**Aspect Identification and Sentiment Rating from Customer Reviews**

Problem Statement: The goal of this task is to identify the various aspects from the customer reviews and rate each aspect on a scale of 0(Extremely unhappy) to 1(Happy). The input for the task: <https://www.trustpilot.com/review/www.sonetel.com>

Sentiment analysis refers to the use of natural language processing, text analysis and statistical learning to determine the attitude of a speaker or a writer with respect to some topic. Sentiment analysis is extremely useful in cases such as social media monitoring, online review analysis, Market Analysis etc. as it allows us to gain an overview of the wider public opinion behind certain topics.

**Approach followed in R:**

1)Data Gathering/Web scraping : Firstly we gather data for our model by scrapping the webpages from the above website with the help of library “rvest”.

**2) Identifying review words and performing Sentiment analysis on them.**

**i)Text cleaning:**Stopwords removal, punctuation removal..etc were few of the techniques used to clean the text. This is common to every text analysis problem.

**ii)Sentiment generation:**This method is capable of classifying the sentence as positive, negative and neutral. All the aspects in a review were detected, also the positive and negatve sentiments words were detected. To rate an aspect the total avarage of sentiment ofthatreview was assigned to the aspect. The assumption made was that even if a negative aspect got more points here, other reviews with negative sentiments would eventally bring this negative aspect down.

Firsly install all required libraries using these commands

Install.packages(“tidytext”)

Install.packages (“dplyr”)

Install.packages (“stringr”)

Install.packages (“tm”)

Install.packages (“rvest”)

1) Data Gathering: Scrapping the webpages with the help of library “rvest”.

CODE:

library(rvest)

datalist = list() #an empty list to append our url dataframes

for (i in 1:23) {

URL.base <- <https://www.trustpilot.com/review/www.sonetel.com?page=> #our web address

URL <- paste0(URL.base, i) #we increment our page no. using this paste cmd and assign to URL

#print(URL) #always check at every stop point if code is giving desired output

webpage<-read\_html(URL) # read\_html() is used to read our html page from the web

#################scrap only the review-body for our content#######################

review\_body\_html<-html\_nodes(webpage,'.review-body') #Specifying to choose class review body

review\_body<-html\_text(review\_body\_html ) #Converting to text

review\_body<-gsub("\n","",review\_body) #Removing new lines and extra spaces with gsub

review\_body<-gsub("\r","",review\_body)

review\_body<-gsub('\\s+',' ',review\_body)

review\_body<-as.character(review\_body)

####################dataframing and adding it to a list####################

df<-data.frame(user = user\_names, body = review\_body)

datalist[[i]] <- df

}

big\_data <- do.call(rbind, datalist)

# Combing our list into a single dataframe.

**2) Identifying review words and performing Sentiment analysis on them.**

**CODE:** # Loading all required Libraries

library(tidytext)

library(dplyr)

library(stringr)

library(tm)

Next we create a corpus of our data(big\_data datframe) we collected. This is provided to us by the package (tm). We need to consider all possible words in our corpus. Then we create a “vectorsource”, that is the input type for the “Corpus”function defined in the package “tm”. We then perform cleaning operations on our data. By cleaning we mean Converting it all to lower case, removing all punctuations and stopwords, removing all whitespaces and numbers.

corpus <- Corpus(VectorSource(big\_data$body))

corpus <- tm\_map(corpus, content\_transformer(tolower))

corpus <- tm\_map(corpus, removePunctuation)

corpus <- tm\_map(corpus, removeWords, stopwords("english"))

corpus <- tm\_map(corpus, stripWhitespace)

corpus <- tm\_map(corpus, removeNumbers)

We next define all our aspect words to be detected in the corpus.

aspect\_word = c("Customer","service","quality","sms","support","setup","customize","phoneservice","purchase","Voip","Troubleshotting","callrates","provider","Voicemail","callforwarding","soundquality")

We now create a corpus for our aspect words defined and perform the same operations as above i.e Converting it all to lower case, removing all punctuations and stopwords, removing all whitespaces and numbers.

aspect\_words <- Corpus(VectorSource(aspect\_word))

aspect\_words <- tm\_map(aspect\_words, content\_transformer(tolower))

aspect\_words <- tm\_map(aspect\_words, removePunctuation)

aspect\_words <- tm\_map(aspect\_words, removeWords, stopwords("english"))

aspect\_words <- tm\_map(aspect\_words, stripWhitespace)

aspect\_words <- tm\_map(aspect\_words, removeNumbers)

Load the positive and negative lexicon data from external file

positive\_lexicon <- read.csv("positive-words.txt")

negative\_lexicon <- read.csv("negative-words.txt")

Initialising our counts and defining empty vectors

total\_pos\_count <- 0

total\_neg\_count <- 0

total\_aspect\_count <- 0

pos\_count\_vector <- c()

neg\_count\_vector <- c()

size <- length(corpus)

We now traverse through all the reviews one by one using a loop, and find all aspect word, positive word and negative words in one review. We also count the total number of aspects , positive words and negative words of all reviews. We find these words by using the intersect() Function.

for(i in 1:size){

corpus\_words<- list(strsplit(corpus[[i]]$content, split = " "))

aspect\_words <- list(aspect\_words)

aspects <- intersect(unlist(corpus\_words),unlist(aspect\_words))

aspect\_count <- length(intersect(unlist(corpus\_words),unlist(aspect\_words)))

positive <- intersect(unlist(corpus\_words), unlist(positive\_lexicon))

pos\_count <- length(intersect(unlist(corpus\_words), unlist(positive\_lexicon)))

negative <- intersect(unlist(corpus\_words), unlist(negative\_lexicon))

neg\_count <- length(intersect(unlist(corpus\_words), unlist(negative\_lexicon)))

total\_count\_for\_current\_review <- pos\_count + neg\_count ## current +ve and -ve count

pos\_percentage <- (pos\_count\*100)/total\_count\_for\_current\_review

neg\_percentage <- (neg\_count\*100)/total\_count\_for\_current\_review

#print(pos\_percentage) ## current positive percentage

#print(neg\_percentage) ## current negtive percentage

#print(paste0("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"))

total\_pos\_count <- total\_pos\_count + pos\_count

total\_neg\_count <- total\_neg\_count + neg\_count

total\_aspect\_count <- total\_aspect\_count + aspect\_count

pos\_count\_vector <- append(pos\_count\_vector, pos\_count)

neg\_count\_vector <- append(neg\_count\_vector, neg\_count)

for(j in 1:length(aspects)) {

if( pos\_count>=neg\_count && length(aspects)>=1 ){

z = (neg\_percentage/100) \*2

rate <- as.numeric(z)

aspects <- c(aspects[j])

rating <- data.frame(aspects, rate)

datalist2 <-rbind(datalist2, rating)

}

**Overall Sentiment of all reviews**

# Sentiment score of all reviews

counts <- data.frame(pos\_count\_vector, neg\_count\_vector)

sentiment <- data.frame(c(1:size),(pos\_count\_vector - neg\_count\_vector) / (pos\_count\_vector + neg\_count\_vector))

total\_count <- total\_pos\_count + total\_neg\_count

overall\_positive\_percentage <- (total\_pos\_count\*100)/total\_count

overall\_negative\_percentage <- (total\_neg\_count\*100)/total\_count

cat(" The overall sentiment is ")

if(overall\_positive\_percentage > overall\_negative\_percentage) {

print(paste0("positive with percentage"))

print(overall\_positive\_percentage)

}

if(overall\_positive\_percentage < overall\_negative\_percentage) {

print(paste0("negative with percentage"))

print(overall\_negative\_percentage)

}

if(neg\_count>pos\_count && length(aspects)>=1 ){

z = (neg\_percentage/100)\*2

rate <- as.numeric(z)

aspects <- c(aspects[j])

rating <- data.frame(aspects, rate)

datalist2 <- rbind(datalist2, rating)

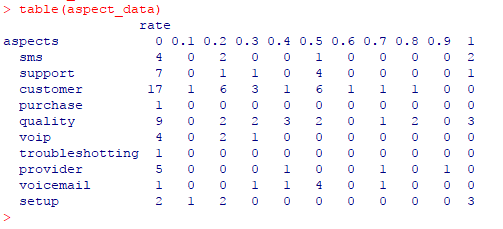
}

}

}

aspect\_data <- data.frame(datalist2)

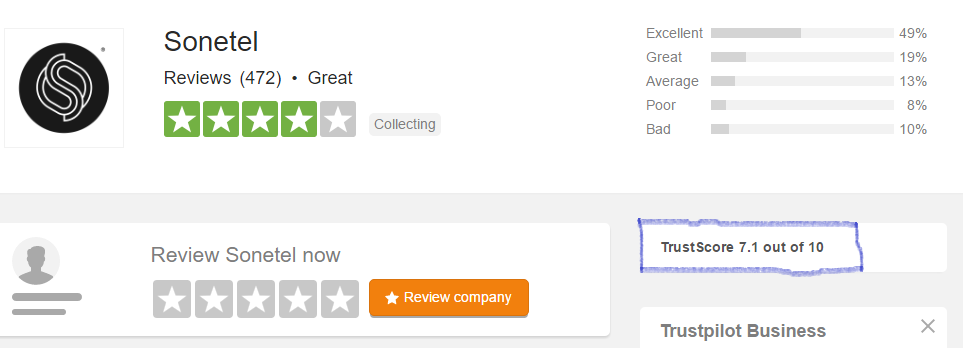
**OUTPUT:**



**The above table shows how many people rated that particular review**

**The overall sentiment of all reviews isC:\Users\me\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2.png**

**Which is close to the star rating(TrustScore) on the website**



**We can get the total average of that aspect by using cmd (change aspects == "provider"accordingly)**

**mean(aspect\_data$rate[aspect\_data$aspects == "provider"],na.rm=TRUE)**