PART TWO CODE

RANDOM FOREST

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"

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

#View(injury.df)

str(injury.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

#Deleting variables that are unneessary

injury.df.clean<-injury.df[,c(-1,-2,-7,-8,-9,-16,-31,-38,-42,-43,-47,-48,-49,-52,-55,-58,-59,-60,-61,-62,-66,-72,-73,-74,-75,-76,-77,-86,-87,-90,-92,-94,-100)]

str(injury.df.clean)

names(injury.df)

# I was not able to positivly impact the confusion matrix with my dummy variables for time

#changing integers into factors

injury.clean.factors<-data.frame(lapply(injury.df.clean,as.factor), stringsAsFactors = FALSE)

str(injury.clean.factors)

#View(injury.clean.factors)

#View(fatalities.clean.factors)

#Bringing target varible to front

injury.clean.factors.front<-injury.clean.factors[,c(13,1:12,14:67)]

#View(injury.clean.factors.front)

str(injury.clean.factors.front)

##Step 4: Reduce the data dimension

## Step 5: Determine the data mining task

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

trainIndex <- createDataPartition(injury.clean.factors.front$injury\_severity, p = .8,list = FALSE,times = 1)

head(trainIndex)

injury\_train.set <- injury.clean.factors.front[trainIndex,]

injury\_validate.set <- injury.clean.factors.front[-trainIndex,]

str(injury.clean.factors.front)

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

## Step 8: Use algorithms to perform the task

RF.model.injuries <-randomForest(injury\_severity ~.,data=injury\_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.injuries)

importance(RF.model.injuries)

varImpPlot(RF.model.injuries)

## Step 9: Interpret the results

#9 MeanDecreaseAccuracy MeanDecreaseGini

#state\_number 10.7704697 46.3203840 1.978391e+02

#vehicle\_number 9.1663066 23.9505443 7.412545e+01

#person\_number 6.3498081 19.9539888 7.231569e+01

#number\_of\_motor\_vehicle\_striking\_non\_motorist 5.2253492 11.9636749 1.929024e+01

#vehicle\_trailing 7.4268545 20.1721013 5.148943e+01

#special\_use 9.1508921 23.7240075 1.579082e+01

#emergency\_motor\_vehicle\_use 9.7544591 9.8081818 2.738062e+00

#rollover 13.6967560 28.7287726 7.640255e+01

#initial\_contact\_point 10.5281923 30.3946316 1.373391e+02

#fire\_occurrence 8.0744760 16.5644729 3.169729e+01

#sex 20.9599722 16.9331221 4.378155e+01

#person\_type 5.7498807 23.3341331 5.217924e+01

#seating\_position 9.4695822 23.3634770 8.519181e+01

#restraint\_system\_helmet\_use 20.3466758 34.7963265 1.912600e+02

#indication\_of\_misuse\_of\_restraint\_system\_helmet 5.0687248 10.3791925 1.625418e+01

#air\_bag\_deployed 22.6584731 40.6051587 1.952304e+02

#ejection 13.4213738 30.6322364 1.444971e+02

#ejection\_path 8.1960470 23.6526910 4.602737e+01

#extrication 10.9891801 38.0524105 1.097828e+02

#police\_reported\_alcohol\_involvement 7.8308631 25.2981217 6.613640e+01

#method\_of\_alcohol\_determination\_by\_police 6.2067393 17.3586412 3.396464e+01

#alcohol\_test\_status1 10.9653226 25.5397691 1.352486e+02

#alcohol\_test\_status2 12.5994673 29.8381368 1.923473e+02

#police\_reported\_drug\_involvement 7.6030497 26.6993105 5.537864e+01

#method\_of\_drug\_determination\_by\_police 5.4506150 13.1305515 2.241378e+01

#drug\_test\_status 10.3206197 21.6118746 1.111983e+02

#drug\_test\_type1 11.3033418 22.6042324 1.118809e+02

#drug\_test\_type2 4.9636743 10.2134268 1.569742e+01

#drug\_test\_type3 3.3010875 7.8187406 5.488957e+00

#drug\_test\_type5 4.7565613 9.2952194 1.795961e+01

#drug\_test\_type6 2.3756539 4.8908513 6.696133e+00

#transported\_to\_first\_treatment\_facility 15.8522370 63.6800628 5.008745e+02

#related\_factors\_person\_level1 7.2549741 5.6895490 2.509848e+00

#related\_factors\_person\_level2 0.0000000 -1.4156214 6.224145e-02

#related\_factors\_person\_level3 0.0000000 0.6160918 6.497949e-02

#hispanic\_origin 23.3772199 63.7509880 1.099441e+03

#non\_motorist\_location\_at\_time\_of\_crash 4.2039756 11.3276197 1.510832e+01

#driver\_maneuvered\_to\_avoid 15.2904738 31.6905601 7.169599e+01

#number\_of\_motor\_vehicles\_in\_transport\_mvit 8.6377934 26.9950099 6.908838e+01

#number\_of\_parked\_working\_vehicles 1.9929760 8.0560826 9.326848e+00

#number\_of\_persons\_not\_in\_motor\_vehicles\_in\_transport\_mvit 18.0008703 34.2373977 1.149821e+02

#number\_of\_persons\_in\_motor\_vehicles\_in\_transport\_mvit 13.4620768 40.8572542 1.731052e+02

#year\_of\_crash 0.0000000 0.0000000 0.000000e+00

#day\_of\_week 7.3897591 30.3809560 1.093434e+02

#hour\_of\_crash 4.9865607 39.4393005 1.572769e+02

#national\_highway\_system 3.7270203 16.6049395 2.587741e+01

#land\_use 8.1412076 22.8659500 3.629198e+01

#functional\_system 6.9067056 28.8863207 8.582655e+01

#ownership 3.6978381 25.9459219 6.071234e+01

#route\_signing 6.8527435 28.4987928 8.558574e+01

#first\_harmful\_event 14.4145805 36.8780777 1.462357e+02

#manner\_of\_collision 15.0249921 28.9032754 8.991237e+01

#relation\_to\_junction\_within\_interchange\_area 4.8487305 10.2558768 1.015024e+01

#relation\_to\_junction\_specific\_location 6.9277703 25.3421195 5.653158e+01

#type\_of\_intersection 2.3954808 21.8083088 3.700372e+01

#work\_zone 5.3456722 9.2454348 7.936609e+00

#relation\_to\_trafficway 5.6514137 26.3689582 8.035977e+01

#light\_condition 1.2000337 21.3524712 5.141156e+01

#atmospheric\_conditions 6.1724500 21.4102638 5.677070e+01

#school\_bus\_related 1.7244504 17.2653270 5.409295e+00

#hour\_of\_notification 5.5549996 39.7786398 1.565245e+02

#hour\_of\_arrival\_at\_scene 7.5318782 41.8916059 1.618491e+02

#related\_factors\_crash\_level\_1 1.1256529 18.8614775 2.889329e+01

#related\_factors\_crash\_level\_2 -0.0303936 3.9472754 6.018298e+00

#related\_factors\_crash\_level\_3 -1.4156299 1.8548457 2.239066e+00

#number\_of\_fatalities 8.9055784 27.1014611 3.959626e+01

#number\_of\_drunk\_drivers 10.0257831 18.4419208 2.884873e+01

# Step 10: Deploy the model

injury\_actual<-injury\_validate.set$injury\_severity

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

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

print(CM.fatalities)

NAÏVE BAYES

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

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

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

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

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

##Step 4: Reduce the data dimension

injury.df.clean<-injury.df[,c(-1,-2,-3,-7,-8,-9,-10,-11,-12,-18,-21,-28,-29,-30,-31,-33,-34,-35,-36,-37,-38,-39,-40,-42,-43,-44,-45,-46,-47,-48,-50,-53,-59,-60,-61,-63,-64,-65,-66,-69,-71,-72,-73,-74,-75,-76,-77,-78,-81,-82,-83,-84,-87,-88,-89,-90,-91,-93,-95,-96,-97,-100)]

dim(injury.df.clean)

injury.df.clean$vehicle\_trailing<-ifelse(injury.df.clean$vehicle\_trailing=="Yes, One Trailing Unit",1,0)

injury.df.clean$rollover<-ifelse(injury.df.clean$rollover=="No Rollover",0,1)

injury.df.clean$fire\_occurrence<-ifelse(injury.df.clean$fire\_occurrence=="Yes",1,0)

injury.df.clean$sex<-ifelse(injury.df.clean$sex=="Male",1,0)

injury.df.clean$indication\_of\_misuse\_of\_restraint\_system\_helmet<-ifelse(injury.df.clean$indication\_of\_misuse\_of\_restraint\_system\_helmet=="Yes",1,0)

injury.df.clean$extrication<-ifelse(injury.df.clean$extrication=="Extricated",1,0)

injury.df.clean$police\_reported\_alcohol\_involvement<-ifelse(injury.df.clean$police\_reported\_alcohol\_involvement=="Yes (Alcohol Involved)",1,0)

injury.df.clean$police\_reported\_drug\_involvement<-ifelse(injury.df.clean$police\_reported\_drug\_involvement=="Yes (Drugs Involved)",1,0)

injury.df.clean$relation\_to\_junction\_within\_interchange\_area<-ifelse(injury.df.clean$relation\_to\_junction\_within\_interchange\_area=="Yes",1,0)

injury.df.clean$injury\_severity<-as.factor(injury.df.clean$injury\_severity)

names(injury.df.clean)

#Rearranging the columns so that our target variable (severity of injury) is first

person.df<- injury.df.clean[,c(9,1:8,10:38)]

## Step 5: Determine the data mining task

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

# Partition data into training and validation sets using a 70-30 split

set.seed(123)

trainIndex <- createDataPartition(person.df$injury\_severity, p = .7,

list = FALSE,

times = 1)

person.train <-person.df[ trainIndex,]

person.valid <- person.df[-trainIndex,]

persons\_train\_predictors <- as.data.frame(scale(person.train[,-1]))

persons\_train\_target <- person.train[,1]

persons\_valid\_predictors <- as.data.frame(scale(person.valid[,-1]))

persons\_valid\_target <- person.valid[,1]

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

person\_nb<- naiveBayes(injury\_severity ~ ., data = person.train)

preds\_nb <- predict(person\_nb, person.valid)

## Step 8: Use algorithms to perform the task

#Deploy Naive Bayes algorithim

person.valid<-as.factor(person.valid)

confusionMatrix(preds\_nb, persons\_valid\_target, positive="yes")

## Step 9: Interpret the results

# Step 10: Deploy the model

person\_nb