Individual_assignment8

Yuhan_Xu_474154

2019/10/31

Problem 8

In the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

```
library(ISLR)
attach(Carseats)

set.seed(1)
train = sample(1:nrow(Carseats), nrow(Carseats)/2)
Sales.test = Carseats[-train,"Sales"]
```

(d)

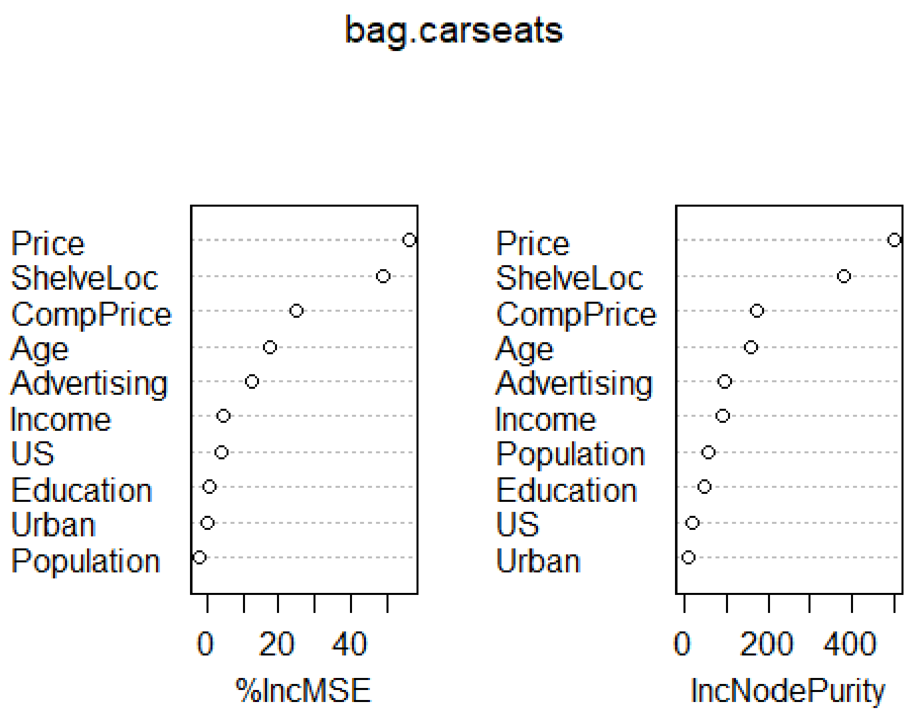
Q: Use the bagging approach in order to analyze this data. What test MSE do you obtain?

```
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

set.seed(1)
bag.carseats = randomForest(Sales~., data = Carseats, subset = train, m
try = 10, importance = TRUE)

yhat.bag = predict(bag.carseats, newdata = Carseats[-train,])
mean((yhat.bag - Sales.test)^2)

## [1] 2.605253
```

A: The test MSE obtained is 2.61.

Q: Use the importance() function to determine which variables are most important.

```
varImpPlot(bag.carseats)
```




```r
importance(bag.carseats)

##                %IncMSE IncNodePurity
## CompPrice   24.8888481    170.182937
## Income       4.7121131     91.264880
## Advertising 12.7692401     97.164338
## Population  -1.8074075     58.244596
## Price       56.3326252    502.903407
## ShelveLoc   48.8886689    380.032715
## Age         17.7275460    157.846774
## Education    0.5962186     44.598731
## Urban        0.1728373      9.822082
## US           4.2172102     18.073863
```

A: From above graphs and scores, the most important variables are Price, the price company charges for car seats at each site and ShelveLoc, the quality of the shelving location for the car seats at each site.

(e)

Q: Use random forests to analyze this data. What test MSE do you obtain?

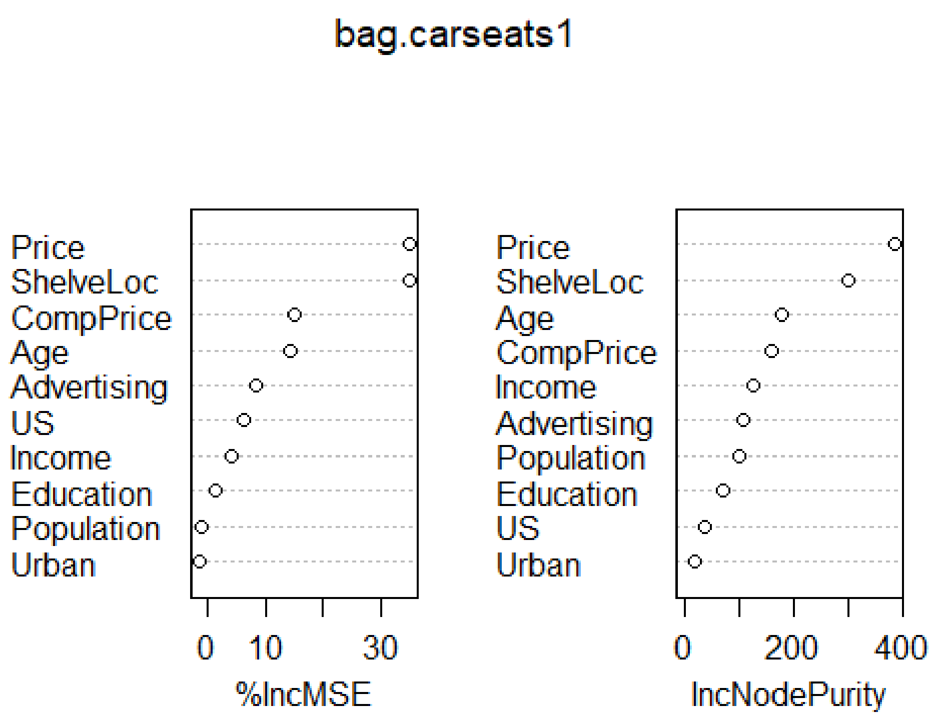
```r
set.seed(1)
bag.carseats1 = randomForest(Sales~., data = Carseats, subset = train,
mtry = 3, importance = TRUE)
```

```r
yhat.bag1 = predict(bag.carseats1, newdata = Carseats[-train,])
mean((yhat.bag1 - Sales.test)^2)

## [1] 2.960559
```

A: The test MSE obtained is 2.96. It's higher than that obtained from bagging.

Q: Use the importance() function to determine which variables are most important.

```r
varImpPlot(bag.carseats1)
```

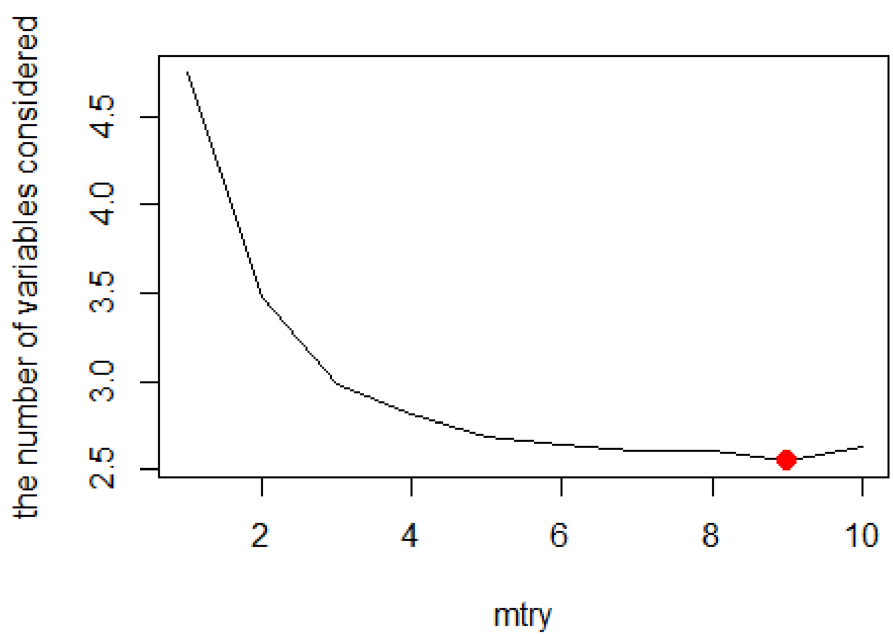



```r
importance(bag.carseats1)

##                %IncMSE IncNodePurity
## CompPrice   14.8840765     158.82956
## Income       4.3293950     125.64850
## Advertising  8.2215192     107.51700
## Population  -0.9488134      97.06024
## Price       34.9793386     385.93142
## ShelveLoc   34.9248499     298.54210
## Age         14.3055912     178.42061
## Education    1.3117842      70.49202
## Urban       -1.2680807      17.39986
## US           6.1139696      33.98963
```

A: The most important variables are also Price and ShelveLoc.

Q: Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

```
set.seed(1)
testMSE.bag = c()

for (m in 1:10){
  bag.carseats = randomForest(Sales~., data = Carseats, subset = train,
 mtry = m, importance = TRUE)
  yhat.bag = predict(bag.carseats, newdata = Carseats[-train,])
  testMSE.bag = c(testMSE.bag, mean((yhat.bag - Sales.test)^2))
}

plot(1:10, testMSE.bag, xlab = "mtry", ylab = "the number of variables
considered", type = "l")

points(which.min(testMSE.bag),min(testMSE.bag),col="red",cex=2,pch=20)
```



A: When we consider more variables at each split, the test MSE will first decrease and then increase after there are more than 9 variables.

Problem 10

We now use boosting to predict Salary in the Hitters data set.

(a)

Q: Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

```
dim(Hitters)

## [1] 322  20

Hitters = na.omit(Hitters)
dim(Hitters)

## [1] 263  20

logSalary = log(Hitters$Salary)
Hitters = data.frame(Hitters, logSalary)
```

(b)

Q: Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

```
set.seed(2)
train1 = sample(1:nrow(Hitters), 200)
Hitters.train = Hitters[train1,]
Hitters.test = Hitters[-train1,]
```

(c)

Q: Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter λ.

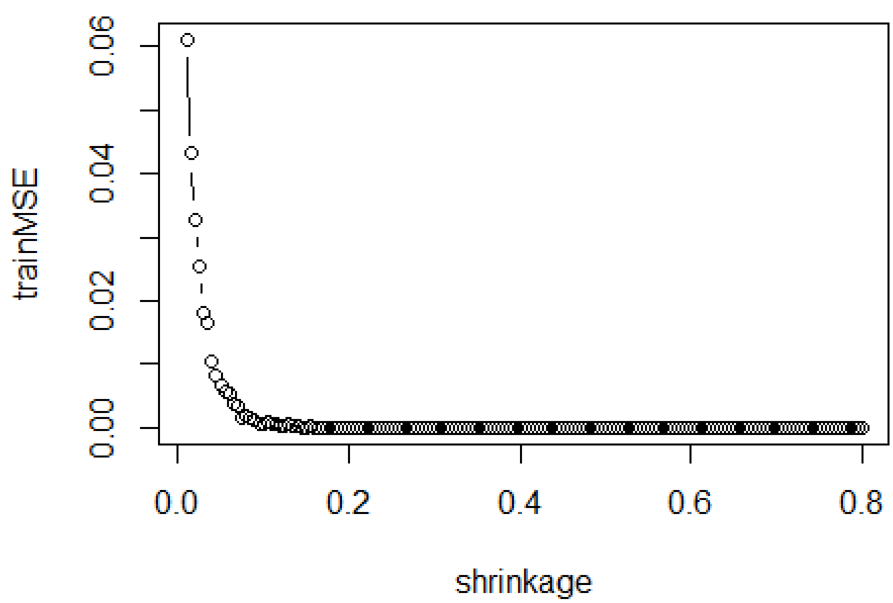
```
library(gbm)

## Loaded gbm 2.1.5

set.seed(1)

lam.series = seq(from=0.01, to=0.8 ,by=0.005)
trainMSE = rep(1,times = length(lam.series))
testMSE = rep(1,times = length(lam.series))

for (i in 1:length(lam.series)){
  boost.Hitters = gbm(logSalary~.-Salary, data = Hitters.train, distrib
ution = "gaussian", n.trees = 1000, interaction.depth =  4, shrinkage =
 lam.series[i], verbose = F)

  yhat.boost1 = predict(boost.Hitters, Hitters.train, n.trees = 1000)
  trainMSE[i] = mean((yhat.boost1 - Hitters.train$logSalary)^2)

  yhat.boost2 = predict(boost.Hitters, Hitters.test, n.trees = 1000)
```

```
  testMSE[i] = mean((yhat.boost2 - Hitters.test$logSalary)^2)
}

min(testMSE)

## [1] 0.21591
```

Q: Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

A:

```
plot(lam.series, trainMSE, xlab = "shrinkage", type = "b")
```

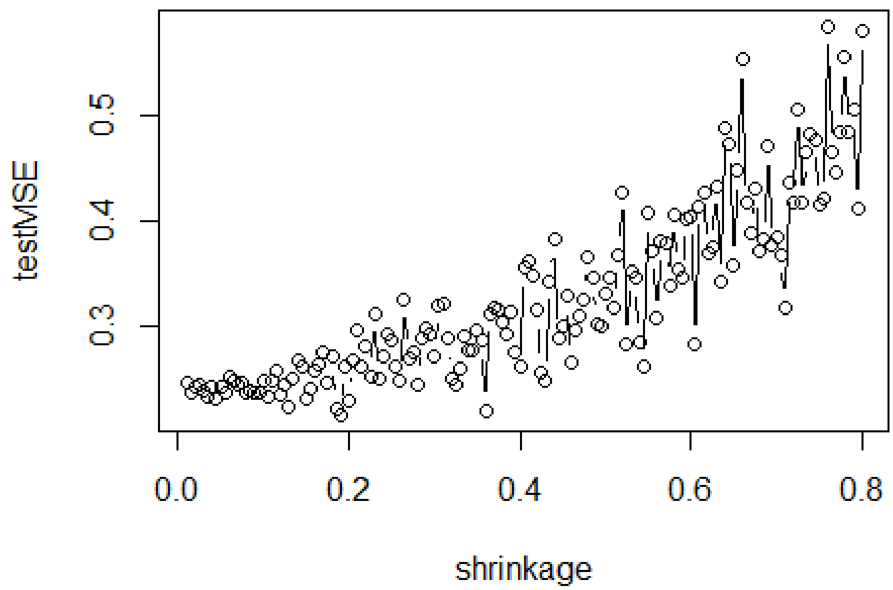


(d)

Q: Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

A:

```
plot(lam.series, testMSE, xlab = "shrinkage", type = "b")
```



(e)

Q: Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6.
A: Applying linear regression:

```
lm.fit = lm(logSalary~.-Salary, data = Hitters.train)
yhat.lm = predict(lm.fit, Hitters.test)
mean((yhat.lm-Hitters.test$logSalary)^2)

## [1] 0.5143062
```

Applying LASSO:

```
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18

set.seed(1)

x.train = model.matrix(logSalary~.-Salary, data = Hitters.train)[,-1]
x.test = model.matrix(logSalary~.-Salary, data = Hitters.test)[,-1]
y.train = Hitters.train$logSalary
y.test = Hitters.test$logSalary
```

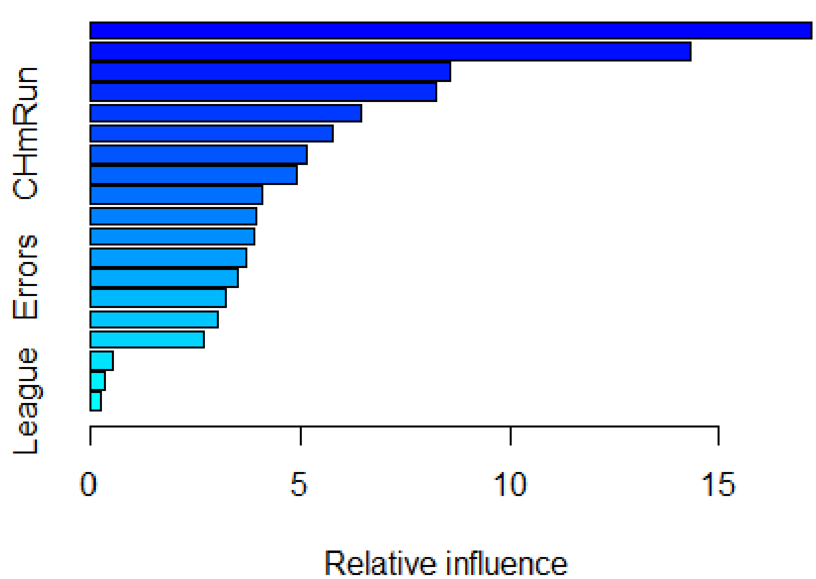
```
grid = 10 ^ seq(10, -2, length=100)

lasso.mod = glmnet(x.train, y.train, alpha = 1, lambda = grid)

cv.out = cv.glmnet(x.train, y.train, alpha = 1)
bestlam = cv.out$lambda.min

lasso.pred = predict(lasso.mod, s = bestlam, newx = x.test, x = x.train,
 y = y.train, exact = TRUE)
mean((lasso.pred - y.test)^2)

## [1] 0.4914141
```

A: The test MSE obtained from boosting is smaller than that obtained from linear regression and LASSO.

(f)

Q: Which variables appear to be the most important predictors in the boosted model?

```
boost.Hitters = gbm(logSalary~.-Salary, data = Hitters.train, distribut
ion = "gaussian", n.trees = 1000, interaction.depth =  4, shrinkage = l
am.series[which.min(testMSE)], verbose = F)
summary(boost.Hitters)
```




```
##                    var    rel.inf
## CRuns            CRuns 17.2042666
```

```
## CAtBat       CAtBat 14.2958476
## CWalks       CWalks  8.5992982
## CHits         CHits  8.2633253
## Walks         Walks  6.4736992
## CHmRun       CHmRun  5.7708001
## PutOuts     PutOuts  5.1836930
## AtBat         AtBat  4.9428118
## RBI             RBI  4.0993017
## Hits           Hits  3.9593152
## Years         Years  3.9245043
## HmRun         HmRun  3.7261732
## Errors       Errors  3.5010100
## Runs           Runs  3.2244363
## CRBI           CRBI  3.0587885
## Assists     Assists  2.6873766
## NewLeague NewLeague  0.5244470
## Division   Division  0.3261154
## League       League  0.2347898
```

A: The most important predictors are CRuns, Number of runs during his career and CRBI, Number of runs batted in during his career.

(g)

Q: Now apply bagging to the training set. What is the test set MSE for this approach?

```r
library(randomForest)
set.seed(1)
bag.Hitters = randomForest(logSalary~.-Salary, data = Hitters.train, mt
ry = 19)
yhat.bag2 = predict(bag.Hitters, Hitters.test)
mean((yhat.bag2-Hitters.test$logSalary)^2)

## [1] 0.1888011
```