**Assignment 6.1**

library(ggplot2)

library(caret)

library(caretEnsemble)

library(ROSE)

library(mlbench)

library(DMwR)

library(rpart)

library(rattle)

library(rpart.plot)

library(RColorBrewer)

#Load the dataset

setwd("~/Desktop/bank")

mydata <- read.csv("bank.csv")

#Summary on dataset

summary(mydata)

str(mydata)

p\_age <- ggplot(mydata, aes(factor(y), age)) + geom\_boxplot(aes(fill = factor(y)))

p\_age

p\_balance <- ggplot(mydata, aes(factor(y), balance)) + geom\_boxplot(aes(fill = factor(y)))

p\_balance

p\_day <- ggplot(mydata, aes(factor(y), day)) + geom\_boxplot(aes(fill = factor(y)))

p\_day

p\_duration <- ggplot(mydata, aes(factor(y), duration)) + geom\_boxplot(aes(fill = factor(y)))

p\_duration

p\_campaign <- ggplot(mydata, aes(factor(y), campaign)) + geom\_boxplot(aes(fill = factor(y)))

p\_campaign

p\_pdays <- ggplot(mydata, aes(factor(y), pdays)) + geom\_boxplot(aes(fill = factor(y)))

p\_pdays

p\_previous <- ggplot(mydata, aes(factor(y), previous)) + geom\_boxplot(aes(fill = factor(y)))

p\_previous

#Generate dummy variables

for(level in unique(mydata$job)){

mydata[paste("job", level, sep = "\_")] <- ifelse(mydata$job == level, 1, 0)

}

for(level in unique(mydata$marital)){

mydata[paste("marital", level, sep = "\_")] <- ifelse(mydata$marital == level, 1, 0)

}

for(level in unique(mydata$education)){

mydata[paste("education", level, sep = "\_")] <- ifelse(mydata$education == level, 1, 0)

}

mydata$default\_yes <- ifelse(mydata$default == "yes", 1, 0)

mydata$housing\_yes <- ifelse(mydata$housing == "yes", 1, 0)

mydata$loan\_yes <- ifelse(mydata$loan == "yes", 1, 0)

for(level in unique(mydata$contact)){

mydata[paste("contact", level, sep = "\_")] <- ifelse(mydata$contact == level, 1, 0)

}

for(level in unique(mydata$month)){

mydata[paste("month", level, sep = "\_")] <- ifelse(mydata$month == level, 1, 0)

}

for(level in unique(mydata$poutcome)){

mydata[paste("poutcome", level, sep = "\_")] <- ifelse(mydata$poutcome == level, 1, 0)

}

mydata$Class <- ifelse(mydata$y == "yes", "Yes", "No")

#Remove unwanted columns

mydata$X <- NULL

mydata$job <- NULL

mydata$marital <- NULL

mydata$education <- NULL

mydata$default <- NULL

mydata$housing <- NULL

mydata$loan <- NULL

mydata$contact <- NULL

mydata$month <- NULL

mydata$poutcome <- NULL

mydata$y <- NULL

mydata$Class <- as.factor((mydata$Class))

colnames(mydata)[11] <- "job\_blue\_collar"

colnames(mydata)[14] <- "job\_admin"

colnames(mydata)[16] <- "job\_self\_employeed"

#Splitting

set.seed(1)

training\_size <- floor(0.80 \* nrow(mydata))

train\_ind <- sample(seq\_len(nrow(mydata)), size = training\_size)

training <- mydata[train\_ind, ]

testing <- mydata[-train\_ind, ]

#Normalizing

preProcValues <- preProcess(training, method = c("center", "scale"))

scaled.training <- predict(preProcValues, training)

scaled.testing <- predict(preProcValues, testing)

#Sampling

ctrl <- trainControl(method = "repeatedcv", repeats = 5,

classProbs = TRUE,

summaryFunction = twoClassSummary)

set.seed(2)

down\_training <- downSample(x = scaled.training[, -ncol(scaled.training)],

y = scaled.training$Class)

up\_training <- upSample(x = scaled.training[, -ncol(scaled.training)],

y = scaled.training$Class)

smote\_training <- SMOTE(Class~., data = scaled.training)

rose\_training <- ROSE(Class~., data = scaled.training, seed=2)$data

#Model training - CART

set.seed(3)

orig\_fit <- train(Class~., data = training,

method = "rpart",

metric = "ROC",

trControl = ctrl)

set.seed(4)

down\_outside <- train(Class~., data = down\_training,

method = "rpart",

metric = "ROC",

trControl = ctrl)

set.seed(5)

up\_outside <- train(Class~., data = up\_training,

method = "rpart",

metric = "ROC",

trControl = ctrl)

set.seed(6)

smote\_outside <- train(Class~., data = smote\_training,

method = "rpart",

metric = "ROC",

trControl = ctrl)

set.seed(7)

rose\_outside <- train(Class~., data = rose\_training,

method = "rpart",

metric = "ROC",

trControl = ctrl)

#Model testing - Original

original\_model <- list(original = orig\_fit)

test\_roc <- function(model, data) {

library(pROC)

roc\_obj <- roc(data$Class,

predict(model, data, type = "prob")[, "Yes"],

levels = c("No", "Yes"))

ci(roc\_obj)

}

original\_test <- lapply(original\_model, test\_roc, data = testing)

original\_test <- lapply(original\_test, as.vector)

original\_test <- do.call("rbind", original\_test)

colnames(original\_test) <- c("lower", "ROC", "upper")

original\_test <- as.data.frame(original\_test)

#Model testing - Resampled

scaled\_models <- list(down = down\_outside,

up = up\_outside,

SMOTE = smote\_outside,

ROSE = rose\_outside)

scaled\_test <- lapply(scaled\_models, test\_roc, data = scaled.testing)

scaled\_test <- lapply(scaled\_test, as.vector)

scaled\_test <- do.call("rbind", scaled\_test)

colnames(scaled\_test) <- c("lower", "ROC", "upper")

scaled\_test <- as.data.frame(scaled\_test)

cart\_test <- rbind(original\_test,scaled\_test)

fancyRpartPlot(up\_outside$finalModel)