auto insurance ML classification models

Josh Bicer

2023-06-01

Part 1: Install libraries and Read in Data

The sections below are used to install the required packages and libraries used for this script and to read in the training and testing datasets for the model.

```
# Install necessary packages and libraries

#install.packages("Amelia")
#install.packages("e1071")
#install.packages("psych")
#install.packages("class")
#install.packages("dplyr")
#install.packages("ROCR")
#install.packages("corrplot")
#install.packages("car")
#install.packages("leaps")
#install.packages("MASS")
#install.packages('glm2')
#install.packages("pROC")
#install.packages("InformationValue")
#install.packages("pbkrtest")
#install.packages("caret")
#install.packages("party")
#install.packages("ipred")
#install.packages("gbm")
library(class)
library(dplyr)
library(zoo)
library(ROCR)
library(corrplot)
library(car)
library(leaps)
library(MASS)
library(glm2)
library(pROC)
library(InformationValue)
library(pbkrtest)
library(caret)
library(Amelia)
library(e1071)
library(psych)
library(party)
library(ipred)
```

```r
library(rpart)
library(randomForest)
library(gbm)
```

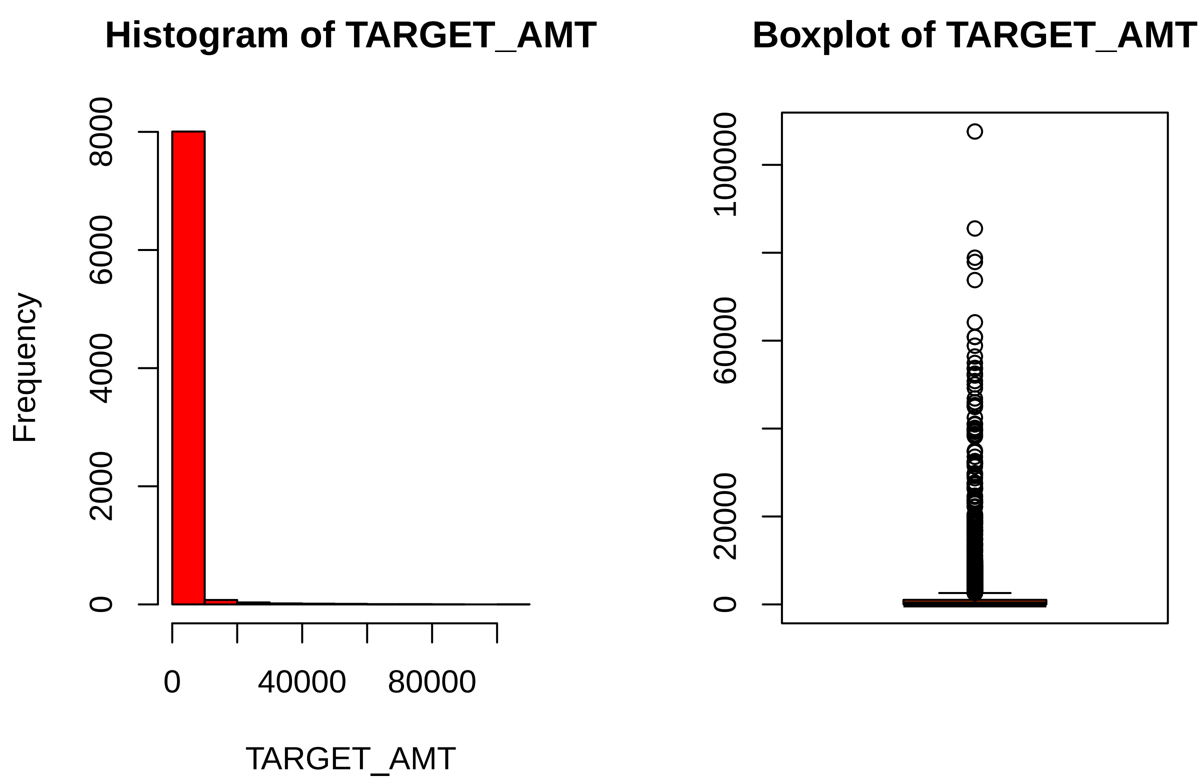
```r
# Import data from CSV for training and testing data sets
data = read.csv("auto_insurance_training.csv")
test = read.csv("auto_insurance_test.csv")


# Read in variables as factors or numeric in training data set
data$INDEX = as.factor(data$INDEX)
data$TARGET_FLAG = as.factor(data$TARGET_FLAG)
data$SEX = as.factor(data$SEX)
data$EDUCATION = as.factor(data$EDUCATION)
data$PARENT1 = as.factor(data$PARENT1)
data$INCOME = suppressWarnings(as.numeric(gsub("[^0-9.]", "", data$INCOME)))
data$HOME_VAL = suppressWarnings(as.numeric(gsub("[^0-9.]", "", data$HOME_VAL)))
data$MSTATUS = as.factor(data$MSTATUS)
data$REVOKED = as.factor(data$REVOKED)
data$RED_CAR = as.factor(ifelse(data$RED_CAR=="yes", 1, 0))
data$URBANICITY = ifelse(data$URBANICITY == "Highly Urban/ Urban", "Urban", "Rural")
data$URBANICITY = as.factor(data$URBANICITY)
data$JOB = as.factor(data$JOB)
data$CAR_USE = as.factor(data$CAR_USE)
data$CAR_TYPE = as.factor(data$CAR_TYPE)
data$DO_KIDS_DRIVE = as.factor(ifelse(data$KIDSDRIV > 0, 1, 0 ))
data$OLDCLAIM = suppressWarnings(as.numeric(gsub("[^0-9.]", "", data$HOME_VAL)))
data$BLUEBOOK = suppressWarnings(as.numeric(gsub("[^0-9.]", "", data$BLUEBOOK)))

# Read in variables as factor or numeric for testing data set
test$INDEX = as.factor(test$INDEX)
test$TARGET_FLAG = as.factor(test$TARGET_FLAG)
test$SEX = as.factor(test$SEX)
test$EDUCATION = as.factor(test$EDUCATION)
test$PARENT1 = as.factor(test$PARENT1)
test$INCOME = suppressWarnings(as.numeric(gsub("[^0-9.]", "", test$INCOME)))
test$HOME_VAL = suppressWarnings(as.numeric(gsub("[^0-9.]", "", test$HOME_VAL)))
test$MSTATUS = as.factor(test$MSTATUS)
test$REVOKED = as.factor(test$REVOKED)
test$RED_CAR = as.factor(ifelse(test$RED_CAR=="yes", 1, 0))
test$URBANICITY = ifelse(test$URBANICITY == "Highly Urban/ Urban", "Urban", "Rural")
test$URBANICITY = as.factor(test$URBANICITY)
test$JOB = as.factor(test$JOB)
test$CAR_USE = as.factor(test$CAR_USE)
test$CAR_TYPE = as.factor(test$CAR_TYPE)
test$DO_KIDS_DRIVE = as.factor(ifelse(test$KIDSDRIV > 0, 1, 0 ))
test$OLDCLAIM = suppressWarnings(as.numeric(gsub("[^0-9.]", "", test$HOME_VAL)))
test$BLUEBOOK = suppressWarnings(as.numeric(gsub("[^0-9.]", "", test$BLUEBOOK)))
```

Part 2: Data Exploration

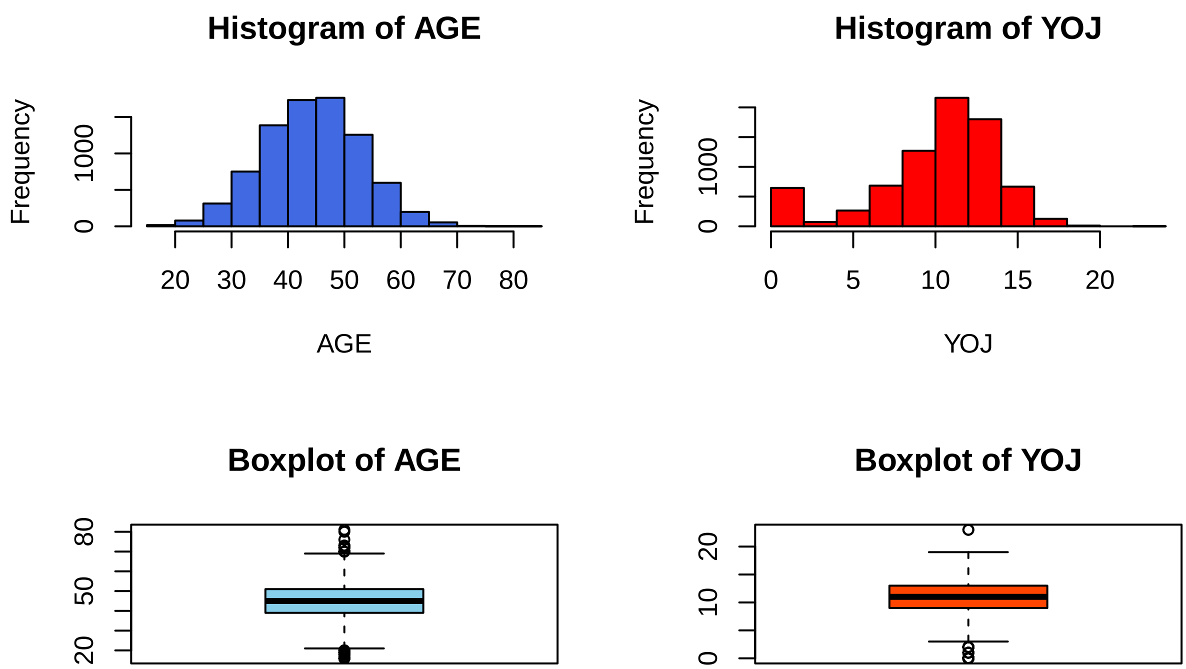
Part 2 of the script explores the data by creating histograms, box plots, and correlation plots of the data. This is meant to gain a better understand of the variables used for the model and how they interact.

The dataset is available online and features a series of auto insurance customers at a given company. The Target Flag represents a [0,1] binary outcome of whether the driver was involved in an accident or not.

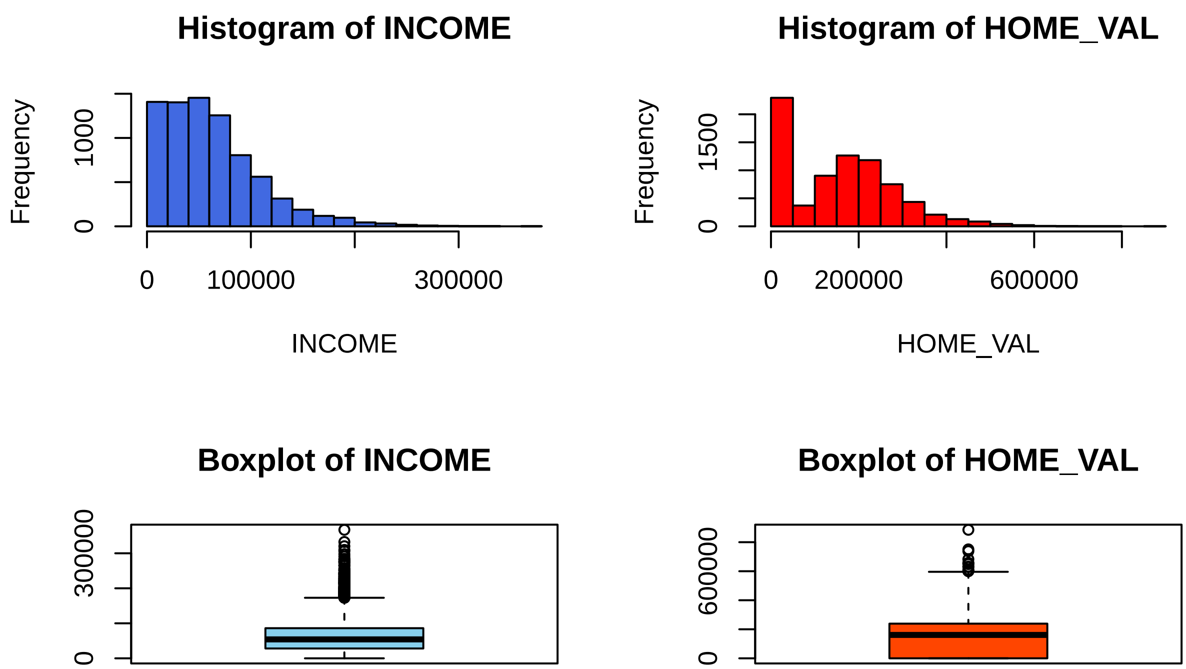
```
### Create histograms and boxplots for response variable and inputs

# Target Amount represents auto insurance claim amount as target variable
par(mfrow=c(1,2))
hist(data$TARGET_AMT, col = "red", xlab = "TARGET_AMT", main = "Histogram of TARGET_AMT")
boxplot(data$TARGET_AMT, col = "orangered", main = "Boxplot of TARGET_AMT")
```



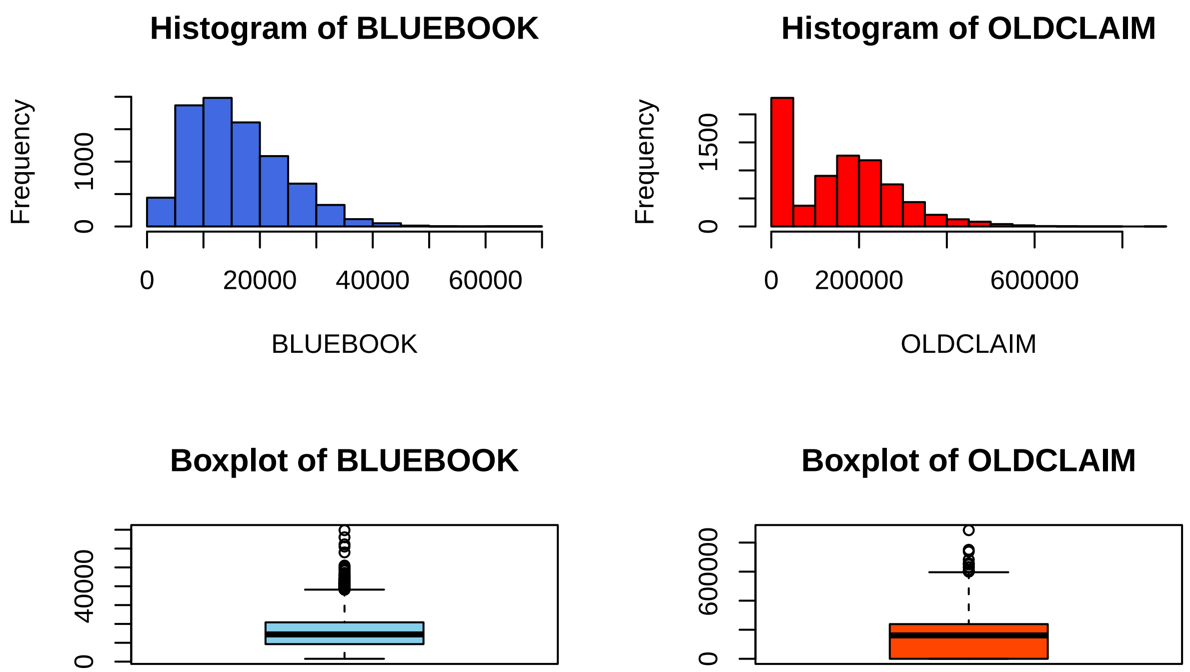

```
par(mfrow=c(1,1))

# Age and Years on Job inputs
par(mfrow=c(2,2))
hist(data$AGE, col = "royalblue", xlab = "AGE", main = "Histogram of AGE")
hist(data$YOJ, col = "red", xlab = "YOJ", main = "Histogram of YOJ")
boxplot(data$AGE, col = "skyblue", main = "Boxplot of AGE")
boxplot(data$YOJ, col = "orangered", main = "Boxplot of YOJ")
```



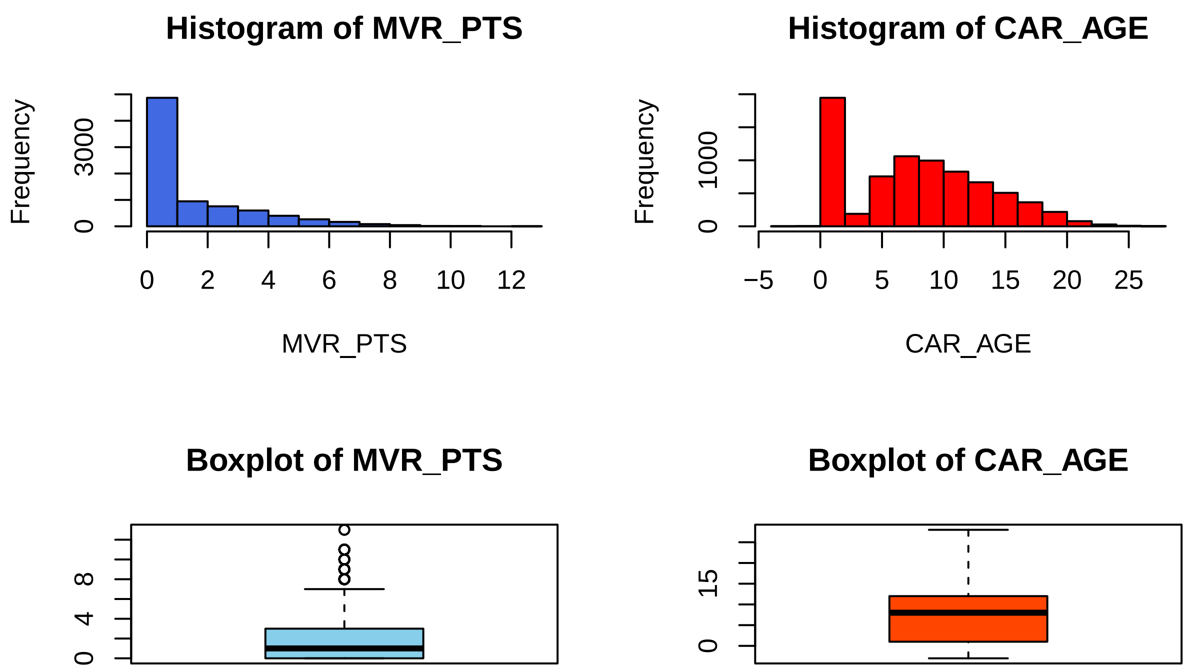

```r
par(mfrow=c(1,1))

# Income and Home Value inputs
par(mfrow=c(2,2))
hist(data$INCOME, col = "royalblue", xlab = "INCOME", main = "Histogram of INCOME")
hist(data$HOME_VAL, col = "red", xlab = "HOME_VAL", main = "Histogram of HOME_VAL")
boxplot(data$INCOME, col = "skyblue", main = "Boxplot of INCOME")
boxplot(data$HOME_VAL, col = "orangered", main = "Boxplot of HOME_VAL")
```



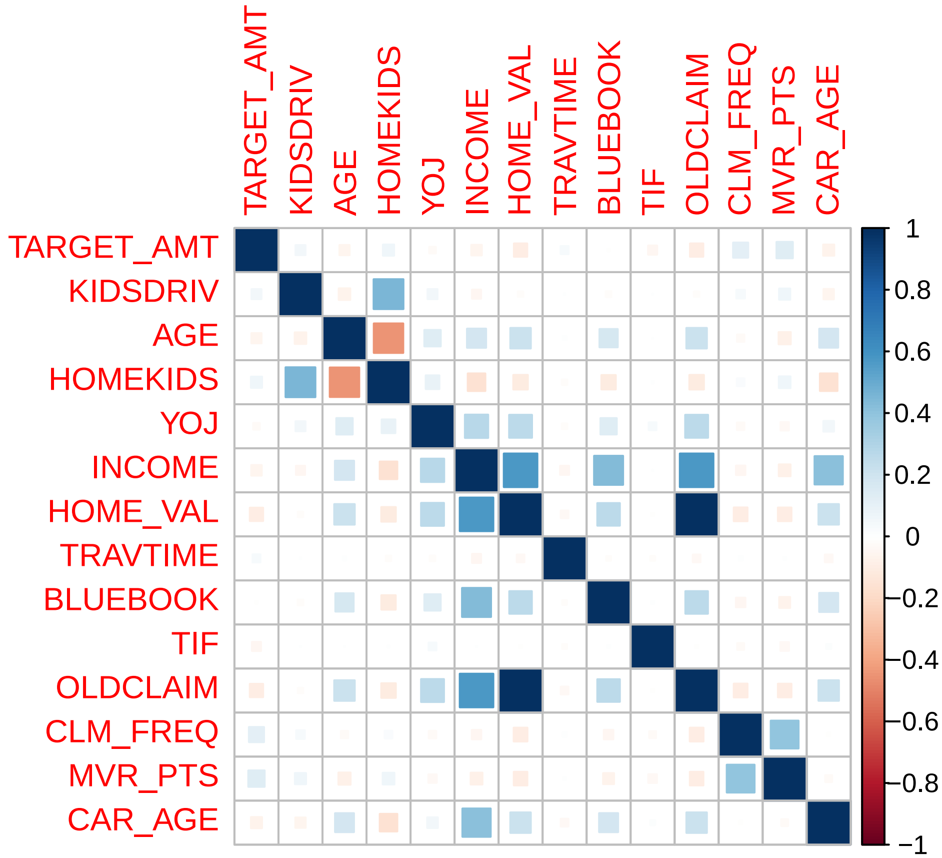

```
par(mfrow=c(1,1))

# Bluebook home value and old claim amount inputs
par(mfrow=c(2,2))
hist(data$BLUEBOOK, col = "royalblue", xlab = "BLUEBOOK", main = "Histogram of BLUEBOOK")
hist(data$OLDCLAIM, col = "red", xlab = "OLDCLAIM", main = "Histogram of OLDCLAIM")
boxplot(data$BLUEBOOK, col = "skyblue", main = "Boxplot of BLUEBOOK")
boxplot(data$OLDCLAIM, col = "orangered", main = "Boxplot of OLDCLAIM")
```




```r
par(mfrow=c(1,1))

# MVR points and car age in years inputs
par(mfrow=c(2,2))
hist(data$MVR_PTS, col = "royalblue", xlab = "MVR_PTS", main = "Histogram of MVR_PTS")
hist(data$CAR_AGE, col = "red", xlab = "CAR_AGE", main = "Histogram of CAR_AGE")
boxplot(data$MVR_PTS, col = "skyblue", main = "Boxplot of MVR_PTS")
boxplot(data$CAR_AGE, col = "orangered", main = "Boxplot of CAR_AGE")
```




```
par(mfrow=c(1,1))

# Explore correlation between input variables
c = na.omit(data)
c1 = cor(c[sapply(c, is.numeric)])
corrplot(c1, method = "square")
```



Part 3: Data Preparation This portion of the script is to prepare the data for the models. Flag variables are created to denote where any missing values have been replaced or imputed with the median value. To compute the median replacement values, the na.aggregate function is applied to impute based on other relevant input variables. Variables for education, income, home value, age, and old claims are put into bins to evaluate the effectiveness in the model. Finally, several squared inputs and interaction terms for home value, income, and bluebook value are created. The same steps are then performed on the testing dataset to ensure consistency.

```
### Training: Fix NA's and replace with median value. Create FLAG variables for missing values
data$AGE_FLAG = as.factor(ifelse(is.na(data$AGE), 1, 0))
data$AGE[is.na(data$AGE)] = median(data$AGE, na.rm = "TRUE")

# Years on Job
# Input missing values from median of Job
data$YOJ_FLAG = as.factor(ifelse(is.na(data$YOJ), 1, 0))
data$YOJ = na.aggregate(data$YOJ, data$JOB, median, na.rm = TRUE)

# Income
# Input missing values from median of Job
data$INCOME_FLAG = as.factor(ifelse(is.na(data$INCOME), 1, 0))
data$INCOME = na.aggregate(data$INCOME, data$JOB, median(), na.rm = TRUE)

# Home Value
# Input missing values from median of job
data$HOME_VAL_FLAG = as.factor(ifelse(is.na(data$HOME_VAL), 1, 0))
data$HOME_VAL = na.aggregate(data$HOME_VAL, data$JOB, median, na.rm = TRUE)
```

```
# Car age in years
data$CAR_AGE[data$CAR_AGE < 0 ] = NA
data$CAR_AGE_FLAG = as.factor(ifelse(is.na(data$CAR_AGE), 1, 0))
data$CAR_AGE = na.aggregate(data$CAR_AGE, data$CAR_TYPE, median, na.rm = TRUE)

# Old claims
data$OLDCLAIM_FLAG = as.factor(ifelse(is.na(data$OLDCLAIM), 1, 0))
data$OLDCLAIM = ifelse(data$CAR_AGE < 5 & !is.na(data$CAR_AGE),0,data$OLDCLAIM)
data$OLDCLAIM = na.aggregate(data$OLDCLAIM, data$CAR_AGE, mean, na.rm = TRUE)

### Training: Create imputed variables and bin variables
data$HOME_OWNER = as.factor(ifelse(data$HOME_VAL == 0, 0, 1))

# Create squared roots for larger numeric values
data$SQRT_TRAVTIME = sqrt(data$TRAVTIME)
data$SQRT_BLUEBOOK = sqrt(data$BLUEBOOK)
data$SQRT_HOME_VAL = sqrt(data$HOME_VAL)

# Bin Income using 1st and 3rd quantiles. Separate NA and Zero values.
data$INCOME_bin[data$INCOME == 0] = "Zero"
data$INCOME_bin[data$INCOME > 0 & data$INCOME < quantile(data$INCOME, c(.25))] = "Low"
data$INCOME_bin[data$INCOME >= quantile(data$INCOME, c(.25)) & data$INCOME < quantile(data$INCOME, c(.75
data$INCOME_bin[data$INCOME >= quantile(data$INCOME, c(.75))] = "High"
data$INCOME_bin[data$INCOME_FLAG == 1] = "NA"
data$INCOME_bin = factor(data$INCOME_bin)
data$INCOME_bin = factor(data$INCOME_bin, levels=c("NA","Zero","Low","Medium","High"))

# Bin Education into 3 Groups
data$EDUCATION_bin[data$EDUCATION == "<High School" | data$EDUCATION == "z_High School"] = "High School
data$EDUCATION_bin[data$EDUCATION == "Bachelors" ] = "Bachelors"
data$EDUCATION_bin[data$EDUCATION == "PhD" | data$EDUCATION == "Masters"] = "Advanced Degree"
data$EDUCATION_bin = factor(data$EDUCATION_bin)
data$EDUCATION_bin = factor(data$EDUCATION_bin, levels = c("High School or Less", "Bachelors", "Advance

# Bin Home Value into 4 Groups
data$HOME_VAL_bin[data$HOME_VAL == 0] = "No Home"
data$HOME_VAL_bin[data$HOME_VAL > 0 & data$HOME_VAL < 150000] = "Low"
data$HOME_VAL_bin[data$HOME_VAL >= 150000 & data$HOME_VAL < 300000] = "Medium"
data$HOME_VAL_bin[data$HOME_VAL >= 300000] = "High"
data$HOME_VAL_bin = factor(data$HOME_VAL_bin)
data$HOME_VAL_bin = factor(data$HOME_VAL_bin, levels = c("No Home", "Low", "Medium", "High"))

# Bin Age into 5 Groups
data$AGE_bin[data$AGE >= 16 & data$AGE <= 19] = "Teenager"
data$AGE_bin[data$AGE >= 20 & data$AGE <= 26] = "Young Adult"
data$AGE_bin[data$AGE >= 27 & data$AGE <= 43] = "Adult"
data$AGE_bin[data$AGE >= 44 & data$AGE <= 62] = "Gen X"
data$AGE_bin[data$AGE >= 62] = "62 and over"
data$AGE_bin = factor(data$AGE_bin)
data$AGE_bin = factor(data$AGE_bin, levels = c("Teenager", "Young Adult", "Adult", "Gen X", "62 and ove

# Bin Old Claims into 3 Groups
data$OLDCLAIM_bin[data$OLDCLAIM == 0] = "No Claims"
```

```r
data$OLDCLAIM_bin[data$OLDCLAIM > 0 & data$OLDCLAIM <= quantile(data$OLDCLAIM, c(.75))] = "Low Claims"
data$OLDCLAIM_bin[data$OLDCLAIM > quantile(data$OLDCLAIM, c(.75))] = "High Claims"
data$OLDCLAIM_bin = factor(data$OLDCLAIM_bin)
data$OLDCLAIM_bin = factor(data$OLDCLAIM_bin, levels = c("No Claims", "Low Claims", "High Claims"))

# Confirm data is clean
summary(data)
```

```
##      INDEX       TARGET_FLAG   TARGET_AMT        KIDSDRIV           AGE
##  1      :   1    0:6008      Min.   :     0   Min.   :0.0000   Min.   :16.00
##  2      :   1    1:2153      1st Qu.:     0   1st Qu.:0.0000   1st Qu.:39.00
##  4      :   1                Median :     0   Median :0.0000   Median :45.00
##  5      :   1                Mean   :  1504   Mean   :0.1711   Mean   :44.79
##  6      :   1                3rd Qu.:  1036   3rd Qu.:0.0000   3rd Qu.:51.00
##  7      :   1                Max.   :107586   Max.   :4.0000   Max.   :81.00
##  (Other):8155
##     HOMEKIDS            YOJ             INCOME        PARENT1       HOME_VAL
##  Min.   :0.0000   Min.   : 0.00   Min.   :     0   No :7084   Min.   :     0
##  1st Qu.:0.0000   1st Qu.: 9.00   1st Qu.: 28299   Yes:1077   1st Qu.:     0
##  Median :0.0000   Median :11.34   Median : 54877              Median :160429
##  Mean   :0.7212   Mean   :10.50   Mean   : 61603              Mean   :154848
##  3rd Qu.:1.0000   3rd Qu.:13.00   3rd Qu.: 86268              3rd Qu.:234300
##  Max.   :5.0000   Max.   :23.00   Max.   :367030              Max.   :885282
##
##  MSTATUS      SEX                EDUCATION               JOB
##  Yes :4894   M   :3786   <High School :1203   z_Blue Collar:1825
##  z_No:3267   z_F:4375   Bachelors    :2242   Clerical     :1271
##                          Masters      :1658   Professional :1117
##                          PhD          : 728   Manager      : 988
##                          z_High School:2330   Lawyer       : 835
##                                               Student      : 712
##                                               (Other)      :1413
##     TRAVTIME             CAR_USE         BLUEBOOK           TIF
##  Min.   :  5.00   Commercial:3029   Min.   : 1500   Min.   : 1.000
##  1st Qu.: 22.00   Private   :5132   1st Qu.: 9280   1st Qu.: 1.000
##  Median : 33.00                     Median :14440   Median : 4.000
##  Mean   : 33.49                     Mean   :15710   Mean   : 5.351
##  3rd Qu.: 44.00                     3rd Qu.:20850   3rd Qu.: 7.000
##  Max.   :142.00                     Max.   :69740   Max.   :25.000
##
##         CAR_TYPE    RED_CAR     OLDCLAIM         CLM_FREQ       REVOKED
##  Minivan    :2145   0:5783   Min.   :     0   Min.   :0.0000   No :7161
##  Panel Truck: 676   1:2378   1st Qu.:     0   1st Qu.:0.0000   Yes:1000
##  Pickup     :1389            Median :105539   Median :0.0000
##  Sports Car : 907            Mean   :122125   Mean   :0.7986
##  Van        : 750            3rd Qu.:218964   3rd Qu.:2.0000
##  z_SUV      :2294            Max.   :885282   Max.   :5.0000
##
##     MVR_PTS          CAR_AGE       URBANICITY   DO_KIDS_DRIVE AGE_FLAG YOJ_FLAG
##  Min.   : 0.000   Min.   : 0.000   Rural:1669   0:7180        0:8155   0:7707
##  1st Qu.: 0.000   1st Qu.: 4.000   Urban:6492   1: 981        1:   6   1: 454
##  Median : 1.000   Median : 8.000
##  Mean   : 1.696   Mean   : 8.331
##  3rd Qu.: 3.000   3rd Qu.:12.000
```

```
##  Max.   :13.000   Max.   :28.000
##
##  INCOME_FLAG HOME_VAL_FLAG CAR_AGE_FLAG OLDCLAIM_FLAG HOME_OWNER
##  0:7716      0:7697        0:7650       0:7697        0:2294
##  1: 445      1: 464        1: 511       1: 464        1:5867
##
##
##
##
##
##  SQRT_TRAVTIME    SQRT_BLUEBOOK    SQRT_HOME_VAL    INCOME_bin
##  Min.   : 2.236   Min.   : 38.73   Min.   :  0.0   NA    : 445
##  1st Qu.: 4.690   1st Qu.: 96.33   1st Qu.:  0.0   Zero  : 615
##  Median : 5.745   Median :120.17   Median :400.5   Low   :1328
##  Mean   : 5.599   Mean   :120.62   Mean   :325.6   Medium:3850
##  3rd Qu.: 6.633   3rd Qu.:144.40   3rd Qu.:484.0   High  :1923
##  Max.   :11.916   Max.   :264.08   Max.   :940.9
##
##              EDUCATION_bin   HOME_VAL_bin        AGE_bin
##  High School or Less:3533   No Home:2294   Teenager   :  14
##  Bachelors          :2242   Low    :1423   Young Adult: 120
##  Advanced Degree    :2386   Medium :3511   Adult      :3432
##                             High   : 933   Gen X      :4393
##                                            62 and over: 202
##
##
##      OLDCLAIM_bin
##  No Claims  :3794
##  Low Claims :2327
##  High Claims:2040
##
##
##
##
```

The same data preparation steps are performed on the testing dataset below

```r
# Age
test$AGE_FLAG = as.factor(ifelse(is.na(test$AGE), 1, 0))
test$AGE[is.na(test$AGE)] = median(data$AGE, na.rm = "TRUE")

# Years on Job
test$YOJ_FLAG = as.factor(ifelse(is.na(test$YOJ), 1, 0))
test$YOJ = na.aggregate(test$YOJ, test$JOB, median(data$YOJ), na.rm = TRUE)

# Income
test$INCOME_FLAG = as.factor(ifelse(is.na(test$INCOME), 1, 0))
test$INCOME = na.aggregate(test$INCOME, test$JOB, median(data$INCOME), na.rm = TRUE)

# Home Value
test$HOME_VAL_FLAG = as.factor(ifelse(is.na(test$HOME_VAL), 1, 0))
test$HOME_VAL = na.aggregate(test$HOME_VAL, test$JOB, median(data$HOME_VAL), na.rm = TRUE)

# Car Age
test$CAR_AGE[test$CAR_AGE < 0 ] = NA
```

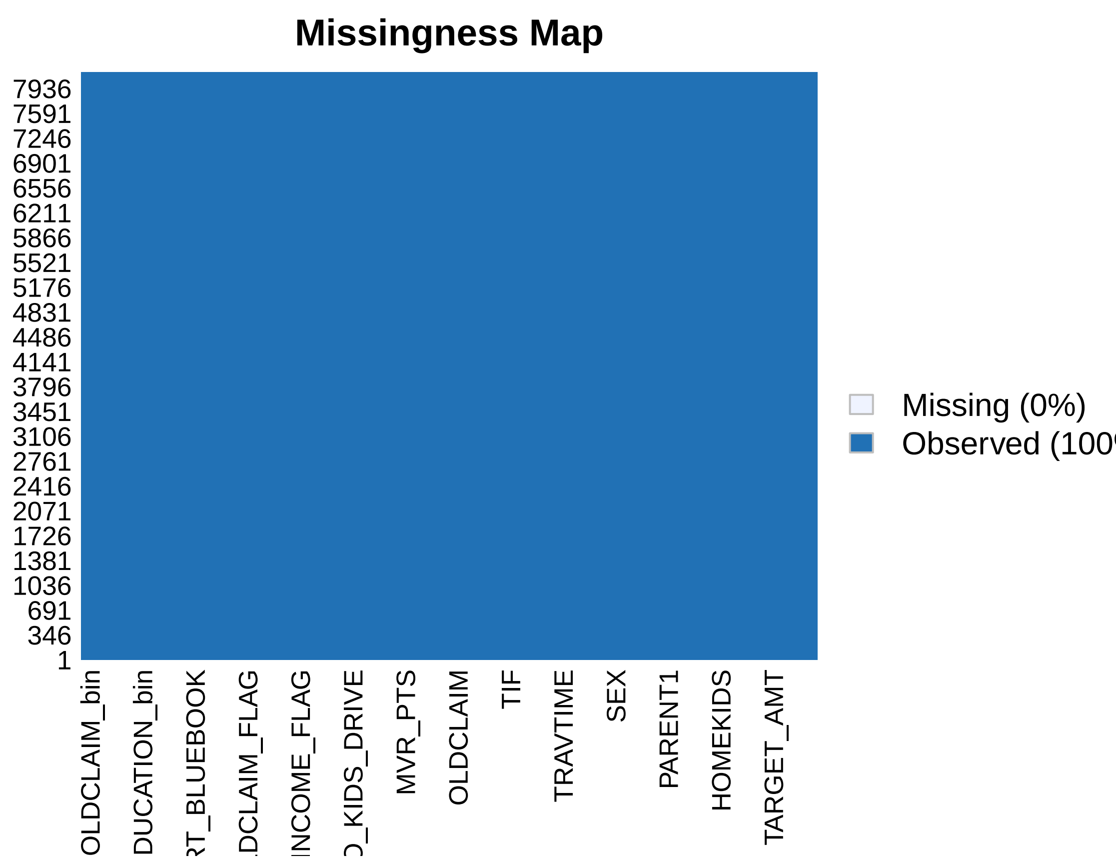
```
test$CAR_AGE_FLAG = as.factor(ifelse(is.na(test$CAR_AGE), 1, 0))
test$CAR_AGE = na.aggregate(test$CAR_AGE, test$CAR_TYPE, median(data$CAR_AGE), na.rm = TRUE)

# Old Claims
test$OLDCLAIM_FLAG = as.factor(ifelse(is.na(test$OLDCLAIM), 1, 0))
test$OLDCLAIM = ifelse(test$CAR_AGE < 5 & !is.na(test$CAR_AGE),0,test$OLDCLAIM)
test$OLDCLAIM = na.aggregate(test$OLDCLAIM, test$CAR_AGE, median(data$OLDCLAIM), na.rm = TRUE)

### Testing: Create imputed variables and bin variables
test$HOME_OWNER = as.factor(ifelse(test$HOME_VAL == 0, 0, 1))

# Create square root values for large numbers
test$SQRT_TRAVTIME = sqrt(test$TRAVTIME)
test$SQRT_BLUEBOOK = sqrt(test$BLUEBOOK)
test$SQRT_HOME_VAL = sqrt(test$HOME_VAL)

# Bin Income using 1st and 3rd quantiles. Separate NA and Zero values.
test$INCOME_bin[test$INCOME == 0] = "Zero"
test$INCOME_bin[test$INCOME > 0 & test$INCOME < quantile(data$INCOME, c(.25))] = "Low"
test$INCOME_bin[test$INCOME >= quantile(data$INCOME, c(.25)) & test$INCOME < quantile(data$INCOME, c(.75
test$INCOME_bin[test$INCOME >= quantile(data$INCOME, c(.75))] = "High"
test$INCOME_bin[test$INCOME_FLAG == 1] = "NA"
test$INCOME_bin = factor(test$INCOME_bin)
test$INCOME_bin = factor(test$INCOME_bin, levels=c("NA","Zero","Low","Medium","High"))

# Bin Education into 3 Groups
test$EDUCATION_bin[test$EDUCATION == "<High School" | test$EDUCATION == "z_High School"] = "High School
test$EDUCATION_bin[test$EDUCATION == "Bachelors" ] = "Bachelors"
test$EDUCATION_bin[test$EDUCATION == "PhD" | test$EDUCATION == "Masters"] = "Advanced Degree"
test$EDUCATION_bin = factor(test$EDUCATION_bin)
test$EDUCATION_bin = factor(test$EDUCATION_bin, levels = c("High School or Less", "Bachelors", "Advanced

# Bin Home Value into 4 Groups
test$HOME_VAL_bin[test$HOME_VAL == 0] = "No Home"
test$HOME_VAL_bin[test$HOME_VAL > 0 & test$HOME_VAL < 150000] = "Low"
test$HOME_VAL_bin[test$HOME_VAL >= 150000 & test$HOME_VAL < 300000] = "Medium"
test$HOME_VAL_bin[test$HOME_VAL >= 300000] = "High"
test$HOME_VAL_bin = factor(test$HOME_VAL_bin)
test$HOME_VAL_bin = factor(test$HOME_VAL_bin, levels = c("No Home", "Low", "Medium", "High"))

# Bin Age into 5 Groups
test$AGE_bin[test$AGE >= 16 & test$AGE <= 19] = "Teenager"
test$AGE_bin[test$AGE >= 20 & test$AGE <= 26] = "Young Adult"
test$AGE_bin[test$AGE >= 27 & test$AGE <= 43] = "Adult"
test$AGE_bin[test$AGE >= 44 & test$AGE <= 62] = "Gen X"
test$AGE_bin[test$AGE >= 62] = "62 and over"
test$AGE_bin = factor(test$AGE_bin)
test$AGE_bin = factor(test$AGE_bin, levels = c("Teenager", "Young Adult", "Adult", "Gen X", "62 and ove

# Bin Old Claims into 3 Groups
test$OLDCLAIM_bin[test$OLDCLAIM == 0] = "No Claims"
test$OLDCLAIM_bin[test$OLDCLAIM > 0 & test$OLDCLAIM <= quantile(data$OLDCLAIM, c(.75))] = "Low Claims"
test$OLDCLAIM_bin[test$OLDCLAIM > quantile(data$OLDCLAIM, c(.75))] = "High Claims"
```

```r
test$OLDCLAIM_bin = factor(test$OLDCLAIM_bin)
test$OLDCLAIM_bin = factor(test$OLDCLAIM_bin, levels = c("No Claims", "Low Claims", "High Claims"))

# Confirm data is clean and no missing observations
summary(test)
```

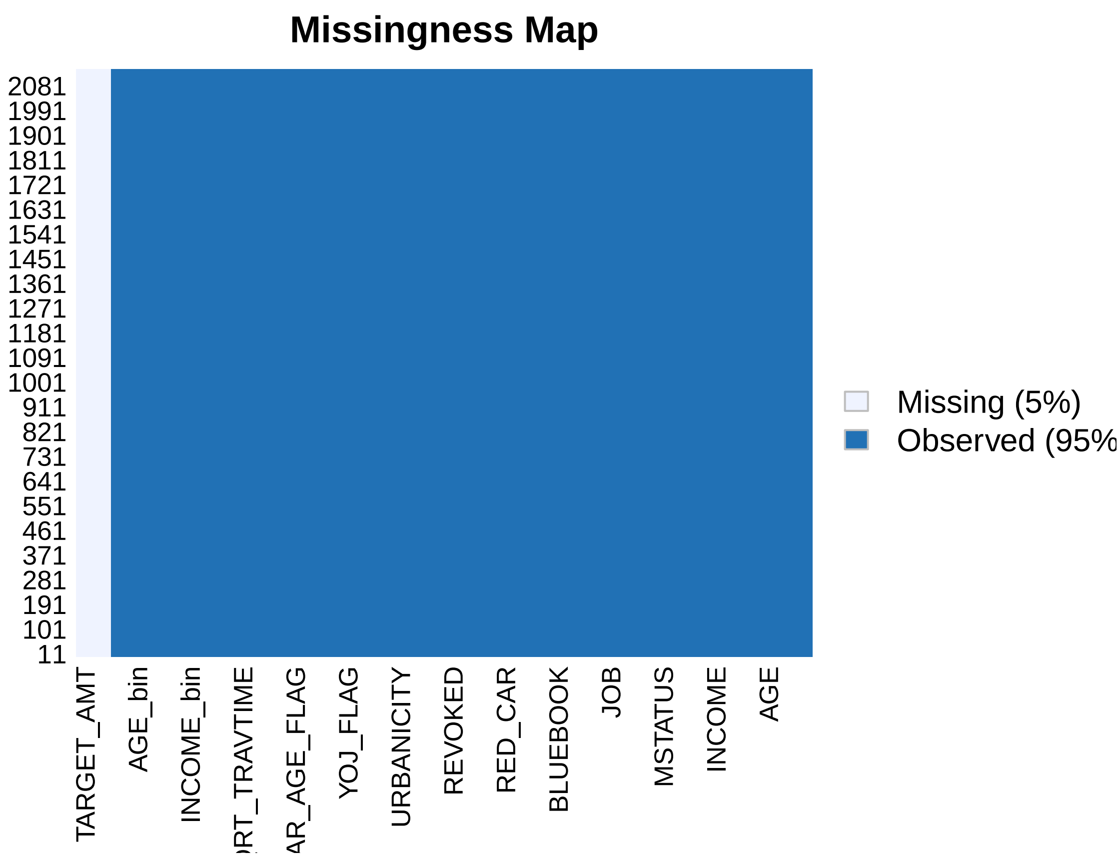
```
##      INDEX      TARGET_FLAG TARGET_AMT       KIDSDRIV          AGE
##  3      :   1   NA's:2141   Mode:logical   Min.   :0.0000   Min.   :17.00
##  9      :   1               NA's:2141      1st Qu.:0.0000   1st Qu.:39.00
##  10     :   1                              Median :0.0000   Median :45.00
##  18     :   1                              Mean   :0.1625   Mean   :45.02
##  21     :   1                              3rd Qu.:0.0000   3rd Qu.:51.00
##  30     :   1                              Max.   :3.0000   Max.   :73.00
##  (Other):2135
##     HOMEKIDS           YOJ            INCOME       PARENT1       HOME_VAL
##  Min.   :0.0000   Min.   : 0.00   Min.   :     0   No :1875   Min.   :     0
##  1st Qu.:0.0000   1st Qu.: 9.00   1st Qu.: 25929   Yes: 266   1st Qu.:     0
##  Median :0.0000   Median :11.18   Median : 53227              Median :159272
##  Mean   :0.7174   Mean   :10.37   Mean   : 60321              Mean   :153020
##  3rd Qu.:1.0000   3rd Qu.:13.00   3rd Qu.: 86541              3rd Qu.:231852
##  Max.   :5.0000   Max.   :19.00   Max.   :291182              Max.   :669271
##
##  MSTATUS      SEX                EDUCATION              JOB
##  Yes :1294   M  : 971   <High School :312   z_Blue Collar:463
##  z_No: 847   z_F:1170   Bachelors    :581   Clerical     :319
##                         Masters      :420   Professional :291
##                         PhD          :206   Manager      :269
##                         z_High School:622   Home Maker   :202
##                                             Lawyer       :196
##                                             (Other)      :401
##     TRAVTIME            CAR_USE        BLUEBOOK          TIF
##  Min.   :  5.00   Commercial: 760   Min.   : 1500   Min.   : 1.000
##  1st Qu.: 22.00   Private   :1381   1st Qu.: 8870   1st Qu.: 1.000
##  Median : 33.00                     Median :14170   Median : 4.000
##  Mean   : 33.15                     Mean   :15469   Mean   : 5.245
##  3rd Qu.: 43.00                     3rd Qu.:21050   3rd Qu.: 7.000
##  Max.   :105.00                     Max.   :49940   Max.   :25.000
##
##         CAR_TYPE   RED_CAR     OLDCLAIM         CLM_FREQ      REVOKED
##  Minivan     :549   0:1543   Min.   :     0   Min.   :0.000   No :1880
##  Panel Truck:177   1: 598   1st Qu.:     0   1st Qu.:0.000   Yes: 261
##  Pickup      :383            Median : 92197   Median :0.000
##  Sports Car :272            Mean   :119259   Mean   :0.809
##  Van         :171            3rd Qu.:215203   3rd Qu.:2.000
##  z_SUV       :589            Max.   :669271   Max.   :5.000
##
##     MVR_PTS          CAR_AGE        URBANICITY   DO_KIDS_DRIVE AGE_FLAG YOJ_FLAG
##  Min.   : 0.000   Min.   : 0.000   Rural: 403   0:1889        0:2140   0:2047
##  1st Qu.: 0.000   1st Qu.: 1.000   Urban:1738   1: 252        1:   1   1:  94
##  Median : 1.000   Median : 8.000
##  Mean   : 1.766   Mean   : 8.186
##  3rd Qu.: 3.000   3rd Qu.:12.000
##  Max.   :12.000   Max.   :26.000
##
```

```
##   INCOME_FLAG HOME_VAL_FLAG CAR_AGE_FLAG OLDCLAIM_FLAG HOME_OWNER
##   0:2016      0:2030        0:2012       0:2030        0: 614
##   1: 125      1: 111        1: 129       1: 111        1:1527
##
##
##
##
##
##   SQRT_TRAVTIME    SQRT_BLUEBOOK    SQRT_HOME_VAL    INCOME_bin
##   Min.   : 2.236   Min.   : 38.73   Min.   :  0.0   NA    :125
##   1st Qu.: 4.690   1st Qu.: 94.18   1st Qu.:  0.0   Zero  :182
##   Median : 5.745   Median :119.04   Median :399.1   Low   :365
##   Mean   : 5.570   Mean   :119.39   Mean   :322.1   Medium:965
##   3rd Qu.: 6.557   3rd Qu.:145.09   3rd Qu.:481.5   High  :504
##   Max.   :10.247   Max.   :223.47   Max.   :818.1
##
##               EDUCATION_bin  HOME_VAL_bin         AGE_bin          OLDCLAIM_bin
##   High School or Less:934   No Home:614   Teenager   :   4   No Claims   :1028
##   Bachelors          :581   Low    :389   Young Adult:  28   Low Claims : 594
##   Advanced Degree    :626   Medium :879   Adult      : 879   High Claims: 519
##                             High   :259   Gen X      :1178
##                                           62 and over:  52
##
##
```

```
missmap(data)
```



```
missmap(test)
```



Part 4: Model Development The model uses several classification Machine Learning models to compare below: 1. Logistic Regression 2. Decision Tree 3. Decision Tree with Bagging 4. Random Forest with Bagging 5. Decision Tree with Boosting

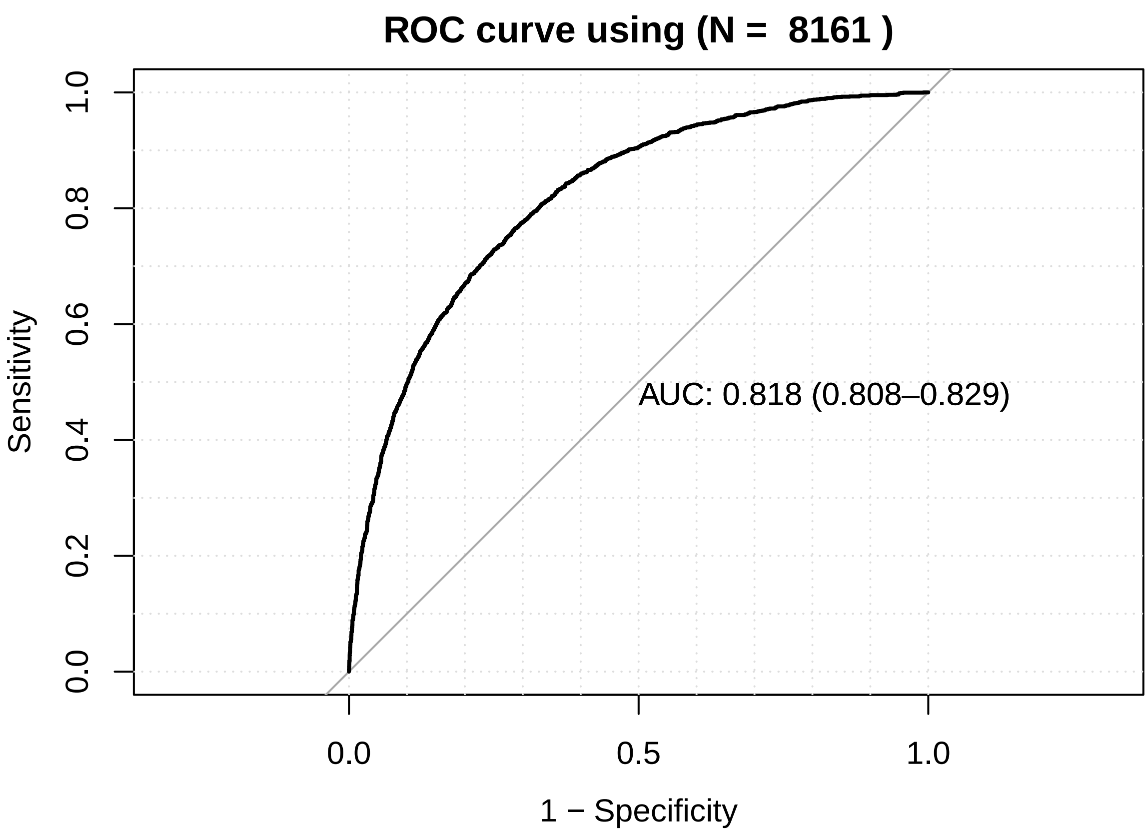
```
### Binary Response Model 1: Standard Logistic Regression
lr = glm(TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + AGE_FLAG + SEX +
               MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER +
               CLM_FREQ + REVOKED + MVR_PTS + URBANICITY + DO_KIDS_DRIVE +
               HOME_OWNER + SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK +
               OLDCLAIM_bin + INCOME_bin + AGE_bin + EDUCATION_bin, data = data, family = binomial())
summary(lr)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + AGE_FLAG +
##     SEX + MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER +
##     CLM_FREQ + REVOKED + MVR_PTS + URBANICITY + DO_KIDS_DRIVE +
##     HOME_OWNER + SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK + OLDCLAIM_bin +
##     INCOME_bin + AGE_bin + EDUCATION_bin, family = binomial(),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4712  -0.7093  -0.3933   0.5926   3.1883
##
```

```
## Coefficients:
##                                Estimate  Std. Error z value Pr(>|z|)
## (Intercept)                 -2.75236022  0.75548818  -3.643 0.000269 ***
## KIDSDRIV                     0.21743862  0.12239876   1.776 0.075654 .
## YOJ                          0.02011737  0.01119287   1.797 0.072282 .
## PARENT1Yes                   0.29669435  0.10062520   2.949 0.003193 **
## AGE_FLAG1                    2.26022885  1.20762467   1.872 0.061258 .
## SEXz_F                      -0.05400800  0.10199035  -0.530 0.596431
## MSTATUSz_No                  0.52154180  0.08529091   6.115 9.66e-10 ***
## JOBClerical                  0.47985073  0.19612645   2.447 0.014419 *
## JOBDoctor                   -0.34222197  0.24872862  -1.376 0.168857
## JOBHome Maker                0.21983265  0.21826120   1.007 0.313839
## JOBLawyer                    0.17173733  0.16441619   1.045 0.296241
## JOBManager                  -0.51673998  0.17080775  -3.025 0.002484 **
## JOBProfessional              0.21242676  0.17754645   1.196 0.231518
## JOBStudent                   0.09443883  0.22524364   0.419 0.675016
## JOBz_Blue Collar             0.36440688  0.18505415   1.969 0.048931 *
## CAR_USEPrivate              -0.76914498  0.08826209  -8.714  < 2e-16 ***
## TIF                         -0.05615736  0.00739728  -7.592 3.16e-14 ***
## CAR_TYPEPanel Truck          0.51909063  0.16661855   3.115 0.001837 **
## CAR_TYPEPickup               0.59578848  0.10091006   5.904 3.54e-09 ***
## CAR_TYPESports Car           0.94938626  0.13357070   7.108 1.18e-12 ***
## CAR_TYPEVan                  0.68275379  0.12709851   5.372 7.79e-08 ***
## CAR_TYPEz_SUV                0.77187414  0.11286982   6.839 8.00e-12 ***
## HOME_OWNER1                 -0.27814122  0.10490464  -2.651 0.008017 **
## CLM_FREQ                     0.15014290  0.02564370   5.855 4.77e-09 ***
## REVOKEDYes                   0.73175633  0.08087597   9.048  < 2e-16 ***
## MVR_PTS                      0.10016645  0.01375559   7.282 3.29e-13 ***
## URBANICITYUrban              2.42295840  0.11355605  21.337  < 2e-16 ***
## DO_KIDS_DRIVE1               0.40797778  0.19597161   2.082 0.037359 *
## SQRT_TRAVTIME                0.17090438  0.02099488   8.140 3.94e-16 ***
## BLUEBOOK                     0.00003976  0.00002114   1.881 0.060021 .
## SQRT_BLUEBOOK               -0.01495263  0.00493713  -3.029 0.002457 **
## OLDCLAIM_binLow Claims      -0.01180649  0.09015853  -0.131 0.895813
## OLDCLAIM_binHigh Claims     -0.16529941  0.11112116  -1.488 0.136867
## INCOME_binZero               0.81856257  0.21397183   3.826 0.000130 ***
## INCOME_binLow                0.02291488  0.15101460   0.152 0.879392
## INCOME_binMedium             0.02327042  0.13597123   0.171 0.864112
## INCOME_binHigh              -0.37462155  0.15515609  -2.414 0.015758 *
## AGE_binYoung Adult           0.56338651  0.65975303   0.854 0.393141
## AGE_binAdult                -0.66909250  0.62352093  -1.073 0.283232
## AGE_binGen X                -0.85680172  0.62453787  -1.372 0.170095
## AGE_bin62 and over          -0.33941280  0.65125870  -0.521 0.602252
## EDUCATION_binBachelors      -0.40745286  0.08515090  -4.785 1.71e-06 ***
## EDUCATION_binAdvanced Degree -0.28370443  0.14205752  -1.997 0.045813 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9418.0  on 8160  degrees of freedom
## Residual deviance: 7213.6  on 8118  degrees of freedom
## AIC: 7299.6
##
```

```
## Number of Fisher Scoring iterations: 5
```

```r
# Calculate ROC Curve and AUC for Model 1
predicted1 = predict(lr, data, type="response")
par(mfrow = c(1, 1))
roc(data$TARGET_FLAG, as.vector(predicted1), percent=F, boot.n=1000, ci.alpha=0.9, stratified=FALSE,
    plot=TRUE, grid=TRUE, show.thres=TRUE, legacy.axes = TRUE, reuse.auc = TRUE,print.auc = TRUE,
    print.thres.col = "blue", ci=TRUE, ci.type="bars", print.thres.cex = 0.7,
    main = paste("ROC curve using","(N = ",nrow(data),")"))
```




```
##
## Call:
## roc.default(response = data$TARGET_FLAG, predictor = as.vector(predicted1),     percent = F, ci = TRU
##
## Data: as.vector(predicted1) in 6008 controls (data$TARGET_FLAG 0) < 2153 cases (data$TARGET_FLAG 1).
## Area under the curve: 0.8184
## 95% CI: 0.8084-0.8285 (DeLong)
```

```r
# Confusion Matrix for Model 1
lrPredict = ifelse(predicted1 > .5, 1, 0)
lrPredict = as.factor(lrPredict)
CM1 = confusionMatrix(lrPredict, data$TARGET_FLAG)

### Binary Response Model 2: Standard Decision Tree
tree = ctree(TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + AGE_FLAG + SEX +
                 MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER +
                 CLM_FREQ + REVOKED + MVR_PTS + URBANICITY + DO_KIDS_DRIVE +
                 HOME_OWNER + SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK +
```

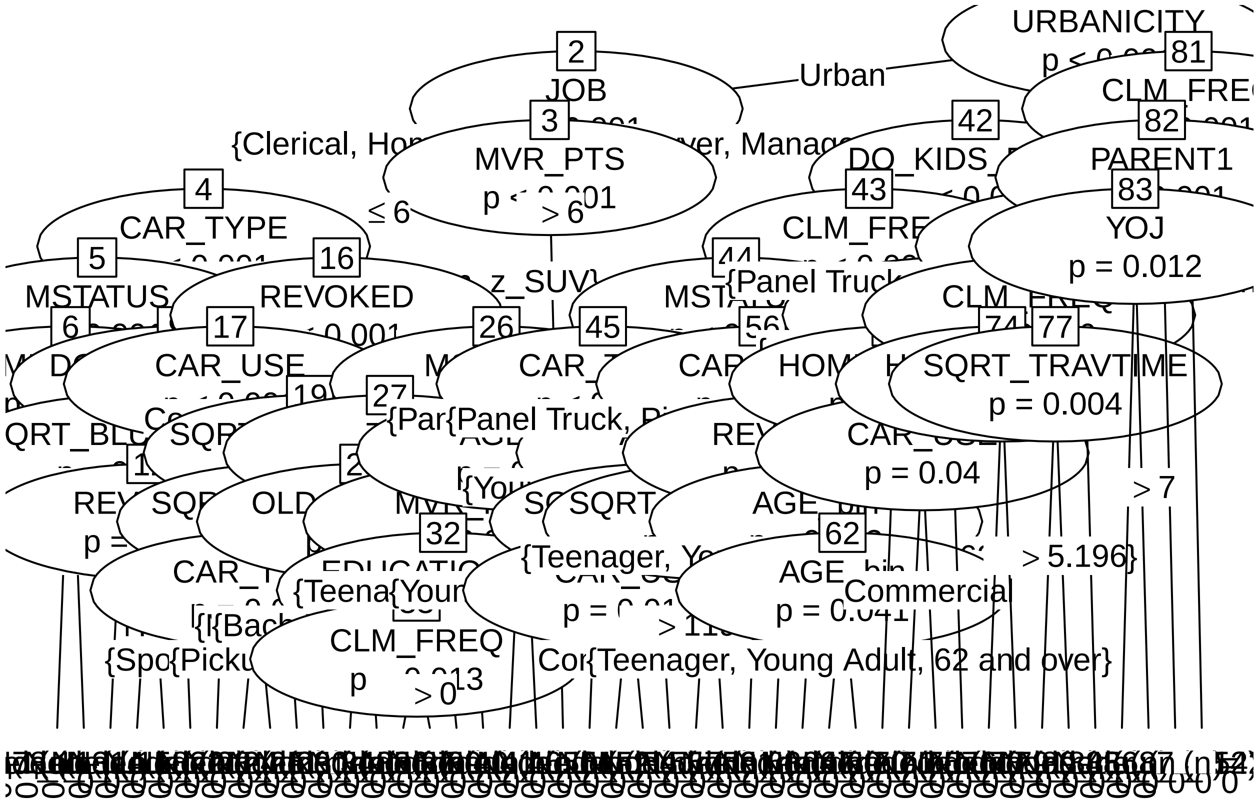
```
                 OLDCLAIM_bin + INCOME_bin + AGE_bin + EDUCATION_bin
                ,data = data)

print(tree)
```

```
##
##   Conditional inference tree with 44 terminal nodes
##
## Response:  TARGET_FLAG
## Inputs:  KIDSDRIV, YOJ, PARENT1, AGE_FLAG, SEX, MSTATUS, JOB, CAR_USE, TIF, CAR_TYPE, HOME_OWNER, CLI
## Number of observations:  8161
##
## 1) URBANICITY == {Urban}; criterion = 1, statistic = 410.354
##   2) JOB == {Clerical, Home Maker, Student, z_Blue Collar}; criterion = 1, statistic = 505.981
##     3) MVR_PTS <= 6; criterion = 1, statistic = 116.463
##       4) CAR_TYPE == {Minivan}; criterion = 1, statistic = 96.484
##         5) MSTATUS == {Yes}; criterion = 1, statistic = 29.734
##           6) HOME_OWNER == {0}; criterion = 0.997, statistic = 14.728
##             7)*  weights = 48
##           6) HOME_OWNER == {1}
##             8)*  weights = 421
##         5) MSTATUS == {z_No}
##           9) DO_KIDS_DRIVE == {0}; criterion = 0.979, statistic = 18.209
##             10) SQRT_BLUEBOOK <= 84.2615; criterion = 0.985, statistic = 14.434
##               11)*  weights = 31
##             10) SQRT_BLUEBOOK > 84.2615
##               12) REVOKED == {No}; criterion = 0.951, statistic = 9.791
##                 13)*  weights = 177
##               12) REVOKED == {Yes}
##                 14)*  weights = 25
##           9) DO_KIDS_DRIVE == {1}
##             15)*  weights = 24
##       4) CAR_TYPE == {Panel Truck, Pickup, Sports Car, Van, z_SUV}
##         16) REVOKED == {Yes}; criterion = 1, statistic = 68.341
##           17) CAR_USE == {Commercial}; criterion = 1, statistic = 18.473
##             18)*  weights = 170
##           17) CAR_USE == {Private}
##             19) SQRT_BLUEBOOK <= 128.6857; criterion = 0.982, statistic = 13.609
##               20) SQRT_TRAVTIME <= 6.244998; criterion = 0.998, statistic = 15.465
##                 21) CAR_TYPE == {Sports Car, z_SUV}; criterion = 0.984, statistic = 17.028
##                   22)*  weights = 72
##                 21) CAR_TYPE == {Pickup, Van}
##                   23)*  weights = 12
##               20) SQRT_TRAVTIME > 6.244998
##                 24)*  weights = 45
##             19) SQRT_BLUEBOOK > 128.6857
##               25)*  weights = 33
##         16) REVOKED == {No}
##           26) MSTATUS == {Yes}; criterion = 1, statistic = 47.128
##             27) TIF <= 2; criterion = 1, statistic = 27.358
##               28) OLDCLAIM_bin == {High Claims}; criterion = 0.98, statistic = 14.072
##                 29)*  weights = 37
##               28) OLDCLAIM_bin == {No Claims, Low Claims}
##                 30)*  weights = 337
```

```
##             27) TIF > 2
##               31) MVR_PTS <= 2; criterion = 0.996, statistic = 18.416
##                 32) EDUCATION_bin == {Bachelors, Advanced Degree}; criterion = 0.974, statistic = 13
##                   33) CLM_FREQ <= 0; criterion = 0.987, statistic = 11.858
##                     34)*  weights = 102
##                   33) CLM_FREQ > 0
##                     35)*  weights = 57
##                 32) EDUCATION_bin == {High School or Less}
##                   36)*  weights = 401
##               31) MVR_PTS > 2
##                 37)*  weights = 204
##           26) MSTATUS == {z_No}
##             38) AGE_bin == {Teenager, Gen X, 62 and over}; criterion = 0.981, statistic = 18.918
##               39)*  weights = 287
##             38) AGE_bin == {Young Adult, Adult}
##               40)*  weights = 396
##     3) MVR_PTS > 6
##       41)*  weights = 187
##   2) JOB == {, Doctor, Lawyer, Manager, Professional}
##     42) DO_KIDS_DRIVE == {0}; criterion = 1, statistic = 190.844
##       43) CLM_FREQ <= 0; criterion = 1, statistic = 68.187
##         44) MSTATUS == {z_No}; criterion = 1, statistic = 48.072
##           45) CAR_TYPE == {Panel Truck, Sports Car, Van}; criterion = 1, statistic = 31.593
##             46)*  weights = 221
##           45) CAR_TYPE == {Minivan, Pickup, z_SUV}
##             47) AGE_bin == {Gen X}; criterion = 0.982, statistic = 16.728
##               48) SQRT_TRAVTIME <= 7.211103; criterion = 0.951, statistic = 9.399
##                 49) CAR_USE == {Commercial}; criterion = 0.981, statistic = 16.336
##                   50)*  weights = 31
##                 49) CAR_USE == {Private}
##                   51)*  weights = 260
##               48) SQRT_TRAVTIME > 7.211103
##                 52)*  weights = 30
##             47) AGE_bin == {Young Adult, Adult, 62 and over}
##               53) SQRT_BLUEBOOK <= 119.708; criterion = 0.992, statistic = 12.89
##                 54)*  weights = 76
##               53) SQRT_BLUEBOOK > 119.708
##                 55)*  weights = 96
##         44) MSTATUS == {Yes}
##           56) CAR_TYPE == {Minivan}; criterion = 0.977, statistic = 20.48
##             57)*  weights = 401
##           56) CAR_TYPE == {Panel Truck, Pickup, Sports Car, Van, z_SUV}
##             58) REVOKED == {Yes}; criterion = 0.996, statistic = 18.124
##               59)*  weights = 93
##             58) REVOKED == {No}
##               60) AGE_bin == {Gen X}; criterion = 0.997, statistic = 22.761
##                 61)*  weights = 489
##               60) AGE_bin == {Teenager, Young Adult, Adult, 62 and over}
##                 62) AGE_bin == {Adult}; criterion = 0.959, statistic = 15.005
##                   63)*  weights = 197
##                 62) AGE_bin == {Teenager, Young Adult, 62 and over}
##                   64)*  weights = 40
##       43) CLM_FREQ > 0
##         65) JOB == {, Lawyer, Professional}; criterion = 1, statistic = 40.423
```

```
##           66) HOME_OWNER == {0}; criterion = 0.998, statistic = 15.877
##             67)*  weights = 231
##           66) HOME_OWNER == {1}
##             68) CAR_USE == {Private}; criterion = 0.96, statistic = 12.851
##               69)*  weights = 352
##             68) CAR_USE == {Commercial}
##               70)*  weights = 225
##         65) JOB == {Doctor, Manager}
##           71)*  weights = 363
##     42) DO_KIDS_DRIVE == {1}
##       72) CAR_TYPE == {Panel Truck, Pickup, Sports Car, Van, z_SUV}; criterion = 0.998, statistic = 1
##         73) CLM_FREQ <= 1; criterion = 0.974, statistic = 10.609
##           74) HOME_OWNER == {1}; criterion = 0.985, statistic = 11.583
##             75)*  weights = 106
##           74) HOME_OWNER == {0}
##             76)*  weights = 32
##         73) CLM_FREQ > 1
##           77) SQRT_TRAVTIME <= 5.196152; criterion = 0.996, statistic = 14.066
##             78)*  weights = 34
##           77) SQRT_TRAVTIME > 5.196152
##             79)*  weights = 64
##       72) CAR_TYPE == {Minivan}
##         80)*  weights = 85
## 1) URBANICITY == {Rural}
##   81) CLM_FREQ <= 0; criterion = 1, statistic = 91.122
##     82) PARENT1 == {Yes}; criterion = 1, statistic = 23.571
##       83) YOJ <= 7; criterion = 0.988, statistic = 15.088
##         84)*  weights = 30
##       83) YOJ > 7
##         85)*  weights = 154
##     82) PARENT1 == {No}
##       86)*  weights = 1276
##   81) CLM_FREQ > 0
##     87)*  weights = 209
```

```
plot(tree)
```



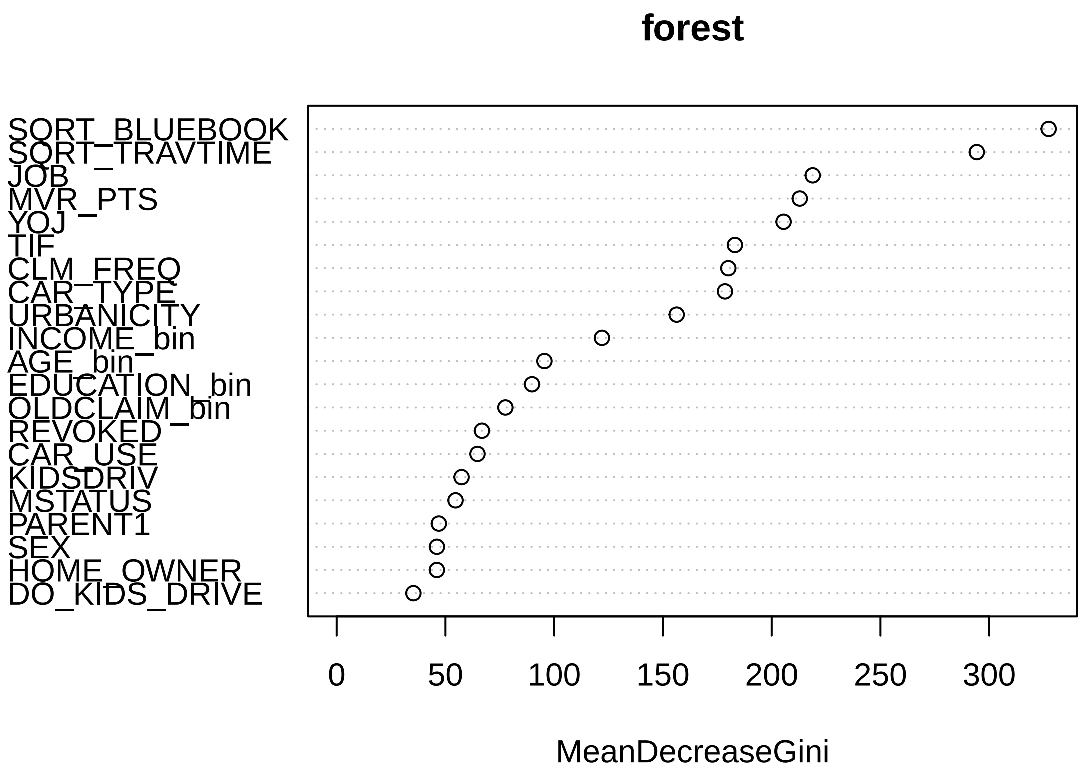

```r
# Confusion Matrix for Model 2
treePredict = predict(tree, type = "response")
CM2 = confusionMatrix(treePredict, data$TARGET_FLAG)

### Binary Response Model 3: Decision Tree with Bagging
tree_bagging = bagging(TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + AGE_FLAG + SEX +
                     MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER +
                     CLM_FREQ + REVOKED + MVR_PTS + URBANICITY + DO_KIDS_DRIVE +
                     HOME_OWNER + SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK +
                     OLDCLAIM_bin + INCOME_bin + AGE_bin + EDUCATION_bin
                    ,data = data, nbagg = 100, coob = TRUE, control = rpart.control(minsplit = 2, cp = 0))

print(tree_bagging)
```

```
##
## Bagging classification trees with 100 bootstrap replications
##
## Call: bagging.data.frame(formula = TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 +
##     AGE_FLAG + SEX + MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE +
##     HOME_OWNER + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY +
##     DO_KIDS_DRIVE + HOME_OWNER + SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK +
##     OLDCLAIM_bin + INCOME_bin + AGE_bin + EDUCATION_bin, data = data,
##     nbagg = 100, coob = TRUE, control = rpart.control(minsplit = 2,
##         cp = 0))
##
## Out-of-bag estimate of misclassification error:  0.2279
```

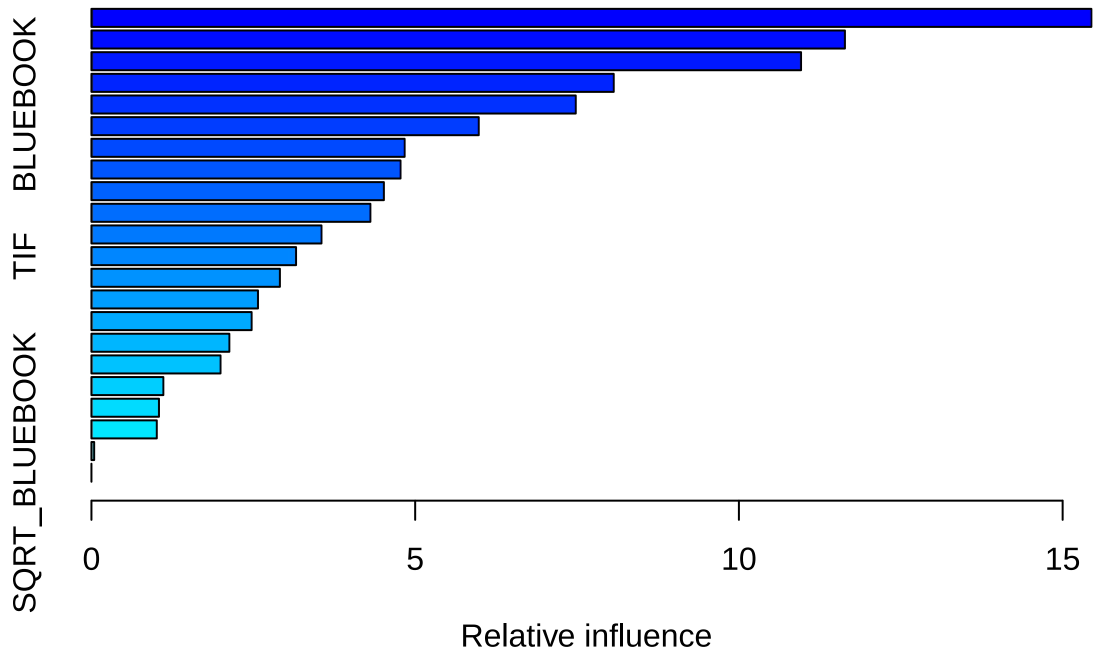
```r
# Confusion Matrix for Model 3
tree_baggingPredict = predict(tree_bagging, type = "class")
CM3 = confusionMatrix(tree_baggingPredict, data$TARGET_FLAG)

### Binary Response Model 4: Random Forests with Bagging
forest = randomForest(TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + SEX + DO_KIDS_DRIVE +
                        MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER +
                        CLM_FREQ + REVOKED + MVR_PTS + URBANICITY +
                        HOME_OWNER + SQRT_TRAVTIME + SQRT_BLUEBOOK +
                        OLDCLAIM_bin + INCOME_bin + AGE_bin + EDUCATION_bin
                      ,data = data, ntree=150, mtry = 3)

print(forest)
```

```
##
## Call:
##  randomForest(formula = TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 +      SEX + DO_KIDS_DRIVE + MSTATUS +
##                Type of random forest: classification
##                      Number of trees: 150
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 21.26%
## Confusion matrix:
##      0   1 class.error
## 0 5656 352  0.05858855
## 1 1383 770  0.64235950
```

```r
varImpPlot(forest)
```




```
# Confusion Matrix for Model 4
forestPredict = predict(forest, type = "class")
CM4 = confusionMatrix(forestPredict, data$TARGET_FLAG)

### Binary Response Model 5: Decision Tree with Boosting

tree_boost = gbm(TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + SEX +
               MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER +
               CLM_FREQ + REVOKED + MVR_PTS + URBANICITY + DO_KIDS_DRIVE +
               HOME_OWNER + SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK +
               OLDCLAIM_bin + INCOME_bin + AGE_bin + EDUCATION_bin
              ,data = data, n.trees = 500, distribution = 'gaussian',
              cv.folds = 5, shrinkage = .1)

summary(tree_boost)
```




```
##                           var     rel.inf
## URBANICITY       URBANICITY 15.44372716
## JOB                     JOB 11.63743113
## CLM_FREQ           CLM_FREQ 10.95971479
## MVR_PTS             MVR_PTS  8.06633483
## BLUEBOOK           BLUEBOOK  7.47972068
## CAR_TYPE           CAR_TYPE  5.98133343
## REVOKED             REVOKED  4.83701097
## CAR_USE             CAR_USE  4.77333638
## PARENT1             PARENT1  4.51653496
## SQRT_TRAVTIME SQRT_TRAVTIME  4.30819319
## HOME_OWNER       HOME_OWNER  3.55269464
## TIF                     TIF  3.16009071
## AGE_bin             AGE_bin  2.91060637
## INCOME_bin       INCOME_bin  2.57220345
## EDUCATION_bin EDUCATION_bin  2.47375490
## KIDSDRIV           KIDSDRIV  2.12971078
## MSTATUS             MSTATUS  1.99338314
## YOJ                     YOJ  1.11026625
## OLDCLAIM_bin   OLDCLAIM_bin  1.04273244
## DO_KIDS_DRIVE DO_KIDS_DRIVE  1.00924023
## SEX                     SEX  0.04197958
## SQRT_BLUEBOOK SQRT_BLUEBOOK  0.00000000
```

```
print(tree_boost)
```

```
## gbm(formula = TARGET_FLAG ~ KIDSDRIV + YOJ + PARENT1 + SEX +
```

```
##     MSTATUS + JOB + CAR_USE + TIF + CAR_TYPE + HOME_OWNER + CLM_FREQ +
##     REVOKED + MVR_PTS + URBANICITY + DO_KIDS_DRIVE + HOME_OWNER +
##     SQRT_TRAVTIME + BLUEBOOK + SQRT_BLUEBOOK + OLDCLAIM_bin +
##     INCOME_bin + AGE_bin + EDUCATION_bin, distribution = "gaussian",
##     data = data, n.trees = 500, shrinkage = 0.1, cv.folds = 5)
## A gradient boosted model with gaussian loss function.
## 500 iterations were performed.
## The best cross-validation iteration was 281.
## There were 22 predictors of which 20 had non-zero influence.
```

```
tree_boostPredict = predict.gbm(tree_boost, type = "response",n.trees = 500)
```

Part 5: Model Evaluation

The portion of the script is used to compare the results of the five models developed above. The following evaluation criteria are used for model evaluation: 1. Confusion Matrix 2. KS Statistic 3. AUC/ROC Curve

```
# ks statistic
ks_stat(actuals=data$TARGET_FLAG, predictedScores=lrPredict)
ks_stat(actuals=data$TARGET_FLAG, predictedScores=treePredict)
ks_stat(actuals=data$TARGET_FLAG, predictedScores=tree_baggingPredict)
ks_stat(actuals=data$TARGET_FLAG, predictedScores=forestPredict)
ks_stat(actuals=data$TARGET_FLAG, predictedScores=tree_boostPredict)



# Compare Confusion Matrices

df = data.frame(row.names = c("Accuracy", "Sensitivity" ,"Specificity", "Pos Pred Value", "Neg Pred Valu

df$CM1 = c(CM1$overall[1], CM1$byClass[1:11])
df$CM2 = c(CM2$overall[1], CM2$byClass[1:11])
df$CM3 = c(CM3$overall[1], CM3$byClass[1:11])
df$CM4 = c(CM4$overall[1], CM4$byClass[1:11])



df
```

Part 6: Model Selection and Testing Prediction

Based on the results of the model evaluation criteria above, I am select the fourth model that uses a Random Forest with bagging applied to apply to the test dataset.

```
# Apply the prediction to the testing dataset
testPredict = predict(forest, newdata = test, type = "class")

claims = sum(as.numeric(testPredict[testPredict==1]))

print("The prediction on the testing dataset indicates the following number of claims out of 2,468 obse
```

```
## [1] "The prediction on the testing dataset indicates the following number of claims out of 2,468 obse
```

```
print(claims)
```

```
## [1] 626
```