Meghana_Nadig_Assignment 4

Q.1 SMS message filtering example

Step 1 - Collecting Data

```
# Importing CSV data

sms_raw <- read.csv("C:/Users/Meghana Nadig/Downloads/Assignment 4/Assignment4.csv", stringsAsFactors =

str(sms_raw)
```

```
## 'data.frame':    5574 obs. of  2 variables:
##  $ type: chr  "ham" "ham" "spam" "ham" ...
##  $ text: chr  "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... C
```

Step 2 - Exploring and preparing the data

```
# Converting "type" which is a character variable into a factor

sms_raw$type <- factor(sms_raw$type)

str(sms_raw$type)
```

```
##  Factor w/ 2 levels "ham","spam": 1 1 2 1 1 2 1 1 2 2 ...
```

```
table(sms_raw$type)
```

```
##
##  ham spam
## 4827  747
```

```
# Installing the text mining package
#install.packages("NLP")
#install.packages("tm")

library(NLP)
library(tm)
```

Data Preparation - Cleaning and Standardizing text data

```
# Creating a corpus

sms_corpus <- VCorpus(VectorSource(sms_raw$text))

print(sms_corpus)
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 5574
```

```
# Viewing summary of first-two messeges

inspect(sms_corpus[1:2])
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 2
```

```
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 111
##
## [[2]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 29
```

```
# Viewing the actual message text

as.character(sms_corpus[[1]])
```

```
## [1] "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there
```

```
# Viewing multiple documents

lapply(sms_corpus[1:2], as.character)
```

```
## $`1`
## [1] "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there
##
## $`2`
## [1] "Ok lar... Joking wif u oni..."
```

```
# Cleaning the corpus and standardizing the messages to use only lowercase characters

sms_corpus_clean <- tm_map(sms_corpus, content_transformer(tolower))

as.character(sms_corpus[[1]])
```

```
## [1] "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there
```

```
as.character(sms_corpus_clean[[1]])
```

```
## [1] "go until jurong point, crazy.. available only in bugis n great world la e buffet... cine there
```

```
#install.packages("SnowballC")

library(SnowballC)

# Removing numbers from SMS messages

sms_corpus_clean <- tm_map(sms_corpus_clean, removeNumbers)

# Removing StopWords

sms_corpus_clean <- tm_map(sms_corpus_clean, removeWords, stopwords())

# Removing Punctuations

sms_corpus_clean <- tm_map(sms_corpus_clean, removePunctuation)

# Stemming
```

```
# Applying WordStem

sms_corpus_clean <- tm_map(sms_corpus_clean, stemDocument)

# Removing blankspaces

sms_corpus_clean <- tm_map(sms_corpus_clean, stripWhitespace)

# Comapring before and after cleaning SMS messages

as.character(sms_corpus[1:3])
```

```
## [1] "list(list(content = \"Go until jurong point, crazy.. Available only in bugis n great world la e
## [2] "list()"
## [3] "list()"
```

```
as.character(sms_corpus_clean[1:3])
```

```
## [1] "list(list(content = \"go jurong point crazi avail bugi n great world la e buffet cine got amor
## [2] "list()"
## [3] "list()"
```

Data Preparation - Splitting text documents into words

```
# Tokenizing by creating DTM matrix

sms_dtm <- DocumentTermMatrix(sms_corpus_clean)
```

Data Preparation- Creating training and test datasets

```
# Training dataset

train <- sms_dtm[1:4180,]

# Testing dataset

test <- sms_dtm[4181:5574,]

# Creating labels of training and testing datasets

sms_train_labels <- sms_raw[1:4180,]$type

sms_test_labels <- sms_raw[4181:5574,]$type

# Checking the subsets

prop.table(table(sms_train_labels))
```

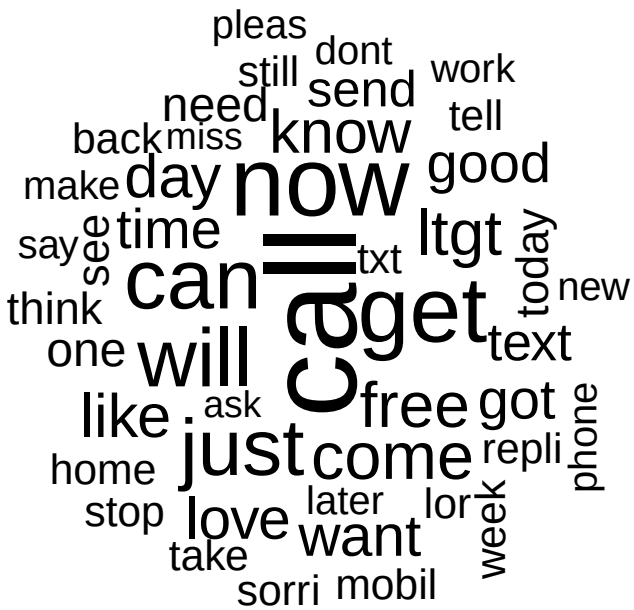
```
## sms_train_labels
##       ham      spam
## 0.8648325 0.1351675
```

```
prop.table(table(sms_test_labels))
```

```
## sms_test_labels
##       ham      spam
## 0.8694405 0.1305595
```

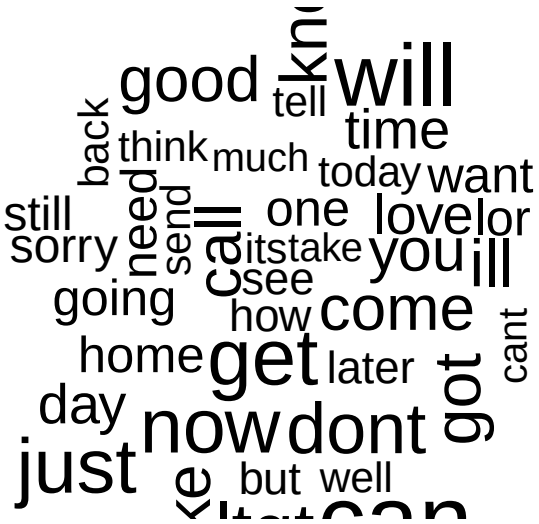
Visualizing text data - Word Clouds

```
#install.packages("wordcloud")
#install.packages("RColorBrewer")
library(RColorBrewer)
library(wordcloud)

wordcloud(sms_corpus_clean, min.freq = 126, random.order = FALSE)
```




```
# creating spam & ham subset

spam<- subset(sms_raw, type == "spam")

ham <- subset(sms_raw, type == "ham")

# WordCloud the spam and ham subset

wordcloud(spam$text, max.words = 40, scale= c(3, 0.5))
```


```
wordcloud(ham$text, max.words = 40, scale= c(3, 0.5))
```



Data Preparation - creating indicator features for frequent words

```
sms_freq_words <- findFreqTerms(train, 5)

str(sms_freq_words)
```

```
##  chr [1:1164] "â£wk" "â€¦" "â€"" "abiola" "abl" "abt" "accept" ...
```

```
# Filtering DTM

sms_dtm_freq_train <- train[,sms_freq_words]

sms_dtm_freq_test <- test[,sms_freq_words]
```

```
# Converting counts to yes or no

convert_counts <- function(x) {
  x <- ifelse(x > 0, "Yes", "No")
}
```

```
# Applying convert_counts() to each of the columns

sms_train <- apply(sms_dtm_freq_train, MARGIN = 2, convert_counts)

sms_test <- apply(sms_dtm_freq_test, MARGIN = 2, convert_counts)
```

Step 3 - Training a model on the data

```r
#install.packages("e1071")

library(e1071)
```

```r
# Using naivebayes function from the package

sms_classifier <- naiveBayes(sms_train, sms_train_labels)
```

Step 4 - Evaluating model performance

```r
# Making prediction

sms_test_pred <- predict(sms_classifier, sms_test)

library(gmodels)

# Comparing predictions

CrossTable(sms_test_pred, sms_test_labels, prop.chisq = FALSE, prop.t = FALSE, dnn = c('predicted', 'ac
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1394
##
##
##              | actual
##    predicted |       ham |      spam | Row Total |
## -------------|-----------|-----------|-----------|
##          ham |      1203 |        20 |      1223 |
##              |     0.984 |     0.016 |     0.877 |
##              |     0.993 |     0.110 |           |
## -------------|-----------|-----------|-----------|
##         spam |         9 |       162 |       171 |
##              |     0.053 |     0.947 |     0.123 |
##              |     0.007 |     0.890 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1212 |       182 |      1394 |
##              |     0.869 |     0.131 |           |
## -------------|-----------|-----------|-----------|
##
##
```

Step 5 - Improving model performance

```r
# Building naivebayes model with laplace = 1

sms_classifier2 <- naiveBayes(sms_train, sms_train_labels, laplace = 1)
```

```
# Making predictions

sms_test_pred2 <- predict(sms_classifier2, sms_test)

# Comparing predictions

CrossTable(sms_test_pred2, sms_test_labels, prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE, dnn = c
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1394
##
##
##              | actual
##    predicted |       ham |      spam | Row Total |
## -------------|-----------|-----------|-----------|
##          ham |      1205 |        28 |      1233 |
##              |     0.994 |     0.154 |           |
## -------------|-----------|-----------|-----------|
##         spam |         7 |       154 |       161 |
##              |     0.006 |     0.846 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1212 |       182 |      1394 |
##              |     0.869 |     0.131 |           |
## -------------|-----------|-----------|-----------|
##
##
```

Q.2 Naive bayes for iris data

```
# Installing the klaR package

#install.packages("klaR")
library(klaR)
```

```
## Loading required package: MASS
```

```
data(iris)
```

```
# Checking first few rows of data
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
# identify indexes to be in testing dataset
# every index of 5th, 10th, 15th..will be the testing dataset
# the rest are training dataset

testidx <- which(1:length(iris[,1]) %% 5 == 0)

testidx
```

```
##  [1]   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85
## [18]  90  95 100 105 110 115 120 125 130 135 140 145 150
```

```r
# seperate into training and testing datasets

iristrain <- iris[-testidx,]

iristest <- iris[testidx,]

iristrain
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 31           4.8         3.1          1.6         0.2     setosa
## 32           5.4         3.4          1.5         0.4     setosa
## 33           5.2         4.1          1.5         0.1     setosa
## 34           5.5         4.2          1.4         0.2     setosa
## 36           5.0         3.2          1.2         0.2     setosa
## 37           5.5         3.5          1.3         0.2     setosa
## 38           4.9         3.6          1.4         0.1     setosa
## 39           4.4         3.0          1.3         0.2     setosa
## 41           5.0         3.5          1.3         0.3     setosa
## 42           4.5         2.3          1.3         0.3     setosa
## 43           4.4         3.2          1.3         0.2     setosa
```

```
## 44          5.0         3.5          1.6         0.6     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
## 103         7.1         3.0          5.9         2.1  virginica
## 104         6.3         2.9          5.6         1.8  virginica
## 106         7.6         3.0          6.6         2.1  virginica
## 107         4.9         2.5          4.5         1.7  virginica
## 108         7.3         2.9          6.3         1.8  virginica
## 109         6.7         2.5          5.8         1.8  virginica
## 111         6.5         3.2          5.1         2.0  virginica
```

```
## 112          6.4         2.7          5.3         1.9  virginica
## 113          6.8         3.0          5.5         2.1  virginica
## 114          5.7         2.5          5.0         2.0  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
```

```
iristest
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 5            5.0         3.6          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 35           4.9         3.1          1.5         0.2     setosa
## 40           5.1         3.4          1.5         0.2     setosa
## 45           5.1         3.8          1.9         0.4     setosa
## 50           5.0         3.3          1.4         0.2     setosa
## 55           6.5         2.8          4.6         1.5 versicolor
## 60           5.2         2.7          3.9         1.4 versicolor
## 65           5.6         2.9          3.6         1.3 versicolor
## 70           5.6         2.5          3.9         1.1 versicolor
## 75           6.4         2.9          4.3         1.3 versicolor
## 80           5.7         2.6          3.5         1.0 versicolor
## 85           5.4         3.0          4.5         1.5 versicolor
## 90           5.5         2.5          4.0         1.3 versicolor
## 95           5.6         2.7          4.2         1.3 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
```

```
## 105          6.5         3.0          5.8         2.2  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
# apply Naive Bayes

nbmodel <- NaiveBayes(Species~., data = iristrain)
nbmodel
```

```
## $apriori
## grouping
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## $tables
## $tables$Sepal.Length
##              [,1]      [,2]
## setosa     4.9975 0.3675892
## versicolor 5.9900 0.5295378
## virginica  6.6100 0.6647922
##
## $tables$Sepal.Width
##              [,1]      [,2]
## setosa     3.4175 0.3960623
## versicolor 2.7775 0.3415556
## virginica  2.9700 0.3081791
##
## $tables$Petal.Length
##              [,1]      [,2]
## setosa     1.4425 0.1583367
## versicolor 4.3100 0.4850588
## virginica  5.5575 0.5930743
##
## $tables$Petal.Width
##              [,1]      [,2]
## setosa     0.2525 0.1109111
## versicolor 1.3325 0.2080280
## virginica  2.0300 0.2355572
##
##
## $levels
## [1] "setosa"     "versicolor" "virginica"
##
## $call
## NaiveBayes.default(x = X, grouping = Y)
##
## $x
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
```

```
## 1          5.1         3.5         1.4         0.2
## 2          4.9         3.0         1.4         0.2
## 3          4.7         3.2         1.3         0.2
## 4          4.6         3.1         1.5         0.2
## 6          5.4         3.9         1.7         0.4
## 7          4.6         3.4         1.4         0.3
## 8          5.0         3.4         1.5         0.2
## 9          4.4         2.9         1.4         0.2
## 11         5.4         3.7         1.5         0.2
## 12         4.8         3.4         1.6         0.2
## 13         4.8         3.0         1.4         0.1
## 14         4.3         3.0         1.1         0.1
## 16         5.7         4.4         1.5         0.4
## 17         5.4         3.9         1.3         0.4
## 18         5.1         3.5         1.4         0.3
## 19         5.7         3.8         1.7         0.3
## 21         5.4         3.4         1.7         0.2
## 22         5.1         3.7         1.5         0.4
## 23         4.6         3.6         1.0         0.2
## 24         5.1         3.3         1.7         0.5
## 26         5.0         3.0         1.6         0.2
## 27         5.0         3.4         1.6         0.4
## 28         5.2         3.5         1.5         0.2
## 29         5.2         3.4         1.4         0.2
## 31         4.8         3.1         1.6         0.2
## 32         5.4         3.4         1.5         0.4
## 33         5.2         4.1         1.5         0.1
## 34         5.5         4.2         1.4         0.2
## 36         5.0         3.2         1.2         0.2
## 37         5.5         3.5         1.3         0.2
## 38         4.9         3.6         1.4         0.1
## 39         4.4         3.0         1.3         0.2
## 41         5.0         3.5         1.3         0.3
## 42         4.5         2.3         1.3         0.3
## 43         4.4         3.2         1.3         0.2
## 44         5.0         3.5         1.6         0.6
## 46         4.8         3.0         1.4         0.3
## 47         5.1         3.8         1.6         0.2
## 48         4.6         3.2         1.4         0.2
## 49         5.3         3.7         1.5         0.2
## 51         7.0         3.2         4.7         1.4
## 52         6.4         3.2         4.5         1.5
## 53         6.9         3.1         4.9         1.5
## 54         5.5         2.3         4.0         1.3
## 56         5.7         2.8         4.5         1.3
## 57         6.3         3.3         4.7         1.6
## 58         4.9         2.4         3.3         1.0
## 59         6.6         2.9         4.6         1.3
## 61         5.0         2.0         3.5         1.0
## 62         5.9         3.0         4.2         1.5
## 63         6.0         2.2         4.0         1.0
## 64         6.1         2.9         4.7         1.4
## 66         6.7         3.1         4.4         1.4
## 67         5.6         3.0         4.5         1.5
```

```
## 68          5.8         2.7         4.1         1.0
## 69          6.2         2.2         4.5         1.5
## 71          5.9         3.2         4.8         1.8
## 72          6.1         2.8         4.0         1.3
## 73          6.3         2.5         4.9         1.5
## 74          6.1         2.8         4.7         1.2
## 76          6.6         3.0         4.4         1.4
## 77          6.8         2.8         4.8         1.4
## 78          6.7         3.0         5.0         1.7
## 79          6.0         2.9         4.5         1.5
## 81          5.5         2.4         3.8         1.1
## 82          5.5         2.4         3.7         1.0
## 83          5.8         2.7         3.9         1.2
## 84          6.0         2.7         5.1         1.6
## 86          6.0         3.4         4.5         1.6
## 87          6.7         3.1         4.7         1.5
## 88          6.3         2.3         4.4         1.3
## 89          5.6         3.0         4.1         1.3
## 91          5.5         2.6         4.4         1.2
## 92          6.1         3.0         4.6         1.4
## 93          5.8         2.6         4.0         1.2
## 94          5.0         2.3         3.3         1.0
## 96          5.7         3.0         4.2         1.2
## 97          5.7         2.9         4.2         1.3
## 98          6.2         2.9         4.3         1.3
## 99          5.1         2.5         3.0         1.1
## 101         6.3         3.3         6.0         2.5
## 102         5.8         2.7         5.1         1.9
## 103         7.1         3.0         5.9         2.1
## 104         6.3         2.9         5.6         1.8
## 106         7.6         3.0         6.6         2.1
## 107         4.9         2.5         4.5         1.7
## 108         7.3         2.9         6.3         1.8
## 109         6.7         2.5         5.8         1.8
## 111         6.5         3.2         5.1         2.0
## 112         6.4         2.7         5.3         1.9
## 113         6.8         3.0         5.5         2.1
## 114         5.7         2.5         5.0         2.0
## 116         6.4         3.2         5.3         2.3
## 117         6.5         3.0         5.5         1.8
## 118         7.7         3.8         6.7         2.2
## 119         7.7         2.6         6.9         2.3
## 121         6.9         3.2         5.7         2.3
## 122         5.6         2.8         4.9         2.0
## 123         7.7         2.8         6.7         2.0
## 124         6.3         2.7         4.9         1.8
## 126         7.2         3.2         6.0         1.8
## 127         6.2         2.8         4.8         1.8
## 128         6.1         3.0         4.9         1.8
## 129         6.4         2.8         5.6         2.1
## 131         7.4         2.8         6.1         1.9
## 132         7.9         3.8         6.4         2.0
## 133         6.4         2.8         5.6         2.2
## 134         6.3         2.8         5.1         1.5
```

```
## 136          7.7         3.0          6.1         2.3
## 137          6.3         3.4          5.6         2.4
## 138          6.4         3.1          5.5         1.8
## 139          6.0         3.0          4.8         1.8
## 141          6.7         3.1          5.6         2.4
## 142          6.9         3.1          5.1         2.3
## 143          5.8         2.7          5.1         1.9
## 144          6.8         3.2          5.9         2.3
## 146          6.7         3.0          5.2         2.3
## 147          6.3         2.5          5.0         1.9
## 148          6.5         3.0          5.2         2.0
## 149          6.2         3.4          5.4         2.3
##
## $usekernel
## [1] FALSE
##
## $varnames
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
##
## attr(,"class")
## [1] "NaiveBayes"
```

```
# check the accuracy

prediction <- predict(nbmodel, iristest[,-5])
prediction
```

```
## $class
##          5         10         15         20         25         30
##     setosa     setosa     setosa     setosa     setosa     setosa
##         35         40         45         50         55         60
##     setosa     setosa     setosa     setosa versicolor versicolor
##         65         70         75         80         85         90
## versicolor versicolor versicolor versicolor versicolor versicolor
##         95        100        105        110        115        120
## versicolor versicolor  virginica  virginica  virginica versicolor
##        125        130        135        140        145        150
##  virginica  virginica versicolor  virginica  virginica  virginica
## Levels: setosa versicolor virginica
##
## $posterior
##            setosa   versicolor    virginica
## 5    1.000000e+00 9.719403e-18 1.269520e-27
## 10   1.000000e+00 5.211703e-17 2.131110e-27
## 15   1.000000e+00 1.089192e-17 2.739030e-27
## 20   1.000000e+00 1.730238e-16 4.646574e-26
## 25   1.000000e+00 2.865240e-13 3.011907e-23
## 30   1.000000e+00 6.068738e-16 5.421070e-26
## 35   1.000000e+00 3.481834e-16 2.742806e-26
## 40   1.000000e+00 1.524952e-16 1.709987e-26
## 45   1.000000e+00 1.903899e-11 1.036818e-20
## 50   1.000000e+00 5.148805e-17 5.434037e-27
## 55  1.908472e-117 9.701551e-01 2.984486e-02
## 60   1.564426e-75 9.998786e-01 1.213884e-04
## 65   6.747139e-60 9.999713e-01 2.866378e-05
```

```
## 70   6.371824e-66 9.999975e-01 2.540877e-06
## 75   2.467655e-93 9.992533e-01 7.467368e-04
## 80   7.752015e-47 9.999996e-01 4.013157e-07
## 85  2.774541e-108 9.932957e-01 6.704323e-03
## 90   4.826124e-77 9.999658e-01 3.422244e-05
## 95   2.976371e-86 9.998855e-01 1.145287e-04
## 100  1.275232e-81 9.998886e-01 1.113538e-04
## 105 8.390839e-235 1.714436e-06 9.999983e-01
## 110 1.896303e-282 1.350340e-10 1.000000e+00
## 115 1.836683e-197 8.431508e-06 9.999916e-01
## 120 1.957497e-139 9.852239e-01 1.477612e-02
## 125 2.535549e-221 6.960364e-06 9.999930e-01
## 130 2.685178e-203 3.112391e-03 9.968876e-01
## 135 3.478603e-174 7.855623e-01 2.144377e-01
## 140 1.999354e-201 2.646789e-05 9.999735e-01
## 145 2.161428e-249 6.370732e-09 1.000000e+00
## 150 7.983561e-159 9.289179e-02 9.071082e-01
```

```
table(prediction$class, iristest[,5])
```

```
##
##              setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         10         2
##   virginica       0          0         8
```

1. How would you make a prediction for a new case with the above package?

prediction <- predict(nbmodel, iristest)

2.How does this package deal with numeric features?

This package works well with both numeric as well as character variables.

3.How does it specify a Laplace estimator?

NaiveBayes(x, grouping, prior, usekernel = FALSE, fL = 0, ...)

fL- Factor for Laplace correction, default factor is 0, i.e. no correction.

Q.3 Laplace Estimator

Adds a small number to each of the counts which ensures that each feature has a nonzero probability of occuring with each class. Typically, the Laplace estimator is set to 1, which ensures that each class-feature combination is found in the data at least once.

Example:

Given: a1, a2, a1, a2,a3, a1, a3, a2

Without laplace estimator:

Probability(a1)= 3/8 Probability(a2)= 3/8 Probability(a3)= 2/8

With laplace estimator: (K=1) Probability(a1)= (3+1)/8+3(1)=4/11 Probability(a2)= (3+1)/8+3(1)=4/11 Probability(a3)= (2+1)/8+3(1)=3/11

The Laplace tends to draw the estimate of probability distribution closer to uniform distribution and larger the value of k, closer will it be to uniform distribution. This makes the estimated probabiliity nonzero.