############################## R CODE APPENDIX ###################

##### THIRD PAPER ASSIGNMENT (974) ########

setwd("~/Desktop/PhD work/Year 3/Spring 2017/STAT 974 - Modern Regression/Assignments/Assignment 3")

load("~/Desktop/PhD work/Year 3/Spring 2017/STAT 974 - Modern Regression/Assignments/Assignment 3/admissions974.rdata")

summary(admissions)

str(admissions)

#detach("package:plyr", unload=TRUE)

#detach("package:dplyr", unload=TRUE)

#install.packages("MAPLES")

library(plyr) #ALWAYS load plyr BEFORE dplyr

library(dplyr)

library(ggplot2)

library(MAPLES)

library(randomForest)

#rm(admissions2)

admissions2 <- admissions #creating a duplicate dataset to work with

str(admissions2)

dim(admissions)

admissions2$admit <- as.factor(admissions2$admit) #converting the admit variable to a factor

admissions2$anglo <- as.factor(admissions2$anglo) #converting the anglo variable to a factor

admissions2$asian <- as.factor(admissions2$asian) #converting the asian variable to a factor

admissions2$black <- as.factor(admissions2$black) #converting the black variable to a factor

admissions2$sex <- as.factor(admissions2$sex) #converting the sex variable to a factor

str(admissions2)

summary(admissions2)

#View(admissions2)

#RELABELING ADMIT

summary(admissions2$admit)

admissions2$admit <- revalue(admissions2$admit, c("0" ="Reject", "1" ="Admit")) #renaming admit factor levels

## CLEANING WHITE

admissions2$anglo[admissions2$anglo== "white"] <- 1 #changing white to 1

admissions2$anglo[admissions2$anglo== 10 ] <- 1 #changing 10 to 1

admissions2$anglo <- droplevels(admissions2$anglo) #removing empty factors

NA\_anglo <- which(is.na(admissions2$anglo)) #identifying the NAs

admissions2 <- admissions2[-NA\_anglo,] #removing the NAs

summary(admissions2$anglo)

length(NA\_anglo)/length(admissions2$anglo) # percentage of missing data

# CLEANING ASIAN

NA\_asian <- which(is.na(admissions2$asian)) #identifying the NAs

admissions2 <- admissions2[-NA\_asian,] #removing the NAs

summary(admissions2$asian)

length(NA\_asian)/length(admissions2$asian)

#CLEANING INCOME

NA\_income <- which(is.na(admissions2$income)) #identifying the NAs

admissions2 <- admissions2[-NA\_income,] #removing the NAs

summary(admissions2)

sort(unique(admissions2$income)) #may want to do something about the very low observations

sort(admissions2$income)

length(NA\_income)/length(admissions2$income)

#CLEANING SEX

NA\_sex <- which(is.na(admissions2$sex)) #identifying the NAs

admissions2 <- admissions2[-NA\_sex,] #removing the NAs

admissions2$sex <- revalue(admissions2$sex, c("0" ="Female", "1" ="Male")) #renaming sex factor levels

summary(admissions2)

dim(admissions2)

(length(NA\_sex)/length(admissions2$sex))\*100

#CLEANING GPA

sort(unique(admissions2$gpa.wtd))

head(sort(admissions2$gpa.wtd))

admissions2$gpa.wtd[admissions2$gpa.wtd== -3.2 ] <- 3.2 #flipping the sign on -3.2

summary(admissions2$gpa.wtd)

str(admissions2)

NA\_gpa.wtd <- which(admissions2$gpa == 0) #identifying the individuals with 0.0 as GPAs

admissions2 <- admissions2[-NA\_gpa.wtd,]

summary(admissions2)

dim(admissions2)

length(NA\_gpa.wtd)/length(admissions2$gpa.wtd)

#CLEANING SAT1.VERB (POSSIBLE RANGE: 200 - 800)

summary(admissions2$sati.verb)

sort(unique(admissions2$sati.verb))

NA\_sati.verb <- which(admissions2$sati.verb == 0) #there are 322 zeros (the same observations have zeros for both math and verbal), can't drop them because then you'd loose a lot of data, including all of the black students in the sample

admissions2 <- admissions2[-NA\_sati.verb,]

summary(admissions2$sati.verb)

length(NA\_sati.verb)/length(admissions2$sati.verb)

#CLEANING SAT1.MATH (POSSIBLE RANGE: 200 - 800)

sort(unique(admissions2$sati.math))

length(which(admissions2$sati.math == 0)) #there are 322 zeros (the same observations have zeros for both math and verbal), can't drop them because then you'd loose a lot of data, including all of the black students in the sample

summary(admissions2$sati.math)

dim(admissions2)

summary(admissions2)

dim(admissions2)

#CREATING A RACE VARIABLE

admissions2$race <- rep(3, length(admissions2$anglo))

white <- which(admissions2$anglo == 1)

asian <- which(admissions2$asian == 1)

black <- which(admissions2$black == 1)

length(admissions2$race)

admissions2$race[white] <- 0

admissions2$race[asian] <- 1

admissions2$race[black] <- 2

admissions2$race <- as.factor(admissions2$race)

admissions2$race <- revalue(admissions2$race, c("0" ="White", "1" ="Asian", "2" ="Black", "3" ="Other")) #renaming race factor levels

admissions2$race <- relevel(admissions2$race, "Other") #changing reference category to Other

summary(admissions2$anglo)

summary(admissions2$black)

summary(admissions2$asian)

summary(admissions2)

sd(admissions2$income)

summary(admissions2)

class(admissions2$race)

names(admissions2)

summary(admissions2)

summary(admissions2)

##RACE CODING SCHEME#

#0 = WHITE

#1 = ASIAN

#2 = BLACK

#3 = OTHER

#CREATING GPA SQUARED VARIABLE

admissions2$gpa.wtd.sq <- (admissions2$gpa.wtd)^2

qqnorm(admissions2$gpa.wtd.sq)

qqline(admissions2$gpa.wtd.sq) #the square of GPA is normally distributed.

summary(admissions2)

#CREATING SAT COMPOSITE VARIABLE

admissions2 <- mutate(admissions2, satcomp = sati.verb + sati.math)

names(admissions2)

#CHECKING NORMALITY OF CONTINUOUS VARIABLES

#GPA

qqnorm(admissions2$gpa.wtd)

qqline(admissions2$gpa.wtd) #GPA not normally distributed

hist(admissions2$gpa.wtd)

p1 <- ggplot(admissions2, aes(x=gpa.wtd)) + geom\_histogram() +

ggtitle("Distribution of Weighted GPA") + theme(plot.title=element\_text(hjust=0.5))

#INCOME

qqnorm(admissions2$income)

qqline(admissions2$income) #income not normally distributed

hist(admissions2$income)

p2 <- ggplot(admissions2, aes(x=income)) + geom\_histogram(bins = 10) +

ggtitle("Distribution of Income") + theme(plot.title=element\_text(hjust=0.5))

#SAT1.VERBAL

qqnorm(admissions2$sati.verb)

qqline(admissions2$sati.verb) #SAT1 VERBAL APPROXIMATELY NORMALLY DISTRBUTED

hist(admissions2$sati.verb)

p3 <- ggplot(admissions2, aes(x=sati.verb)) + geom\_histogram(bins = 20) +

ggtitle("Distribution of SAT1 Verbal") + theme(plot.title=element\_text(hjust=0.5))

#SAT1.MATH

qqnorm(admissions2$sati.math)

qqline(admissions2$sati.math) #SAT1 MATH LEFT SKEWED

hist(admissions2$sati.math)

p4 <- ggplot(admissions2, aes(x=sati.math)) + geom\_histogram(bins = 30) +

ggtitle("Distribution of SAT1 Math") + theme(plot.title=element\_text(hjust=0.5))

#SAT1.COMP

qqnorm(admissions2$sat.comp)

qqline(admissions2$sat.comp)

ggplot(admissions2, aes(x=sat.comp)) + geom\_histogram(bins = 30) +

ggtitle("Distribution of SAT1 Math") + theme(plot.title=element\_text(hjust=0.5))

#DESIGNATING THE MULTI-PLOT FUNCTION

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))

}

}

}

multiplot(p1, p2, p3, p4, cols=2) #univariate plots for continuous variables

#### BIVARIATE DESCRIPTIVE STATISTICS ####

#Admit X GPA Wtd

bip1 <- ggplot(admissions2, aes(factor(admit), gpa.wtd)) + geom\_boxplot() + xlab("Admit") + ylab("Weighted GPA") +

ggtitle("Admission by GPA") + theme(plot.title=element\_text(hjust=0.5))

#Admit X SAT1.verb:

bip2 <- ggplot(admissions2, aes(factor(admit), sati.verb)) + geom\_boxplot() + xlab("Admit") + ylab("SAT 1 Verbal") +

ggtitle("Admission by SAT 1 Verbal") + theme(plot.title=element\_text(hjust=0.5))

#Admit X SAT1.math:

bip3 <- ggplot(admissions2, aes(factor(admit), sati.math)) + geom\_boxplot() + xlab("Admit") + ylab("SAT 1 Math") +

ggtitle("Admission by SAT 1 Math") + theme(plot.title=element\_text(hjust=0.5))

#Admit X Income

bip4 <- ggplot(admissions2, aes(factor(admit), income)) + geom\_boxplot() + xlab("Admit") + ylab("Income") +

ggtitle("Admission by Income") + theme(plot.title=element\_text(hjust=0.5))

#Admit X Composite SAT Score

ggplot(admissions2, aes(factor(admit), sat.comp)) + geom\_boxplot() + xlab("Admit") + ylab("SAT Composite Score") +

ggtitle("Admission by Composite SAT Score") + theme(plot.title=element\_text(hjust=0.5))

multiplot(bip1, bip2, bip3, bip4, cols=2) #univariate plots for continuous variables

#Admit X Sex

tabx(admissions2$sex, admissions2$admit, prow = TRUE)

#Admit X Race

tabx(admissions2$race, admissions2$admit, prow = TRUE)

pairs(admissions2[,-c(1:4,9:11)],panel = panel.smooth) #pairwise scatterplots of continuous variables

admissions2[,-c(1:4,9:11)] #taking out the categorical variables (just continuous variables)

names(admissions2)

### SPLITTING THE DATA ###

N <- length(admissions2$admit)

set.seed(10)

index.train <- sample(N, (N/2))

data.train <- admissions2[index.train,] # Set the N/2 randomly chosen subjects as a training data

data.test <- admissions2[-index.train,] # The remaining subjects will be reserved for testing purposes.

nrow(data.test)

(summary(data.train$admit)[2])\*2/3 #two thirds of the sample size in the smaller outcome (p. 223)

summary(admissions2$admit) #reject, admit

#target cost ratio is 4 to 1.

#### RANDOM FORESTS ####

### TUNING COST RATIO IN TRAINING DATA###

#start with 50/50

rf0 <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.train, importance=T, sampsize=c(623,623)) #increase the first number

print(rf0) #c(623,623) gives me a cost ratio of about 2 to 1

confuse <- rf0$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

rf1 <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.train, importance=T, sampsize=c(275,623)) #increase the first number

#c(275,623) gives me a cost ratio of about 5 (5.03) to 1

print(rf1) #get classification table from this output

names(rf1)

confuse <- rf1$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

rf2 <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.train, importance=T, sampsize=c(175,623)) #increase the first number

#c(175,623) gives me a cost ratio of about 10 (9.92) to 1

print(rf2) #get classification table from this output

names(rf2)

confuse <- rf2$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

rf3 <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.train, importance=T, sampsize=c(400,623)) #increase the first number

#c(400,623) gives me a cost ratio of about 3 (3.02) to 1

print(rf3) #get classification table from this output

names(rf3)

confuse <- rf3$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

rf4 <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.train, importance=T, sampsize=c(525,623)) #increase the first number

#c(525,623) gives me a cost ratio of about 2 (2.08) to 1

print(rf4) #get classification table from this output

names(rf4)

confuse <- rf4$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

par(mfrow=c(2,2))

names(rfalt)

rfalt$classes

rf5 <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.train, importance=T, sampsize=c(300,623)) #increase the first number

#c(300,623) gives me a cost ratio of about 4 (4.10) to 1

print(rf5) #get classification table from this output

names(rf5)

confuse <- rf5$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

dim(data.train)

############## FINAL MODEL RUN IN TEST DATA ##################

set.seed(10)

final.rf <- randomForest(admit ~ sex + race + income + gpa.wtd.sq + sati.verb + sati.math, data = data.test, importance=T, sampsize=c(300,623))

print(final.rf) #get classification table from this output

confuse <- final.rf$confusion #need to change this for each model

confuse[1,3] #Rejection Model Error (b/a+b)

confuse[2,3] #Acceptance Model Error (c/c+d)

confuse[2,1]/(confuse[1,1]+confuse[2,1]) #Rejection use error (c/a+b)

confuse[1,2]/(confuse[1,2]+confuse[2,2]) #Acceptance use error (b/b+d)

(confuse[1,2]+confuse[2,1])/(confuse[1,1]+confuse[2,1]+confuse[2,1]+confuse[2,2]) #Overall error (b+c)/(a+b+c+d)

dim(data.test)

#Variable Importance Plots

#Use Unstandardized Plots

par(mfrow=c(2,1))

varImpPlot(final.rf, type=1, scale = F, class="Admit", main="Forecasting Importance Plot for Admittance (Unstandardized)", col="blue",

cex = 1, pch = 19)

varImpPlot(final.rf, type = 1, scale = T, class= "Admit", main = "Forecasting Importance Plot for Admittance (Standardized)", col = "blue",

cex = 1, pch = 19)

varImpPlot(final.rf, type = 1, scale = F, class="Reject", main = "Forecasting Importance Plot for Rejection (Unstandardized)", col = "blue",

cex = 1, pch = 19)

varImpPlot(final.rf, type = 1, scale = T, class="Reject", main = "Forecasting Importance Plot for Rejection (Standardized)", col = "blue",

cex = 1, pch = 19)

par(mfrow=c(1,1))

# PARTIAL PLOTS

#SEX

part1 <- partialPlot(final.rf, pred.data= data.train, x.var = sex, rug = T, which.class = "Admit",

main = "Partial Dependence Plot for Admittance on Sex", xlab = "Sex", ylab = "Centered Log Odds of Admittance")

#RACE

part2 <- partialPlot(final.rf, pred.data = data.train, x.var = race, rug = T, which.class = "Admit",

main = "Partial Dependence Plot for Admittance on Race", xlab = "Race/Ethnicity", ylab = "Centered Log Odds of Admittance")

#GPA

part3 <- partialPlot(final.rf, pred.data = data.train, x.var = gpa.wtd.sq, rug = T, which.class = "Admit",

main = "Partial Dependence Plot for Admittance on Weighted GPA", xlab = "Weighted GPA (Squared)", ylab = "Centered Log Odds of Admittance")

par(mfrow=c(2,1))

scatter.smooth(part3$x,part3$y, span= 1/3, xlab = "Weighted GPA",

ylab = "Centered Log Odds of Admittance", main = "Partial Dependence Plot for Admittance on Weighted GPA",

col = "blue", pch = 19)

### COMPUTING PROPORTIONS ####

part3$ytimes2 <- (part3$y)\*2

part3$yexp <- exp(part3$ytimes2)

part3$yprop <- part3$yexp/(1+part3$yexp)

### GRAPHING PROPORTIONS

scatter.smooth(part3$x,part3$yprop, span= 1/3, xlab = "Weighted GPA",

ylab = "Proportion", main = "Partial Dependence Plot for Admittance on Weighted GPA",

col = "blue", pch = 19)

#SAT VERBAL

part4 <- partialPlot(final.rf, pred.data = admissions2, x.var = sati.verb, rug = T, which.class = "Admit",

main = "Partial Dependence Plot for Admittance on SAT Verbal", xlab = "SAT Verbal", ylab = "Centered Log Odds of Admittance")

scatter.smooth(part4$x,part4$y, span= 1/3, xlab = "SAT Verbal",

ylab = "Centered Log Odds of Admittance", main = "Partial Dependence Plot for Admittance on SAT Verbal",

col = "blue", pch = 19)

### COMPUTING PROPORTIONS ####

part4$ytimes2 <- (part4$y)\*2

part4$yexp <- exp(part4$ytimes2)

part4$yprop <- part4$yexp/(1+part4$yexp)

names(part4)

### GRAPHING PROPORTIONS

scatter.smooth(part4$x,part4$yprop, span= 1/3, xlab = "SAT Verbal",

ylab = "Proportion", main = "Partial Dependence Plot for Admittance on SAT Verbal",

col = "blue", pch = 19)

#SAT MATH

part5 <- partialPlot(final.rf, pred.data = admissions2, x.var = sati.math, rug = T, which.class = "Admit",

main = "Partial Dependence Plot for Admittance on SAT Math", xlab = "SAT Math", ylab = "Centered Log Odds of Admittance")

scatter.smooth(part5$x,part5$y, span= 1/3, xlab = "SAT Math",

ylab = "Centered Log Odds of Admittance", main = "Partial Dependence Plot for Admittance on SAT Math",

col = "blue", pch = 19)

### COMPUTING PROPORTIONS ####

part5$ytimes2 <- (part5$y)\*2

part5$yexp <- exp(part5$ytimes2)

part5$yprop <- part5$yexp/(1+part5$yexp)

names(part5)

### GRAPHING PROPORTIONS

scatter.smooth(part5$x,part5$yprop, span= 1/3, xlab = "SAT Math",

ylab = "Proportion", main = "Partial Dependence Plot for Admittance on SAT Math",

col = "blue", pch = 19)

#INCOME

part6 <- partialPlot(final.rf, pred.data = admissions2, x.var = income, rug = T, which.class = "Admit",

main = "Partial Dependence Plot for Admittance on Income", xlab = "Income", ylab = "Centered Log Odds of Admittance")

scatter.smooth(part6$x,part6$y, span= 1/3, xlab = "Income",

ylab = "Centered Log Odds of Admittance", main = "Partial Dependence Plot for Admittance on Income",

col = "blue", pch = 19)

### COMPUTING PROPORTIONS ####

part6$ytimes2 <- (part6$y)\*2

part6$yexp <- exp(part6$ytimes2)

part6$yprop <- part6$yexp/(1+part6$yexp)

names(part6)

### GRAPHING PROPORTIONS

scatter.smooth(part6$x,part6$yprop, span= 1/3, xlab = "Income",

ylab = "Proportion", main = "Partial Dependence Plot for Admittance on Income",

col = "blue", pch = 19)

names(part6)

part6$y