Homework 4.3 - Carseats dataset

Hamed

2/26/2020

# Use the Carseats data set in ISLR package

library(ISLR)  
library(tree)  
  
carseats=as.data.frame(Carseats)  
str(carseats)

## 'data.frame': 400 obs. of 11 variables:  
## $ Sales : num 9.5 11.22 10.06 7.4 4.15 ...  
## $ CompPrice : num 138 111 113 117 141 124 115 136 132 132 ...  
## $ Income : num 73 48 35 100 64 113 105 81 110 113 ...  
## $ Advertising: num 11 16 10 4 3 13 0 15 0 0 ...  
## $ Population : num 276 260 269 466 340 501 45 425 108 131 ...  
## $ Price : num 120 83 80 97 128 72 108 120 124 124 ...  
## $ ShelveLoc : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...  
## $ Age : num 42 65 59 55 38 78 71 67 76 76 ...  
## $ Education : num 17 10 12 14 13 16 15 10 10 17 ...  
## $ Urban : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...  
## $ US : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...

dim(carseats)

## [1] 400 11

## (a) Split the data set into a training set and a test set.

set.seed(123)  
smp\_size <-floor(0.80 \* nrow(carseats))  
smp\_size

## [1] 320

train\_ind <-sample(seq\_len(nrow(carseats)), size = smp\_size)  
train.carseats = carseats[train\_ind,]  
dim(train.carseats)

## [1] 320 11

test.carseats = carseats[-train\_ind,]  
dim(test.carseats)

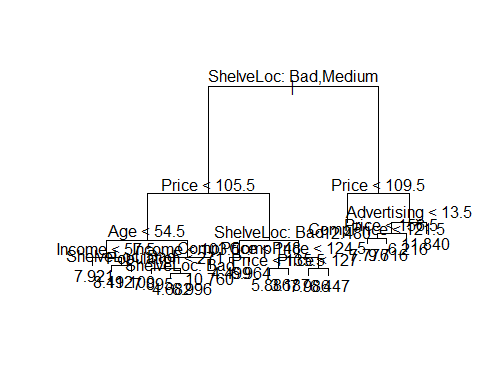
## [1] 80 11

## (b) Fit a regression tree to the training set by selecting an appropriate target variable to predict. Plot the tree, and interpret the results. What test MSE do you obtain?

#Regression model tree   
tree.carseats = tree(Sales~.,data=train.carseats)  
summary.carseats = summary(tree.carseats)  
summary.carseats

##   
## Regression tree:  
## tree(formula = Sales ~ ., data = train.carseats)  
## Variables actually used in tree construction:  
## [1] "ShelveLoc" "Price" "Age" "Income" "Population"   
## [6] "CompPrice" "Advertising"  
## Number of terminal nodes: 18   
## Residual mean deviance: 2.541 = 767.3 / 302   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -4.12900 -1.11200 -0.01104 0.00000 0.95020 5.64100

#Plot the tree  
plot(tree.carseats)  
text(tree.carseats,pretty=0)



#Prediction  
tree.pred = predict(tree.carseats, newdata=test.carseats)  
head(tree.pred)

## 1 3 6 15 17 18   
## 4.498810 7.894737 10.760000 7.776429 7.776429 9.716364

#Mean squared error  
mean((tree.pred-test.carseats$Sales)^2)

## [1] 4.1593

## (c) Use cross-validation in order to determine the optimal level of tree complexity.

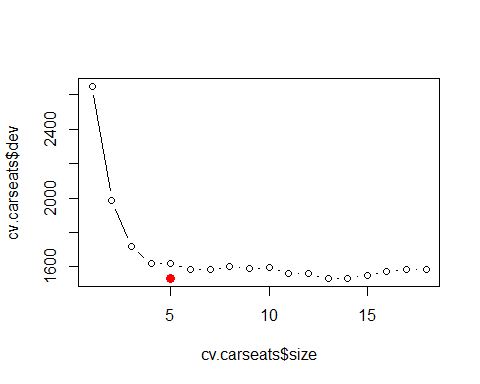
#Cross validation  
cv.carseats = cv.tree(tree.carseats)  
names(cv.carseats)

## [1] "size" "dev" "k" "method"

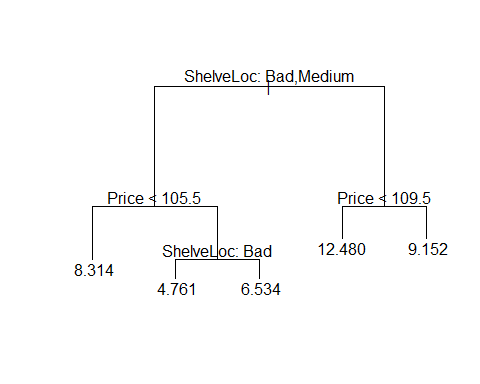
cv.carseats

## $size  
## [1] 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1  
##   
## $dev  
## [1] 1587.056 1586.685 1570.479 1551.676 1533.205 1533.205 1559.156 1562.609  
## [9] 1595.755 1590.495 1602.464 1583.848 1583.848 1617.345 1617.345 1720.102  
## [17] 1987.530 2649.990  
##   
## $k  
## [1] -Inf 27.15329 29.32100 32.19753 33.10336 33.87276 37.31269  
## [8] 37.62423 42.97521 46.35501 66.34292 69.10022 69.39554 104.48783  
## [15] 104.91492 178.86486 293.28670 665.98532  
##   
## $method  
## [1] "deviance"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

#plot the deviance against the size of the tree model  
plot(cv.carseats$size, cv.carseats$dev,type="b")  
points(which.min(cv.carseats$dev),cv.carseats$dev[which.min(cv.carseats$dev)],col="red",pch=19,cex=1.25)



#Prune and Plot the pruned cross validated tree  
prune.carseats = prune.tree(tree.carseats,best = which.min(cv.carseats$dev))  
plot(prune.carseats)  
text(prune.carseats,pretty=0)



#Prediction  
prune.pred = predict(prune.carseats, test.carseats)  
head(prune.pred)

## 1 3 6 15 17 18   
## 4.761064 8.313529 8.313529 9.151633 9.151633 9.151633

#Mean squared error  
mean((prune.pred-test.carseats$Sales)^2)

## [1] 4.387291

## (d) Use the bagging approach to do the prediction. What test MSE do you obtain? Which variables are most important? Plot the variable importance.

library(randomForest)

## randomForest 4.6-14

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

set.seed(1)  
  
#Bagging  
bag.carseats = randomForest(Sales~.,data=train.carseats,mtry=10,importance=T)  
bag.carseats

##   
## Call:  
## randomForest(formula = Sales ~ ., data = train.carseats, mtry = 10, importance = T)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 10  
##   
## Mean of squared residuals: 2.457478  
## % Var explained: 70.21

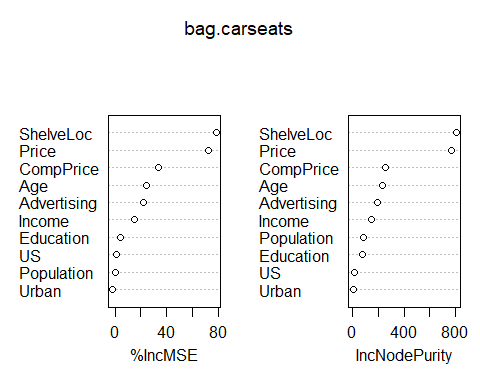
#Prediction and mean squared error (MSE)  
yhat.bag = predict(bag.carseats,newdata=test.carseats)  
mean((yhat.bag - test.carseats$Sales)^2)

## [1] 2.102364

#Variable importance  
importance(bag.carseats)

## %IncMSE IncNodePurity  
## CompPrice 33.9984190 252.265173  
## Income 15.3397079 147.056418  
## Advertising 22.2968622 189.210197  
## Population 0.3734011 79.123364  
## Price 72.1674452 767.508773  
## ShelveLoc 78.8565462 809.298696  
## Age 24.5332679 234.353078  
## Education 4.5163483 76.864069  
## Urban -1.6226503 8.071738  
## US 1.1817709 10.384082

varImpPlot(bag.carseats)



# (e) Use random forest to do the prediction. What test MSE do you obtain?

#Value of m=5  
rf5.carseats = randomForest(Sales~.,data=train.carseats,mtry=5,importance=T)  
rf5.carseats

##   
## Call:  
## randomForest(formula = Sales ~ ., data = train.carseats, mtry = 5, importance = T)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## Mean of squared residuals: 2.524748  
## % Var explained: 69.39

#Prediction and MSE  
yhat5.rf = predict(rf5.carseats,newdata=test.carseats)  
mean((yhat5.rf-test.carseats$Sales)^2)

## [1] 2.232687

## Try various values of m, the number of variables considered at each split and describe the effect of m on the error rate.

#Value of m=3  
rf3.carseats = randomForest(Sales~.,data=train.carseats,mtry=3,importance=T)  
rf3.carseats

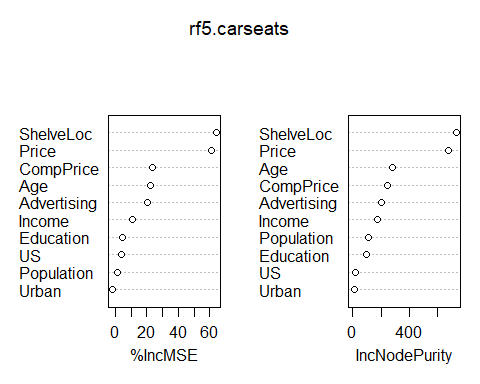
##   
## Call:  
## randomForest(formula = Sales ~ ., data = train.carseats, mtry = 3, importance = T)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 2.799985  
## % Var explained: 66.06

#Prediction and MSE  
yhat3.rf = predict(rf3.carseats,newdata=test.carseats)  
mean((yhat3.rf-test.carseats$Sales)^2)

## [1] 2.613921

## Which variables are most important. Plot the variable importance.

par(mfrow=c(2,2))  
varImpPlot(rf5.carseats)



varImpPlot(rf3.carseats)

