PART ONE CODE

LDA

PATH="C:/Users/dsand/OneDrive/Documents/Graduate School/ADM"

fatalities.df<- read.csv(file.path(PATH,"./fatalities\_data\_accident.csv"), header = TRUE)

fatalities.df.clean<-fatalities.df[c(-1,-2,-3,-4,-7,-10,-11,-12,-13,-14,-15,-18,-20,-21,-24,-25,-26,-27,-28,-29,-32,-34,-35,-38,-39,-41,-42,-52,-53)]

str(fatalities.df.clean)

dim(fatalities.df.clean)

fatalities.df.clean$number\_of\_fatalities<-factor(fatalities.df.clean$number\_of\_fatalities)

# Partition Data into training and validation sets using a 70/30 split

set.seed(123)

trainIndex <- createDataPartition(fatalities.df.clean$number\_of\_fatalities, p = .7,

list = FALSE,

times = 1)

fatal.train <-fatalities.df.clean[ trainIndex,]

fatal.valid <- fatalities.df.clean[-trainIndex,]

plot(fatal.train$number\_of\_fatalities, data = fatal.train, pch=ifelse(fatal.train$number\_of\_fatalities==1,1,3))

legend("topright", c("Acceptor", "non-Acceptor"), pch=c(1,3))

segments(x0=85, x1=45,y0=10,y1=25, col = "red")

lda.model <- linDA(fatal.train[,1:22],fatal.train$number\_of\_fatalities)

lda.model$functions

lda.model$scores

lda.model$classification

# Making Predictions using the MASS function

# Training model

lda.model.MASS <- lda(number\_of\_fatalities~.,fatal.train)

lda.model.MASS

#Validation Model

lda.preds <- predict(lda.model.MASS,fatal.valid)

lda.preds$class

lda.preds$posterior

#CONFUSION MATRIX

#confusionMatrix using the validation model

# Accuracy = .95

confusionMatrix(lda.preds$class, fatal.valid$number\_of\_fatalities, positive="1")

CONFUSION MATRIX

library(caret)

library(rpart.plot)

library(adabag)

library(uplift)

library(pls)

library(ggplot2)

library(ggrepel)

library(randomForest)

PATH="C:/Users/dsand/OneDrive/Documents/Graduate School/ADM"

fatalities.df<- read.csv(file.path(PATH,"./fatalities\_data\_accident.csv"), header = TRUE)

str(fatalities.df)

## Step 1: Develop an understanding of the data mining project

## Step 2: Obtain the dataset to be used in the analysis

## Step 3: Explore, clean, and preprocess the data

#### Correlation plot

#### Missing Values

#summary of inital data set

# there is only one data point that has a fatality of 6, this will skew our result, so to minimize variance i will delete this entry

####deleting variables that are unnecessary, or already included in other variables

#v1: deleting entry ID, not relevent

#v2: deleting state number, road conditions and long/lat will tell us more

#v3: deleting case #, tells us nothing about nature of incident

#v4: deleting, same as #vehicles in transport

#v7: deleting, same as # of non motorists in crash

#v10: deleding, tells us same as number of persons in motor vehicle v9

#v11: deleting, long/lat will tell us this

#v12: corresponds to geographical codes, irrelevent with long/lat

#13: day of crash, day of month does not tell us, weather and long/lat will tell more

#15: year is only 2015

#v14: month of crash: long/lat and conditions will tell us more

#v18: min of crash tells us nothing

#v20: only data after 2015, ownership tells us this

#v21: functional system: only 2015- on data, ownership tells us this

#v24: traffic id, route tells us more

#v25: cross raods tell us nothing, ownership tells us more

#v26: route tells us if its on an interstate or not, irrelavent to bus ?

#v29: ownership tells us this, this only tells who is legally patrolling area

#v32: specific location tells us more and is derived from this data

#v38: first condition

#v39: second condition, atmos 40 is derived from these two, tells us same info

#v42: not full data set, understand this based on route signing

#v52: we have hour of day, conditions, and region, tells us more than var. with many levels

#v53: gid is not associated with this

str(fatalities.df)

dim(fatalities.df)

fatalities.df.clean<-fatalities.df[-61,c(-1,-2,-3,-4,-7,-10,-11,-12,-13,-14,-15,-18,-20,-21,-24,-25,-26,-27,-28,-29,-32,-38,-39,-42,-52,-53)]

str(fatalities.df.clean)

dim(fatalities.df.clean)

#View(fatalities.df.clean)

#dummying out the time of notification and time of arrival

W\_in\_10min\_notif<-ifelse(fatalities.df.clean$minute\_of\_notification==1:10, 5,0)

W\_in\_20min\_notif<-ifelse(fatalities.df.clean$minute\_of\_notification==11:20, 15,0)

W\_in\_30min\_notif<-ifelse(fatalities.df.clean$minute\_of\_notification==21:30, 25,0)

W\_in\_40min\_notif<-ifelse(fatalities.df.clean$minute\_of\_notification==31:40, 35,0)

W\_in\_50min\_notif<-ifelse(fatalities.df.clean$minute\_of\_notification==41:50, 45,0)

w\_in\_60min\_notif<-ifelse(fatalities.df.clean$minute\_of\_notification==51:60, 55,0)

W\_in\_10min\_arr<-ifelse(fatalities.df.clean$minute\_of\_arrival\_at\_scene==1:10, 5,0)

W\_in\_20min\_arr<-ifelse(fatalities.df.clean$minute\_of\_arrival\_at\_scene==11:20, 15,0)

W\_in\_30min\_arr<-ifelse(fatalities.df.clean$minute\_of\_arrival\_at\_scene==21:30, 25,0)

W\_in\_40min\_arr<-ifelse(fatalities.df.clean$minute\_of\_arrival\_at\_scene==31:40, 35,0)

W\_in\_50min\_arr<-ifelse(fatalities.df.clean$minute\_of\_arrival\_at\_scene==41:50, 45,0)

w\_in\_60min\_arr<-ifelse(fatalities.df.clean$minute\_of\_arrival\_at\_scene==51:60, 55,0)

#setting dummys to factors

W\_in\_10min\_notif<-as.factor(W\_in\_10min\_notif)

W\_in\_20min\_notif<-as.factor(W\_in\_20min\_notif)

W\_in\_30min\_notif<-as.factor(W\_in\_30min\_notif)

W\_in\_40min\_notif<-as.factor(W\_in\_40min\_notif)

W\_in\_50min\_notif<-as.factor(W\_in\_50min\_notif)

w\_in\_60min\_notif<-as.factor(w\_in\_60min\_notif)

W\_in\_10min\_arr<-as.factor(W\_in\_10min\_arr)

W\_in\_20min\_arr<-as.factor(W\_in\_20min\_arr)

W\_in\_30min\_arr<-as.factor(W\_in\_30min\_arr)

W\_in\_40min\_arr<-as.factor(W\_in\_40min\_arr)

W\_in\_50min\_arr<-as.factor(W\_in\_50min\_arr)

w\_in\_60min\_arr<-as.factor(w\_in\_60min\_arr)

#setting variables to factors

drunk\_drivers<-as.factor(fatalities.df.clean$number\_of\_drunk\_drivers)

fatalities<- as.factor(fatalities.df.clean$number\_of\_fatalities)

crash\_level\_3<- as.factor(fatalities.df.clean$related\_factors\_crash\_level\_3)

crash\_level\_2<-as.factor(fatalities.df.clean$related\_factors\_crash\_level\_2)

crash\_level\_1<-as.factor(fatalities.df.clean$related\_factors\_crash\_level\_1)

minute\_of\_arrival<-as.factor(fatalities.df.clean$minute\_of\_arrival\_at\_scene)

hour\_of\_arrival<-as.factor(fatalities.df.clean$hour\_of\_arrival\_at\_scene)

minute\_notif<-as.factor(fatalities.df.clean$minute\_of\_notification)

hour\_notif<-as.factor(fatalities.df.clean$hour\_of\_notification)

atmospheric<-as.factor(fatalities.df.clean$atmospheric\_conditions)

light<-as.factor(fatalities.df.clean$light\_condition)

trafficway<-as.factor(fatalities.df.clean$relation\_to\_trafficway)

junction\_specific\_location<-as.factor(fatalities.df.clean$relation\_to\_trafficway)

mancollision<-as.factor(fatalities.df.clean$manner\_of\_collision)

firstevent<-as.factor(fatalities.df.clean$first\_harmful\_event)

route<-as.factor(fatalities.df.clean$route\_signing)

owner<-as.factor(fatalities.df.clean$ownership)

highway<-as.factor(fatalities.df.clean$national\_highway\_system)

hourcrash<-as.factor(fatalities.df.clean$hour\_of\_crash)

day<-as.factor(fatalities.df.clean$day\_of\_week)

persons\_in\_motor\_vehicles\_in\_transport\_mvit<-as.factor(fatalities.df.clean$number\_of\_persons\_in\_motor\_vehicles\_in\_transport\_mvit)

persons\_not\_in\_motor\_vehicles\_in\_transport\_mvit<-as.factor(fatalities.df.clean$number\_of\_persons\_not\_in\_motor\_vehicles\_in\_transport\_mvit)

parked\_working\_vehicles<-as.factor(fatalities.df.clean$number\_of\_parked\_working\_vehicles)

motor\_vehicles\_in\_transport\_mvit<-as.factor(fatalities.df.clean$number\_of\_motor\_vehicles\_in\_transport\_mvit)

#creating data frame with all varibles as factors

accident.df.factors<-data.frame(drunk\_drivers,fatalities,crash\_level\_3,crash\_level\_2,crash\_level\_1,hour\_of\_arrival,

hour\_notif,atmospheric,light,trafficway,junction\_specific\_location,mancollision,firstevent,

route,owner,highway,hourcrash,day,persons\_in\_motor\_vehicles\_in\_transport\_mvit,persons\_not\_in\_motor\_vehicles\_in\_transport\_mvit,

parked\_working\_vehicles,motor\_vehicles\_in\_transport\_mvit,fatalities.df.clean$school\_bus\_related,

fatalities.df.clean$work\_zone,fatalities.df.clean$type\_of\_intersection,W\_in\_10min\_notif,

W\_in\_20min\_notif,W\_in\_30min\_notif,W\_in\_40min\_notif,W\_in\_50min\_notif,w\_in\_60min\_notif,

W\_in\_10min\_arr,W\_in\_20min\_arr,W\_in\_30min\_arr,W\_in\_40min\_arr,W\_in\_50min\_arr, w\_in\_60min\_arr)

View(accident.df.factors)

str(accident.df.factors)

#Bringing target variable to front of data...

fatalities.df.clean.factors<-accident.df.factors[,c(2,1,3:37)]

#turning school bus related from categorical to binanry

fatalities.df.clean.factors$school\_bus\_related<-ifelse(fatalities.df.clean.factors$school\_bus\_related=="Yes",1,0)

##Step 4: Reduce the data dimension

#done in preprocessing step 3

## Step 5: Determine the data mining task

#predicting fatalities using random forests

## Step 6: Partition the data (for supervised tasks)

#random forests use a 80/20 split

trainIndex <- createDataPartition(fatalities.df.clean.factors$fatalities, p = .8,list = FALSE,times = 1)

head(trainIndex)

fatalities\_train.set <- fatalities.df.clean.factors[trainIndex,]

fatalities\_validate.set <- fatalities.df.clean.factors[-trainIndex,]

## Step 7: Choose the data mining techniques to be used

###use random forest algorithm

## Step 8: Use algorithms to perform the task

RF.model.fatalities <-randomForest(fatalities ~.,data=fatalities\_train.set, mtry=3, ntree=1000,na.action = na.omit, importance=TRUE) #default to try three predictors at a time and create 500 trees.

print(RF.model.fatalities)

importance(RF.model.fatalities)

varImpPlot(RF.model.fatalities)

## Step 9: Interpret the results

#MeanDecreaseAccuracy MeanDecreaseGini

#drunk\_drivers 6.05013152 9.9485771

#crash\_level\_3 0.13122604 0.5079941

#crash\_level\_2 3.30553882 1.7288890

#crash\_level\_1 4.29392793 9.1821329

#hour\_of\_arrival 25.41930002 31.0822661

#hour\_notif 28.54930535 29.5162247

#atmospheric 1.96885102 17.1196746

#light 9.93313410 11.4548182

#trafficway 7.23055880 6.3138547

#junction\_specific\_location 6.75607202 6.4106792

#mancollision 7.79480559 16.7005497

#firstevent 12.75764184 18.0138030

#route 2.18818301 19.9806124

#owner -0.32708172 14.3473608

#highway 1.10866811 5.5442194

#hourcrash 29.26547788 31.0333993

#day -0.16945143 26.7451815

#persons\_in\_motor\_vehicles\_in\_transport\_mvit 27.61561468 48.7640994

#persons\_not\_in\_motor\_vehicles\_in\_transport\_mvit 9.02962793 7.4554825

#parked\_working\_vehicles 3.81780663 3.4772889

#motor\_vehicles\_in\_transport\_mvit 13.99140387 11.9373198

#fatalities.df.clean.school\_bus\_related -1.54048987 0.4604764

#fatalities.df.clean.work\_zone 3.61067808 2.4563134

#fatalities.df.clean.type\_of\_intersection 4.87377428 7.3902709

#W\_in\_10min\_notif 0.07489583 0.7070676

#W\_in\_20min\_notif 1.13851550 2.1877834

#W\_in\_30min\_notif 0.30926507 1.4755855

#W\_in\_40min\_notif -3.77525583 1.5956124

#W\_in\_50min\_notif -0.68785807 2.2787055

#w\_in\_60min\_notif -0.53885309 1.1887538

#W\_in\_10min\_arr 0.36315076 1.5957993

#W\_in\_20min\_arr -2.25081612 1.8239257

#W\_in\_30min\_arr 0.44623685 2.0982875

#W\_in\_40min\_arr -3.11744967 1.7438301

#W\_in\_50min\_arr -0.92716204 0.9009859

#w\_in\_60min\_arr 0.98079267 0.9613588

## Step 10: Deploy the model

fatalities\_actual<-fatalities\_validate.set$fatalities

fatalities\_predicted <-predict(RF.model.fatalities, fatalities\_validate.set, type="class")

CM.fatalities<-confusionMatrix(fatalities\_predicted, fatalities\_actual, positive="1")

print(CM.fatalities)