**ITMD 525- DATA MINING**

**CODE DOCUMENTATION**

**Predicting Twitter Sentiment Of The US Public Towards 2016 Presidential Election**

**By**

**Hemanth Inukonda Kamalakannan – A20369096**

**SreeKumar Selvam – A20372235**

**Srilekha Napa Ugandhar – A20370759**

**CODE TO PERFORM KNN CLASSIFIER**

**#KNN CLASSIFIERS**

#KNN CLASSIFIERS

setwd("/Users/sree/Desktop/Semester2/data mining/DataMining/KNN/Bag of Words/Fourth Fold")

#Libraries required to perform the classification operation including the preprocessing operations, and also for the accuracy operations

library(NLP)

library(wordcloud)

library(tm)

library(SnowballC)

library(plyr)

library(stringr)

library(class)

# Reading the input file or importing the input file containing the Twitter data

input <- read.csv(file="/Users/sree/Desktop/Semester2/data mining/DataMining/KNN/Bag of Words/Fourth Fold/Fourth fold.csv",head=TRUE,sep=",")

# converting the input data into a vector source

input.corp<-Corpus(VectorSource(input$text))

#function to read the inpult file and the input under the label text

CleanTweets<-function(input)

input.corp<-input$text

# the below ode is run for preprocessing the input data which includes removal of https tages, stemming , removal of stops words, punctuation, alphanumeric letters

input.corp<-gsub("\\.","",input.corp)

input.corp<-gsub(" http.\* \*","",input.corp)

input.corp<-gsub("(f|ht)tp(s?) \*://(.\*)[.][a-z]+", "", input.corp)

input.corp<- gsub("#\\w+","",input.corp)

input.corp <- gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", input.corp)

input.corp<- gsub("@\\w+", "", input.corp)

input.corp<- gsub("[[:punct:]]", "", input.corp)

#input.corp<- gsub("[[:cntrl:]]", "", input.corp)

input.corp<- gsub("[[:digit:]]","",input.corp)

input.corp<- gsub("[^a-zA-Z0-9]"," ",input.corp)

input.corp<- gsub("^\\s+", "", input.corp)

input.corp<- gsub("[\r\eval]", "", input.corp)

input.corp<-gsub("\\.","",input.corp)

# Convereting the preprocessed data into vector corpuse source which is used again for further processing and text transformation if in case if there is any extra

input.corp<- VCorpus(VectorSource(input.corp))

input.corp <- tm\_map(input.corp,content\_transformer(function(x) iconv(x, to='UTF-8-MAC', sub='byte')),mc.cores=1)

input.corp <- tm\_map(input.corp, content\_transformer(tolower), mc.cores=1)

input.corp <- tm\_map(input.corp, removeNumbers)

input.corp <- tm\_map(input.corp, stripWhitespace)

input.corp <- tm\_map(input.corp, removePunctuation, mc.cores=1)

input.corp <- tm\_map(input.corp, stemDocument, language = "english")

input.corp <- tm\_map(input.corp, function(x)removeWords(x,stopwords("english")), mc.cores=1)

#user entered stop words which we found less frequently appearing in the input data

#input.corp<- tm\_map(input.corp, removeWords,c(stopwords("english")))

mystopwords<-c("a","able","about","across","after","all","almost","also","am","among",

"an","and","any","are","as","at","be","because","been","but","by","can",

"cannot","could","dear","did","do","does","either","else","ever","every","for",

"from","get","got","had","has","have","he","her","hers","him","his","how","however","i","if",

"in","into","is","it","its","just","least","let","like","likely","may","me","might","most","must",

"my","neither","nor","of","off","often","on","only","or","other","our","own","rather","said",

"say","says","she","should","since","so","some","than","that","the","their","them","then","there","these",

"they","this","tis","to","too","twas","us","wants","was","we","were","what","when","where","which","while",

"who","whom","why","will","with","would","yet","you","your","last","amp","night","fox","gop","one","can","amp","just",

"get","going","still","term","now","httpstcobhvimxjew","even","anything","back","done","gonna","keep","know","make", "much",

"nothing","rep","right","see","thats","really","yall","thats","want","pass", "two","thing","things","though","today","tonight",

"take","rep","run","running","ryan","scotus","remember","potus", "please","next","needs","made","makes","many","looking","lot",

"look","lets","gets","give","goes","happen","hes","forget","end","everyone","everything","dems","day","delaware","dem", "come",

"check","another","actually","gotta")

input.corp=tm\_map(input.corp,removeWords,mystopwords)

input.corp

input.corp<-tm\_map(input.corp,PlainTextDocument)

input.corp[[1019]]$content

#converting the cleaned data into a datfrm

datfrm<-data.frame(text=unlist(sapply(input.corp, `[`, "content")), stringsAsFactors=F)

datfrm[1:15,]

#converting the datfrm into a document term matrix

doctrmmatrix <- DocumentTermMatrix(input.corp)

m <- as.matrix(doctrmmatrix)

dim(m)

#now the converted matrix variable is again changed to a data frame

matrix.df <- as.data.frame(data.matrix(doctrmmatrix),row.names=NULL, stringsAsfactors = FALSE)

dim(doctrmmatrix)

# Binding the column category with the know classification

matrix.df <- cbind(matrix.df, input$sentiment,row.names=NULL)

#changing the name of the newly added column to labels

colnames(matrix.df)[ncol(matrix.df)] <- "labels"

# training and testdataing the input data in the ration of 80:20

traindata <- sample(nrow(matrix.df), ceiling(nrow(matrix.df) \* 0.80),replace=TRUE)

testdata <- (1:nrow(matrix.df))[- traindata]

classi <- matrix.df[,"labels"]

#Data model is created and after tht the newly named category is removed data

modelingdat <- matrix.df[,!colnames(matrix.df) %in% "labels"]

modelingdat

#producing the knn classifier model

knn.predi <- knn(modelingdat[traindata, ], modelingdat[testdata, ],classi[traindata],k=20)

knn.predi

write.csv(knn.predi,file="BagOfwordsPredictedFOld.csv")

#producing the confusion matrix

confusio.mat <- table("Predictions" = knn.predi, Actual = classi[testdata])

confusio.mat

# lable to produce the accuracy precison and recall

eval = sum(confusio.mat)

noofcol = nrow(confusio.mat) # number of classiasses

diagonl = diag(confusio.mat)

sumofrow = apply(confusio.mat, 1, sum)

# number of instanoofcoles per classiass

sumofcolumn = apply(confusio.mat, 2, sum)

precvalue = (sumofrow / eval) # distribution of instances over the actual classiasses

reclval = (sumofcolumn / eval)

accuracy = (sum(diagonl) / eval)

accuracy

(accuracy <- sum(diag(confusio.mat))/length(testdata) \* 100)

#code to calculate the precision and recall

precision = diagonl / sumofcolumn

recall = diagonl / sumofrow

f1 = 2 \* precision \* recall / (precision + recall)

data.frame(sum((precision)/3))

data.frame(sum((recall)/3))

data.frame(sum((f1)/3))

**CODE TO PERFORM MAXIMUM ENTROPY CLASSIFIER**

setwd("/Users/sree/Desktop/Semester2/datamining/DataMining/KNN/Bag of Words/Fourth Fold")

**#Libraries required to perform the classification operation including the preprocessing** **operations, and also for the accuracy operations**

library(NLP)

library(wordcloud)

library(tm)

library(SnowballC)

library(plyr)

library(stringr)

library(class)

**# Reading the input file or importing the input file containing the Twitter data**

input <- read.csv(file="/Users/sree/Desktop/Semester2/data mining/DataMining/KNN/Bag of Words/Fourth Fold/Fourth fold.csv",head=TRUE,sep=",")

**# converting the input data into a vector source**

input.corp<-Corpus(VectorSource(input$text))

**#function to read the inpult file and the input under the label text**

CleanTweets<-function(input)

input.corp<-input$text

**# the below ode is run for preprocessing the input data which includes removal of https tages, stemming , removal of stops words, punctuation, alphanumeric letters**

input.corp<-gsub("\\.","",input.corp)

input.corp<-gsub(" http.\* \*","",input.corp)

input.corp<-gsub("(f|ht)tp(s?) \*://(.\*)[.][a-z]+", "", input.corp)

input.corp<- gsub("#\\w+","",input.corp)

input.corp <- gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", input.corp)

input.corp<- gsub("@\\w+", "", input.corp)

input.corp<- gsub("[[:punct:]]", "", input.corp)

#input.corp<- gsub("[[:cntrl:]]", "", input.corp)

input.corp<- gsub("[[:digit:]]","",input.corp)

input.corp<- gsub("[^a-zA-Z0-9]"," ",input.corp)

input.corp<- gsub("^\\s+", "", input.corp)

input.corp<- gsub("[\r\eval]", "", input.corp)

input.corp<-gsub("\\.","",input.corp)

**# Convereting the preprocessed data into vector corpuse source which is used again for further processing and text transformation if in case if there is any extra**

input.corp<- VCorpus(VectorSource(input.corp))

input.corp <- tm\_map(input.corp,content\_transformer(function(x) iconv(x, to='UTF-8-MAC', sub='byte')),mc.cores=1)

input.corp <- tm\_map(input.corp, content\_transformer(tolower), mc.cores=1)

input.corp <- tm\_map(input.corp, removeNumbers)

input.corp <- tm\_map(input.corp, stripWhitespace)

input.corp <- tm\_map(input.corp, removePunctuation, mc.cores=1)

input.corp <- tm\_map(input.corp, stemDocument, language = "english")

input.corp <- tm\_map(input.corp, function(x)removeWords(x,stopwords("english")), mc.cores=1)

**#user entered stop words which we found less frequently appearing in the input data**

#input.corp<- tm\_map(input.corp, removeWords,c(stopwords("english")))

mystopwords<-c("a","able","about","across","after","all","almost","also","am","among",

"an","and","any","are","as","at","be","because","been","but","by","can",

"cannot","could","dear","did","do","does","either","else","ever","every","for",

"from","get","got","had","has","have","he","her","hers","him","his","how","however","i","if",

"in","into","is","it","its","just","least","let","like","likely","may","me","might","most","must",

"my","neither","nor","of","off","often","on","only","or","other","our","own","rather","said",

"say","says","she","should","since","so","some","than","that","the","their","them","then","there","these",

"they","this","tis","to","too","twas","us","wants","was","we","were","what","when","where","which","while",

"who","whom","why","will","with","would","yet","you","your","last","amp","night","fox","gop","one","can","amp","just",

"get","going","still","term","now","httpstcobhvimxjew","even","anything","back","done","gonna","keep","know","make", "much",

"nothing","rep","right","see","thats","really","yall","thats","want","pass", "two","thing","things","though","today","tonight",

"take","rep","run","running","ryan","scotus","remember","potus", "please","next","needs","made","makes","many","looking","lot",

"look","lets","gets","give","goes","happen","hes","forget","end","everyone","everything","dems","day","delaware","dem", "come",

"check","another","actually","gotta")

input.corp=tm\_map(input.corp,removeWords,mystopwords)

input.corp

input.corp<-tm\_map(input.corp,PlainTextDocument)

input.corp[[1019]]$content

**#to print the data into a matrix form**

corp <- create\_matrix(input[,1))

matrix

mat <-as.matrix(corp)

mat

crecontain <create\_container(corp,as.numeric(as.factor(input[,2])),trainSize = 1:4000,testSize = 4001:5000, virgin=FALSE)

crecontain

**# function to perform the maximum entropy function which is divided into the ratio of 80% and 20%**

~~trmodel~~ = train\_model(crecontain ,algorithm = c("MAXENT"))

trmodel

finres = classify\_model(crecontain, trmodel)

finres

write.csv(finres,file="FutureFold.csv")

**# function to classify the labels into maximum entropy classifier**

t<-table(as.numeric(as.factor(input[4001:5000, 2])), finres[,"MAXENTROPY\_LABEL" ])

finres[,"MAXENTROPY\_LABEL"])

**#function to exhibit the evaluation metrics such as precision, recall and f1 score**

analytics = create\_analytics(crecontain, finres)

summary(analytics)

numberofins <-sum(t)

diagonal <- diag(t)

accuracy <-sum(diagonal)/numberofins

accuracy

**#to print the accuracy of the algorithm**

recall\_accuracy(as.numeric(as.factor(input[4001:5000, 2])),

(accuracy <- sum(diag(t))/length(numberofins) \* 100/1000)

**CODE TO PERFORM SUPPORT VECTOR MACHINE**

setwd("/Users/sree/Desktop/Semester2/datamining/DataMining/KNN/Bag of Words/Fourth Fold")

**#Libraries required to perform the classification operation including the preprocessing** **operations, and also for the accuracy operations**

library(NLP)

library(wordcloud)

library(tm)

library(SnowballC)

library(plyr)

library(stringr)

library(class)

LIBRARY(rtext)

Library(RtextTools)

**# Reading the input file or importing the input file containing the Twitter data**

input <- read.csv(file="/Users/sree/Desktop/Semester2/data mining/DataMining/KNN/Bag of Words/Fourth Fold/Fourth fold.csv",head=TRUE,sep=",")

**# converting the input data into a vector source**

input.corp<-Corpus(VectorSource(input$text))

**#function to read the inpult file and the input under the label text**

CleanTweets<-function(input)

input.corp<-input$text

**# the below ode is run for preprocessing the input data which includes removal of https tages, stemming , removal of stops words, punctuation, alphanumeric letters**

input.corp<-gsub("\\.","",input.corp)

input.corp<-gsub(" http.\* \*","",input.corp)

input.corp<-gsub("(f|ht)tp(s?) \*://(.\*)[.][a-z]+", "", input.corp)

input.corp<- gsub("#\\w+","",input.corp)

input.corp <- gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", input.corp)

input.corp<- gsub("@\\w+", "", input.corp)

input.corp<- gsub("[[:punct:]]", "", input.corp)

#input.corp<- gsub("[[:cntrl:]]", "", input.corp)

input.corp<- gsub("[[:digit:]]","",input.corp)

input.corp<- gsub("[^a-zA-Z0-9]"," ",input.corp)

input.corp<- gsub("^\\s+", "", input.corp)

input.corp<- gsub("[\r\eval]", "", input.corp)

input.corp<-gsub("\\.","",input.corp)

**# Convereting the preprocessed data into vector corpuse source which is used again for further processing and text transformation if in case if there is any extra**

input.corp<- VCorpus(VectorSource(input.corp))

input.corp <- tm\_map(input.corp,content\_transformer(function(x) iconv(x, to='UTF-8-MAC', sub='byte')),mc.cores=1)

input.corp <- tm\_map(input.corp, content\_transformer(tolower), mc.cores=1)

input.corp <- tm\_map(input.corp, removeNumbers)

input.corp <- tm\_map(input.corp, stripWhitespace)

input.corp <- tm\_map(input.corp, removePunctuation, mc.cores=1)

input.corp <- tm\_map(input.corp, stemDocument, language = "english")

input.corp <- tm\_map(input.corp, function(x)removeWords(x,stopwords("english")), mc.cores=1)

**#user entered stop words which we found less frequently appearing in the input data**

#input.corp<- tm\_map(input.corp, removeWords,c(stopwords("english")))

mystopwords<-c("a","able","about","across","after","all","almost","also","am","among",

"an","and","any","are","as","at","be","because","been","but","by","can",

"cannot","could","dear","did","do","does","either","else","ever","every","for",

"from","get","got","had","has","have","he","her","hers","him","his","how","however","i","if",

"in","into","is","it","its","just","least","let","like","likely","may","me","might","most","must",

"my","neither","nor","of","off","often","on","only","or","other","our","own","rather","said",

"say","says","she","should","since","so","some","than","that","the","their","them","then","there","these",

"they","this","tis","to","too","twas","us","wants","was","we","were","what","when","where","which","while",

"who","whom","why","will","with","would","yet","you","your","last","amp","night","fox","gop","one","can","amp","just",

"get","going","still","term","now","httpstcobhvimxjew","even","anything","back","done","gonna","keep","know","make", "much",

"nothing","rep","right","see","thats","really","yall","thats","want","pass", "two","thing","things","though","today","tonight",

"take","rep","run","running","ryan","scotus","remember","potus", "please","next","needs","made","makes","many","looking","lot",

"look","lets","gets","give","goes","happen","hes","forget","end","everyone","everything","dems","day","delaware","dem", "come",

"check","another","actually","gotta")

input.corp=tm\_map(input.corp,removeWords,mystopwords)

input.corp

input.corp<-tm\_map(input.corp,PlainTextDocument)

input.corp[[1019]]$content

nb <-ncol(input)

**# retrieve the id column**

ids <- input[,1]

# retrieve the class column

class <-as.character(input[,nb])

**#creating the matrix of the input**

corp <- create\_matrix(input[,1])

mat <-as.matrix(corp)

mat

**# creating a container of variables and dividing the rwation into 80 and 20 percent**

crecontain <- create\_container(corp,as.numeric(as.factor(input[,2])),trainSize = 1:5000,testSize = 5001:6000, virgin=FALSE)

crecontain

**#creating a model for the algorithm**

trmodel = train\_model(crecontain ,algorithm = c("SVM"))

trmodel

**#classifying the models based on the container models and the models that is used for training**

finres = classify\_model(crecontain, trmodel)

finres

write.csv(finres,file="BagofWordsSecondFold.csv")

**# function to classify the labels into maximum entropy classifier**

t<-table(as.numeric(as.factor(input[4001:5000, 2])), results[,"SVM\_LABEL" ])

t

typeof(t)

recall\_accuracy(as.numeric(as.factor(input[4001:5000, 2])), results[,"SVM\_LABEL"])

**#function to exhibit the evaluation metrics such as precision, recall and f1 score**

analytics = create\_analytics(crecontain, results)

summary(analytics)

n = sum(conf.mat) # number of instances

nc = nrow(conf.mat) # number of classes

diag = diag(conf.mat) # number of correctly classified instances per class

rowsums = apply(conf.mat, 1, sum) # number of instances per class

colsums = apply(conf.mat, 2, sum) # number of predictions per class

p = rowsums / n # distribution of instances over the actual classes

q = colsums / n # distribution of instances over the predicted classes

accuracy = sum(diag) / n

accuracy

precision = diag / colsums

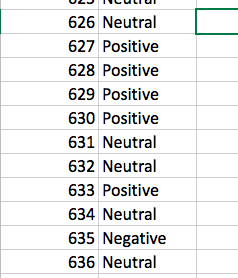
recall = diag / rowsums

f1 = 2 \* precision \* recall / (precision + recall)

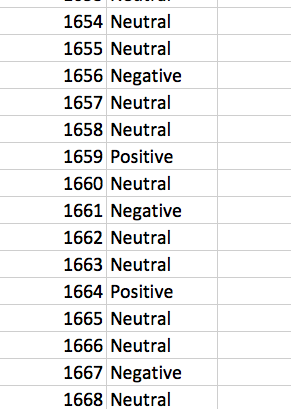
data.frame(precision, recall, f1)

**SAMPLE CODE OUTPUT FOR KNN CLASSIFIER OF THE FUTURE PREDICTION IN CSV FORMAT**

**METHOD 1 MANUAL**

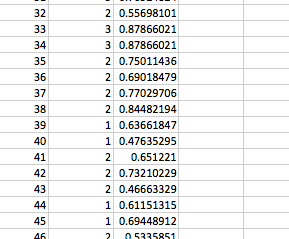


**METHOD 2 BAG OF WORDS**

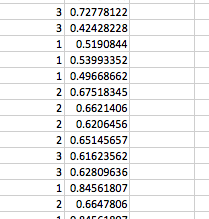
****

**SAMPLE CODE OUTPUT OF THE SVM CLASSIIFIER OF FUTURE PREDICTION IN CSV FORMAT**

**METHOD 1 MANUAL**

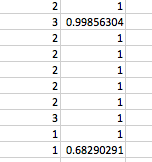
****

**METHOD 2 BAG OF WORDS**

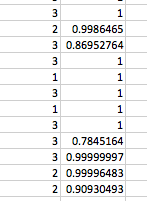
****

**SAMPLE CODE OUTPUT OF THE MAX ENTROPY CLASSIIFIER OF FUTURE PREDICTION IN CSV FORMAT**

**METHOD 1 MANUAL**

****

**METHOD 2 BAG OF WORDS**

****