````
---
title: "Week Five Part 2 - Document Classification"
author: "Brian K. Liles"
date: "July 7, 2019"
output:
  html_document: default
  pdf_document: default
---
#Overview
It can be useful to be able to classify new "test" documents using already classified "training" documents.  A common example is using a corpus of labeled spam and ham (non-spam) e-mails to predict whether or not a new document is spam.

#Goal
For this assignment I will be using the **spam** data set, which was downloaded from www.kaggle.com

#Libraries
```{r, include = FALSE}
library(tidyverse)
library(pROC)
library(quanteda)
```
#Data
Data was copied from **Kaggle** and added to the github page listed within the **read.csv** statement.

With the **tbl_df** statement from the **dplyr** package a data frame was created.
```{r}
spam <- read.csv("https://raw.githubusercontent.com/LilesB/Data-620/master/spam.csv", header=TRUE, sep = ",", quote = '\"\"', stringsAsFactors=FALSE)
```
Using the **glimpse** function from the **tidyverse** we will look at the **spam** dataset
```{r}
glimpse(spam)
```
Based off the information provided by the **glimpse** function, we will remove the last three variables.
```{r}
spam <- spam %>%
    select(v1,v2)

glimpse(spam)
```
Now we have a manageable data set with 5,572 observations and 2 variables to work with. Next, we will change the names of the columns.

```{r}
colnames(spam) <- c("email","contents")
glimpse(spam)
```

Next, we will use the **table** function to determine the tally of ham vs spam
```{r}
cat("Frequency of Ham & Spam Emails","\n")
table(spam$email)
```
To create a visual, we will utilize the **ggplot** package to view a bar chart of the data
```{r}
ggplot(data = spam, aes(x = email)) +
  geom_bar(fill = "gray", width = 0.5) +
  xlab("Email Variable") +
  ylab("Number of Emails") +
  ggtitle("Distribution of Emails \n Ham versus Spam")
```

#Training & Test Data Sets
```{r}
spamTrain <- spam[1:4458,]
spamTest <- spam[4458:nrow(spam),]
```

```{r}
# check the allocation of spam/no spam data for the training data set
table(spamTrain$email)
```
Here we see that 3,856 ham emails and 602 spam emails

```{r}
# check the allocation of spam/no spam data for the testing data set
table(spamTest$email)
```
Here we see that 970 ham emails and 145 spam emails

#Naive Bayes Classifier
According to https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54 **Naive Bayes** classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.
````

With the **quanteda** package, our first step is to create a corpus based on the **content** column

```{r}
# construct a corpus object based on data in the content column
contentCorpus <- corpus(spam$contents)

# assign value to the email column
docvars(contentCorpus) <- spam$email
```

Next, we will create a document-feature matrix based off **contentCorpus**
```{r}
dfm <- dfm(contentCorpus, tolower=TRUE)

# set the minimum and maximum frequencies
dfm <- dfm_trim(dfm, min_docfreq = 3)

dfm <- dfm_weight(dfm)
```

```{r}
dfmTrain <- dfm[1:4458,]
dfmTest <- dfm[4458:nrow(spam),]
```

We can now run the Naive Bayes classifier
```{r}
(naiveBayes <- textmodel_nb(dfmTrain,spamTrain[,1]))
```
Next, we will run a prediction utilizing the **predict** function
```{r}
prediction <- predict(naiveBayes,dfmTest)
```

Next, using the **table** function we will view the predictions
```{r}
table(prediction, actual = spamTest[,1])
```

Lastly, we will check the accuracy of the model by using the **pROC** package and also checking the accuracy of the test
```{r}
mean(prediction == spamTest[,1])*100
```

```{r}
predictNum <- ifelse(prediction == "spam",1,2)
aucTest <- roc(as.factor(spamTest[,1]),predictNum)
plot(aucTest)
```