library(pastecs)

library(tidyverse)

# https://www.rdocumentation.org/packages/naniar/versions/0.6.1

library(naniar)

library(reshape2)

library(ggplot2)

library(readr)

library(dplyr)

# Used for skewness

library(moments)

# Log Scale

library(scales)

# Correlation corrplot

# https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html

library(corrplot)

# https://www.rdocumentation.org/packages/car/versions/3.0-11/topics/vif

library(car)

library(caret)

library(leaps)

library(pROC)

# To help prevent scientific notation for viewed values

options(scipen=100)

# To set the number of decimal places

options(digits=4)

# From http://www.cookbook-r.com/Graphs/Multiple\_graphs\_on\_one\_page\_(ggplot2)/

# Multiple plot function

#

# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)

# - cols: Number of columns in layout

# - layout: A matrix specifying the layout. If present, 'cols' is ignored.

#

# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),

# then plot 1 will go in the upper left, 2 will go in the upper right, and

# 3 will go all the way across the bottom.

#

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {

library(grid)

# Make a list from the ... arguments and plotlist

plots <- c(list(...), plotlist)

numPlots = length(plots)

# If layout is NULL, then use 'cols' to determine layout

if (is.null(layout)) {

# Make the panel

# ncol: Number of columns of plots

# nrow: Number of rows needed, calculated from # of cols

layout <- matrix(seq(1, cols \* ceiling(numPlots/cols)),

ncol = cols, nrow = ceiling(numPlots/cols))

}

if (numPlots==1) {

print(plots[[1]])

} else {

# Set up the page

grid.newpage()

pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

# Make each plot, in the correct location

for (i in 1:numPlots) {

# Get the i,j matrix positions of the regions that contain this subplot

matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,

layout.pos.col = matchidx$col))

}

}

}

# Reading in the csv files

trainData <- read\_csv("https://github.com/logicalschema/Fall-2021/raw/main/DATA621/hw4/insurance\_training\_data.csv.gz")

evalData <- read.csv("https://raw.githubusercontent.com/logicalschema/Fall-2021/main/DATA621/hw4/insurance-evaluation-data.csv")

# Remove the INDEX column

trainData = subset(trainData, select = -c(INDEX) )

evalData = subset(evalData, select = -c(INDEX) )

head(trainData)

uniqueValues <- lapply(trainData[, c("MSTATUS","SEX","EDUCATION","JOB","CAR\_USE","CAR\_TYPE","RED\_CAR","REVOKED","URBANICITY")], unique)

print(uniqueValues)

strip\_dollars <- function(x){

x <- as.character(x)

x <- gsub(",", "", x)

x <- gsub("\\$", "", x)

as.numeric(x)

}

fix\_data\_types <- function(cleanMe){

cleanMe %>%

rowwise() %>%

mutate(INCOME = strip\_dollars(INCOME),

HOME\_VAL = strip\_dollars(HOME\_VAL),

BLUEBOOK = strip\_dollars(BLUEBOOK),

OLDCLAIM = strip\_dollars(OLDCLAIM)) %>%

ungroup()

}

clean\_values <- function(cleanMe){

cleanMe %>%

rowwise() %>%

mutate(CAR\_AGE = ifelse(CAR\_AGE < 0, NA, CAR\_AGE)) %>%

mutate(MSTATUS = ifelse(MSTATUS == 'z\_No', 0, 1)) %>%

mutate(RED\_CAR = ifelse(RED\_CAR == 'no', 0, 1)) %>%

mutate(REVOKED = ifelse(REVOKED == 'No', 0, 1)) %>%

mutate(URBANICITY = ifelse(URBANICITY == 'z\_Highly Rural/ Rural', 0, 1)) %>%

ungroup()

}

trainData <- trainData %>%

fix\_data\_types() %>%

clean\_values()

evalData <- evalData %>%

fix\_data\_types() %>%

clean\_values()

head(trainData)

summary(trainData)

# Replace YOJ, INCOME, AGE, HOME\_VAL, and CAR\_AGE for train and eval Data

trainData$YOJ[is.na(trainData$YOJ)] <- mean(trainData$YOJ, na.rm=TRUE)

trainData$INCOME[is.na(trainData$INCOME)] <- mean(trainData$INCOME, na.rm=TRUE)

trainData$AGE[is.na(trainData$AGE)] <- mean(trainData$AGE, na.rm=TRUE)

trainData$HOME\_VAL[is.na(trainData$HOME\_VAL)] <- mean(trainData$HOME\_VAL, na.rm=TRUE)

trainData$CAR\_AGE[is.na(trainData$CAR\_AGE)] <- mean(trainData$CAR\_AGE, na.rm=TRUE)

evalData$YOJ[is.na(evalData$YOJ)] <- mean(evalData$YOJ, na.rm=TRUE)

evalData$INCOME[is.na(evalData$INCOME)] <- mean(evalData$INCOME, na.rm=TRUE)

evalData$AGE[is.na(evalData$AGE)] <- mean(evalData$AGE, na.rm=TRUE)

evalData$HOME\_VAL[is.na(evalData$HOME\_VAL)] <- mean(evalData$HOME\_VAL, na.rm=TRUE)

evalData$CAR\_AGE[is.na(evalData$CAR\_AGE)] <- mean(evalData$CAR\_AGE, na.rm=TRUE)

# Log transformations for the variables

trainData$logTARGET\_AMT <- log1p(trainData$TARGET\_AMT)

hist(trainData$logTARGET\_AMT)

trainData$logINCOME <- log1p(trainData$INCOME)

hist(trainData$logINCOME)

trainData$logHOME\_VAL <- log1p(trainData$HOME\_VAL)

hist(trainData$logHOME\_VAL)

# Perform the transformations on the evaluation data

evalData$logTARGET\_AMT <- log1p(evalData$TARGET\_AMT)

evalData$logINCOME <- log1p(evalData$INCOME)

evalData$logHOME\_VAL <- log1p(evalData$HOME\_VAL)

par(mfrow = c(3, 3))

plotData <- melt(trainData)

ggplot(plotData, aes(x= value)) +

theme(panel.border = element\_blank(), panel.background = element\_blank(),

panel.grid.major = element\_blank(), panel.grid.minor = element\_blank()) +

geom\_density(fill='dodgerblue') + facet\_wrap(~variable, scales = 'free')

p1 <- ggplot(trainData) +

aes(y =logTARGET\_AMT, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p2 <- ggplot(trainData) +

aes(y =KIDSDRIV, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p3 <- ggplot(trainData) +

aes(y =AGE, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p4 <- ggplot(trainData) +

aes(y =HOMEKIDS, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p5 <- ggplot(trainData) +

aes(y =YOJ, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p6 <- ggplot(trainData) +

aes(y =logINCOME, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p7 <- ggplot(trainData) +

aes(y =PARENT1, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p8 <- ggplot(trainData) +

aes(y =logHOME\_VAL, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p9 <- ggplot(trainData) +

aes(y =MSTATUS, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p10 <- ggplot(trainData) +

aes(y =SEX, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p11 <- ggplot(trainData) +

aes(y =EDUCATION, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p12 <- ggplot(trainData) +

aes(y =JOB, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p13 <- ggplot(trainData) +

aes(y =TRAVTIME, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p14 <- ggplot(trainData) +

aes(y =CAR\_USE, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p15 <- ggplot(trainData) +

aes(y =RED\_CAR, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p16 <- ggplot(trainData) +

aes(y =OLDCLAIM, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p17 <- ggplot(trainData) +

aes(y =CLM\_FREQ, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p18 <- ggplot(trainData) +

aes(y =REVOKED, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p19 <- ggplot(trainData) +

aes(y =MVR\_PTS, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p20 <- ggplot(trainData) +

aes(y =CAR\_AGE, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

p21 <- ggplot(trainData) +

aes(y =URBANICITY, x = as.factor(TARGET\_FLAG)) +

geom\_boxplot(fill="dodgerblue", alpha=0.3) +

theme\_minimal() + xlab("TARGET\_FLAG")

multiplot(p1, p2, p3, p4, p5, p6, cols=3)

multiplot(p7, p8, p9, p10, p11, p12, cols=3)

multiplot(p13, p14, p15, p16, p17, p18, cols=3)

multiplot(p19, p20, p21, cols=3)

# Create a target variable with the numeric value of TARGET\_FLAG

trainData$target <- as.numeric(trainData$TARGET\_FLAG)

evalData$target <- as.numeric(evalData$TARGET\_FLAG)

trainData %>%

complete.cases() %>%

trainData[., c("target", "logTARGET\_AMT", "KIDSDRIV", "AGE", "HOMEKIDS", "YOJ", "logINCOME", "logHOME\_VAL", "MSTATUS", "TRAVTIME", "BLUEBOOK", "TIF", "RED\_CAR", "OLDCLAIM", "CLM\_FREQ", "REVOKED", "MVR\_PTS", "CAR\_AGE", "URBANICITY" )] %>%

cor() %>%

corrplot(method = "number")

cor(trainData[sapply(trainData, is.numeric)])

skewness(trainData[sapply(trainData, is.numeric)])

set.seed(11192021)

trainIndex <- createDataPartition(trainData$target, p = 0.8,

list = FALSE)

training <- trainData[trainIndex, ]

evaluation <- trainData[-trainIndex, ]

head(training)

head(evaluation)

reduced\_model <- glm(target ~ SEX + logINCOME + logHOME\_VAL, data=training, family = 'binomial')

summary(reduced\_model)

predictionProbabilities <- predict(reduced\_model, evaluation, type="response")

actuals\_predictions <- data.frame(cbind(actual=evaluation$target, predictedProb=predictionProbabilities))

actuals\_predictions$prediction <- ifelse(actuals\_predictions$predictedProb < 0.5, 0, 1)

head(actuals\_predictions)

confusionMatrix(as.factor(actuals\_predictions$prediction), as.factor(actuals\_predictions$actual), positive = '1')

proc = roc(actuals\_predictions$actual, actuals\_predictions$prediction)

plot(proc)

print(proc$auc)

reduced\_model <- glm(target ~ SEX + AGE + RED\_CAR + URBANICITY + CAR\_TYPE + YOJ, data=training, family = 'binomial')

summary(reduced\_model)

predictionProbabilities <- predict(reduced\_model, evaluation, type="response")

actuals\_predictions <- data.frame(cbind(actual=evaluation$target, predictedProb=predictionProbabilities))

actuals\_predictions$prediction <- ifelse(actuals\_predictions$predictedProb < 0.5, 0, 1)

head(actuals\_predictions)

confusionMatrix(as.factor(actuals\_predictions$prediction), as.factor(actuals\_predictions$actual), positive = '1')

proc = roc(actuals\_predictions$actual, actuals\_predictions$prediction)

plot(proc)

print(proc$auc)

reduced\_model <- glm(target ~ SEX + AGE + RED\_CAR + URBANICITY + CAR\_TYPE + EDUCATION + KIDSDRIV + MSTATUS + MVR\_PTS + CAR\_USE + JOB + TIF + YOJ, data=training, family = 'binomial')

summary(reduced\_model)

predictionProbabilities <- predict(reduced\_model, evaluation, type="response")

actuals\_predictions <- data.frame(cbind(actual=evaluation$target, predictedProb=predictionProbabilities))

actuals\_predictions$prediction <- ifelse(actuals\_predictions$predictedProb < 0.5, 0, 1)

head(actuals\_predictions)

confusionMatrix(as.factor(actuals\_predictions$prediction), as.factor(actuals\_predictions$actual), positive = '1')

proc = roc(actuals\_predictions$actual, actuals\_predictions$prediction)

plot(proc)

print(proc$auc)

reduced\_model <- glm(target ~ AGE + SEX + MSTATUS + CLM\_FREQ + REVOKED + MVR\_PTS + TRAVTIME + CAR\_USE, data=training, family = 'binomial')

summary(reduced\_model)

predictionProbabilities <- predict(reduced\_model, evaluation, type="response")

actuals\_predictions <- data.frame(cbind(actual=evaluation$target, predictedProb=predictionProbabilities))

actuals\_predictions$prediction <- ifelse(actuals\_predictions$predictedProb < 0.5, 0, 1)

head(actuals\_predictions)

confusionMatrix(as.factor(actuals\_predictions$prediction), as.factor(actuals\_predictions$actual), positive = '1')

proc = roc(actuals\_predictions$actual, actuals\_predictions$prediction)

plot(proc)

print(proc$auc)

baseline\_lm <- lm(TARGET\_AMT ~ BLUEBOOK + CAR\_AGE + CAR\_TYPE + logINCOME, training)

summary(baseline\_lm)

data.frame(yhat = predict(baseline\_lm, evaluation), actual = evaluation$TARGET\_AMT) %>%

ggplot(., aes(actual, yhat)) +

geom\_point()

second\_lm <- lm(TARGET\_AMT ~ EDUCATION + BLUEBOOK + CAR\_AGE + CAR\_TYPE + TRAVTIME + logINCOME + logHOME\_VAL, training)

summary(second\_lm)

data.frame(yhat = predict(second\_lm, evaluation), actual = evaluation$TARGET\_AMT) %>%

ggplot(., aes(actual, yhat)) +

geom\_point()

third\_lm <- lm(TARGET\_AMT ~ BLUEBOOK + CAR\_AGE + TRAVTIME, training)

summary(third\_lm)

data.frame(yhat = predict(third\_lm, evaluation), actual = evaluation$TARGET\_AMT) %>%

ggplot(., aes(actual, yhat)) +

geom\_point()

reduced\_model <- glm(target ~ SEX + AGE + RED\_CAR + URBANICITY + CAR\_TYPE + EDUCATION + KIDSDRIV + MSTATUS + MVR\_PTS + CAR\_USE + JOB + TIF + YOJ, data=training, family = 'binomial')

predictionProbabilities <- predict(reduced\_model, evaluation, type="response")

actuals\_predictions <- data.frame(cbind(actual=evalData$target, predictedProb=predictionProbabilities))

actuals\_predictions$prediction <- ifelse(actuals\_predictions$predictedProb < 0.5, 0, 1)

head(actuals\_predictions)

df <- data.frame(prediction = predict(second\_lm, evalData), actual = evalData$TARGET\_AMT)

head(df)