**BEFORE PREPROCESSING**

library(tm)

library(stringr)

data<-read.csv(file = "e:/sen\_ten.csv",header = TRUE,sep = ",")

some\_txt9<-0

pos.x1<-0

neg.y1<-0

total<-0

pos.words<-scan('positive-words.txt',what='character',comment.char=";")

neg.words<-scan('negative-words.txt',what='character',comment.char=";")

for(i in 1:1048588)

{

some\_txt1<-gsub('[[:punct:]]'," ",data[[3]][i])

#removing the control words

some\_txt2<-gsub('[[:cntrl:]]'," ",some\_txt1)

#removing the digits(numbers)

some\_txt3<-gsub('\\d+'," ",some\_txt2)

#removing the hyper links

some\_txt4<-gsub('http.\* \*','',some\_txt3)

#to remove any alphabets

some\_txt5<-gsub(pattern=" [a-z] ",replace=" ",some\_txt4)

some\_txt6<-gsub(pattern=" [A-Z] ",replace=" ",some\_txt5)

#converting to lower

some\_txt7<-tolower(some\_txt6)

#to get list of stopwords

#stopwords()

some\_txt8<-removeWords(some\_txt7,stopwords())

#to remove additional blankspaces

some\_txt9[i]<-stripWhitespace(some\_txt8)

#converting the dataframe into list

word.list<-str\_split(some\_txt9[i],'\\s+')

words<-unlist(word.list)

pos.x1[i]<-sum(!is.na(match(words,pos.words)))

neg.y1[i]<-sum(!is.na(match(words,neg.words)))

total[i]<-pos.x1[i]-neg.y1[i]

}

data1<-data.frame(pos.x1,neg.y1,total,some\_txt9)

write.csv(data1,file='e:/with\_total.csv',row.names=F)

data2<-data.frame(pos.x1,neg.y1,data[2],some\_txt9)

write.csv(data1,file='e:/with\_sentiment.csv',row.names=F)

**NAÏVE BAYES CODE**

data<-read.csv(file = "e:/with\_sentiment.csv",header = TRUE,sep = ",")

library(RTextTools)

library(e1071)

library(caret)

actual\_score<-ifelse(data$Sentiment=="1","POS","NEG")

data1<-data[,c(-3,-4)]

data1<-data.frame(data1,actual\_score)

# various scales of tweeets for analysis

j<-0

k<-0

for(i in 1:30)

{

if(i<=10)

{

j<-j+100

sentdata<-data1[1:j,]

}

if((i>=11)&&(i<=19))

{

j<-j+1000

sentdata<-data1[1:j,]

}

if((i>=20)&&(i<=29))

{

k<-k+100000

sentdata<-data1[1:k,]

}

if(i==30)

{

sentdata<-data1[1:1048588,]

}

#random sampling for training and testing

set.seed(2)

id=sample(2,nrow(sentdata),prob = c(0.7,0.3),replace = T)

sentrain<-sentdata[id==1,]

sentest<-sentdata[id==2,]

nrow(sentrain)

nrow(sentest)

#Naive bayes model

sent\_naive\_per<-naiveBayes(actual\_score~.,data=sentrain)

predicted\_score<-predict(sent\_naive\_per,sentest)

result<-confusionMatrix(table(predicted\_score,sentest$actual\_score))

a<-data.frame(result$table)

TN<-a$Freq[1]

FP<-a$Freq[2]

FN<-a$Freq[3]

TP<-a$Freq[4]

table1<-data.frame(TN,FP,FN,TP)

#tocsv <- data.frame(cbind(t(result$overall),t(result$byClass)))

#You can then use

#write.csv(tocsv,file="file.csv",row.names = F)

#pridicted outputs for test data

f=paste("e:/",lengths(sentdata),"tweets.csv",sep="")

output\_NB=data.frame(sentest,predicted\_score)

write.csv(output\_NB,file=f[1],row.names=F)

#statistical measures for test data

measures<-rbind(result$overall)

sensitivity1<-result$byClass["Sensitivity"]

specificity1<-result$byClass["Specificity"]

naiveBayes\_measures<-data.frame(measures,sensitivity1,specificity1,table1)

# appending to file

f=paste(lengths(sentdata),sep="")

if(i==1)

{

write.csv(naiveBayes\_measures,file='e:/naiveB\_measures.csv',row.names=f[1])

}

if(i>1)

write.table(naiveBayes\_measures,file="e:/naiveB\_measures.csv",sep=",",append=TRUE,quote=FALSE,col.names=FALSE,row.names =f[1])

}

**SVM CODE**

data<-read.csv(file = "e:/with\_sentiment.csv",header = TRUE,sep = ",")

library(RTextTools)

library(e1071)

library(caret)

#library(ggplot2)

#library(RColorBrewer)

actual\_score<-ifelse(data$Sentiment=="1","POS","NEG")

data1<-data[,c(-3,-4)]

data1<-data.frame(data1,actual\_score)

#various scales of tweeets for analysis

j<-0

for(i in 1:20)

{

if(i<=10)

{

j<-j+100

sentdata<-data1[1:j,]

}

if((i>=11)&&(i<=19))

{

j<-j+1000

sentdata<-data1[1:j,]

}

if(i==20)

{

sentdata<-data1[1:100000,]

}

#random sampling for training and testing

set.seed(2)

id=sample(2,nrow(sentdata),prob = c(0.7,0.3),replace = T)

sentrain<-sentdata[id==1,]

sentest<-sentdata[id==2,]

nrow(sentrain)

nrow(sentest)

##support vector machine

svm\_model<-svm(actual\_score~.,data=sentrain,kernel="sigmoid")#this default radial basis kernel

predicted<- predict(svm\_model,sentest)

result1<-confusionMatrix(table(predicted,sentest$actual\_score))

#table

a<-data.frame(result1$table)

TN<-a$Freq[1]

FP<-a$Freq[2]

FN<-a$Freq[3]

TP<-a$Freq[4]

table1<-data.frame(TN,FP,FN,TP)

#pridicted outputs for test data

output\_svm=data.frame(sentest,predicted)

f=paste("e:/",lengths(sentdata),"tweets.csv",sep="")

write.csv(output\_svm,file=f[1],row.names=F)

#statistical measures for test data

measures1<-rbind(result1$overall)

sensitivity2<-result1$byClass["Sensitivity"]

specificity2<-result1$byClass["Specificity"]

svm\_measures<-data.frame(measures1,sensitivity2,specificity2,table1)

#appending to file

f=paste(lengths(sentdata),sep="")

if(i==1)

{

write.csv(svm\_measures,file='e:/svm\_measures.csv',row.names=f[1])

}

if(i>1)

write.table(svm\_measures,file="e:/svm\_measures.csv",sep=",",append=TRUE,quote=FALSE,col.names=FALSE,row.names =f[1])

}

#data<-read.csv(file = "e:/svm\_measures.csv",header = TRUE,sep = ",")

**REAL – TIME IMPLEMENTATION**

library(twitteR)#httr packaGE

library(ROAuth)

library(stringr)

library(plyr)

library(tm)

options(max.print=5000)

requestURL <- "https://api.twitter.com/oauth/request\_token"

accessURL <- "https://api.twitter.com/oauth/access\_token"

authURL <- "https://api.twitter.com/oauth/authorize"

consumerKey <-"cLSXJU1kSMjIdUM9DukheLT31"

consumerSecret <- "FRFb1STdDFFIxyqwDvixM9hLHHFApaLaEdxN0d3aNgKdcZxFav"

access\_token<-"904728592722092032-oGUdbewFV0JlrbDZzExg6wGhhENVvWP"

access\_secret<-"a5G5oIaqiGIQ51VJvo0AA7UzRzDxxM6Ruosruhzds1BbS"

twitCred <- OAuthFactory$new(consumerKey=consumerKey,

consumerSecret=consumerSecret,

requestURL=requestURL,

accessURL=accessURL,

authURL=authURL)

twitCred$handshake()

setup\_twitter\_oauth(consumerKey,consumerSecret,access\_token,access\_secret)

some\_tweets<-searchTwitter(“Annamalai University",n=400,lang="en")

#saving the tweets in csv file

#write.csv(some\_tweets,"tweets.csv")

#for getting only the texts in the csv file

tweets=sapply(some\_tweets,function(x) x$getText())#error

# converting to dataframe

tweets1<-data.frame(tweets)

n<-nrow(tweets1)

pos.words<-scan('positive-words.txt',what='character',comment.char=";")

neg.words<-scan('negative-words.txt',what='character',comment.char=";")

pos<-0

neg<-0

some\_txt9<-0

for(i in 1:n)

{

some\_txt1<-gsub('[[:punct:]]'," ",tweets1[i,1])

#removing the control words

some\_txt2<-gsub('[[:cntrl:]]'," ",some\_txt1)

#removing the digits(numbers)

some\_txt3<-gsub('\\d+'," ",some\_txt2)

#remving the hyper links

some\_txt4<-gsub('http.\* \*','',some\_txt3)

#to remove any alphabets

some\_txt5<-gsub(pattern=" [a-z] ",replace=" ",some\_txt4)

some\_txt6<-gsub(pattern=" [A-Z] ",replace=" ",some\_txt5)

#converting to lower

some\_txt6a<- gsub("1/2", "",some\_txt6)

some\_txt7<-tolower(some\_txt6a)

#to get list of stopwords

#stopwords()

some\_txt8<-removeWords(some\_txt7,stopwords())

#to remove additional blankspaces

some\_txt9[i]<-stripWhitespace(some\_txt8)

#converting the dataframe into list

word.list<-str\_split(some\_txt9[i],'\\s+')

words<-unlist(word.list)

#words

pos[i]<-sum(!is.na(match(words,pos.words)))

neg[i]<-sum(!is.na(match(words,neg.words)))

}

data2<-data.frame(pos,neg,some\_txt9)

write.csv(data2,file='e:AU.csv',row.names=F)

#source('preprocess\_new.R')

data<-read.csv(file = "e:/with\_sentiment.csv",header = TRUE,sep = ",")

library(RTextTools)

library(e1071)

library(caret)

actual\_score<-ifelse(data$Sentiment=="1","POS","NEG")

data1<-data[,c(-3,-4)]

data1<-data.frame(data1,actual\_score)

#naive bayes model

#training

traindata<-data1[1:4000,]

set.seed(2)

id=sample(2,nrow(traindata),prob = c(0.7,0.3),replace = T)

sentrain<-traindata[id==1,]

#testing

data3<-data2[,-3]

actual\_score<-rep("NEG",400)

data4<-data.frame(data3,actual\_score)

sentest<-data4[1:400,]

sent\_naive\_per<-naiveBayes(actual\_score~.,data=sentrain)

predicted\_score<-predict(sent\_naive\_per,sentest)

res<-data.frame(sentest,predicted\_score)

write.csv(res,"NB\_prediction \_with\_negative\_dummy.csv",row.names = FALSE)

#svm model

pos<-0

neg<-0

svm\_model<-svm(actual\_score~.,data=sentrain,kernel="radial")

predicted<- predict(svm\_model,sentest)

result<-data.frame(sentest,predicted)

write.csv(result,"prediction \_with\_negative\_dummy.csv",row.names = FALSE)

data3<-data2[,-3]

actual\_score<-rep("POS",400)

data4<-data.frame(data3,actual\_score)

sentest<-data4[1:400,]

sent\_naive\_per<-naiveBayes(actual\_score~.,data=sentrain)

predicted\_score1<-predict(sent\_naive\_per,sentest)

res1<-data.frame(sentest,predicted\_score1)

write.csv(res1,"SVM\_prediction \_with\_negative\_dummy.csv",row.names = FALSE)

#svm model

pos<-0

neg<-0

svm\_model<-svm(actual\_score~.,data=sentrain,kernel="radial")

predicted<- predict(svm\_model,sentest)

result1<-data.frame(sentest,predicted)

write.csv(result,"prediction \_with\_positive\_dummy.csv",row.names = FALSE)