**CS688 C1 Web analytics and Mining**
**Homework #3**

## Part-A - kNN classification:

a-[a,b,d,e,f]) The data is read and then the Corpus is created with 400 documents as mentioned.

a-c) Preprocessing: Here all the numbers, punctuations and stop words are removed to create a document term matrix.

```
Doc.corpus <- Corpus(URISource(c(Doc1.Train.path$filelist[1:100],Doc1.Test.path$filelist[1:100],
                                 Doc2.Train.path$filelist[1:100],Doc2.Test.path$filelist[1:100])),
                     readerControl=list(reader=readPlain))

Doc.corpus.temp <- tm_map(Doc.corpus, removeNumbers)
Doc.corpus.temp <- tm_map(Doc.corpus.temp, removePunctuation)
Doc.corpus.temp <- tm_map(Doc.corpus.temp, removeWords, stopwords("english"))

Doc.corpus.stemmed.temp <- tm_map(Doc.corpus.temp, stemDocument)

Doc.corpus.conditioned.dtm <-DocumentTermMatrix(Doc.corpus.stemmed.temp,
                                                control=list(wordLengths =c(2,Inf),bounds=list(global = c(5,Inf))))
dtm.conditioned.counts <-colSums(as.matrix(Doc.corpus.conditioned.dtm))
```

a-g) The kNN() function is applied

```
> train.doc <- Doc.corpus.conditioned.dtm[c(c(1:100), c(201:300)),]
> test.doc <- Doc.corpus.conditioned.dtm[c(101:200, 301:400),]
> tags <- factor(c(rep("Sci",100), rep("Reg",100)))
> prob.test <- knn(train.doc, test.doc, tags, k = 2, prob = TRUE)
> head(prob.test,100)
  [1] Sci Sci Sci Reg Sci Reg Sci Sci Reg Sci Reg Sci Reg Reg Sci Reg Reg Sci Sci Sci Reg Reg Sci Reg Sci Sci Sci Reg Sci
 [30] Sci Reg Sci Sci Sci Sci Sci Reg Reg Sci Reg Reg Sci Sci Sci Sci Reg Sci Sci Sci Reg Sci Reg Sci Reg Sci Reg Sci Reg
 [59] Reg Sci Sci Sci Sci Reg Reg Reg Sci Sci Reg Sci Reg Sci Reg Sci Reg Sci Reg Sci Sci Reg Reg Sci Reg Reg Sci Sci Reg
 [88] Reg Reg Sci Sci Sci Reg Sci Sci Sci Reg Sci Reg Sci
Levels: Reg Sci
>
```

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

```
> result <- data.frame(Doc=a, Predict=b, Prob=c, Correct=d)
> head(result,10)
   Doc Predict      Prob Correct
1    1     Sci 1.0000000    TRUE
2    2     Sci 0.5000000    TRUE
3    3     Sci 0.5000000    TRUE
4    4     Reg 1.0000000   FALSE
5    5     Sci 1.0000000    TRUE
6    6     Reg 0.6666667   FALSE
7    7     Sci 1.0000000    TRUE
8    8     Sci 0.5000000    TRUE
9    9     Reg 0.5000000   FALSE
10  10     Sci 1.0000000    TRUE
>
```

a-i) What is percentage of correct (TRUE) classifications?

```
> true.classifications <- sum(c)/ length(tags)
> sprintf("The true classification is %.4f", true.classifications)
[1] "The true classification is 0.7075"
>
```

**Part-B Effectiveness of the classification**

Confusion Matrix:

```
> confusion.matrix <- table(matrix(tags,ncol=1), matrix(prob.test,ncol=1))
> colnames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
> rownames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
> confusion.matrix

        Reg(1) Sci(0)
Reg(1)     70     30
Sci(0)     42     58
```

TP, TN, FP, FN values

```
> TP <- confusion.matrix[[1,1]]
> TN <- confusion.matrix[[2,2]]
> FP <- confusion.matrix[[1,2]]
> FN <- confusion.matrix[[2,1]]
> sprintf("TP = %d, TN = %d, FP = %d, FN = %d",TP, TN, FP, FN)
[1] "TP = 70, TN = 58, FP = 30, FN = 42"
```

Precision, recall and F-score

```
> precision <- TP /sum(confusion.matrix[1,])
> recall <- TP /sum(confusion.matrix[,1])
> f.score <- 2*precision*recall/ (precision+recall)
> sprintf("The precision is %.4f", precision)
[1] "The precision is 0.7000"
> sprintf("The recall is %.4f", recall)
[1] "The recall is 0.6250"
> sprintf("The f-score is %.4f", f.score)
[1] "The f-score is 0.6604"
>
```

**Additional tests:**

1) Checking the effectiveness of the classification without stemming the data and using k=2 neighbors

```
> confusion.matrix

        Reg(1) Sci(0)
  Reg(1)     77     23
  Sci(0)     54     46
> TP <- confusion.matrix[[1,1]]
> TN <- confusion.matrix[[2,2]]
> FP <- confusion.matrix[[1,2]]
> FN <- confusion.matrix[[2,1]]
> sprintf("TP = %d, TN = %d, FP = %d, FN = %d",TP, TN, FP, FN)
[1] "TP = 77, TN = 46, FP = 23, FN = 54"
> precision <- TP /sum(confusion.matrix[1,])
> recall <- TP /sum(confusion.matrix[,1])
> f.score <- 2*precision*recall/ (precision+recall)
> sprintf("The precision is %.4f", precision)
[1] "The precision is 0.7700"
> sprintf("The recall is %.4f", recall)
[1] "The recall is 0.5878"
> sprintf("The f-score is %.4f", f.score)
[1] "The f-score is 0.6667"
```

2) Checking the effectiveness of the classification with stemming the data and using k-value raised to 4 neighbors

```
> confusion.matrix

        Reg(1) Sci(0)
  Reg(1)     83     17
  Sci(0)     49     51
> TP <- confusion.matrix[[1,1]]
> TN <- confusion.matrix[[2,2]]
> FP <- confusion.matrix[[1,2]]
> FN <- confusion.matrix[[2,1]]
> sprintf("TP = %d, TN = %d, FP = %d, FN = %d",TP, TN, FP, FN)
[1] "TP = 83, TN = 51, FP = 17, FN = 49"
> precision <- TP /sum(confusion.matrix[1,])
> recall <- TP /sum(confusion.matrix[,1])
> f.score <- 2*precision*recall/ (precision+recall)
> sprintf("The precision is %.4f", precision)
[1] "The precision is 0.8300"
> sprintf("The recall is %.4f", recall)
[1] "The recall is 0.6288"
> sprintf("The f-score is %.4f", f.score)
[1] "The f-score is 0.7155"
```

The precision level is increased when the data is processed without stemming and even increased by 6% when it is done with k=4 neighbors. The results have significant improvement over the F-score.

R-code:

```
Part- A
rm(list = ls())
cat("\014")
set.seed(123)
setwd("Documents/Fall-2016/CS688/20Newsgroups/")
library(tm)
library(SnowballC)
library(class)
# ---------- Read the directory source files ----------
Doc1.Train.path <- DirSource("20news-bydate-train/sci.space/")
Doc1.Test.path <- DirSource("20news-bydate-test/sci.space/")
Doc2.Train.path <- DirSource("20news-bydate-train/rec.autos/")
Doc2.Test.path <- DirSource("20news-bydate-test/rec.autos/")
# ---------- Creating corpus with 100 files from each folders ----------
Doc.corpus <- Corpus(URISource(c(Doc1.Train.path$filelist[1:100],Doc1.Test.path$filelist[1:100],
Doc2.Train.path
                                 $filelist[1:100], Doc2.Test.path$filelist[1:100])),
readerControl=list(reader=readPlain))
# ---------- Preprocessing ----------
# Removing the numbers, punctuations, and stopwords
Doc.corpus.temp <- tm_map(Doc.corpus, removeNumbers)
Doc.corpus.temp <- tm_map(Doc.corpus.temp, removePunctuation)
Doc.corpus.temp <- tm_map(Doc.corpus.temp, removeWords, stopwords("english"))
# Stemming the document
Doc.corpus.stemmed.temp <- tm_map(Doc.corpus.temp, stemDocument)
# ---------- Creating a DTM ----------
Doc.corpus.conditioned.dtm <-DocumentTermMatrix(Doc.corpus.stemmed.temp,
                                      control=list(wordLengths =c(2,Inf),bounds=list(global = c(5,Inf))))
dtm.conditioned.counts <-colSums(as.matrix(Doc.corpus.conditioned.dtm))
# class(dtm.conditioned.counts)
# head(dtm.conditioned.counts)
# length(dtm.conditioned.counts)
#
# ord<-order(dtm.conditioned.counts)
# dtm.conditioned.counts[head(ord)]
# dtm.conditioned.counts[tail(ord)]
#
# m <- as.matrix(dtm.conditioned.counts)
# dim(m)
# head(m)
# ---------- Creating a test and train document for kNN() ----------
train.doc <- Doc.corpus.conditioned.dtm[c(c(1:100), c(201:300)),]
test.doc <- Doc.corpus.conditioned.dtm[c(101:200, 301:400),]
dim(Doc.corpus.conditioned.dtm)
dim(train.doc)
dim(test.doc)
tags <- factor(c(rep("Sci",100), rep("Reg",100)))
# kNN function is applied over the first test and train set
prob.test <- knn(train.doc, test.doc, tags, k = 2, prob = TRUE)
head(prob.test,100)
# Classification result
```

```
a <- 1:length(prob.test)
b <- levels(prob.test)[prob.test]
c <- attributes(prob.test)$prob
d <- prob.test == tags
result <- data.frame(Doc=a, Predict=b, Prob=c, Correct=d)
head(result,10)
# True Classification result
true.classifications <- sum(c)/ length(tags)
sprintf("The true classification is %.4f", true.classifications)


Part- B
# Confusion Matrix
confusion.matrix <- table(matrix(tags,ncol=1), matrix(prob.test,ncol=1))
colnames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
rownames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
confusion.matrix
## Reg as positive and Sci as negative
TP <- confusion.matrix[[1,1]]
TN <- confusion.matrix[[2,2]]
FP <- confusion.matrix[[1,2]]
FN <- confusion.matrix[[2,1]]
sprintf("TP = %d, TN = %d, FP = %d, FN = %d",TP, TN, FP, FN)
# Calculating the precision, recall and f-score
precision <- TP /sum(confusion.matrix[1,])
recall <- TP /sum(confusion.matrix[,1])
f.score <- 2*precision*recall/ (precision+recall)
sprintf("The precision is %.4f", precision)
sprintf("The recall is %.4f", recall)
sprintf("The f-score is %.4f", f.score)


## ---------- Additional tests ----------
# Without Stemming the document AND conditioned

Doc.corpus.dtm <- DocumentTermMatrix(Doc.corpus.temp,control=list(wordLengths =c(2,Inf),
                                bounds=list(global = c(5,Inf))))
dtm.counts <- colSums(as.matrix(Doc.corpus.dtm))
class(dtm.counts)
head(dtm.counts)
length(dtm.counts)
# ---------- Test-1 and results ----------
train.doc <- Doc.corpus.dtm[c(c(1:100), c(201:300)),]
test.doc <- Doc.corpus.dtm[c(101:200, 301:400),]
prob.test.1 <- knn(train.doc, test.doc, tags, k = 2, prob = TRUE)
confusion.matrix <- table(matrix(tags,ncol=1), matrix(prob.test.1,ncol=1))
colnames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
rownames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
confusion.matrix
TP <- confusion.matrix[[1,1]]
TN <- confusion.matrix[[2,2]]
FP <- confusion.matrix[[1,2]]
FN <- confusion.matrix[[2,1]]
sprintf("TP = %d, TN = %d, FP = %d, FN = %d",TP, TN, FP, FN)
```

```
precision <- TP /sum(confusion.matrix[1,])
recall <- TP /sum(confusion.matrix[,1])
f.score <- 2*precision*recall/ (precision+recall)
sprintf("The precision is %.4f", precision)
sprintf("The recall is %.4f", recall)
sprintf("The f-score is %.4f", f.score)

# With Stemming the document AND conditioned and k-value raised to 4

Doc.corpus.stemmed.dtm <- DocumentTermMatrix(Doc.corpus.stemmed.temp,
                                  control=list(wordLengths =c(2,Inf),bounds=list(global = c(5,Inf))))
dtm.stemmed.counts <- colSums(as.matrix(Doc.corpus.stemmed.dtm))
class(dtm.stemmed.counts)
head(dtm.stemmed.counts)
length(dtm.stemmed.counts)
# ---------- Test-2 and results ----------
train.doc <- Doc.corpus.stemmed.dtm[c(c(1:100), c(201:300)),]
test.doc <- Doc.corpus.stemmed.dtm[c(101:200, 301:400),]
prob.test.2 <- knn(train.doc, test.doc, tags, k = 4, prob = TRUE)
confusion.matrix <- table(matrix(tags,ncol=1), matrix(prob.test.2,ncol=1))
colnames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
rownames(confusion.matrix) <- c("Reg(1)", "Sci(0)")
confusion.matrix
TP <- confusion.matrix[[1,1]]
TN <- confusion.matrix[[2,2]]
FP <- confusion.matrix[[1,2]]
FN <- confusion.matrix[[2,1]]
sprintf("TP = %d, TN = %d, FP = %d, FN = %d",TP, TN, FP, FN)
precision <- TP /sum(confusion.matrix[1,])
recall <- TP /sum(confusion.matrix[,1])
f.score <- 2*precision*recall/ (precision+recall)
sprintf("The precision is %.4f", precision)
sprintf("The recall is %.4f", recall)
sprintf("The f-score is %.4f", f.score)
```