Exercise 2.1

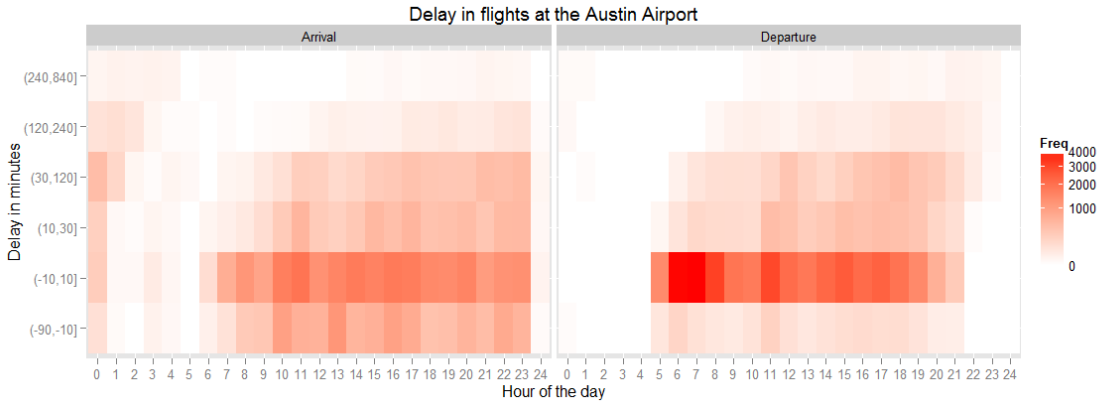
Anagh

August 17, 2015

Question 1

```
adata <- read.csv("../data/ABIA.csv", header = TRUE)

adata1<-
adata[,c("Month","DayofMonth","DayOfWeek","DepTime","ArrTime","Origin","Dest"
,"ArrDelay","DepDelay")]

adata1$ftype<-NA
adata1$ftype[adata1$Origin=="AUS"]="Departure"
adata1$ftype[adata1$Dest=="AUS"]="Arrival"



adata1$fdel<-NA
adata1$fdel[adata1$Origin=="AUS"]=adata1$DepDelay[adata1$Origin=="AUS"]
adata1$fdel[adata1$Dest=="AUS"]=adata1$ArrDelay[adata1$Dest=="AUS"]




adata1$fhour<-NA
adata1$fhour[adata1$Origin=="AUS"]=floor(adata1$DepTime[adata1$Origin=="AUS"]
/100)
adata1$fhour[adata1$Dest=="AUS"]=floor(adata1$ArrTime[adata1$Dest=="AUS"]/100
)

adata2<-na.omit(adata1[,c("ftype","fdel","fhour")])


adata2$qdel <- with(adata2, cut(fdel, breaks = c(-90,-10,10,30,120,240,840)))

atab <- xtabs(~fhour + qdel + ftype, adata2)

atabdf <- as.data.frame.table(atab)

library(ggplot2)

pg <- ggplot(atabdf, aes(fhour, qdel, fill = Freq))+facet_grid(~ftype) +
geom_tile() + scale_fill_gradient( trans="sqrt",low = "white",high="red")
```

```
pg + labs(title = "Delay in flights at the Austin Airport")+ xlab("Hour of
the day")+ ylab("Delay in minutes")
```



- The graph shows the likely delay that might occur at a spcific time of the day at the Austin Airport for both arriving and departing flights
- Darker the shade of red higher is the probability of the delay (magnitude on the Y Axis)
- It will be riskier(more probability to encounter delays) to fly during times of the day which has darker shades of red for delays
- Delays has been split into logical groups : -10 to 10 minutes : No inconvience ; 10-30 minutes : mild inconvinience and so on
- The graph can be read wrt to time of the day(X Axis) as well as with respect to the acceptable delay (Y axis)

Some observations

- There is a white patch from midnight to 5am which denotes the fact that there arent any flights which depart during this time
- The best time to fly from Austin airport is early morning and the worst is around evening at 5-7pm
- There are no specific pattern for delays for arrival other than the fact that there are slightly higher probability of higher delays during later in the day

Question 2

Naive Bayes Model

- Loading the tm library and executing the wrapper function to utilise the 'read files' utilities which helps in reading and arranging all the input files

```
library(tm)

readerPlain = function(fname)
{
  readPlain(elem=list(content=readLines(fname)),id=fname, language='en')
}
```

- Filenames from the Train and Test folder are read separately. The filenames are appended in a vector that is used to produce the corpus
- Both the Train and Test data is used to form the DTM. This has been done to account for any new words/tokens that the predictor might encounter in the Test data . It also ensures that the terms that are 'jointly sparse' i.e., Test terms that might be classified as sparse in comparison to Train data, are dealt with properly

```
adirs1 = Sys.glob('../data/ReutersC50/C50train/*')
adirs2 = Sys.glob('../data/ReutersC50/C50test/*')

flist1 = NULL
flist2 = NULL

labels = NULL
labels1 = NULL
labels2 = NULL

for(author in adirs1)
{
  files_to_add = Sys.glob(paste0(author, '/*.txt'))
  flist1 = append(flist1, files_to_add)
}

for(author in adirs2)
{
  files_to_add = Sys.glob(paste0(author, '/*.txt'))
  flist2 = append(flist2, files_to_add)
}

flist3 = append(flist1,flist2)
```

- The order of the append is important because we want to ensure that the order of filenames and the subsequent corpus remains at the 'Test/Train' x 'Author' x 'Article' level

```
for(author in adirs1)
{
  author_name = substring(author, first=29)
  files_to_add = Sys.glob(paste0(author, '/*.txt'))
  labels1 = append(labels1, rep(author_name, length(files_to_add)))
}
```

```
for(author in adirs2)
{
  author_name = substring(author, first=28)
  files_to_add = Sys.glob(paste0(author, '/*.txt'))
  labels2 = append(labels2, rep(author_name, length(files_to_add)))
}

labels <- unique(append(labels1, labels2))
```

- We extract the corpus using the readerPlain function and the mega-filelist that we compiled earlier. The corpus is also put through some basic cleaning using the utilities provided within tm. This helps in addressing any redundancies and limits the noise-chasing that might occur with the subsequent model

```
all_docs = lapply(flist3, readerPlain)
names(all_docs) = flist3
names(all_docs) = sub('.txt', '', names(all_docs))

corp = Corpus(VectorSource(all_docs))
names(corp) = names(all_docs)

# Preprocessing
corp = tm_map(corp, content_transformer(tolower)) # make everything lowercase
corp = tm_map(corp, content_transformer(removeNumbers)) # remove numbers
corp = tm_map(corp, content_transformer(removePunctuation)) # remove
punctuation
corp = tm_map(corp, content_transformer(stripWhitespace)) ## remove excess
white-space
corp = tm_map(corp, content_transformer(removeWords), stopwords("SMART"))

DTM = DocumentTermMatrix(corp)
DTM

## <<DocumentTermMatrix (documents: 5000, terms: 44234)>>
## Non-/sparse entries: 858721/220311279
## Sparsity           : 100%
## Maximal term length: 45
## Weighting          : term frequency (tf)
```

As expected, the sparsity of the Document Terms matrix is very high for any meaningful next step. We can use the inbuilt tm functionality of reduce the sparsity by a defined parameter

```
DTM = removeSparseTerms(DTM, 0.975)
DTM

## <<DocumentTermMatrix (documents: 5000, terms: 1386)>>
## Non-/sparse entries: 494509/6435491
## Sparsity           : 93%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

Next, the DTM is cnoverted to a data matrix, and the Train data is separated from the entire DTM. We then calculate the 'weight vector' for each Author after Laplace smoothing. These are the word/token level weights (for each author) that are multiplied with the word frequencies in the Test data to calculate the log probabilities eventually

The weight vector for each Author is named "w_"

```
X = as.matrix(DTM)

X_train <- X[1:2500,]
labels <- unique(labels)
smooth_count = 1/nrow(X_train)

for(i in 1:50)
{
  nam1 <- paste("w",labels[i], sep = "_")
  temp <- colSums(X_train[(50*i-49):(50*i),] + smooth_count)
  assign(nam1, temp/sum(temp))
}
```

- We predict the author name on the Test data by calculating the log probabilities for each document across all authors and finding the highest value

```
X_test <- X[2501:5000,]

result = matrix(, nrow = 2500, ncol = 51)
for(i in 1:2500)
{ for(j in 1:50)
 {
  nam1 <- paste("w",labels[j], sep = "_")
  #check <- log(get(nam1))
  result[i,j] = sum(X_test[i,]*log(get(nam1)))
 }
}

result[1:5,1:10]
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,]  -968.6253 -1205.9818 -1054.3074 -1213.7225 -1158.2985 -1136.5043
## [2,]  -591.0338  -741.1685  -681.2818  -670.1952  -762.9471  -689.4315
## [3,] -2185.4947 -2835.3524 -2494.5284 -2845.0260 -2532.8456 -2488.4620
## [4,]  -604.7419  -902.3168  -800.7661  -775.4366  -781.8511  -788.4264
## [5,] -1182.3740 -1663.7194 -1507.1808 -1531.8951 -1488.0430 -1461.1148
##            [,7]       [,8]       [,9]      [,10]
## [1,] -1110.1601 -1176.3673 -1119.1908 -1163.0840
## [2,]  -683.7230  -773.4523  -747.6904  -718.8658
## [3,] -2595.0482 -2846.5662 -2591.4150 -2200.6135
## [4,]  -790.6297  -856.0517  -793.5156  -803.1398
## [5,] -1532.5642 -1696.8449 -1547.7865 -1316.8529
```

- We then append the predicted authors to the results dataset and then form a new dataset containing only the original author and the predicted author

```
for (i in 1:2500)
{
  result[i,51] = which.max(result[i,])
}

result1 = NULL
result1 = cbind((rep(1:50, each=50)),result[,51])
result1$auth <- rep(1:50, each=50)

## Warning in result1$auth <- rep(1:50, each = 50): Coercing LHS to a list

result1$pred_auth <- result[,51]
```

- We predict the accuracy of classification for each of the authors using confusion matrix in caret package

```
library(caret)

## Warning: package 'caret' was built under R version 3.2.2

## Loading required package: lattice

confusionMatrix(result1$pred_auth,result1$auth)

## Confusion Matrix and Statistics
##
##            Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##         1  42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         2   0 25  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
##         3   0  0 20  0  2  0  0  0  3  0  0  0  0  0  0  0  3  5  0  2  0
##         4   0  0  0 11  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0
##         5   0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   1  0  0  0  0 43  0  8  0  1  0  0  0  0  0  0  0  0  0  0  0
##         7   1  0  0  0  0  0 14  0  0  0  0  0  7  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  3  0  0  1  0
##         10  0  0  0  0  0  2  0  0  0 25  0  0  0  0  0  0  0  0  0  0  0
##         11  0  0  0  0  0  0  0  0  0  0 49  0  0  0  0  0  0  0  0  0  0
##         12  0  0  0  2  0  0  0  0  0  0  0 39  0  0  2  0  0  0  0  0  0
##         13  0  0  0  0  3  0 36  0  0  0  0  0 17  0  0  0  0  0  0  0  0
##         14  0  8  0  0  0  0  0  0  0  0  0  0  1 26  0  0  0  0 11  0  0
##         15  0  0  0 13  0  0  0  1  0  0  0  3 12  0 18  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0 50  0  0  0  0  0
##         17  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0 33  0  0  2  0
##         18  1  0 28  0  1  0  0  1  0  0  0  0  0  0  0  0  1 39  0  0  0
##         19  0 17  0  0  0  0  0  0  0  0  0  0  3 23  0  0  0  0 32  0  0
##         20  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  2  0  0 37  0
##         21  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  2  0 47
```

```
##          22  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0
##          23  0  0  0  0  0  1  0  3  0  3  0  0  0  0  0  0  0  0  0  0  0
##          24  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          25  0  0  0  0  0  0  0  0  5  2  0  0  0  0  0  0  0  0  0  1  0
##          26  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##          27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          29  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0
##          30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          31  0  0  0  0  3  0  0  0  1  0  0  0  4  0  0  0  0  0  0  0  0
##          32  0  0  0  0  0  1  0  2  0  1  0  0  0  0  0  0  0  0  0  0  0
##          33  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2
##          34  0  0  0  0  2  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##          35  0  0  0  3  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0
##          36  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0
##          37  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          38  0  0  0  3  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0
##          39  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  1  0  0
##          40  0  0  0  0  0  1  0  4  0  1  0  0  0  0  0  0  0  0  0  0  0
##          41  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0
##          42  3  0  1  0  0  1  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0
##          43  0  0  0  0  4  0  0  0  0  0  1  4  0  0  0  0  0  1  0  0  0
##          44  0  0  0  7  0  0  0  0  0  0  0  1  1  0 14  0  0  0  1  0  1
##          45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  2  0
##          46  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          47  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0
##          48  0  0  0  0  2  0  0  0  6  0  0  0  0  0  0  0  0  8  1  0  3  0
##          49  0  0  0  0  0  1  0 23  6  0  0  0  0  0  0  0  0  0  0  1  0  0
##          50  2  0  0 11  0  0  0  0  0  0  0  0  2  2  0  2  0  0  0  0  0  0
##            Reference
## Prediction 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
##          1   0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  1
##          2   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          3   0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
##          4   0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##          5   0  0  6  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0
##          6   0  6  0  0  0  5  0  0  0  0  3  0  0  0  1  4  0  0  2  0  0
##          7   0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          8   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          9   5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
##          10  0  0  0  0 10  1  0  0  0  0  0  0  0  0  3  2  0  0  1  0  5
##          11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          12  0  0  0  0  0  1  1  0  0  0  0  0  0  4  0  0  0  0  0  0  0
##          13  0  0  0  0  0  1  0  0  0  2  0  2  0  0  0  0  0  0  0  0  0
##          14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          15  0  0  2  0  0  0  4  1  0  1  0  3  0  9  0  0  1  0  0  1  0
##          16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
##          17  1  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          18  0  0  0  1  0  0  0  0  1  1  0  0  4  0  0  0  0  0  0  0  0
##          19  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
##         20  1  0  0  2  0  0  0  0  5  0  0  0  0  0  0  1  0  1  2  0  0
##         21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         22 37  1  1  7  0  0  0  0  1  0  0  0  2  0  0  0  0  0  0  2  0
##         23  0 26  0  0  1  2  0  0  2  0  2  0  1  0  0  6  0  0  1  0  2
##         24  0  0 28  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  1  0  0
##         25  5  0  2 34  0  0  0  0  1  2  0  0  0  0  2  0  0  0  0  0  0
##         26  0  0  0  0 32  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1
##         27  0  0  0  0  0 31  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         28  0  0  0  0  0  0 39  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##         29  0  0  0  0  0  0  0 49  4  0  0  2  0  1  0  0  0  0  0  0  0
##         30  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0 13  1  0  0
##         31  0  0  4  0  0  0  0  0  0 21  0  0  0  0  0  0  0  0  0  0  0
##         32  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  1  0  2
##         33  0  0  0  0  0  0  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0
##         34  0  2  0  0  0  1  0  0  0  0  0  0 39  0  0  1  0  0  1  2  1
##         35  1  0  0  0  0  0  2  0  0  1  0  0  0 13  0  0  0  0  0  0  0
##         36  0  0  1  0  1  0  0  0  0  0  0  0  0  0 41  3  0  0  0  2  0
##         37  0 10  0  0  1  0  0  0  0  0  1  0  1  0  1 31  0  0  0  0  0
##         38  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0 33  0  0  0  0
##         39  0  0  0  0  0  0  0  0  3  2  0  0  0  1  0  0  0 34  0  0  0
##         40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0
##         41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 38  0
##         42  0  4  0  0  1  2  0  0  1  0  7  0  0  0  1  0  0  0  0  1 32
##         43  0  0  2  1  0  1  0  0  0  5  1  0  0  0  0  0  3  0  0  0  0
##         44  0  0  0  0  0  0  0  0  0  1  1  0  0  8  0  0  1  0  0  2  0
##         45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
##         46  0  0  0  0  0  0  1  0  0  0  0  1  0  1  0  0 11  0  0  0  0
##         47  0  1  0  0  4  0  0  0  2  0  7  0  1  0  0  1  0  0  0  0  6
##         48  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
##         49  0  0  0  0  0  0  1  0  0  2  2  0  1  0  0  0  0  0  0  0  0
##         50  0  0  3  0  0  0  2  0  0  0  0  0  0  8  0  0  0  0  0  0  0
##           Reference
## Prediction 43 44 45 46 47 48 49 50
##         1   0  0  0  0  4  0  0  0
##         2   0  0  0  0  0  0  0  0
##         3   0  0 16  0  0  0  0  0
##         4   0  0  0  0  0  0  0  4
##         5   0  0  0  1  0  0  0  0
##         6   0  0  0  0  1  0  2  0
##         7   0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0 13  0
##         9   0  0  0  0  0  2  0  0
##         10  0  0  0  0  8  0  0  0
##         11  0  0  1  0  0  0  0  2
##         12  1  1  0  1  0  0  0  4
##         13  0  0  0  0  0  0  0  0
##         14  0  0  0  0  0  0  0  0
##         15  1 28  0  0  0  0  1  9
##         16  1  0  0  0  0  0  1  0
##         17  0  0  0  0  0  0  0  0
```

```
##         18  0  0  4  0  0  0  0  0
##         19  0  0  0  0  0  0  0  0
##         20  0  0  0  0  0  1  0  0
##         21  0  0  0  0  0  0  7  0
##         22  0  0  0  0  0  2  0  0
##         23  0  0  0  0  4  0  3  0
##         24  0  0  0  0  0  0  0  0
##         25  0  0  0  0  0  0  0  0
##         26  0  0  0  0  1  0  0  0
##         27  0  0  0  0  0  0  0  0
##         28  0  1  0  0  0  0  0  1
##         29  0  0  0  0  0  0  0  0
##         30  0  0  0  0  0  0  0  0
##         31  0  0  0  0  0  0  0  0
##         32  0  0  0  0  1  0  0  0
##         33  0  0  0  0  0  0  0  0
##         34  0  0  0  0  0  1  1  0
##         35  0  2  0  0  0  0  0  2
##         36  0  0  0  0  0  0  0  0
##         37  0  0  0  0  0  0  0  0
##         38 11  4  0 18  0  0  0  1
##         39  0  0  0  0  0  0  0  0
##         40  0  0  0  0  0  0  0  0
##         41  0  0  0  0  0  0  0  0
##         42  0  0  0  0  5  0  1  0
##         43 27  0  0  8  0  0  1  1
##         44  1 13  0  1  0  0  0  5
##         45  0  0 28  0  0  5  0  0
##         46  8  1  0 20  0  0  0  4
##         47  0  0  0  0 26  0  0  0
##         48  0  0  1  0  0 39  0  0
##         49  0  0  0  0  0  0 20  0
##         50  0  0  0  1  0  0  0 17
##
## Overall Statistics
##
##                Accuracy : 0.6024
##                  95% CI : (0.5829, 0.6217)
##     No Information Rate : 0.02
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5943
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity            0.8400   0.5000   0.4000   0.2200   0.5400   0.8600
## Specificity            0.9963   0.9996   0.9869   0.9939   0.9955   0.9861
## Pos Pred Value         0.8235   0.9615   0.3846   0.4231   0.7105   0.5584
```

```
## Neg Pred Value         0.9967   0.9899   0.9877   0.9842   0.9907   0.9971
## Prevalence             0.0200   0.0200   0.0200   0.0200   0.0200   0.0200
## Detection Rate         0.0168   0.0100   0.0080   0.0044   0.0108   0.0172
## Detection Prevalence   0.0204   0.0104   0.0208   0.0104   0.0152   0.0308
## Balanced Accuracy      0.9182   0.7498   0.6935   0.6069   0.7678   0.9231
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 11
## Sensitivity            0.2800   0.1400   0.4000    0.5000    0.9800
## Specificity            0.9947   0.9947   0.9951    0.9869    0.9988
## Pos Pred Value         0.5185   0.3500   0.6250    0.4386    0.9423
## Neg Pred Value         0.9854   0.9827   0.9878    0.9898    0.9996
## Prevalence             0.0200   0.0200   0.0200    0.0200    0.0200
## Detection Rate         0.0056   0.0028   0.0080    0.0100    0.0196
## Detection Prevalence   0.0108   0.0080   0.0128    0.0228    0.0208
## Balanced Accuracy      0.6373   0.5673   0.6976    0.7435    0.9894
##                      Class: 12 Class: 13 Class: 14 Class: 15 Class: 16
## Sensitivity             0.7800    0.3400    0.5200    0.3600    1.0000
## Specificity             0.9931    0.9820    0.9918    0.9633    0.9984
## Pos Pred Value          0.6964    0.2787    0.5652    0.1667    0.9259
## Neg Pred Value          0.9955    0.9865    0.9902    0.9866    1.0000
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0156    0.0068    0.0104    0.0072    0.0200
## Detection Prevalence    0.0224    0.0244    0.0184    0.0432    0.0216
## Balanced Accuracy       0.8865    0.6610    0.7559    0.6616    0.9992
##                      Class: 17 Class: 18 Class: 19 Class: 20 Class: 21
## Sensitivity             0.6600    0.7800    0.6400    0.7400    0.9400
## Specificity             0.9967    0.9824    0.9824    0.9931    0.9959
## Pos Pred Value          0.8049    0.4756    0.4267    0.6852    0.8246
## Neg Pred Value          0.9931    0.9955    0.9926    0.9947    0.9988
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0132    0.0156    0.0128    0.0148    0.0188
## Detection Prevalence    0.0164    0.0328    0.0300    0.0216    0.0228
## Balanced Accuracy       0.8284    0.8812    0.8112    0.8665    0.9680
##                      Class: 22 Class: 23 Class: 24 Class: 25 Class: 26
## Sensitivity             0.7400    0.5200    0.5600    0.6800    0.6400
## Specificity             0.9927    0.9873    0.9943    0.9918    0.9984
## Pos Pred Value          0.6727    0.4561    0.6667    0.6296    0.8889
## Neg Pred Value          0.9947    0.9902    0.9910    0.9935    0.9927
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0148    0.0104    0.0112    0.0136    0.0128
## Detection Prevalence    0.0220    0.0228    0.0168    0.0216    0.0144
## Balanced Accuracy       0.8663    0.7537    0.7771    0.8359    0.8192
##                      Class: 27 Class: 28 Class: 29 Class: 30 Class: 31
## Sensitivity             0.6200    0.7800    0.9800    0.6000    0.4200
## Specificity             1.0000    0.9988    0.9959    0.9943    0.9951
## Pos Pred Value          1.0000    0.9286    0.8305    0.6818    0.6364
## Neg Pred Value          0.9923    0.9955    0.9996    0.9919    0.9882
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0124    0.0156    0.0196    0.0120    0.0084
## Detection Prevalence    0.0124    0.0168    0.0236    0.0176    0.0132
## Balanced Accuracy       0.8100    0.8894    0.9880    0.7971    0.7076
```

```
##                      Class: 32 Class: 33 Class: 34 Class: 35 Class: 36
## Sensitivity             0.5000    0.8400    0.7800    0.2600    0.8200
## Specificity             0.9967    0.9992    0.9947    0.9939    0.9931
## Pos Pred Value          0.7576    0.9545    0.7500    0.4643    0.7069
## Neg Pred Value          0.9899    0.9967    0.9955    0.9850    0.9963
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0100    0.0168    0.0156    0.0052    0.0164
## Detection Prevalence    0.0132    0.0176    0.0208    0.0112    0.0232
## Balanced Accuracy       0.7484    0.9196    0.8873    0.6269    0.9065
##                      Class: 37 Class: 38 Class: 39 Class: 40 Class: 41
## Sensitivity             0.6200    0.6600    0.6800    0.8000    0.7600
## Specificity             0.9943    0.9829    0.9955    0.9976    0.9988
## Pos Pred Value          0.6889    0.4400    0.7556    0.8696    0.9268
## Neg Pred Value          0.9923    0.9930    0.9935    0.9959    0.9951
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0124    0.0132    0.0136    0.0160    0.0152
## Detection Prevalence    0.0180    0.0300    0.0180    0.0184    0.0164
## Balanced Accuracy       0.8071    0.8214    0.8378    0.8988    0.8794
##                      Class: 42 Class: 43 Class: 44 Class: 45 Class: 46
## Sensitivity             0.6400    0.5400    0.2600    0.5600    0.4000
## Specificity             0.9873    0.9865    0.9816    0.9951    0.9890
## Pos Pred Value          0.5079    0.4500    0.2241    0.7000    0.4255
## Neg Pred Value          0.9926    0.9906    0.9848    0.9911    0.9878
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0128    0.0108    0.0052    0.0112    0.0080
## Detection Prevalence    0.0252    0.0240    0.0232    0.0160    0.0188
## Balanced Accuracy       0.8137    0.7633    0.6208    0.7776    0.6945
##                      Class: 47 Class: 48 Class: 49 Class: 50
## Sensitivity             0.5200    0.7800    0.4000    0.3400
## Specificity             0.9902    0.9902    0.9849    0.9865
## Pos Pred Value          0.5200    0.6190    0.3509    0.3400
## Neg Pred Value          0.9902    0.9955    0.9877    0.9865
## Prevalence              0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0104    0.0156    0.0080    0.0068
## Detection Prevalence    0.0200    0.0252    0.0228    0.0200
## Balanced Accuracy       0.7551    0.8851    0.6924    0.6633
```

- We can see that the overall accuracy rate of the Naive-Bayes classifier based on the articles is around 60%
- From the confusion matrix we can observe that author 8 and author 49 have similar topics(Automobile industry). That is the reason one has been predicted as the other pretty often

For testing another classifier, converting the data matrix to a dataframe to be used in subsequent models

```
auth = rep(rep(1:50,each=50),2)
author = as.data.frame(X)
```

```r
colnames(author) = make.names(colnames(author))
str(author)
```

```
## 'data.frame':    5000 obs. of  1386 variables:
##  $ ability           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abroad            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accept            : num  0 0 0 1 1 0 0 0 0 0 ...
##  $ access            : num  1 0 2 0 0 0 0 0 0 4 ...
##  $ account           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accounting        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accounts          : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ accused           : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ achieve           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acquire           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acquired          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acquisition       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acquisitions      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ act               : num  0 0 0 0 0 0 1 5 1 0 ...
##  $ action            : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ actions           : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ active            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ activities        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ activity          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ add               : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ added             : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ adding            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ addition          : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ additional        : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ address           : num  0 1 0 3 2 1 0 0 0 0 ...
##  $ administration    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ administrative    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ advantage         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ advertising       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ affairs           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ affect            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ affected          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ afternoon         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ agency            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ aggressive        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ago               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ agree             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ agreed            : num  0 0 0 0 0 0 0 2 0 0 ...
##  $ agreement         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ agreements        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ahead             : num  0 0 0 1 1 0 0 0 0 0 ...
##  $ aim               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ aimed             : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ aims              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ air               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ aircraft          : num  0 1 0 0 0 0 0 0 0 0 ...
```

```
##  $ airline           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ airlines          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ airport           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ alliance          : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ allowed           : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ allowing          : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ america           : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ american          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ americas          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ amid              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ amount            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ amp               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ analyst           : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ analysts          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ announce          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ announced         : num  1 1 0 1 1 0 0 0 0 0 ...
##  $ announcement      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ annual            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ appeal            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ appeared          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ appears           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ applications      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ appointed         : num  0 0 0 1 1 0 0 0 0 0 ...
##  $ approach          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ approval          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ approved          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ april             : num  0 0 0 0 0 0 0 0 0 2 ...
##  $ area              : num  0 1 0 0 0 0 1 0 0 0 ...
##  $ areas             : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ arm               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ army              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ asia              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ asian             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ asked             : num  0 0 0 0 0 0 2 0 0 0 ...
##  $ asset             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ assets            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ association       : num  0 0 1 1 1 1 0 0 0 0 ...
##  $ atampt            : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ attempt           : num  0 0 0 1 1 0 0 0 0 0 ...
##  $ attention         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ attractive        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ august            : num  0 0 0 3 2 0 0 0 0 0 ...
##  $ australia         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ australian        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ australias        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ authorities       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ authority         : num  0 0 0 4 4 0 1 0 0 0 ...
##  $ auto              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ average           : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ avoid             : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ back              : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ bad               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ balance           : num  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

```
author$auth=auth
author$auth=as.factor(author$auth)
```

- The data is then divided into Train and Test as below:

```
author_train=author[1:2500,]
author_test=author[2501:5000,]
```

- We then run a Random Forest model on the Train data and test is on the Test data

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(3)
authorRF=randomForest(auth~.,data=author_train)
preds=predict(authorRF,newdata=author_test)
confusionMatrix(preds,author_test$auth)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##         1  45  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##         2   0 31  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  2  0  0
##         3   0  0 15  0  0  0  0  0  3  0  0  0  0  0  0  0  0  2  0  0  0
##         4   0  0  0 25  0  0  0  0  0  0  0  0  0  0 18  0  0  0  0  0  0
##         5   0  0  0  0 25  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   1  0  0  0  0 29  0  5  0  1  0  0  0  0  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0 14  0  0  0  0  0 26  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  6  2  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  1  0  0  0  0  0 16  0  0  0  0  0  0  0  1  3  0  2  0
##         10  0  0  0  0  0  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0
##         11  0  0  0  0  0  0  0  0  0  1 50  0  0  0  0  0  0  0  0  0  0
##         12  0  0  0  2  0  0  0  0  0  0  0 47  0  0  3  0  0  0  0  0  0
##         13  0  0  1  0  2  0 36  0  0  0  0  0 20  0  0  0  0  0  0  0  0
##         14  0  2  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0 14  0  0
##         15  0  0  0  3  0  0  0  0  0  0  0  0  0  0 13  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0 50  0  0  0  0  0
##         17  0  0  2  0  0  0  0  0  4  0  0  0  0  0  0  0 43  0  0  1  0
##         18  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  2 31  0  0  0
##         19  0 16  0  0  0  0  0  0  0  0  0  0  0 19  0  0  0  0 32  0  0
##         20  0  0  1  0  0  0  0  0  4  0  0  0  0  0  0  0  1  1  0 36  0
##         21  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 46
##         22  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         23  0  0  0  0  0  9  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
##          24  0  0  0  0 13  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          25  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
##          26  1  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0
##          27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          29  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  2  1  0
##          30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          31  0  0  0  0 10  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0
##          32  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          33  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4
##          34  0  0  0  0  0  0  0  2  1  0  0  0  0  0  0  0  0  0  0  0  0
##          35  0  0  0  3  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
##          36  0  0  0  0  0  1  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0
##          37  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          38  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          39  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0
##          40  0  0  0  0  0  2  0  1  4  0  0  0  0  0  0  0  0  1  0  0  0
##          41  0  0  0  0  0  0  0  0  1  4  0  0  0  0  0  0  0  0  0  0  0
##          42  2  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0
##          43  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  3  0  0  0
##          44  0  0  0  4  0  0  0  0  0  0  0  1  0  0  7  0  0  0  0  0  0
##          45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9  0  1  0
##          46  0  0  0  1  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0
##          47  0  0  1  0  0  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0
##          48  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0  0  3  0  0  8  0
##          49  0  0  0  0  0  1  0 34  7  0  0  0  0  0  0  0  0  0  0  0  0
##          50  0  0  0  8  0  0  0  0  0  0  0  1  1  0  2  0  0  0  0  0  0
##            Reference
## Prediction 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
##          1   0  0  1  2  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  5  3
##          2   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          3   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          4   0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  1  0  0  0  0
##          5   1  0 10  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0
##          6   2  5  0  0  0  0  0  0  0  0  3  0  0  0  0  1  0  0  0  0  0
##          7   0  1  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          8   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          9   1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          10  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3
##          11  0  0  0  1  0  0  1  0  0  0  1  0  0  1  0  0  0  0  0  0  0
##          12  0  0  0  0  0  0  9  0  0  1  0  0  0  2  0  0  1  0  0  0  0
##          13  0  0  0  0  0  0  0  0  0  3  0  1  0  1  0  0  0  0  0  0  0
##          14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          15  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0
##          16  0  0  0  1  0  0  0  0  0  0  1  0  1  0  0  0  1  0  0  2  0
##          17  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          18  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          19  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          20  3  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          21  0  2  0  0  0  0  0  1  0  0  1  2  3  0  0  0  0  0  0  0  0
```

```
##         22 39  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         23  0 30  0  0  0  0  0  0  0  0  1  0  4  0  0  2  0  1  0  0  0
##         24  0  0 29  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0
##         25  2  0  0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         26  0  0  0  1 43  1  0  0  0  0  7  0  1  0  1  2  0  0  0  0 13
##         27  0  0  0  0  0 31  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         28  0  0  0  0  0  0 39  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##         29  0  0  0  0  0  0  0 47  1  0  0  0  0  0  0  0  0  0  0  0  0
##         30  0  0  0  1  0  0  0  0 47  0  0  2  0  0  0  2  0 25  1  0  0
##         31  0  0  4  0  0  1  0  0  0 41  0  0  0  0  1  1  0  0  0  0  0
##         32  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  1  0  0  0
##         33  0  0  0  0  0  0  0  2  0  0  0 45  1  0  0  0  0  0  0  0  0
##         34  0  0  0  1  0  0  0  0  0  0  1  0 34  0  2  4  0  0  3  0  0
##         35  0  0  0  0  0  0  0  0  0  1  0  0  0 15  0  0  0  0  0  3  0
##         36  0  0  3  4  2  0  0  0  0  0  1  0  0  0 44  0  0  0  0  0  1
##         37  0  6  0  0  0  0  0  0  0  0  1  0  2  0  0 35  0  0  0  0  0
##         38  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0 39  0  0  0  0
##         39  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0 22  1  0  0
##         40  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0 45  0  1
##         41  0  0  0  0  0  0  0  0  0  0  2  0  2  0  0  0  0  0  0 40  0
##         42  0  0  0  0  1  0  0  0  0  0  5  0  0  0  1  0  0  0  0  0 19
##         43  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         44  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0  0
##         45  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
##         46  0  0  0  1  0  0  0  0  0  0  0  0  0  4  0  0  8  0  0  0  0
##         47  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10
##         48  1  1  0  2  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
##         49  0  1  0  0  0  0  1  0  0  0  1  0  2  0  0  0  0  0  0  0  0
##         50  0  0  0  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0  0  0  0
##           Reference
## Prediction 43 44 45 46 47 48 49 50
##         1   0  0  1  0  1  0  2  0
##         2   0  0  0  0  0  0  0  0
##         3   0  0  0  0  0  0  0  0
##         4   1 17  0  0  0  0  0  7
##         5   0  0  0  0  0  0  0  0
##         6   0  0  0  0  0  0  2  0
##         7   0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0 10  0
##         9   0  0  0  0  0  6  1  0
##         10  0  0  0  0  3  0  0  0
##         11  0  0  0  0  0  0  0  0
##         12  0  0  0  0  0  0  0  2
##         13  0  0  1  0  0  0  0  0
##         14  0  0  0  0  0  0  0  0
##         15  0  4  0  0  0  0  0  5
##         16  0  0  0  0  0  0  2  1
##         17  0  0  1  0  0  0  0  0
##         18  0  0  6  0  0  1  0  0
##         19  0  0  0  0  0  0  0  0
```

```
##         20  0  0  0  0  0  1  0  0
##         21  0  1  0  0  0  0  5  0
##         22  0  0  1  0  0  1  0  0
##         23  0  0  0  0  1  0  0  0
##         24  0  0  0  0  0  0  0  0
##         25  0  0  0  0  0  0  0  0
##         26  0  0  0  0  6  0  0  1
##         27  0  0  0  0  0  0  0  0
##         28  0  0  0  0  0  0  0  1
##         29  0  1  0  0  0  0  0  0
##         30  0  0  0  0  0  0  0  0
##         31  0  0  0  0  0  0  0  1
##         32  0  0  0  0  2  0  0  0
##         33  0  0  0  0  0  0  0  0
##         34  0  0  0  0  3  0  0  0
##         35  0  5  0  0  0  0  0  4
##         36  0  0  0  0  5  0  0  0
##         37  0  0  0  0  4  0  0  0
##         38  9  4  0 12  0  0  0  2
##         39  0  0  0  0  0  0  0  0
##         40  0  0  0  0  0  0  0  0
##         41  0  0  0  0  0  0  1  0
##         42  0  0  0  0  2  0  0  0
##         43 26  0  0 10  0  0  0  0
##         44  0 12  0  0  0  0  0  4
##         45  0  0 38  0  0  0  0  0
##         46 14  3  0 28  0  0  0  5
##         47  0  0  0  0 23  0  0  0
##         48  0  0  2  0  0 41  0  0
##         49  0  0  0  0  0  0 27  0
##         50  0  3  0  0  0  0  0 17
##
## Overall Statistics
##
##                Accuracy : 0.6224
##                  95% CI : (0.6031, 0.6415)
##     No Information Rate : 0.02
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6147
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity            0.9000   0.6200   0.3000   0.5000   0.5000   0.5800
## Specificity            0.9927   0.9976   0.9980   0.9808   0.9947   0.9918
## Pos Pred Value         0.7143   0.8378   0.7500   0.3472   0.6579   0.5918
## Neg Pred Value         0.9979   0.9923   0.9859   0.9897   0.9898   0.9914
## Prevalence             0.0200   0.0200   0.0200   0.0200   0.0200   0.0200
```

```
## Detection Rate         0.0180   0.0124   0.0060   0.0100   0.0100   0.0116
## Detection Prevalence   0.0252   0.0148   0.0080   0.0288   0.0152   0.0196
## Balanced Accuracy      0.9463   0.8088   0.6490   0.7404   0.7473   0.7859
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 11
## Sensitivity            0.2800   0.1200   0.3200    0.3000    1.0000
## Specificity            0.9820   0.9951   0.9939    0.9967    0.9980
## Pos Pred Value         0.2414   0.3333   0.5161    0.6522    0.9091
## Neg Pred Value         0.9853   0.9823   0.9862    0.9859    1.0000
## Prevalence             0.0200   0.0200   0.0200    0.0200    0.0200
## Detection Rate         0.0056   0.0024   0.0064    0.0060    0.0200
## Detection Prevalence   0.0232   0.0072   0.0124    0.0092    0.0220
## Balanced Accuracy      0.6310   0.5576   0.6569    0.6484    0.9990
##                      Class: 12 Class: 13 Class: 14 Class: 15 Class: 16
## Sensitivity             0.9400    0.4000    0.5200    0.2600    1.0000
## Specificity             0.9918    0.9816    0.9935    0.9939    0.9955
## Pos Pred Value          0.7015    0.3077    0.6190    0.4643    0.8197
## Neg Pred Value          0.9988    0.9877    0.9902    0.9850    1.0000
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0188    0.0080    0.0104    0.0052    0.0200
## Detection Prevalence    0.0268    0.0260    0.0168    0.0112    0.0244
## Balanced Accuracy       0.9659    0.6908    0.7567    0.6269    0.9978
##                      Class: 17 Class: 18 Class: 19 Class: 20 Class: 21
## Sensitivity             0.8600    0.6200    0.6400    0.7200    0.9200
## Specificity             0.9959    0.9845    0.9857    0.9943    0.9935
## Pos Pred Value          0.8113    0.4493    0.4776    0.7200    0.7419
## Neg Pred Value          0.9971    0.9922    0.9926    0.9943    0.9984
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0172    0.0124    0.0128    0.0144    0.0184
## Detection Prevalence    0.0212    0.0276    0.0268    0.0200    0.0248
## Balanced Accuracy       0.9280    0.8022    0.8129    0.8571    0.9567
##                      Class: 22 Class: 23 Class: 24 Class: 25 Class: 26
## Sensitivity             0.7800    0.6000    0.5800    0.4400    0.8600
## Specificity             0.9955    0.9922    0.9935    0.9988    0.9841
## Pos Pred Value          0.7800    0.6122    0.6444    0.8800    0.5244
## Neg Pred Value          0.9955    0.9918    0.9914    0.9887    0.9971
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0156    0.0120    0.0116    0.0088    0.0172
## Detection Prevalence    0.0200    0.0196    0.0180    0.0100    0.0328
## Balanced Accuracy       0.8878    0.7961    0.7867    0.7194    0.9220
##                      Class: 27 Class: 28 Class: 29 Class: 30 Class: 31
## Sensitivity             0.6200    0.7800    0.9400    0.9400    0.8200
## Specificity             1.0000    0.9992    0.9976    0.9873    0.9918
## Pos Pred Value          1.0000    0.9512    0.8868    0.6026    0.6721
## Neg Pred Value          0.9923    0.9955    0.9988    0.9988    0.9963
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0124    0.0156    0.0188    0.0188    0.0164
## Detection Prevalence    0.0124    0.0164    0.0212    0.0312    0.0244
## Balanced Accuracy       0.8100    0.8896    0.9688    0.9637    0.9059
##                      Class: 32 Class: 33 Class: 34 Class: 35 Class: 36
## Sensitivity             0.4600    0.9000    0.6800    0.3000    0.8800
```

```
## Specificity             0.9984    0.9967    0.9931    0.9931    0.9882
## Pos Pred Value          0.8519    0.8491    0.6667    0.4688    0.6027
## Neg Pred Value          0.9891    0.9980    0.9935    0.9858    0.9975
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0092    0.0180    0.0136    0.0060    0.0176
## Detection Prevalence    0.0108    0.0212    0.0204    0.0128    0.0292
## Balanced Accuracy       0.7292    0.9484    0.8365    0.6465    0.9341
##                      Class: 37 Class: 38 Class: 39 Class: 40 Class: 41
## Sensitivity             0.7000    0.7800    0.4400    0.9000    0.8000
## Specificity             0.9918    0.9865    0.9980    0.9935    0.9959
## Pos Pred Value          0.6364    0.5417    0.8148    0.7377    0.8000
## Neg Pred Value          0.9939    0.9955    0.9887    0.9979    0.9959
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0140    0.0156    0.0088    0.0180    0.0160
## Detection Prevalence    0.0220    0.0288    0.0108    0.0244    0.0200
## Balanced Accuracy       0.8459    0.8833    0.7190    0.9467    0.8980
##                      Class: 42 Class: 43 Class: 44 Class: 45 Class: 46
## Sensitivity             0.3800    0.5200    0.2400    0.7600    0.5600
## Specificity             0.9935    0.9931    0.9910    0.9947    0.9833
## Pos Pred Value          0.5429    0.6047    0.3529    0.7451    0.4058
## Neg Pred Value          0.9874    0.9902    0.9846    0.9951    0.9910
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0076    0.0104    0.0048    0.0152    0.0112
## Detection Prevalence    0.0140    0.0172    0.0136    0.0204    0.0276
## Balanced Accuracy       0.6867    0.7565    0.6155    0.8773    0.7716
##                      Class: 47 Class: 48 Class: 49 Class: 50
## Sensitivity             0.4600    0.8200    0.5400    0.3400
## Specificity             0.9922    0.9902    0.9808    0.9894
## Pos Pred Value          0.5476    0.6308    0.3649    0.3953
## Neg Pred Value          0.9890    0.9963    0.9905    0.9866
## Prevalence              0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0092    0.0164    0.0108    0.0068
## Detection Prevalence    0.0168    0.0260    0.0296    0.0172
## Balanced Accuracy       0.7261    0.9051    0.7604    0.6647
```

We achieve an accuracy of 62.24 which is slightly higher than the naive bayes and would there fore prefer to use random forest for predicting authors among a corpus

Question 3

Question 3

- The *arules* library is needed for the apriori calculation and the reshape library for data preparation

```
library(arules)  # has a big ecosystem of packages built around it
library(reshape)
```

- The data is read and the transaction numbers is added to the list
- The data frame is melted to get all the variables in the row(unstacked) format
- NA's are omittted and the transaction number is added as a categorical variable

```
coldata= max(count.fields("../data/groceries.txt",sep=','))
gdata <- read.csv("../data/groceries.txt", header = FALSE,col.names =
paste0("V",seq_len(coldata)),fill = TRUE)


tno<-1:nrow(gdata)
gdata<-cbind(tno,gdata)

gdata1<-melt(gdata,id=c("tno"))
gdata1<-gdata1[order(gdata1$tno),]

gdata1[gdata1==""] <- NA
gdata1 <- na.omit(gdata1)


gdata1$tno <- factor(gdata1$tno)
```

- First create a list of baskets: vectors of items by trasaction analagous to bags of words
- Duplicates are removed and cast as a special arules "transaction class"

```
# First split data into a list of items for each transaction
gdata1 <- split(x=gdata1$value, f=gdata1$tno)

## Remove duplicates ("de-dupe")
gdata1 <- lapply(gdata1, unique)

## Cast this variable as a special arules "transactions" class.
gtrans <- as(gdata1, "transactions")


# Now run the 'apriori' algorithm
# Look at rules with support > .05 & confidence >.5
grules <- apriori(gtrans, parameter=list(support=.05, confidence=.5,
maxlen=5))

##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support minlen maxlen
##         0.5    0.1    1 none FALSE            TRUE    0.05      1      5
##  target   ext
##   rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
```

```
##
## apriori - find association rules with the apriori algorithm
## version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [0 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Look at the output
inspect(grules)
```

```
## NULL
```

- We notice that we do not get any set for support of .05. Reducing it to .009 still keeping the cond=fidence of 0.5 which would give us a better quality of the rules (and not capture noise in the customer buying pattern due to low confidence levels)

```
# Look at rules with support > .01 & confidence >.5 & length (# items) <= 4
grules1 <- apriori(gtrans, parameter=list(support=.009, confidence=.55,
maxlen=4))
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support minlen maxlen
##        0.55    0.1    1 none FALSE            TRUE   0.009      1      4
##  target   ext
##   rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## apriori - find association rules with the apriori algorithm
## version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [93 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Look at the output
inspect(grules1)
```

```
##    lhs                     rhs                   support confidence
lift
## 1  {curd,
##     yogurt}             => {whole milk}       0.010066090  0.5823529
```

```
2.279125
## 2  {curd,
##     other vegetables}   => {whole milk}       0.009862735  0.5739645
2.246296
## 3  {butter,
##     yogurt}             => {whole milk}       0.009354347  0.6388889
2.500387
## 4  {butter,
##     other vegetables}   => {whole milk}       0.011489578  0.5736041
2.244885
## 5  {domestic eggs,
##     other vegetables}   => {whole milk}       0.012302999  0.5525114
2.162336
## 6  {root vegetables,
##     whipped/sour cream} => {whole milk}       0.009456024  0.5535714
2.166484
## 7  {citrus fruit,
##     root vegetables}    => {other vegetables} 0.010371124  0.5862069
3.029608
## 8  {root vegetables,
##     tropical fruit}     => {other vegetables} 0.012302999  0.5845411
3.020999
## 9  {root vegetables,
##     tropical fruit}     => {whole milk}       0.011997966  0.5700483
2.230969
## 10 {root vegetables,
##     yogurt}             => {whole milk}       0.014539908  0.5629921
2.203354
```

- Whole milk being bough with {Curd, Yogurt} or {Butter, Yogurt} is intuitive because the customers who come in for dairy products will buy these together. Also, what people buy together is also a function of what is kept beside what item. So Yogurt butter and Milk are all kept in the refridgerators
- 'Other Vegetables' which is a category has various kind of vegetables is bought with vegetables and fruits. Here we can see that these are bought with {Citrus Fruit, Root vegetables},{Root Vegetables, Tropical Fruit}
- Also families who do their weekly shop at the retail store tend to buy everything together which will explain the association of dairy products like Milk to categories like vegetables, fruits