**##################################################################################**

**###################### ANALYSIS (SECTION 8.2: 39th Congress) ###########################**

**################ SIMULATION WITH 2 RANDOMLY SELECTED GROUPS #####################**

**################### Reading the data from file PrelimData.RData #########################**

**##################################################################################**

**rm(list = ls())**

**library(tm)**

**library(slam)**

**## LOADING DATA (39th Congress)**

**load("C:\\Johannes Ledolter\\2020March01Book\\Chapter8WEB\\PrelimData.RData")**

**## creating corpus**

**corpus <- VCorpus(VectorSource(data),readerControl = list(reader = readPlain))**

**corpus1 <- tm\_map(corpus, stripWhitespace)**

**corpus2 <- tm\_map(corpus1, content\_transformer(tolower))**

**corpus3 <- tm\_map(corpus2, removePunctuation)**

**corpus4 <- tm\_map(corpus3, removeNumbers)**

**corpus5 <- tm\_map(corpus4, removeWords, stopwords("english"))**

**corp.dtm <- DocumentTermMatrix(corpus5,control=list(stemming=FALSE)) ## no stemming is the default**

**dim(corp.dtm)**

**## omitting a certain term from the corpus such as "globe" or "unitedstates"**

**## take out senfrom gentfrom …. ???**

**dim(corp.dtm)**

**corp.dtm=corp.dtm[,!labels(corp.dtm)$Terms=="globe"]**

**dim(corp.dtm)**

**corp.dtm=corp.dtm[,!labels(corp.dtm)$Terms=="unitedstates"]**

**dim(corp.dtm)**

**corp.dtm=corp.dtm[,!labels(corp.dtm)$Terms=="gentfrom"]**

**dim(corp.dtm)**

**corp.dtm=corp.dtm[,!labels(corp.dtm)$Terms=="senfrom"]**

**dim(corp.dtm)**

**corp.dtm=corp.dtm[,!labels(corp.dtm)$Terms=="friendfrom"]**

**dim(corp.dtm)**

**corp.dtm=corp.dtm[,!labels(corp.dtm)$Terms=="sensreps"]**

**dim(corp.dtm)**

**quantile(len)**

**## cutting out short letters**

**## could do that, but the same letters may get cut when stripping words**

**indexlen=len>=32 ## can pick a length to omit short letters**

**nnn=dim(corp.dtm)[1]**

**number=dim(nnn)**

**for (i in 1:nnn) {**

**number[i]=i**

**}**

**corp.dtm=corp.dtm[number[indexlen],]**

**dim(corp.dtm)**

**meta2=meta2[number[indexlen]]**

**meta3=meta3[number[indexlen]]**

**meta4=meta4[number[indexlen]]**

**meta5=meta5[number[indexlen]]**

**meta6=meta6[number[indexlen]]**

**len=len[number[indexlen]]**

**## cutting out short letters**

**## working with / simplifying corpus**

**dim(corp.dtm)**

**## as.matrix(corp.dtm)**

**findFreqTerms(corp.dtm,10)**

**findFreqTerms(corp.dtm,2000)**

**dim(corp.dtm)**

**corpStripped.dtm=removeSparseTerms(corp.dtm,0.98)**

**## terms that show up in at least 2 percent of documents**

**dim(corpStripped.dtm)**

**## may want to include this**

**## Check for documents with zero words**

**dim(corpStripped.dtm)**

**index=row\_sums(corpStripped.dtm)==0**

**nnn=dim(corpStripped.dtm)[1]**

**number=dim(nnn)**

**for (i in 1:nnn) {**

**number[i]=i**

**}**

**number[index]**

**## do following only if there are empty documents**

**corpStripped.dtm=corpStripped.dtm[-number[index],]**

**dim(corpStripped.dtm)**

**meta2=meta2[-number[index]]**

**meta3=meta3[-number[index]]**

**meta4=meta4[-number[index]]**

**meta5=meta5[-number[index]]**

**meta6=meta6[-number[index]]**

**group=group[-number[index]]**

**len=len[-number[index]]**

**meta2orig=meta2**

**################ SIMULATION WITH 2 RANDOMLY SELECTED GROUPS #####################**

**####################### THE TWO GROUPS ARE NOT DIFFERENT #########################**

**meta2=meta2orig**

**length(meta2)**

**dim(corpStripped.dtm)**

**ndoc=dim(corpStripped.dtm)[1]**

**nt=round(ndoc\*0.5)**

**ndoc**

**nt**

**number=c(1:ndoc)**

**number**

**set.seed(1) ## to make the calculations reproducible in repeated runs**

**train <- sample(1:ndoc,nt)**

**numbersel=number[train]**

**numbersel**

**for (i in 1:ndoc) {**

**meta2[i]="group1"**

**}**

**for (i in 1:length(numbersel)) {**

**meta2[numbersel[i]]="group2"**

**}**

**meta2**

**table(meta2)**

**cor1=corpStripped.dtm**

**ndoc=dim(cor1)[1]**

**## considering occurrences**

**Bcor1=weightBin(cor1) ## to make binary**

**Bxx=as.matrix(Bcor1)**

**dim(Bxx)**

**## considering occurrences**

**## considering frequencies**

**Fxx=as.matrix(cor1)**

**dim(Fxx)**

**## considering frequencies**

**## determining test and evaluation documents**

**nt=round(ndoc\*0.50)**

**ndoc**

**nt**

**set.seed(2) ## to make the calculations reproducible in repeated runs**

**training <- sample(1:ndoc,nt)**

**## nearest neighbor method**

**library(class)**

**set.seed(2) ## as ties broken at random**

**nearest1 <- knn(train=Bxx[training,],test=Bxx[-training,],cl=meta2[training],k=1)**

**nearest5 <- knn(train=Bxx[training,],test=Bxx[-training,],cl=meta2[training],k=5)**

**nearest10 <- knn(train=Bxx[training,],test=Bxx[-training,],cl=meta2[training],k=10)**

**nearest25 <- knn(train=Bxx[training,],test=Bxx[-training,],cl=meta2[training],k=25)**

**data.frame(meta2[-training],nearest1,nearest5,nearest10,nearest25)[1:50,]**

**## calculate the proportion of correct classifications on this one training set**

**pcorrn1=100\*sum(meta2[-training]==nearest1)/(ndoc-nt)**

**pcorrn5=100\*sum(meta2[-training]==nearest5)/(ndoc-nt)**

**pcorrn10=100\*sum(meta2[-training]==nearest10)/(ndoc-nt)**

**pcorrn25=100\*sum(meta2[-training]==nearest25)/(ndoc-nt)**

**mpcorrn1=100-pcorrn1**

**mpcorrn5=100-pcorrn5**

**mpcorrn10=100-pcorrn10**

**mpcorrn25=100-pcorrn25**

**pcorrn1**

**pcorrn5**

**pcorrn10**

**pcorrn25**

**mpcorrn1**

**mpcorrn5**

**mpcorrn10**

**mpcorrn25**

**## NAÏVE BAYES Bernouilli (Package: naivebayes)**

**library(naivebayes)**

**pastdata=Bxx[training,]**

**pastcl=meta2[training]**

**testdata=Bxx[-training,]**

**testcl=meta2[-training]**

**bnb <- bernoulli\_naive\_bayes(x=pastdata,y=pastcl,laplace=1)**

**bnb**

**head(predict(bnb,newdata=testdata,type="class"))**

**head(predict(bnb,newdata=testdata,type="prob"))**

**clpred=predict(bnb,newdata=testdata,type="class")**

**tt=table(testcl,clpred)**

**tt**

**misprop=(tt[1,2]+tt[2,1])/(tt[1,1]+tt[1,2]+tt[2,1]+tt[2,2])**

**misprop**

**## NAÏVE BAYES Binomial (Package: naivebayes)**

**pastdata=Fxx[training,]**

**pastcl=meta2[training]**

**testdata=Fxx[-training,]**

**testcl=meta2[-training]**

**binomnb=multinomial\_naive\_bayes(pastdata,pastcl,prior=NULL,laplace=1)**

**binomnb**

**head(predict(binomnb,newdata=testdata,type="class"))**

**head(predict(binomnb,newdata=testdata,type="prob"))**

**clpred=predict(binomnb,newdata=testdata,type="class")**

**tt=table(testcl,clpred)**

**tt**

**misprop=(tt[1,2]+tt[2,1])/(tt[1,1]+tt[1,2]+tt[2,1]+tt[2,2])**

**misprop**