```
library(caret)
library(randomForest)
library(e1071)
library(quanteda)
library(irlba)
library(tidyverse)
library(data.table)
library(tm)
library(ggplot2)
library(RWeka)
library(wordcloud)
library(stringr)
library(babynames)
library(doSNOW)
library(lsa)

options(warn=-1)
prgm.start<-Sys.time()
set.seed(123456789)

###
loadData<-function(fileToOpen,stringAsFactorValue){
  fileData<-read.csv(fileToOpen,stringsAsFactors = stringAsFactorValue)
  return(fileData)
}

trimSpace<-function(fileData){

  fileData$ShortDesc <- stringr::str_replace_all(fileData$ShortDesc,"[^a-
zA-Z\\s]", "  ")
  fileData$ShortDesc <-
stringr::str_replace_all(fileData$ShortDesc,"[\\s]+", " ")

  # Get rid of trailing "" if necessary
  indexes <- which(fileData$ShortDesc== " ")
  if(length(indexes) > 0){
    fileData$ShortDesc <- fileData$ShortDesc[-indexes]
  }

  i<-1
  while(i <= nrow(fileData)){
    fileData$ShortDesc[i]<-gsub("\\b[a-zA-Z]{1,1}\\b", " ",
fileData$ShortDesc[i])
    i<-i+1
  }
  fileData$ShortDesc <-
stringr::str_replace_all(fileData$ShortDesc,"[\\s]+", " ")
  return(fileData)
}

cleanTokens<-function(rawTokens){

  rawTokens <- tokens_tolower(rawTokens)
  rawTokens <- tokens_select(rawTokens, allStopwords, selection =
"remove")
  rawTokens <- tokens_wordstem(rawTokens, language = "english")
  return(rawTokens)

}
```

```
term.frequency <- function(row) {
  row / sum(row)
}

inverse.doc.freq <- function(col) {
  corpus.size <- length(col)
  doc.count <- length(which(col > 0))

  log10(corpus.size / doc.count)
}

tf.idf <- function(x, idf) {
  x * idf
}


tickets<-loadData("C:/4class_Larger-Changed-Labels.csv",FALSE)

tickets <- tickets[,-c(10,11)]
dim(tickets)
names(tickets) <- c( "Ticket Number","Ticket Create Date","Ticket Solve
Date",
                  "Ticket Start Month","Ticket Close Month","Ticket
Solver Org",
                  "Ticket Solver","Ticket Closer Org","Ticket
Type","ShortDesc","Log","Label","Label 2")

ticketsCopy<-tickets

#All stopwords and excpetions
#stopwords.names<-unique(babynames$name)
stopwords.common<-
c("subject","incident","assignment","message","service","ticket","priorit
y","cism","crm","defect","mbj","mbk","jp","kr")
custom.data.stopwords<-c(tickets$`Ticket Solver`,tickets$`Ticket Start
Month`,tickets$`Ticket Close Month`)
stopwords_exceptions<-c("Case","vin","access","job")
allStopwords<-
c(stopwords("en"),stopwords("SMART"),custom.data.stopwords,stopwords.comm
on)
allStopwords<-setdiff(allStopwords, stopwords_exceptions)


length(which(!complete.cases(tickets)))
tickets$Label <- as.factor(tickets$Label)
str(tickets$Label)

###
tickets<-trimSpace(tickets)

index<-createDataPartition(tickets$Label,p=0.80,times=1,list=FALSE)
trainTickets<-tickets[index,]
testTickets<-tickets[-index,]
dim(trainTickets)
dim(testTickets)

levels(trainTickets)<-c("Access Mgmt","Report","UI-BE","Interface")


trainTickets.tokens <- tokens(trainTickets$ShortDesc, what = "word",
```

```
                            remove_numbers = TRUE, remove_punct = TRUE,
                            remove_symbols = TRUE, remove_hyphens = TRUE)

trainTickets.tokens<-cleanTokens(trainTickets.tokens)

cv.folds <- createMultiFolds(trainTickets$ShortDesc, k = 10, times = 3)
cv.cntrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3,
index = cv.folds)

trainTickets.ngram<-tokens_ngrams(trainTickets.tokens,n =1:3)

trainTickets.ngram.dfm <- dfm(trainTickets.ngram,stem = TRUE,tolower =
FALSE)
trainTickets.ngram.matrix <- as.matrix(trainTickets.ngram.dfm)

trainTickets.ngram.tf <- apply(trainTickets.ngram.matrix, 1,
term.frequency)
trainTickets.ngram.idf <- apply(trainTickets.ngram.matrix, 2,
inverse.doc.freq)
trainTickets.ngram.tfidf <-  apply(trainTickets.ngram.tf, 2, tf.idf, idf
= trainTickets.ngram.idf)
trainTickets.ngram.tfidf <- t(trainTickets.ngram.tfidf)

# #making clean data frame again
trainTickets.ngram.tfidf.df<-
cbind(Label=trainTickets$Label,data.frame(trainTickets.ngram.tfidf))
names(trainTickets.ngram.tfidf.df)<-
make.names(names(trainTickets.ngram.tfidf.df))

c<-ncol(trainTickets.ngram.tfidf.df)

i<-1
while(i <= c){
  colnames(trainTickets.ngram.tfidf.df)[i] <- gsub("\\.$", "",
colnames(trainTickets.ngram.tfidf.df)[i])
  i<-1+i
}


#Garbage collection ie free memory
gc()


# Create a cluster to work on 3 logical cores.
cl <- makeCluster(3, type = "SOCK")
registerDoSNOW(cl)

# Time the code execution
start.time <- Sys.time()


# gives the best result!
model_rf.1_l1 <- train(Label ~ .,  data = trainTickets.ngram.tfidf.df,
method = "rf",
                       trControl = cv.cntrl, tuneLength = 3)

# Processing is done, stop cluster.
stopCluster(cl)

# Total time of execution on workstation was
```

```
 total.time <- Sys.time() - start.time
 total.time


 #Check out our results.
 model_rf.1_l1
```

```
model_rf.1_l1
cm_l1<-
confusionMatrix(trainTickets.ngram.tfidf.df$Label,model_rf.1_l1$finalMode
l$predicted)
cm_l1
```