I will create a comparison cloud to visualize the key words.

**6.1 separating text by polarity**

pol = levels(factor(air\_sentiment$polarity))

nemo = length(pol)

pol.docs = rep("", nemo)

for (i in 1:nemo)

{

tmp = air\_txt[polarity == pol[i]]

pol.docs[i] = paste(tmp, collapse=" ")

}

**6.2 remove stopwords**

pol.docs <- removeWords(pol.docs, stopwords("english"))

# create corpus

corpus <- Corpus(VectorSource(pol.docs))

# create a copy of corpus to use later as a dictionary for stem completion

mycorpus <- corpus

**6.3 Stem words**

library(SnowballC)

library(tm)

# Stemming

corpus <- tm\_map(corpus, stemDocument, language = "english")

**6.4 Create Term Document Matrix**

tdm <- TermDocumentMatrix(corpus)

colnames(tdm) <- pol

class(tdm)

# Convert it to matrix

tdm <- as.matrix(tdm)

**6.5 Frequent Words and Association**

I will identify terms based on frequency of occurrences. Later on I will use ggplot to visualize the most important terms.

# Find which term has the most occurrences?

max(apply(tdm,1,sum))

which(apply(tdm,1,sum)==3273)

**6.6 Word Cloud**

library(wordcloud)

# Comparison Cloud

comparison.cloud(tdm, colors = brewer.pal(nemo, "Dark2"),

scale = c(3,.5), random.order = FALSE, title.size = 1.5)

# WordCloud

word.freq <- sort(rowSums(tdm), decreasing = T)

wordcloud(words = names(word.freq), freq = word.freq, min.freq = 400,

random.order = F)

**Step-7: Tokenize texts**

In this step, I will tokenize the texts from “Airline” dataset. After tokenization, I will create a data frame with ngrams. This data frame will be used as an input later to develop the model.

**Step-8: Build Model**

In this final step, I will build a model by using Naïve Bayes algorithm.

**8.1 Naive Bayes Model**

The Naive Bayes makes the simplifying assumption that all the features are independent. In other words and in the context of Sentiment Analysis, each token (word or group of words) contributes independently to the sentiment of the whole sentence. Even if this assumption may seem too restrictive, Naive Bayes gives good results as it does not over fit.

**8.2 Evaluation phase**

* Predict the classes of the test instances.

**8.3 Performance criteria**

* Confusion matrix

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .