Project Report Code

sahil

October 19, 2016

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Data Cleaning and Transformation

First we load the dataset for the year of 2015. We have added an extra column which signifies whether the violation is related to sanitation or not.

```
library(chron)
library(lubridate)
library(caret)

#df <- read.csv("https://s3-us-west-2.amazonaws.com/fds2016sahil/default2015+(1).csv",na.strings=c("","

df <- read.table("data.txt",na.strings=c("","NA"))
#only select the rows where decision hearing result is either against the respondent("IN VIOLATION")
#or for the respondent("DISMISSED")
selected <- c("DISMISSED","IN VIOLATION")
df1 <- df[df$Hearing.Result %in% selected,]
df_unique <- unique(df$Issuing.Agency)
indx <- sapply(df1, is.factor)
#remove dollar sign
df1[indx] <- lapply(df1[indx], function(x)
                           as.character(gsub("\\$", "", x)))
#convert every column to factor
df1<-data.frame(lapply(df1,factor))
#convert date time to Datetime Format
df1$Violation.Date<- as.Date(df1$Violation.Date, "%m/%d/%Y")
df1$Violation.Time <- chron(times. = df1$Violation.Time)



newdf <- data.frame(df1[,c("Violation.Date","Violation.Time","Issuing.Agency","Violation.Location..Borou
#convert hearing date and time to date time format
newdf$Hearing.Time <- chron(times. = newdf$Hearing.Time)
newdf$Hearing.Date <- as.Date(newdf$Hearing.Date,"%m/%d/%Y")
#remove Issuing Agency, Hearing Location and Violation Locations whose violations are less than 500,10,
newdf <- newdf[!(as.numeric(newdf$Issuing.Agency) %in% which(table(newdf$Issuing.Agency)<500)),]
newdf <- newdf[!(as.numeric(newdf$Scheduled.Hearing.Location) %in% which(table(newdf$Scheduled.Hearing.
newdf <- newdf[!(as.numeric(newdf$Violation.Location..Borough.) %in% which(table(newdf$Violation.Locati
newdf <- droplevels(newdf)
#convert Violation Amount to numeric
```

```
newdf$Total.Violation.Amount <- as.numeric(as.character(newdf$Total.Violation.Amount))
set.seed(22)
pIndex <- createDataPartition(newdf$Issuing.Agency, p = .1,
                                  list = FALSE,
                                  times = 1)
```

In our initial exploration of which predictors to use we find that Violation Time, Issuing Agency, Violation Location, Hearing Location, Violation Time and Total Violation Amount have highest coefficient values.

```
newdf <- newdf[pIndex,]
model1 <-glm(Hearing.Result~.,data = newdf,family = 'binomial')
summary(model1)
```

```
##
## Call:
## glm(formula = Hearing.Result ~ ., family = "binomial", data = newdf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8641  -0.8134   0.2357   0.6121   2.3715
##
## Coefficients:
##                                              Estimate Std. Error z value
## (Intercept)                                 1.006e+00  3.586e+00   0.281
## Violation.Date                              5.383e-03  4.512e-04  11.929
## Violation.Time                              1.502e-01  1.456e-01   1.032
## Issuing.AgencyDEP - BUREAU OF ENV. COMPLIANC  5.244e+00  6.148e-01   8.529
## Issuing.AgencyDEP - IWC                     4.611e+00  7.434e-01   6.203
## Issuing.AgencyDEPT. OF BUILDINGS            4.517e+00  5.995e-01   7.533
## Issuing.AgencyDEPT OF TRANSPORTATION        3.426e+00  5.996e-01   5.713
## Issuing.AgencyDOH MENTAL HEALTH             2.826e+00  6.006e-01   4.705
## Issuing.AgencyDOITT                         2.903e+00  6.791e-01   4.275
## Issuing.AgencyDOS - ENFORCEMENT AGENTS      2.265e+00  5.980e-01   3.787
## Issuing.AgencyFIRE DEPARTMENT OF NYC        5.952e+00  6.020e-01   9.886
## Issuing.AgencyNYPD TRANSPORT INTELLIGENCE DI  3.842e+00  6.189e-01   6.208
## Issuing.AgencyPARKS DEPARTMENT              2.176e+00  6.229e-01   3.493
## Issuing.AgencyPCS - DOHMH                   2.569e+00  6.080e-01   4.225
## Issuing.AgencyPOLICE DEPARTMENT             2.200e+00  6.032e-01   3.646
## Issuing.AgencySANITATION OTHERS             1.887e+00  5.984e-01   3.154
## Issuing.AgencySANITATION POLICE             1.827e+00  6.220e-01   2.938
## Issuing.AgencySANITATION RECYCLING          2.172e+00  6.090e-01   3.566
## Issuing.AgencyVETERINARY-DOHMH              2.050e+00  6.622e-01   3.096
## Violation.Location..Borough.BROOKLYN        -1.534e-01  1.107e-01  -1.385
## Violation.Location..Borough.MANHATTAN       -1.055e-01  1.006e-01  -1.048
## Violation.Location..Borough.QUEENS          -5.110e-02  1.137e-01  -0.449
## Violation.Location..Borough.STATEN IS       -1.383e-01  1.817e-01  -0.761
## Hearing.Date                                -5.660e-03  4.053e-04 -13.965
## Total.Violation.Amount                      -4.314e-05  1.127e-05  -3.829
## Scheduled.Hearing.LocationBROOKLYN           5.358e-01  1.292e-01   4.148
## Scheduled.Hearing.LocationBY PHONE           8.686e-01  1.606e-01   5.409
## Scheduled.Hearing.LocationMANHATTAN          6.294e-01  1.106e-01   5.693
## Scheduled.Hearing.LocationONE-CLICK          4.496e+00  3.392e-01  13.254
## Scheduled.Hearing.LocationQUEENS             9.408e-01  1.371e-01   6.862
## Scheduled.Hearing.LocationSAU: BX            2.158e-01  1.009e+00   0.214
```

```
## Scheduled.Hearing.LocationSAU: MANH           9.153e-01  1.176e-01   7.784
## Scheduled.Hearing.LocationSTATEN IS           8.092e-01  2.232e-01   3.625
## Hearing.Time                                  1.443e+00  4.095e-01   3.524
## Compliance.StatusBoth Due                     1.559e+01  1.744e+02   0.089
## Compliance.StatusCompliance Due               1.507e+01  1.205e+02   0.125
## Compliance.StatusPenalty Due                  7.778e+00  1.001e+00   7.770
##                                              Pr(>|z|)
## (Intercept)                                  0.779066
## Violation.Date                                < 2e-16 ***
## Violation.Time                               0.302155
## Issuing.AgencyDEP - BUREAU OF ENV. COMPLIANC  < 2e-16 ***
## Issuing.AgencyDEP - IWC                      5.55e-10 ***
## Issuing.AgencyDEPT. OF BUILDINGS             4.95e-14 ***
## Issuing.AgencyDEPT OF TRANSPORTATION         1.11e-08 ***
## Issuing.AgencyDOH MENTAL HEALTH              2.54e-06 ***
## Issuing.AgencyDOITT                          1.91e-05 ***
## Issuing.AgencyDOS - ENFORCEMENT AGENTS       0.000152 ***
## Issuing.AgencyFIRE DEPARTMENT OF NYC          < 2e-16 ***
## Issuing.AgencyNYPD TRANSPORT INTELLIGENCE DI 5.37e-10 ***
## Issuing.AgencyPARKS DEPARTMENT               0.000477 ***
## Issuing.AgencyPCS - DOHMH                    2.39e-05 ***
## Issuing.AgencyPOLICE DEPARTMENT              0.000266 ***
## Issuing.AgencySANITATION OTHERS              0.001613 **
## Issuing.AgencySANITATION POLICE              0.003305 **
## Issuing.AgencySANITATION RECYCLING           0.000362 ***
## Issuing.AgencyVETERINARY-DOHMH               0.001964 **
## Violation.Location..Borough.BROOKLYN         0.165926
## Violation.Location..Borough.MANHATTAN        0.294482
## Violation.Location..Borough.QUEENS           0.653084
## Violation.Location..Borough.STATEN IS        0.446397
## Hearing.Date                                  < 2e-16 ***
## Total.Violation.Amount                       0.000129 ***
## Scheduled.Hearing.LocationBROOKLYN           3.36e-05 ***
## Scheduled.Hearing.LocationBY PHONE           6.33e-08 ***
## Scheduled.Hearing.LocationMANHATTAN          1.24e-08 ***
## Scheduled.Hearing.LocationONE-CLICK           < 2e-16 ***
## Scheduled.Hearing.LocationQUEENS             6.78e-12 ***
## Scheduled.Hearing.LocationSAU: BX            0.830675
## Scheduled.Hearing.LocationSAU: MANH          7.05e-15 ***
## Scheduled.Hearing.LocationSTATEN IS          0.000289 ***
## Hearing.Time                                 0.000425 ***
## Compliance.StatusBoth Due                    0.928783
## Compliance.StatusCompliance Due              0.900456
## Compliance.StatusPenalty Due                 7.85e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20642  on 16283  degrees of freedom
## Residual deviance: 13546  on 16247  degrees of freedom
##   (41 observations deleted due to missingness)
## AIC: 13620
##
```

```
## Number of Fisher Scoring iterations: 15
```

We select the predictor columns and put into new dataframe and then replace missing values with modes as knn doesn't remove missing values by default.

```
sampledf<-newdf[ , -which(names(newdf) %in% c("Respondent.Last.Name","Balance.Due","Violation.Date", "Vi
sampledf <- droplevels(sampledf)

Mode <- function (x, na.rm) {
    xtab <- table(x)
    xmode <- names(which(xtab == max(xtab)))
    if (length(xmode) > 1) xmode <- ">1 mode"
    return(xmode)
}
#impute missing values with mode...
for (var in 1:ncol(sampledf)) {
  sampledf[is.na(sampledf[,var]),var] <- Mode(sampledf[,var], na.rm = TRUE)
}


sampledf$Total.Violation.Amount <- as.numeric(sampledf$Total.Violation.Amount)
```

Model

Next we start with implementing the algorithms by first splitting the data into 10 folds for k-fold cross validation. Iteratively each fold is used as test set and remaining 9 are used as train. We find accuracy and execution time for each fold.

K-NN Classification

```
require(class)
require(knncat)
idx <- createFolds(sampledf$Hearing.Result,k = 10)
sapply(idx, length)
```

```
## Fold01 Fold02 Fold03 Fold04 Fold05 Fold06 Fold07 Fold08 Fold09 Fold10
##   1633   1633   1633   1633   1631   1633   1631   1633   1633   1632
```

```
accuracy <- vector()
timeknn <- vector()
for (i in 1:10) {
  start.time <- Sys.time()
  #knncat tests for each k value given and selects the model with best k value.
  model <- knncat(sampledf[ -idx[[i]] , ], sampledf[ idx[[i]], ], k=c(1,3,5,7,9,11,13,15,19),classcol =
  end.time <- Sys.time()
  timeknn[i]<-end.time - start.time
  pred <- predict(model,sampledf[ -idx[[i]] , ],sampledf[ idx[[i]], ],train.classcol = 3,newdata.classc
  cm <-table(pred,sampledf[ idx[[i]], ]$Hearing.Result)
  accuracy[i]<-sum(diag(cm))/sum(cm)
}
cm
```

```
##
```

```
## pred           DISMISSED IN VIOLATION
##   DISMISSED          314          208
##   IN VIOLATION       224          886
```

Linear Regression

We carry out similar steps for regression.

```
accuracylm <- vector()
timelm <- vector()
for (i in 1:10) {
  start.time <- Sys.time()
  model <-glm(Hearing.Result~.,family = binomial,data =sampledf[ -idx[[i]] , ] )
  end.time <- Sys.time()
  timelm[i]<-end.time - start.time
  pred <- predict(model,sampledf[ idx[[i]], ],type = 'response')
  cm <- table(sampledf[ idx[[i]], ]$Hearing.Result, pred>0.5)

  accuracylm[i]<-sum(diag(cm))/sum(cm)
}
cm
```

```
##
##                FALSE TRUE
##   DISMISSED      304  234
##   IN VIOLATION   192  902
```

Random Forests

We carry out similar steps for Random Forests.

```
library(randomForest)
accuracyrf <- vector()
timerf <- vector()
for (i in 1:10) {
  start.time <- Sys.time()
  rf <-randomForest(Hearing.Result~.,data =sampledf[ -idx[[i]] , ] ,importance = TRUE)
  end.time <- Sys.time()
  timerf[i]<-end.time - start.time
  pred <- predict(rf,sampledf[ idx[[i]], ])
  cm <- table(sampledf[ idx[[i]], ]$Hearing.Result, pred)
  accuracyrf[i]<-sum(diag(cm))/sum(cm)
}
cm
```

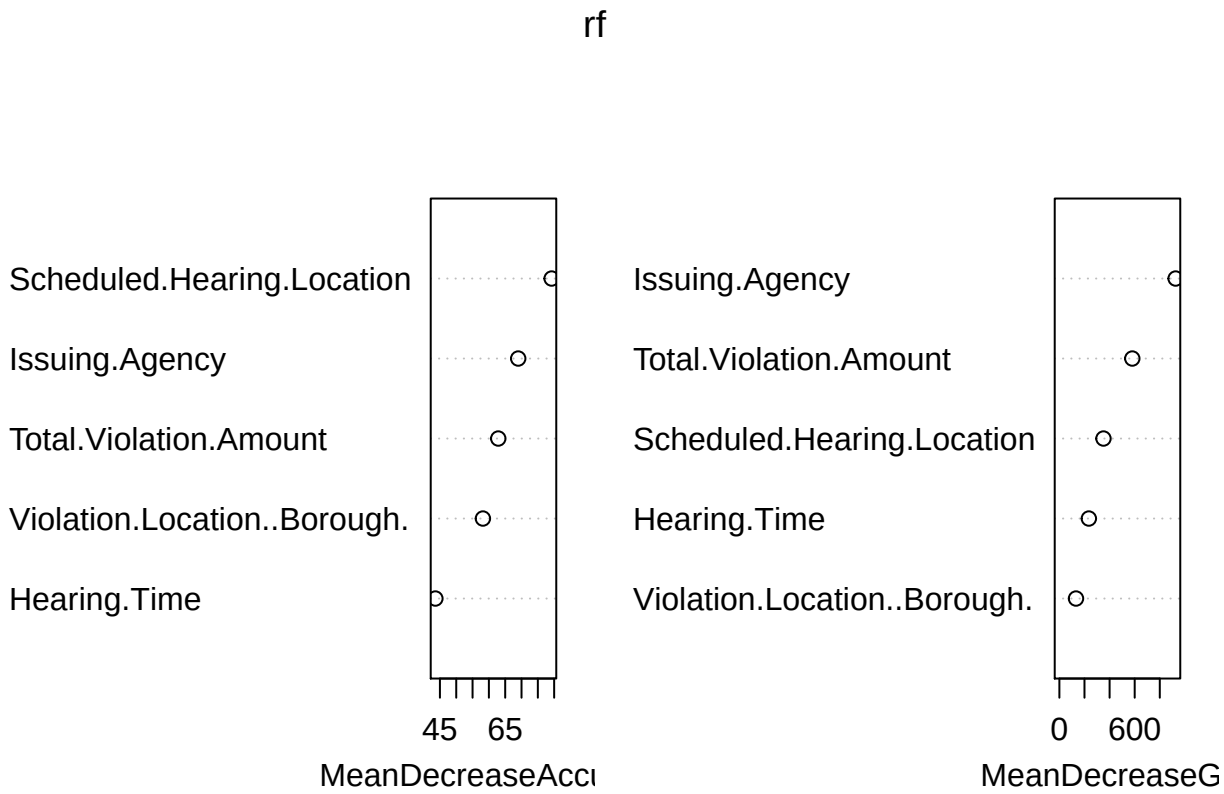
```
##                pred
##                 DISMISSED IN VIOLATION
##   DISMISSED          333          205
##   IN VIOLATION       185          909
```

Importance Plots.

The mean decrease in accuracy a variable causes is determined during the out of bag error calculation phase. The more the accuracy of the random forest decreases due to the exclusion (or permutation) of a single variable, the more important that variable is deemed, and therefore variables with a large mean decrease in accuracy are more important for classification of the data. The mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. Each time a particular variable is used to split a node, the Gini coefficient for the child nodes are calculated and compared to that of the original node. The Gini coefficient is a measure of homogeneity from 0 (homogeneous) to 1 (heterogeneous). The changes in Gini are summed for each variable and normalized at the end of the calculation. Variables that result in nodes with higher purity have a higher decrease in Gini coefficient.

A type 1 variable importance plot shows the mean decrease in accuracy, while a type 2 plot shows the mean decrease in Gini. We see both the plot agree that hearing location is the most important feature and Scheduled Hearing Location and Violation Amount are other important features.

```
trainIndex <- createDataPartition(sampledf$Hearing.Result, p = .8,
                                  list = FALSE,
                                  times = 1)
traindf <- sampledf[trainIndex,]
rf <- randomForest(Hearing.Result~., data=traindf,  importance = TRUE)
varImpPlot(rf)
```




```
importance(rf)
```

```
##                              DISMISSED IN VIOLATION MeanDecreaseAccuracy
## Issuing.Agency                51.973941     25.78017             68.98964
## Violation.Location..Borough.  17.567816     40.45671             58.16869
## Total.Violation.Amount        38.007096     20.53023             62.88185
```

```
## Scheduled.Hearing.Location   34.614426     53.45491             79.22372
## Hearing.Time                  6.513887     32.91663             43.61015
##                            MeanDecreaseGini
## Issuing.Agency                      926.0557
## Violation.Location..Borough.        132.3577
## Total.Violation.Amount              580.9702
## Scheduled.Hearing.Location          351.2797
## Hearing.Time                        234.7495
```

SVM Implementation

There are close to 170000 rows in the dataset. Also some description occur less than 10 times. We remove those descriptions. Then similar to the step we did for Issuing Agency we balance split on Code Description and then use that for our ngram analysis.

```
df2 <- df1[!(as.numeric(df1$Charge..1..Code.Description) %in% which(table(df1$Charge..1..Code.Descripti
df2 <- droplevels(df2)
pIndex <- createDataPartition(df2$Charge..1..Code.Description, p = .1,
                                  list = FALSE,
                                  times = 1)
dfa <- df2[pIndex,]
```

Let's create document term matrix for svm implementation

```
library(tm)
library(SnowballC)
library(tau)
library(RWeka)
descriptions <- data.frame((dfa$Charge..1..Code.Description))
descriptions <- as.character(descriptions$X.dfa.Charge..1..Code.Description.)
documents <- VCorpus(VectorSource(descriptions))
documents <- tm_map(documents,
                    content_transformer(function(x) iconv(x, to='UTF-8', sub='byte')),
                    mc.cores=1)
#remove symbols
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
documents <- tm_map(documents, toSpace, "/",mc.cores = 1)
documents <- tm_map(documents, toSpace, "\\|", mc.cores = 1)
removeURL <- function(x) gsub("http[[:alnum:]]*", "", x)
documents <- tm_map(documents, content_transformer(removeURL))
#convert to lowercase
documents <- tm_map(documents, content_transformer(tolower),mc.cores = 1)
# Remove numbers
documents <- tm_map(documents, removeNumbers,mc.cores = 1)

# Remove punctuations
documents <- tm_map(documents, removePunctuation,mc.cores = 1)

# Remove english common stopwords
documents <- tm_map(documents, removeWords, stopwords("english"))
# specify your stopwords as a character vector
#documents <- tm_map(documents, removeWords, c("pizza","tco","eat"))
# Eliminate extra white spaces
documents <- tm_map(documents, stripWhitespace, mc.cores = 1)
```

```
# Text stemming
documents <- tm_map(documents, stemDocument, lazy = T)
#create unigram matrix
dtm <-(DocumentTermMatrix(documents))
```

```
#Bigrams
BigramTokenizer <- function(x) {RWeka::NGramTokenizer(x, RWeka::Weka_control(min=2, max=2))}
options(mc.cores=1)
dtm2 <- DocumentTermMatrix(documents, control=list(tokenize=BigramTokenizer))
#Trigrams
#ThreegramTokenizer <- function(x) {RWeka::NGramTokenizer(x, RWeka::Weka_control(min=3, max=3))}
#options(mc.cores=1)
#dtm3 <- DocumentTermMatrix(documents, control=list(tokenize=ThreegramTokenizer))
NgramTokenize <- function(x) {RWeka::NGramTokenizer(x, RWeka::Weka_control(min=1, max=2))}
dtmn <- DocumentTermMatrix(documents, control=list(tokenize=NgramTokenize))
trainIndex <- createDataPartition(dfa$Hearing.Result, p = .75,
                                  list = FALSE,
                                  times = 1)
```

```
m1 <- as.matrix(dtm)#unigram
m2 <- as.matrix(dtm2)#bigram
mn <- as.matrix(dtmn)
df1 <- data.frame(m1,dfa$Hearing.Result) #unigramdf
df2 <- data.frame(m2,dfa$Hearing.Result) #bigramdf
dfn <- data.frame(mn,dfa$Hearing.Result) #bigramdf

names(df1)[names(df1) == 'dfa.Hearing.Result'] <- 'Result'
names(df2)[names(df2) == 'dfa.Hearing.Result'] <- 'Result'
names(dfn)[names(dfn) == 'dfa.Hearing.Result'] <- 'Result'
traindf1 <- df1[trainIndex,]
testdf1 <- df1[-trainIndex,]
traindf2 <- df2[trainIndex,]
testdf2 <- df2[-trainIndex,]

library("e1071")
library("kernlab")
svm_model1 <- svm(Result ~ ., data=traindf1) #unigramdf
pred1 <- predict(svm_model1,testdf1)
cm1 <- table(pred1,testdf1$Result)
cm1
```

```
##
## pred1          DISMISSED IN VIOLATION
##   DISMISSED          227          166
##   IN VIOLATION      1133         2614
```

```
sum(diag(cm1))/sum(cm1) #Accuracy unigram linear svm
```

```
## [1] 0.6862319
```

```
svm_model2 <- svm(Result ~ ., data=traindf2)#bigramdf
pred2 <- predict(svm_model2,testdf2)
cm2 <- table(pred2,testdf2$Result)
cm2
```

```
##
```

```
## pred2          DISMISSED IN VIOLATION
##   DISMISSED            0            0
##   IN VIOLATION      1360         2780
```

```
sum(diag(cm2))/sum(cm2) #Accuracy bigram linear sum
```

```
## [1] 0.6714976
```

The accuracy for unigram is clearly low. So let's change the kernel from linear to rbf

```
#rbf kernel
svp <- ksvm(Result~., data= traindf1,type="C-svc",kernel='rbf',kpar=list(sigma=1),C=1)
pred1 <- predict(svp,testdf1)
cm1 <- table(pred1,testdf1$Result)
cm1
```

```
##
## pred1          DISMISSED IN VIOLATION
##   DISMISSED          798          591
##   IN VIOLATION       562         2189
```

```
sum(diag(cm1))/sum(cm1)#Accuracy unigram rbf kernel
```

```
## [1] 0.7214976
```

```
svp <- ksvm(Result~., data= traindf2,type="C-svc",kernel='rbf',kpar=list(sigma=1),C=1)
pred2 <- predict(svp,testdf2)
cm2 <- table(pred2,testdf2$Result)
cm2
```

```
##
## pred2          DISMISSED IN VIOLATION
##   DISMISSED          779          574
##   IN VIOLATION       581         2206
```

```
sum(diag(cm2))/sum(cm2) #Accuracy Bigram rbf
```

```
## [1] 0.7210145
```

We observe that the accuracy for unigram and bigram improves by about 0.05 when we change the kernel.

Visualization

First Visualitation is for the Accuracies of 3 of the algorithms i.e Logistic Regression, Random Forest, K-NN Classification.It is a grouped bar plot and we observe the accuracy of Random Forest is slightly better than the other two.
##Accuracy Comparison

```
library(reshape2)
folds <- 1:10
mean(accuracyrf)#random forest accuracy averaged across folds
```

```
## [1] 0.7475046
```

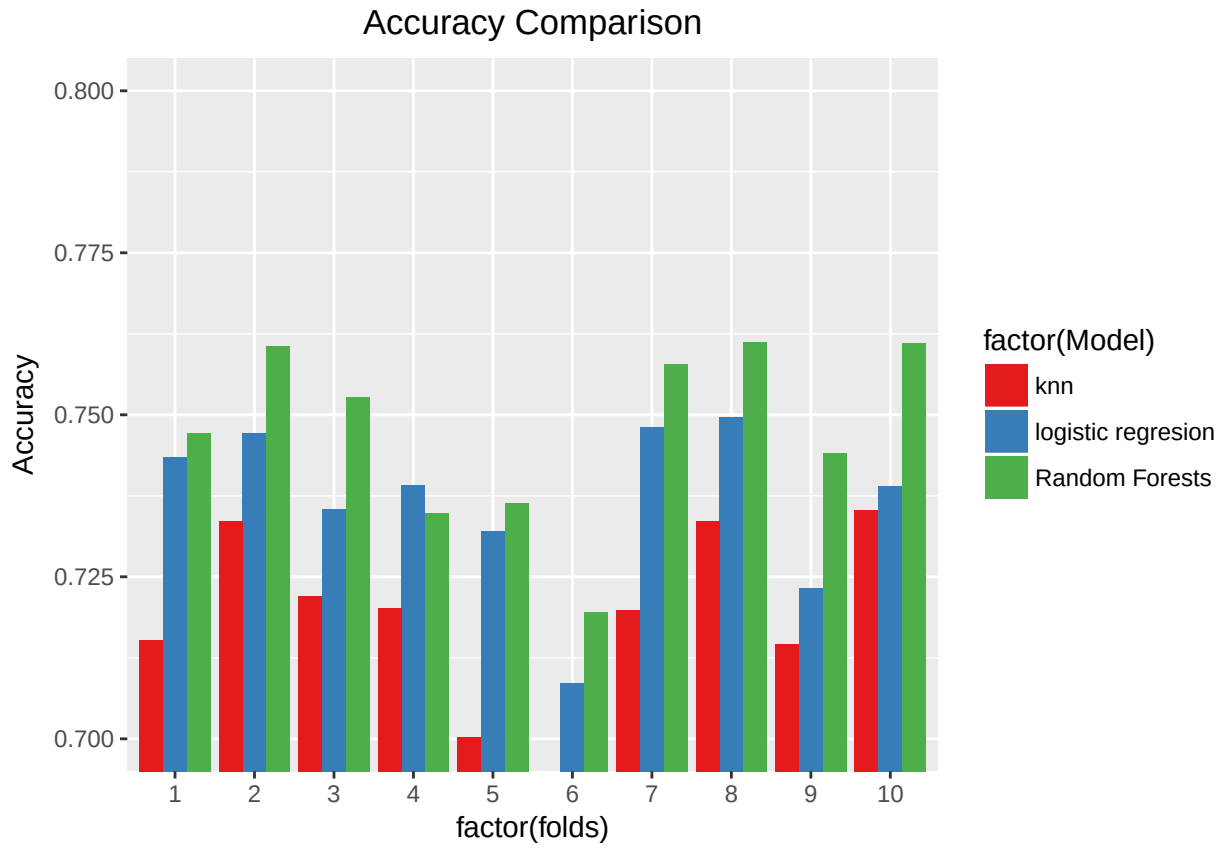
```
mean(accuracylm)#logistic regression acccuracy averaged across folds
```

```
## [1] 0.7365401
```

```
mean(accuracy)#knn accuracy averaged across folds...
```

```
## [1] 0.7184676
```

```
comp<-data.frame(accuracy,accuracylm,accuracyrf,folds)
colnames(comp)<- c("knn","logistic regresion","Random Forests", "folds")
compfull <- melt(comp,id = c("folds"))
colnames(compfull) <- c("folds","Model","Accuracy")
compfull$folds <- as.factor(compfull$folds)
ggplot(compfull, aes(factor(folds), Accuracy, fill = factor(Model)))+
  labs(title="Accuracy Comparison") +
  coord_cartesian(ylim=c(0.7,0.8)) +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_bar(stat="identity", position = "dodge") +
  scale_fill_brewer(palette = "Set1")
```

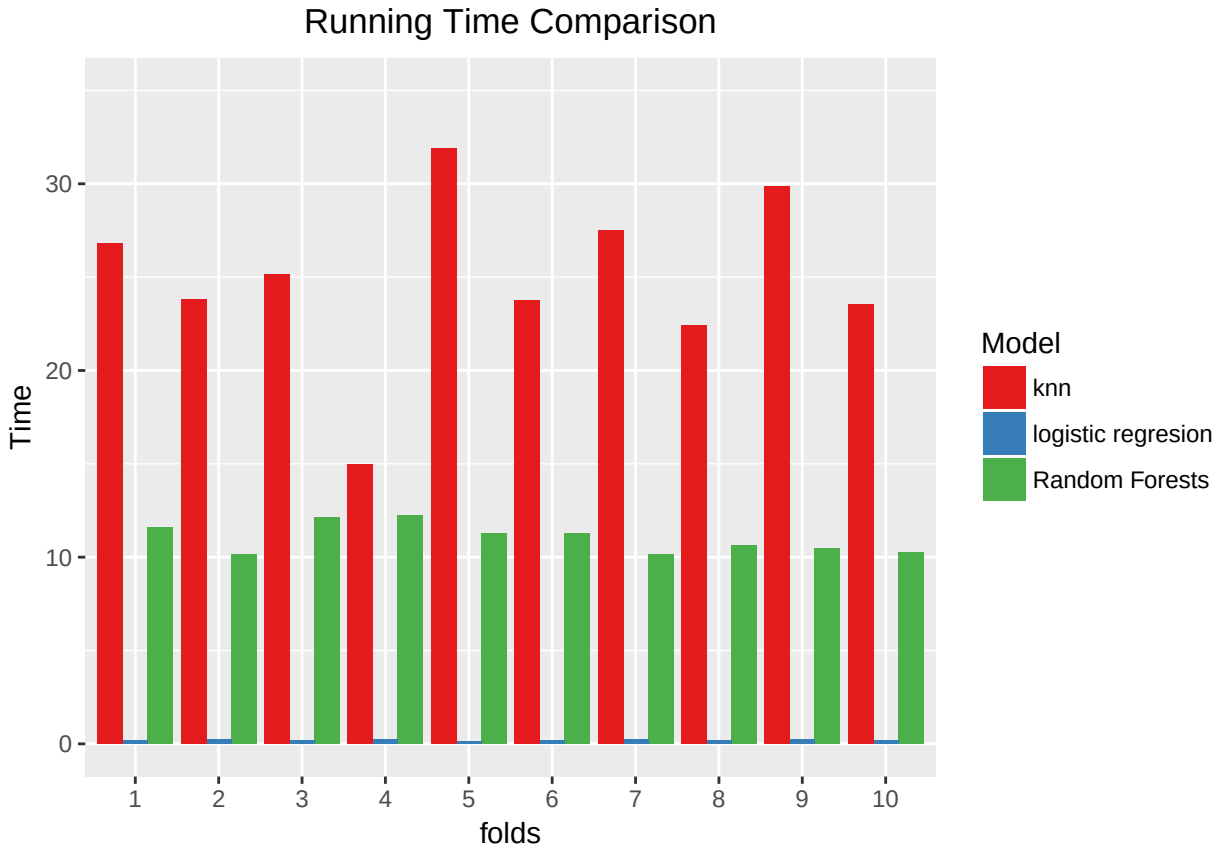


Execution Time Comparison

Secondly we identify the execution time of each algorithm and plot them on the grouped bar plot. We find the logistic regression performs the best followed by random forest and knn.

```
folds <- 1:10
comp<-data.frame(timeknn,timelm,timerf,folds)
colnames(comp)<- c("knn","logistic regresion","Random Forests", "folds")
compfull <- melt(comp,id = c("folds"))
colnames(compfull) <- c("folds","Model","Time")
```

```
compfull$folds <- as.factor(compfull$folds)
ggplot(compfull, aes(folds, Time, fill = Model))+labs(title="Running Time Comparison") +ylim(0,35)+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_bar(stat="identity", position = "dodge") +
  scale_fill_brewer(palette = "Set1")
```



Issuing Agencies Comparison

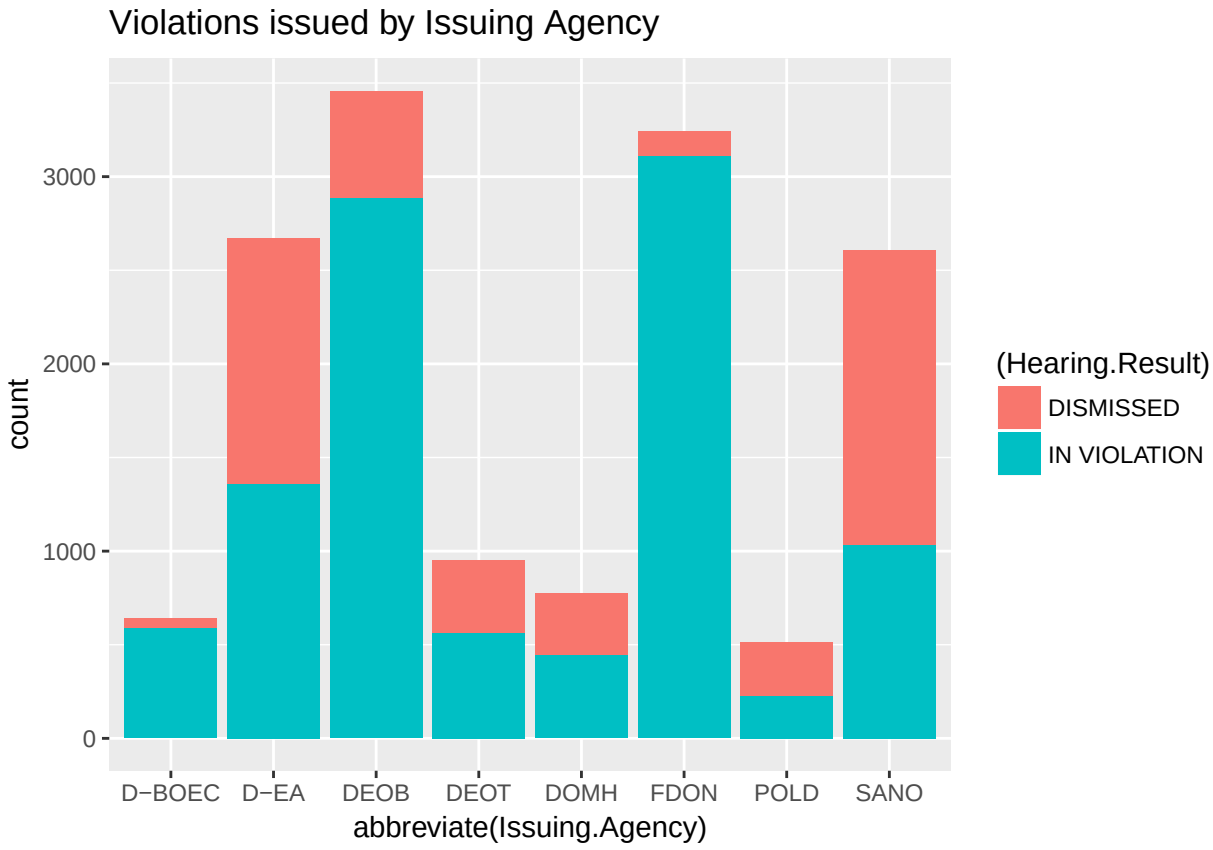
Last we plot stacked bar plot of top Issuing Agencies with The violation count on the y axis and the hearing result as the differentiatior. This graph helps us identify the issuing agencies who are more consistent in their Violation tickets and the we can also find out which agency's violations are dismissed more frequently which signifies they have higher error rate.

```
dfbar <- sampledf[!(as.numeric(sampledf$Issuing.Agency) %in% which(table(sampledf$Issuing.Agency)<500))
dfbar <- droplevels(dfbar)
issuingagency<-data.frame(unique(dfbar$Issuing.Agency),unique(abbreviate(dfbar$Issuing.Agency)))
colnames(issuingagency)<-c("Name","Abbreviation")
issuingagency
```

```
##                            Name Abbreviation
## 1             DEPT. OF BUILDINGS         DEOB
## 2       DOS - ENFORCEMENT AGENTS         D-EA
## 3         FIRE DEPARTMENT OF NYC         FDON
## 4              DOH MENTAL HEALTH         DOMH
## 5         DEPT OF TRANSPORTATION         DEOT
## 6              SANITATION OTHERS         SANO
```

```
## 7 DEP - BUREAU OF ENV. COMPLIANC       D-BOEC
## 8              POLICE DEPARTMENT         POLD
```

```
qplot(abbreviate(Issuing.Agency), data=dfbar,geom="bar", fill=(Hearing.Result))+labs(title = "Violations
```



Contribution of n-grams to the outcome

The y axis shows the relevant terms that are important in determining the Hearing Results...

```
library(dplyr)
library(tidytext)

classes <- data.frame(rownames(dfa),dfa$Hearing.Result)
colnames(classes) <- c("document","result")
dtmn1 <- tidy(dtmn)
mergedtmn <- merge(dtmn1,classes)
dismissed<- dfn[dfn$Result %in% c("DISMISSED"),]
violation <- dfn[dfn$Result %in% c("IN VIOLATION"),]
dtmndis <-as.matrix( dtmn[rownames(dismissed),])
dtmnvio <- as.matrix(dtmn[rownames(violation),])
freqd <- sort(colSums(as.matrix(dtmndis)), decreasing=TRUE)
freqv <- sort(colSums(as.matrix(dtmnvio)), decreasing=TRUE)
head(freqd, 10) #10 most frequent features for Dismissed
```

```
##      dirti     failur       area dirti area    sidewalk     street
##       1168        927        700        677        661        636
```

```
##       unit     improp  receptacl     offens
##        625        616        577        520
```

```
head(freqv, 10) #10 most frequent features for Violation
```

```
##       failur       permit         fail         fire      prevent
##         1627         1476         1379         1227         1154
##      protect         test        dirti         fire      protect
##         1005          979          912          891          824
```

```
mergedtmn %>%
  count(result, term, wt = count) %>%
  ungroup() %>%
  filter(n >= 40) %>%
  mutate(n = ifelse(result == "IN VIOLATION", -n, n)) %>%
  mutate(term = reorder(term, n)) %>%
  ggplot(aes(term, n, fill = result)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylab("Contribution to Hearing Result")
```