#HOW TO CONVERT TO UTF-8

str(iconv(enc2utf8(tweet\_df$text),sub="byte"))

#INITIALIZE PACKAGES

library("gmodels", lib.loc="~/R/win-library/3.2")

library("e1071", lib.loc="~/R/win-library/3.2")

library("NLP", lib.loc="~/R/win-library/3.2")

library("jsonlite", lib.loc="~/R/win-library/3.2")

library("RCurl", lib.loc="~/R/win-library/3.2")

library("plyr", lib.loc="~/R/win-library/3.2")

library("rjson", lib.loc="~/R/win-library/3.2")

library("rJava", lib.loc="~/R/win-library/3.2")

library("streamR", lib.loc="~/R/win-library/3.2")

library("ROAuth", lib.loc="~/R/win-library/3.2")

library("RJSONIO", lib.loc="~/R/win-library/3.2")

library("twitteR", lib.loc="~/R/win-library/3.2")

library("tm", lib.loc="~/R/win-library/3.2")

library("wordcloud", lib.loc="~/R/win-library/3.2")

#SET WORKING DIRECTORY

setwd ("C:/Users/fassster/Documents/R/Rsample")

#REMOVE PUNCTUATIONS EXCEPT FOR @ AND #

#TO USE WRAP THROUGH: content\_transformer(removeMostPunctuation)

removeMostPunctuation<-

function (x, preserve\_intra\_word\_dashes = FALSE)

{

rmpunct <- function(x) {

x <- gsub("#", "\002", x)

x <- gsub("@", "\003", x)

x <- gsub("[[:punct:]]+", " ", x)

gsub("\003", "@", x, fixed = TRUE)

gsub("\002", "#", x, fixed = TRUE)

}

if (preserve\_intra\_word\_dashes) {

x <- gsub("(\\w)-(\\w)", "\\1\001\\2", x)

x <- rmpunct(x)

gsub("\001", "-", x, fixed = TRUE)

} else {

rmpunct(x)

}

}

#IMPORT CSV

tagged\_tweets\_df <- read.csv("tl\_tweets\_tagged.csv", header = TRUE, sep = ",")

#PUT TEXT INTO CORPUS

tweet\_texts <- tagged\_tweets\_df$Text

tweet\_corpus <- VCorpus (VectorSource(tweet\_texts))

#CLEANING OF CORPUS

tweet\_corpus\_clean <- tm\_map(tweet\_corpus, content\_transformer(tolower))

tweet\_corpus\_clean <- tm\_map(tweet\_corpus\_clean, removeNumbers)

tweet\_corpus\_clean <- tm\_map(tweet\_corpus\_clean, removeWords,

c(stopwords(),"ang","sa","mga","https","nga","otwol","aldub","hahaha","haha","opo","mag","yes","lang","pero","kakatawa","votekathyrnfpp","tagos","#aldubthmonthsary","#kca"))

tweet\_corpus\_clean <- tm\_map(tweet\_corpus\_clean, content\_transformer(removeMostPunctuation),preserve\_intra\_word\_dashes = TRUE)

tweet\_corpus\_clean <- tm\_map(tweet\_corpus\_clean, stripWhitespace)

as.character(tweet\_corpus[[180]])

as.character(tweet\_corpus\_clean[[180]])

tweet\_corpus\_clean\_df<-data.frame(text=unlist(sapply(tweet\_corpus\_clean, `[`, "content")), stringsAsFactors=F)

#PUTTING INTO A DATA TERM MATRIX

tweet\_dtm <- DocumentTermMatrix(tweet\_corpus\_clean)

str(tweet\_dtm)

#TRAINING

tweet\_total\_count <- 4292

tweet\_training\_count <- 3219

tweet\_dtm\_train <- tweet\_dtm[1:tweet\_training\_count, ]

tweet\_dtm\_test <- tweet\_dtm [(tweet\_training\_count+1):tweet\_total\_count, ]

tweet\_train\_labels <- tagged\_tweets\_df[1:tweet\_training\_count, ]$Tag

tweet\_test\_labels <- tagged\_tweets\_df[(tweet\_training\_count+1):tweet\_total\_count, ]$Tag

prop.table(table(tweet\_train\_labels))

prop.table(table(tweet\_test\_labels))

tweet\_freq\_words <- findFreqTerms(tweet\_dtm\_train, 5)

tweet\_dtm\_freq\_train<- tweet\_dtm\_train[ , tweet\_freq\_words]

tweet\_dtm\_freq\_test <- tweet\_dtm\_test[ , tweet\_freq\_words]

#CONVERT COUNT FUNCTION

convert\_counts <- function(x) {

x <- ifelse(x > 0, "Yes", "No")

}

tweet\_train <- apply(tweet\_dtm\_freq\_train, MARGIN = 2,

convert\_counts)

tweet\_test <- apply(tweet\_dtm\_freq\_test, MARGIN = 2,

convert\_counts)

#START NAIVE BAYES

tweet\_classifier <- naiveBayes(tweet\_train, as.factor(tweet\_train\_labels))

tweet\_test\_pred <- predict(tweet\_classifier, tweet\_test)

CrossTable(tweet\_test\_pred, tweet\_test\_labels,

prop.chisq = FALSE, prop.t = FALSE,

dnn = c('predicted', 'actual'))

#IMPROVED NAIVE BAYES

tweet\_classifier2 <- naiveBayes(tweet\_train, as.factor(tweet\_train\_labels),

laplace = 1)

tweet\_test\_pred2 <- predict(tweet\_classifier2, tweet\_test)

CrossTable(tweet\_test\_pred2, tweet\_test\_labels,

prop.chisq = FALSE, prop.t = FALSE,

dnn = c('predicted', 'actual'))