```
# Rudy Meza
# Machine Learning in R
# Michael Chang
# Spring
# '***Lab and Quiz***'
# '*******************Machine Learning*********************'
# 8.3 Lab: Decision Trees
library(tree)
library(ISLR)
attach(Carseats)
High=ifelse(Sales<=8,"No","Yes")
Carseats$High <- factor(ifelse(Carseats$Sales <= 8, "No", "Yes"))
Carseats=data.frame(Carseats ,High)

tree.carseats <- tree::tree(High ~ . -Sales, Carseats)
summary(tree.carseats)
plot(tree.carseats)
text(tree.carseats, pretty=0)

tree.carseats
set.seed(2)
train <- sample(1:nrow(Carseats), 200)
Carseats.test <- Carseats[-train, ]
High.test <- High[-train]
tree.carseats <- tree(High ~ . - Sales, Carseats, subset = train)
tree.pred <- predict(tree.carseats, Carseats.test, type = "class")
table(tree.pred, High.test)
(86 + 57)/200

set.seed(3)
cv.carseats <- cv.tree(tree.carseats, FUN = prune.misclass)
names(cv.carseats)

par(mfrow = c(1, 2))
plot(cv.carseats$size, cv.carseats$dev, type = "b")
plot(cv.carseats$k, cv.carseats$dev, type = "b")

prune.carseats <- prune.misclass(tree.carseats, best = 9)
plot(prune.carseats)
text(prune.carseats, pretty = 0)

tree.pred <- predict(prune.carseats, Carseats.test, type = "class")
table(tree.pred, High.test)

(94 + 60)/200

prune.carseats <- prune.misclass(tree.carseats, best = 15)
plot(prune.carseats)
```

```
text(prune.carseats, pretty = 0)

tree.pred <- predict(prune.carseats, Carseats.test, type = "class")
table(tree.pred, High.test)

(86 + 62)/200

# 8.3.2 Fitting Regression Trees

library(MASS)
set.seed(1)
train <- sample(1:nrow(Boston), nrow(Boston)/2)
tree.boston <- tree(medv ~ ., Boston, subset = train)
summary(tree.boston)

plot(tree.boston)
text(tree.boston, pretty = 0)

cv.boston <- cv.tree(tree.boston)
plot(cv.boston$size, cv.boston$dev, type = "b")

prune.boston <- prune.tree(tree.boston, best = 5)
plot(prune.boston)
text(prune.boston, pretty = 0)

yhat <- predict(tree.boston, newdata = Boston[-train, ])
boston.test <- Boston[-train, "medv"]
plot(yhat, boston.test)
abline(0, 1)

mean((yhat - boston.test)^2)

# 8.3.3 Bagging and Random Forests

library(randomForest)
set.seed(1)
bag.boston =randomForest(medv~.,data=Boston ,subset =train,
mtry=13, importance =TRUE)
bag.boston

yhat.bag <- predict(bag.boston, newdata = Boston[-train, ])
plot(yhat.bag, boston.test)
abline(0, 1)

mean((yhat.bag - boston.test)^2)

bag.boston <- randomForest(medv ~ ., data = Boston, subset = train, mtry = 13, ntree = 25)
yhat.bag <- predict(bag.boston, newdata = Boston[-train, ])
```

```
mean((yhat.bag - boston.test)^2)

set.seed(1)
rf.boston <- randomForest(medv ~ ., data = Boston, subset = train, mtry = 6, importance = TRUE)
yhat.rf <- predict(rf.boston, newdata = Boston[-train, ])
mean((yhat.rf - boston.test)^2)

importance(rf.boston)

# 8.3.4 Boosting

library(gbm)

set.seed(1)
boost.boston <- gbm(medv ~ ., data = Boston[train, ], distribution = "gaussian", n.trees = 5000,
interaction.depth = 4)
summary(boost.boston)

par(mfrow = c(1, 2))
plot(boost.boston, i = "rm")
plot(boost.boston, i = "lstat")

yhat.boost <- predict(boost.boston, newdata = Boston[-train, ], n.trees = 5000)
mean((yhat.boost - boston.test)^2)

boost.boston =gbm(medv~.,data=Boston [train ,], distribution=
"gaussian",n.trees =5000 , interaction.depth =4, shrinkage =0.2,
verbose =F)

yhat.boost=predict (boost.boston ,newdata =Boston [-train ,],
n.trees =5000)

mean(( yhat.boost -boston.test)^2)

 # ***********Quiz*****************

 # Question 1

prune.carseats <- prune.misclass(tree.carseats, best=9)
plot(prune.carseats)
text(prune.carseats, pretty = 0)

# Question 2

tree.pred=predict(prune.carseats , Carseats.test ,type="class")
table(tree.pred, High.test)
94+60+24+22
(94+60)/200
```

```
# Question 3

tree.carseats_a=tree(High~., Carseats ,subset =train)
summary(tree.carseats_a)
tree.pred=predict(tree.carseats_a, Carseats.test ,type="class")
table(tree.pred, High.test)

#Question 4

set.seed(1)

train_b<- sample(1:nrow(Boston), nrow(Boston)/2)

tree.boston <- tree(medv ~ ., data = Boston, subset = train_b)

summary(tree.boston)
# Lstat, rm, dis
# Question 5
set.seed (1)
rf.boston =randomForest(medv~.,data=Boston, subset =train_b,importance =TRUE)
yhat.rf = predict(rf.boston ,newdata=Boston[-train,])
mean(( yhat.rf-boston.test)^2)


# 13.5

# Question 6

set.seed(1)
boost.boston=gbm(medv~.,data=Boston[train ,], distribution=
"gaussian",n.trees =5000 , interaction.depth =5, shrinkage =0.1)
yhat.boost=predict(boost.boston ,newdata =Boston[-train ,],
n.trees =5000)

mean(( yhat.boost-boston.test)^2)
# 10.4

# Question 7

set.seed(1)
boost.boston=gbm(medv~.,data=Boston[train ,], distribution=
"gaussian",n.trees =5000 , interaction.depth =5, shrinkage =0.01)
yhat.boost=predict(boost.boston ,newdata =Boston[-train ,],
n.trees =5000)

mean(( yhat.boost-boston.test)^2)
# The error is lower as the model is shrinking lower and able to learn more about the data.
```