DATA 607 - Project 4

Chitrarth Kaushik

4/27/2020

Creating dataframes including ham and spam files

```
ham_files="C:/MSDS/DATA 607/Project 4/easy_ham"
files_nms_ham = list.files(ham_files)
spam_files="C:/MSDS/DATA 607/Project 4/spam_2"
files_nms_spam = list.files(spam_files)
```

Creating a list of docs and creating data frames

```
docs_ham <- NA
for(i in 1:length(files_nms_ham))
{
  file_path_h<-paste(ham_files, sep="/", files_nms_ham[1])
  text_ham <-readLines(file_path_h)
  text_list_ham<- list(paste(text_ham, collapse="\n"))
  docs_ham = c(docs_ham,text_list_ham)

}

docs_spam <- NA
for(i in 1:length(files_nms_spam))
{
  file_path_s<-paste(spam_files, sep="/", files_nms_spam[1])
  text_spam <-readLines(file_path_s)
  text_list_spam<- list(paste(text_spam, collapse="\n"))
  docs_spam = c(docs_spam,text_list_spam)

}

# creating ham data frame
ham_data <-as.data.frame(unlist(docs_ham),stringsAsFactors = FALSE)
ham_data$type <- "ham"
colnames(ham_data) <- c("text","type")

# creating spam data frame
spam_data <-as.data.frame(unlist(docs_spam),stringsAsFactors = FALSE)
spam_data$type <- "spam"
colnames(spam_data) <- c("text","type")
```

```r
#combining data frames
combined_data <- rbind(ham_data, spam_data)
```

Cleaning the data to create the corpus

```r
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.6.3
```

```
## Loading required package: NLP
```

```r
library(SnowballC)
```

```
## Warning: package 'SnowballC' was built under R version 3.6.3
```

```r
corpus_clean = VCorpus(VectorSource(combined_data$text))
corpus_clean = tm_map(corpus_clean, content_transformer(tolower))
corpus_clean = tm_map(corpus_clean, removeNumbers)
corpus_clean = tm_map(corpus_clean, removePunctuation)
corpus_clean = tm_map(corpus_clean, removeWords, stopwords())
corpus_clean = tm_map(corpus_clean, stemDocument)
corpus_clean = tm_map(corpus_clean, stripWhitespace)
```

creating document matrix and removing sparse terms

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
doc_matrix <- DocumentTermMatrix(corpus_clean)
doc_matrix = removeSparseTerms(doc_matrix, 0.97)
final_data = as.data.frame(as.matrix(doc_matrix))
final_data$type = combined_data$type
final_data<-final_data %>% mutate(class=if_else(`type`== "spam",1,0))
final_data <- subset(final_data, select = -type )

spam_data_clean <- final_data %>% filter(`class` == 1 )
nrow(spam_data_clean)
```

```
## [1] 1398
```

```
ham_data_clean <- final_data %>% filter(`class` == 0 )
nrow(ham_data_clean)

## [1] 2502
```

#splitting data into development and validation sample

```
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)

## Warning: package 'caTools' was built under R version 3.6.3

set.seed(123)
flag <- sample.split(final_data$class, SplitRatio = 0.7)

development_sample = subset(final_data, flag == TRUE)
validation_sample = subset(final_data, flag == FALSE)

num_obs_d<-nrow(development_sample)
num_obs_d

## [1] 2730

num_obs_v<-nrow(validation_sample)
num_obs_v

## [1] 1170

num_obs<-ncol(validation_sample) - 1
num_obs

## [1] 343
```

using random forest as the classifier algorithm

```
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.6.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.6.3

rf = randomForest(x = development_sample[-num_obs],
                          y = development_sample$class,
                          ntree = 3, keep.forest = TRUE)

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

classifier <- naiveBayes(development_sample,
factor(development_sample$class))

#predicting using the random forest created
pred = predict(classifier, newdata = validation_sample)

#preparing the confusion matrix
table(pred, validation_sample$class)

##
## pred   0   1
##    0 751   1
##    1   0 418
```