a) For each subject select:

o 100 documents for training from the train folder

o 100 documents for testing from the test folder

**Doc1.Train.Source <- DirSource(paste(getwd(),"/DemoData/News2/20news-bydate-train/sci.space",sep=""))**

**Doc1.Train <- Corpus(URISource(Doc1.Train.Source$filelist[1:100]), readerControl=list(reader=readPlain))**

**Doc2.Train.Source <- DirSource(paste(getwd(),"/DemoData/News2/20news-bydate-train/rec.autos",sep=""))**

**Doc2.Train <- Corpus(URISource(Doc1.Train.Source$filelist[1:100]), readerControl=list(reader=readPlain))**

**Doc1.Test.Source <- DirSource(paste(getwd(),"/DemoData/News2/20news-bydate-test/sci.space",sep=""))**

**Doc1.Test <- Corpus(URISource(Doc1.Train.Source$filelist[1:100]), readerControl=list(reader=readPlain))**

**Doc2.Test.Source <- DirSource(paste(getwd(),"/DemoData/News2/20news-bydate-test/rec.autos",sep=""))**

**Doc2.Test <- Corpus(URISource(Doc1.Train.Source$filelist[1:100]), readerControl=list(reader=readPlain))**

b) Obtain the merged Corpus (of 400 documents), please keep the order as

o Doc1.Train from the "sci.space" newsgroup train data

o Doc1.Test from the "sci.space" newsgroup test data

o Doc2.Train from the " rec.autos" newsgroup train data

o Doc2.Test from the " rec.autos" newsgroup test data

**merge.corpus <- c(Doc1.Train,Doc1.Test,Doc2.Train,Doc2.Test)**

**merge.corpus**

c) Implement preprocessing (clearly indicate what you have used)

**#Preprocessing**

**# Convert to lower case**

**merge.corpus.tranf <- tm\_map(merge.corpus, content\_transformer(tolower))**

**# Remove Punctuation**

**merge.corpus.tranf <- tm\_map(merge.corpus.tranf, removePunctuation)**

**# Remove stop words**

**merge.corpus.tranf <- tm\_map(merge.corpus.tranf, removeWords, stopwords("english"))**

**# specify your stopwords as a character vector**

**merge.corpus.tranf <- tm\_map(merge.corpus.tranf, removeWords, c("since", "let", "yes", "every", "yeah"))**

d) Create the Document-Term Matrix using the following arguments

o minWordLength=2

o minDocFreq=5

**# ignore extremely rare words i.e. terms that appear in less then 1% of the documents**

**minTermFreq <- 5**

**dtm = DocumentTermMatrix(merge.corpus,**

**control = list(**

**wordLengths=c(2, Inf),**

**bounds = list(global = c(minTermFreq, Inf))**

**))**

**dtm.matrix = as.matrix(dtm)**

e) Split the Document-Term Matrix into

o train dataset containing rows (1:100,201:300)

o test dataset containing rows (101:200,301:400)

**train.doc <- dtm.matrix[c(1:100,201:300),]**

**test.doc <- dtm.matrix[c(101:200,301:400),]**

f) Use the abbreviations "Sci" and "Rec" as tag factors in your classification.

**Tags <- factor(c(rep("Sci",100), rep("Rec",100)))**

g) Classify text using the kNN() function (use k=3)

**prob.test<- knn(train.doc, test.doc, Tags, k = 3, prob=TRUE)**

**prob.test**

h) Display classification results as a R dataframe and name the columns as:

o "Doc"

o "Predict" - Tag factors of predicted subject ("Sci" or "Rec")

o "Prob" - The classification probability

o "Correct' - TRUE/FALSE

**# Display Classification Results**

**a <- c(1:400) #document ids**

**b <- prob.test #predicts by the algorithm**

**c <- attributes(prob.test)$prob #proportion of the votes for the winning class**

**d <- c(rep("Sci", 100), rep("Rec", 100)) # comparison between predicted & actual**

**result <- data.frame(Doc=a, Predict=b,Prob=c,Correct= (b == d))**

**result**