Project4

Shariq Mian

11/15/2021

Libraries

```
library(tm)
library(knitr)
library(plyr)
library(wordcloud)
library(tidyverse)
library(tm)
library(magrittr)
library(data.table)
library(dplyr)
library(randomForest)
library(tidymodels)
library(caTools)
library(gmailr)
```

Accessing ham email messages

```
ham="~/Shariq School/SPS/Data 607/SpamHam/easy_ham"
count_ham=length(list.files(path = ham))
ham_list=list.files(ham)
count_ham
```

```
## [1] 2550
```

Accessing spam email messages

```
spam= "~/Shariq School/SPS/Data 607/SpamHam/spam_2"
count_spam=length(list.files(path = spam))
spam_list=list.files(spam)
count_spam
```

```
## [1] 1397
```

Turing into text terms

```
spam_list=list.files(spam)
ham_text = NA
for(i in 1:length(ham_list))
{
  path=paste0(ham, "/", ham_list[i])
  text =readLines(path)
  list= list(paste(text, collapse="\n"))
  ham_text = c(ham_text,list)

}

spam_text = NA
for(i in 1:length(spam_list))
{
  path=paste0(spam, "/", spam_list[i])
  text =readLines(path)
  list= list(paste(text, collapse="\n"))
  spam_text = c(spam_text,list)
}
```

seperating email body from message

```
email_body <- function(ham_text){
  message = str_split(ham_text,"\n\n") %>% unlist()
  body = paste(message[2:length(message)], collapse=' ' )
  return(body)
}

ham_text <- email_body(ham_text)
```

Creating Corpus

```
# Building a new corpus
ham_corpus =VCorpus(VectorSource(unlist(lapply(ham_text, as.character))))
ham_terms_matrix = TermDocumentMatrix(ham_corpus,control= list(removePunctuation=TRUE, removeNumbers=TRU
ham_corpus = tm_map(ham_corpus, removeNumbers)
ham_corpus = tm_map(ham_corpus, removeWords, stopwords())
ham_corpus = tm_map(ham_corpus, removePunctuation)
ham_corpus = tm_map(ham_corpus, stemDocument)
ham_corpus = tm_map(ham_corpus, stripWhitespace)
```

creating TDM

```
ham_terms_matrix = TermDocumentMatrix(ham_corpus)
```

```
spam_corpus= VCorpus(VectorSource(spam_text))
spam_terms_matrix= TermDocumentMatrix(spam_corpus,control=list(removePunctuation=TRUE, removeNumbers=TRU
spam_corpus = tm_map(spam_corpus, removeNumbers)
spam_corpus = tm_map(spam_corpus, removeWords, stopwords())
spam_corpus = tm_map(spam_corpus, removePunctuation)
spam_corpus = tm_map(spam_corpus, stemDocument)
spam_corpus = tm_map(spam_corpus, stripWhitespace)
```

```
spam_terms_matrix = TermDocumentMatrix(spam_corpus)
```

Creating Spam Data Frame

```
spam_df = as.data.frame(as.table(spam_terms_matrix))
spam_df$spam_ham = "1"
colnames(spam_df) = c('TERM', 'SPAM_DOCS', 'FREQ', 'CLASS')
spam_df = subset(spam_df, select = -c(2) )
spam_df$FREQ[is.na(spam_df$FREQ)] = '0'
```

```
spam_df = ddply(spam_df, .(TERM, CLASS), summarize, FREQ = sum(as.numeric(FREQ)))
```

Creating Ham Data Frame

```
ham_df = as.data.frame(as.table(ham_terms_matrix))
ham_df$spam_ham = "0"
colnames(ham_df) = c('TERM', 'HAM_DOCS', 'FREQ', 'CLASS')
ham_df = subset(ham_df, select = -c(2) )
ham_df$FREQ[is.na(ham_df$FREQ)] = '0'
```

```
ham_df = ddply(ham_df, .(TERM, CLASS), summarize, FREQ = sum(as.numeric(FREQ)))
ham_dfsort=arrange(ham_df, FREQ)
head(ham_dfsort,15)
```

```
##             TERM CLASS FREQ
## 1  \006argotech     0    1
## 2     \023c\024     0    1
## 3    comments       0    1
## 4      quizzes      0    1
## 5            ¿ll     0    1
## 6         ‘adolf     0    1
## 7          ‘boot     0    1
## 8          ‘dear     0    1
## 9         ‘don’t     0    1
## 10        ‘he’ll     0    1
## 11       ‘hello’     0    1
## 12          ‘how     0    1
## 13        ‘its    0    1
## 14     ‘johnni    0    1
## 15        ‘ma’    0    1
```

Combining DF

```
# Bind the data frames
spam_ham_df = rbind(spam_df,ham_df)
head(spam_ham_df)
```

```
##                                                                                         TERM
## 1                                                                                      -even
## 2                                                                                       font
## 3                                                                                       most
## 4                                                                                       ¡¡¡¡
## 5 ¡¡¡¡¡¶ä§¹íó¢óï¡·ôãû½ð¡°í»ææó¢óï¡±£¬ó¢îäãû³æ½ð¡°eenglish¡±£¬êçò»öö×¢öøêµð§µäó¢óï×ôðþ¿î³ì¡£
## 6                                                             ¡¡¡¡¡¶ä§¹íó¢óï¡·òôæä²»óãñ§óï·¨
##   CLASS FREQ
## 1     1    2
## 2     1    1
## 3     1    1
## 4     1    8
## 5     1    2
## 6     1    2
```

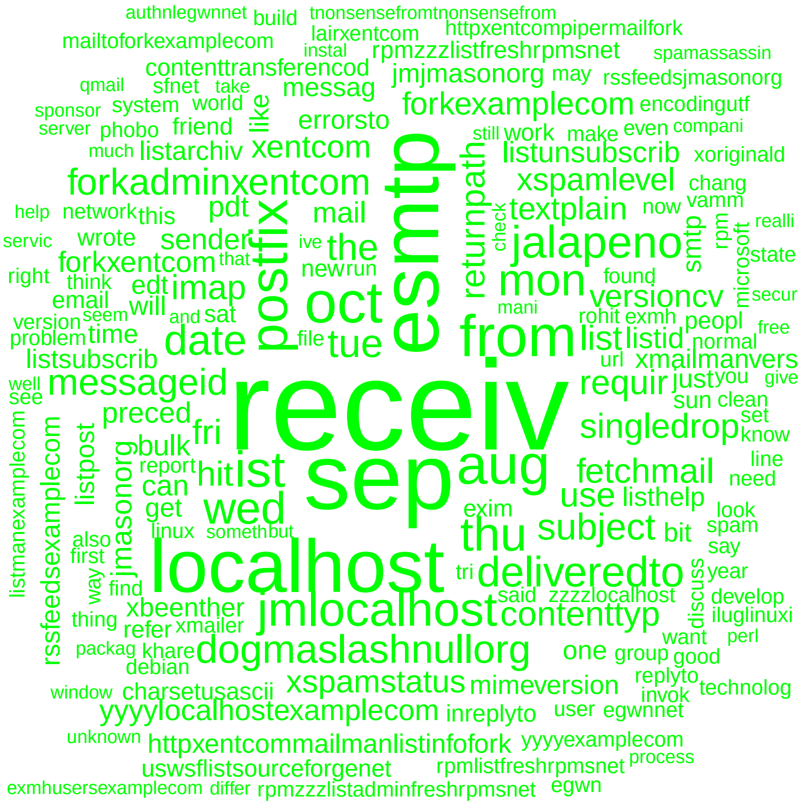
Randomizing Data/Shuffle

```
spam_ham_df<- spam_ham_df[sample(nrow(spam_ham_df)),]
head(spam_ham_df, n=20)
```

```
##                                                                  TERM CLASS FREQ
## 20699                                                ffeeaeabcafcfddd     1    1
## 80133                                                           jitsr     0    3
## 15986   dgvdchbglnbjpjzwzxinpjxzcgfudqpzdhlsztnysbicmvkjzivtpziahcvw     1    1
## 44727                                                       programsb     1    1
## 63632                                                           back’     0    2
## 75522                                                           gsbaz     0    1
## 50036                                                 sizedissuedfont     1    1
## 14196                                                            curb     1    3
## 9650                                                    ccbbabfeeacdd     1    2
## 79210                                                         illplac     0    1
## 94203                                                            word     0  258
## 91661                                                       thermomet     0   15
## 73251                                                            ggtu     0    1
## 6419   biizxlvbmrpbwvuclvbijbvblyvzdgtzxivzwhawwvawwbjlzeynipbwfnzxmv     1    1
## 79591                                                          instal     0  433
## 67281                                                   ctypejusthtml     0    2
## 34650                                                            kvpb     1    1
## 45778                                                    quotstandard     1    4
## 69614                                                           edead     0    1
## 12637                                              colorffffffbbeauti     1    1
```

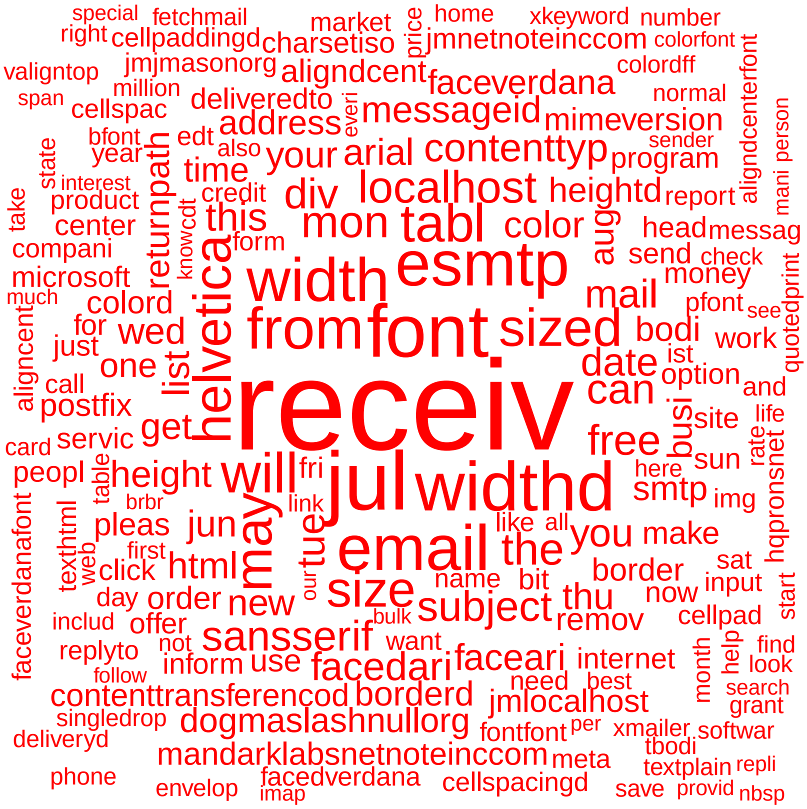
Wordcloud Ham Corpus

```r
wordcloud(ham_corpus, max.words = 200, random.order = FALSE, colors=c('green'))
```



Wordcloud Spam Corpus

```r
wordcloud(spam_corpus, max.words = 200, random.order = FALSE, colors=c('red'))
```



```
spam_ham_df<- spam_ham_df[sample(nrow(spam_ham_df)),]
```

Changing data to ready for Model use

```
spam_ham_df$CLASS=factor(spam_ham_df$CLASS)
spam_ham_df$CLASS <- as.numeric(as.character(spam_ham_df$CLASS))
spam_ham_df=spam_ham_df[c("TERM", "CLASS")]
```

Split data into train test

```
set.seed(1024)
split = sample.split(spam_ham_df$CLASS, SplitRatio = 0.8)
training = subset(spam_ham_df, split == TRUE)
testing = subset(spam_ham_df, split == FALSE)
noob =  ncol(training) - 1
```

```
head(training$CLASS)
```

```
## [1] 1 0 1 1 1 1
```

Random Forest Classifier

```
classifier = randomForest(x = training[-noob],y = training$CLASS,ntree = 3)
```

```
## Warning in randomForest.default(x = training[-noob], y = training$CLASS, :
## The response has five or fewer unique values. Are you sure you want to do
## regression?
```

Accuracy

In confusion matrix, I didnt have any false negative and my accuracy was 100%.

```
y_predictor = predict(classifier, newdata = testing[-noob])
confusion_matrix <- table(y_predictor>0,testing$CLASS)
confusion_matrix
```

```
##
##            0     1
##   TRUE  6683 12345
```

I want to connect to my own gmail to test it on some spam and ham emails in my gmail and see what accuracy I get.

```
gm_auth_configure(path  = "credentials.json")
gm_auth(email = TRUE, cache = ".secret")
```

```
## ! Using an auto-discovered, cached token.

##   To suppress this message, modify your code or options to clearly consent to
##   the use of a cached token.

##   See gargle's "Non-interactive auth" vignette for more details:

##   <https://gargle.r-lib.org/articles/non-interactive-auth.html>

## i The gmailr package is using a cached token for 'mianshariq@gmail.com'.
```

```
msgs = gm_messages(search="before:2021/15/11 after:2021/11/01", num_results = 5, label_ids="Spam")
```

```
msgs
```

```
ids = gmailr::gm_id(msgs, what="message_id")
o = gmail.sentiment(ids)

write.table(o, "./gmail_text_analysis.csv", sep=",", row.names=F)
```

```
gmail="~/Shariq School/SPS/Data 607/SpamHam/Gmail"
count_gmail=length(list.files(path = gmail))
gmail_list=list.files(gmail)
count_gmail
```

```
## [1] 2
```

```
gmail_text = NA
for(i in 1:length(gmail_list))
{
  path=paste0(gmail, "/", gmail_list[i])
  text =readLines(path)
  list= list(paste(text, collapse="\n"))
  gmail_text = c(gmail_text,list)

}
```

Building a new Gmail corpus

```
gmail_corpus =VCorpus(VectorSource(unlist(lapply(gmail_text, as.character))))
gmail_terms_matrix = TermDocumentMatrix(gmail_corpus,control= list(removePunctuation=TRUE, removeNumbers
gmail_corpus = tm_map(gmail_corpus, removeNumbers)
gmail_corpus = tm_map(gmail_corpus, removeWords, stopwords())
gmail_corpus = tm_map(gmail_corpus, removePunctuation)
gmail_corpus = tm_map(gmail_corpus, stemDocument)
gmail_corpus = tm_map(gmail_corpus, stripWhitespace)
gmail_terms_matrix = TermDocumentMatrix(gmail_corpus)
```

```
gmail_df = as.data.frame(as.table(gmail_terms_matrix))
gmail_df$gmail_spam = "1"
colnames(gmail_df) = c('TERM', 'gmail_DOCS', 'FREQ', 'CLASS')
gmail_df = subset(gmail_df, select = -c(2) )
gmail_df$FREQ[is.na(gmail_df$FREQ)] = '0'
gmail_df = ddply(gmail_df, .(TERM, CLASS), summarize, FREQ = sum(as.numeric(FREQ)))
head(gmail_df, n = 20)
```

```
##               TERM CLASS FREQ
## 1             abus     1    1
## 2           access     1    1
## 3          account     1    2
## 4            addit     1    1
## 5          address     1    2
## 6         advertis     1    2
## 7  advertisement     1    1
## 8            among     1    1
## 9            appli     1    2
## 10          applic     1    1
## 11          approv     1    2
## 12             are     1    1
## 13          attn     1    1
## 14         autom     1    1
## 15         avail     1    2
## 16          bank     1    2
## 17      benefici     1    1
## 18           box     1    1
## 19        button     1    1
## 20           can     1    1
```

```
gmail1="~/Shariq School/SPS/Data 607/SpamHam/Gmail1"
count_gmail1=length(list.files(path = gmail1))
gmail1_list=list.files(gmail1)
count_gmail1
```

```
## [1] 2
```

```
gmail1_text = NA
for(i in 1:length(gmail1_list))
{
  path=paste0(gmail1, "/", gmail1_list[i])
  text =readLines(path)
  list= list(paste(text, collapse="\n"))
  gmail1_text = c(gmail1_text,list)

}
```

```
## Warning in readLines(path): incomplete final line found on '~/Shariq School/SPS/
## Data 607/SpamHam/Gmail1/gmail1.txt'
```

Building a new Gmail corpus

```
gmail1_corpus =VCorpus(VectorSource(unlist(lapply(gmail1_text, as.character))))
gmail1_terms_matrix = TermDocumentMatrix(gmail1_corpus,control= list(removePunctuation=TRUE, removeNumbe
gmail1_corpus = tm_map(gmail1_corpus, removeNumbers)
gmail1_corpus = tm_map(gmail1_corpus, removeWords, stopwords())
gmail1_corpus = tm_map(gmail1_corpus, removePunctuation)
gmail1_corpus = tm_map(gmail1_corpus, stemDocument)
gmail1_corpus = tm_map(gmail1_corpus, stripWhitespace)
gmail1_terms_matrix = TermDocumentMatrix(gmail1_corpus)
```

```
gmail1_df = as.data.frame(as.table(gmail1_terms_matrix))
gmail1_df$gmail1_ham = "0"
colnames(gmail1_df) = c('TERM', 'gmail1_DOCS', 'FREQ', 'CLASS')
gmail1_df = subset(gmail1_df, select = -c(2) )
gmail1_df$FREQ[is.na(gmail1_df$FREQ)] = '0'
gmail1_df = ddply(gmail1_df, .(TERM, CLASS), summarize, FREQ = sum(as.numeric(FREQ)))
head(gmail1_df, n = 20)
```

```
##                         TERM CLASS FREQ
## 1                       â\200"     0    1
## 2                   abubakar     0    3
## 3                     accept     0    1
## 4                      addit     0    1
## 5                     alissa     0    1
## 6                   american     0    1
## 7              amianyahoocom     0    1
## 8                     attach     0    1
## 9                 attachment     0    1
## 10                     begin     0    1
## 11                       big     0    1
## 12                       can     0    1
## 13                    candid     0    1
## 14                   certifi     0    1
## 15              chemistrysci     0    1
## 16                     chose     0    1
## 17              completeâ\200¯     0    1
## 18 completeâ\200¯thisâ\200¯within     0    1
## 19                   congrat     0    1
## 20                      date     0    1
```

```
gmail_ham_df = rbind(gmail_df,gmail1_df)
gmail_ham_df$CLASS=factor(gmail_ham_df$CLASS)
gmail_ham_df$CLASS <- as.numeric(as.character(gmail_ham_df$CLASS))
gmail_ham_df=gmail_ham_df[c("TERM", "CLASS")]
gmail_ham_df<- gmail_ham_df[sample(nrow(gmail_ham_df)),]
head(gmail_ham_df)
```

```
##        TERM CLASS
## 217   iphon     0
## 203 forward     0
## 137    sent     1
## 35    creek     1
## 97     name     1
## 34   credit     1
```

```
testing_gmail = gmail_ham_df
head(testing_gmail)
```

```
##        TERM CLASS
## 217   iphon     0
## 203 forward     0
## 137    sent     1
## 35    creek     1
## 97     name     1
## 34   credit     1
```

Random Forest Classifier

```
classifier = randomForest(x = training[-noob],y = training$CLASS,ntree = 3)
```

```
## Warning in randomForest.default(x = training[-noob], y = training$CLASS, :
## The response has five or fewer unique values. Are you sure you want to do
## regression?
```

Accuracy

In confusion matrix, I dint have any false negative and my accuracy was 100%.

```
y_predictor1 = predict(classifier, newdata = testing_gmail[-noob])
confusion_matrix1 <- table(y_predictor>0,testing_gmail$CLASS)
confusion_matrix1
```