# [STAT 4610] HW - 9

Michael Ghattas

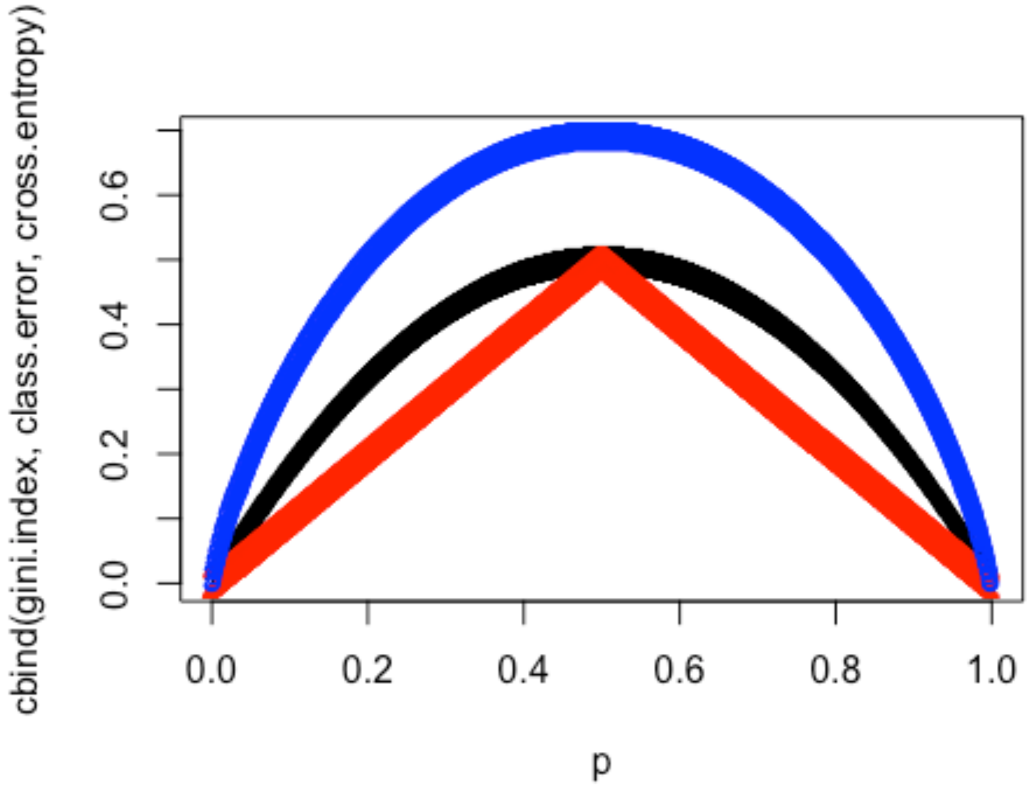
11/6/2022

## Chapter 8

```
library(ISLR)
library(tree)

## Warning: package 'tree' was built under R version 4.1.2

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.2

## randomForest 4.7-1.1

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

### Problem-3

```
p <- seq(0, 1, 0.001)

gini.index <- 2 * (p * (1 - p))
class.error <- 1 - pmax(p, 1 - p)
cross.entropy <- - ((p * log(p)) + ((1 - p) * log(1 - p)))

matplot(p, cbind(gini.index, class.error, cross.entropy), col = c("black",
"red", "blue"))
```




### Problem-8

```
data(Carseats)
```

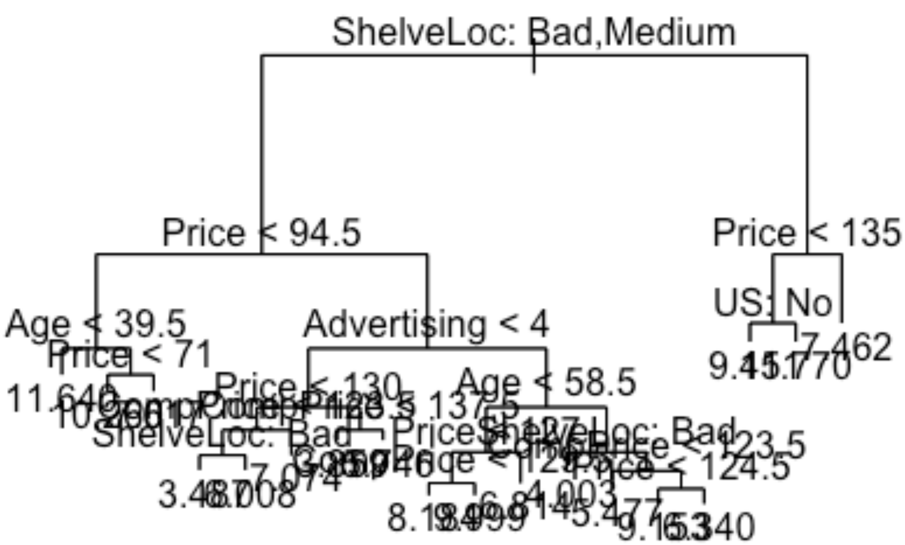
#### part (a)

```
set.seed(1)

train = sample(1:nrow(Carseats), nrow(Carseats) / 2)
Car.train = Carseats[train, ]
Car.test = Carseats[-train, ]
```

#### Part (b)

```
reg.tree = tree(Sales~.,data = Carseats, subset=train)
reg.tree = tree(Sales~.,data = Car.train)
summary(reg.tree)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Car.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "CompPrice"
## [6] "US"
## Number of terminal nodes:  18
## Residual mean deviance:  2.167 = 394.3 / 182
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900

plot(reg.tree)
text(reg.tree, pretty = 0)
```

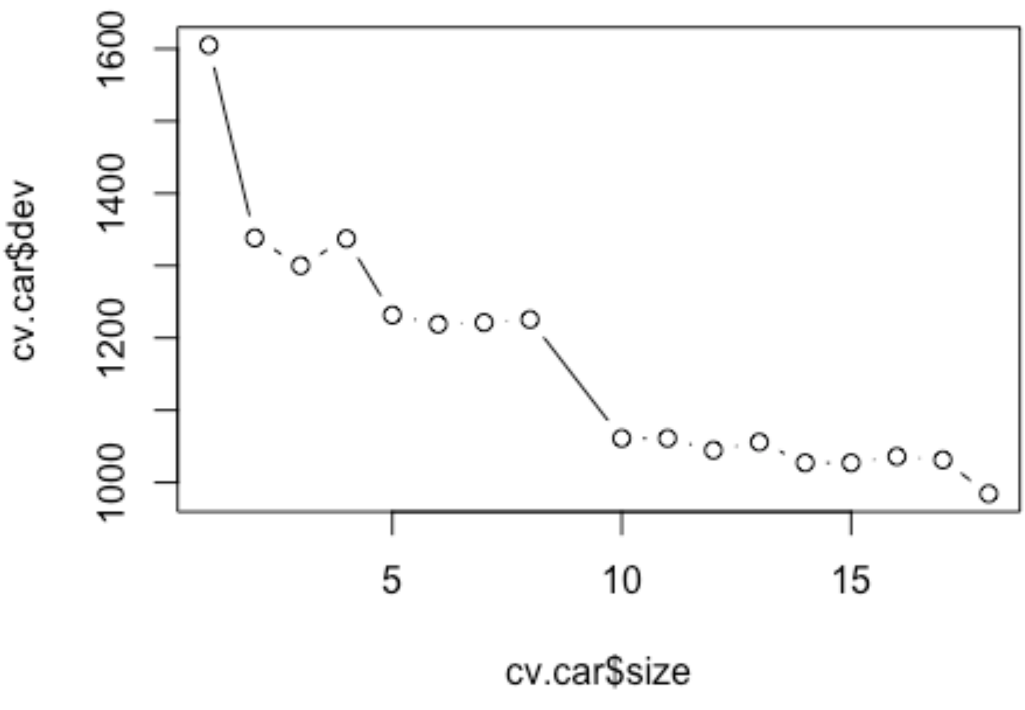


```
yhat = predict(reg.tree,newdata = Car.test)
mean((yhat - Car.test$Sales)^2)

## [1] 4.922039
```

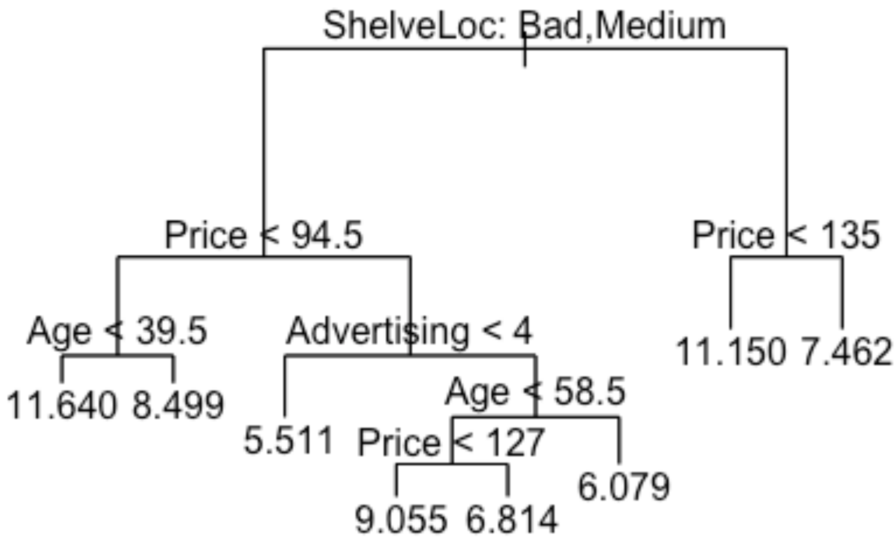
-> Test MSE = 4.922

### Part (c)

```
set.seed(1)


cv.car = cv.tree(reg.tree)
plot(cv.car$size, cv.car$dev, type = "b")
```




```
prune.car = prune.tree(reg.tree, best = 8)
plot(prune.car)
text(prune.car, pretty = 0)
```




```
yhat = predict(prune.car, newdata = Car.test)
mean((yhat - Car.test$Sales)^2)

## [1] 5.113254
```

-> Optimal tree complexity level = 8
-> Pruning tree increases MSE to 5.113
-> Worse performance

### Part (d)

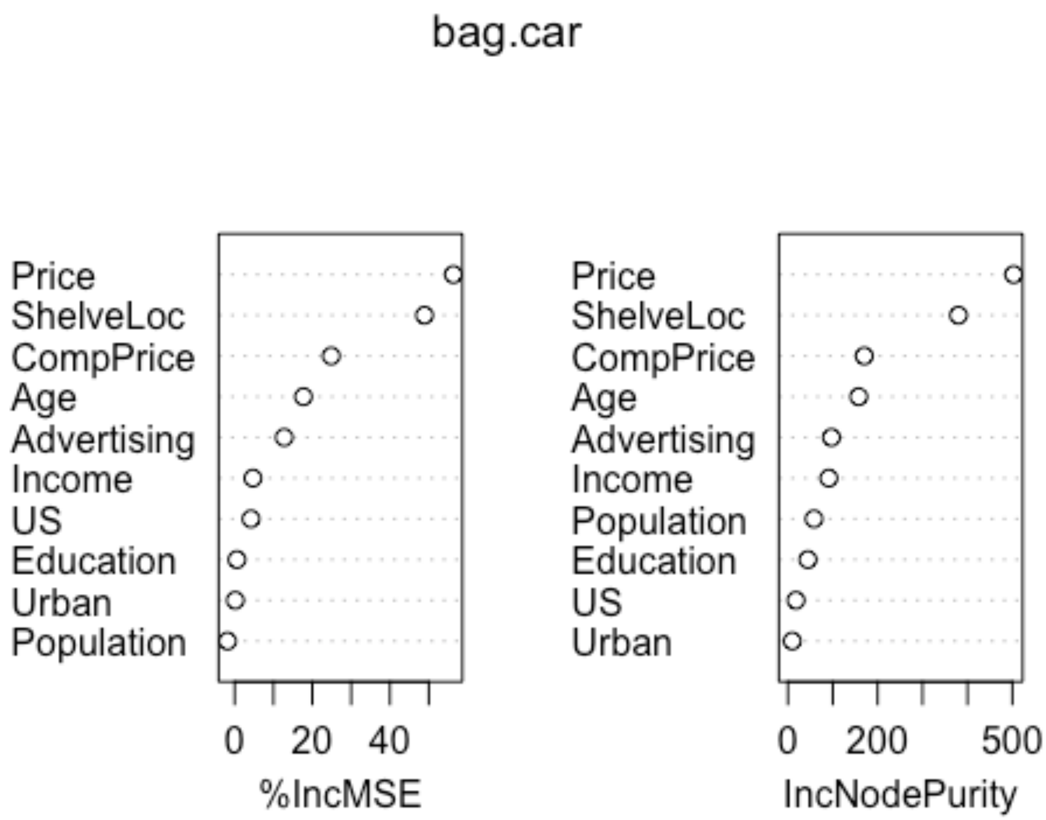
```
set.seed(1)

bag.car = randomForest(Sales ~ ., data = Car.train, mtry = 10, importance =
TRUE)
yhat.bag = predict(bag.car, newdata = Car.test)
mean((yhat.bag - Car.test$Sales)^2)
```

```
## [1] 2.605253
```

```
importance(bag.car)
```

```
##                %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
```

```
varImpPlot(bag.car)
```




-> MSE = 2.605
-> Better performance
-> Most important variables are: (1) Price | (2) Shelving Location

**part (e)**

```
set.seed(1)

rf.car = randomForest(Sales ~ ., data = Car.train, mtry = 3, importance =
TRUE)
yhat.rf = predict(rf.car, newdata = Car.test)
mean((yhat.rf - Car.test$Sales)^2)

## [1] 2.960559
```

-> $m = \sqrt(p)$
-> MSE = 2.961
-> Worse performance

## End.