HW8

David Schultheiss

11/4/2020

##Problem 1

a)

```
tumor <- read.csv("/Users/davidschultheiss/Downloads/tumor.csv")
tumor$Diagnosis = factor(tumor$Diagnosis)

set.seed(1)
train = sample(1:nrow(tumor), .9*nrow(tumor))
test = tumor[-train, ]
training = tumor[train, ]
```

b)

```
library(tree)
tumor.tree = tree(Diagnosis~., data= training)
summary(tumor.tree)
```

```
##
## Classification tree:
## tree(formula = Diagnosis ~ ., data = training)
## Variables actually used in tree construction:
## [1] "Concave.Points" "Area"           "Texture"        "Perimeter"
## Number of terminal nodes:  9
## Residual mean deviance:  0.1964 = 98.81 / 503
## Misclassification error rate: 0.03906 = 20 / 512
```

The training error rate is 3.91%, with 9 terminal nodes.

c)

```
tumor.tree
```

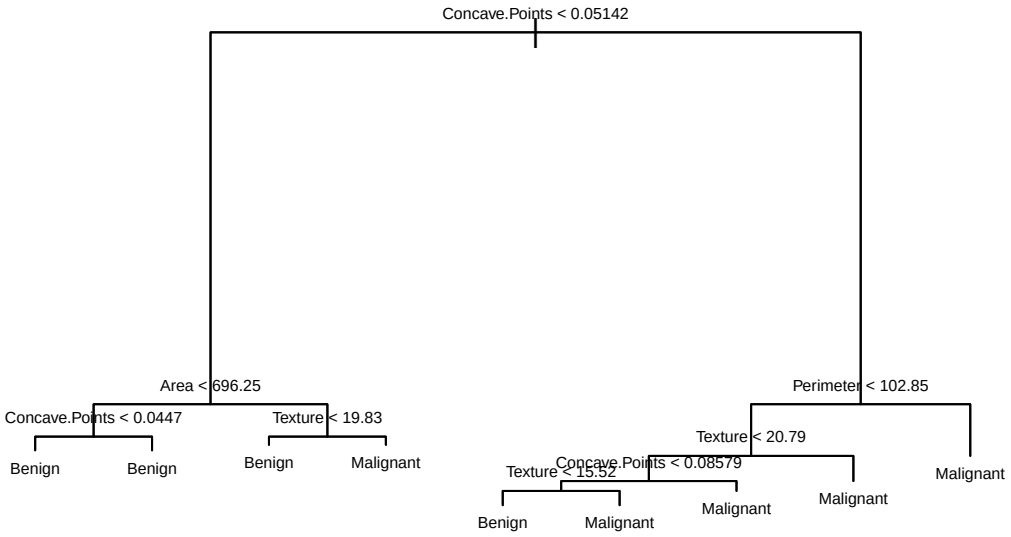
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 512 674.300 Benign ( 0.63086 0.36914 )
##    2) Concave.Points < 0.05142 316 126.600 Benign ( 0.94937 0.05063 )
##      4) Area < 696.25 303  73.940 Benign ( 0.97360 0.02640 )
```

```
##        8) Concave.Points < 0.0447 289  42.190 Benign ( 0.98616 0.01384 ) *
##        9) Concave.Points > 0.0447 14  16.750 Benign ( 0.71429 0.28571 ) *
##      5) Area > 696.25 13  17.320 Malignant ( 0.38462 0.61538 )
##       10) Texture < 19.83 7   8.376 Benign ( 0.71429 0.28571 ) *
##       11) Texture > 19.83 6   0.000 Malignant ( 0.00000 1.00000 ) *
##    3) Concave.Points > 0.05142 196 141.700 Malignant ( 0.11735 0.88265 )
##      6) Perimeter < 102.85 66  85.340 Malignant ( 0.34848 0.65152 )
##       12) Texture < 20.79 42  57.840 Benign ( 0.54762 0.45238 )
##         24) Concave.Points < 0.08579 36  47.090 Benign ( 0.63889 0.36111 )
##           48) Texture < 15.52 13   0.000 Benign ( 1.00000 0.00000 ) *
##           49) Texture > 15.52 23  31.490 Malignant ( 0.43478 0.56522 ) *
##         25) Concave.Points > 0.08579 6   0.000 Malignant ( 0.00000 1.00000 ) *
##       13) Texture > 20.79 24   0.000 Malignant ( 0.00000 1.00000 ) *
##      7) Perimeter > 102.85 130   0.000 Malignant ( 0.00000 1.00000 ) *
```

Our teminal point for Concave.Points < 0.0447 has a result of Benign with 98.6% probability. 289 observations were in this category.

d)

```
plot(tumor.tree)
text(tumor.tree, cex = .5)
```



Concave points $<$ .05142 leads us to 4 terminal nodes, with 3/4 resulting in a diagnosis of Benign. When this condition is not met, we have 5 terminal nodes with 4/5 resulting in a diagnosis of Malignant.

e)

```
tumor.pred = predict(tumor.tree, test, type = 'class')
table(tumor.pred, test$Diagnosis)
```

```
##
## tumor.pred  Benign Malignant
##   Benign        30         2
##   Malignant      4        21
```

```
mean(tumor.pred != test$Diagnosis)
```

```
## [1] 0.1052632
```

10.53% error rate.

f)

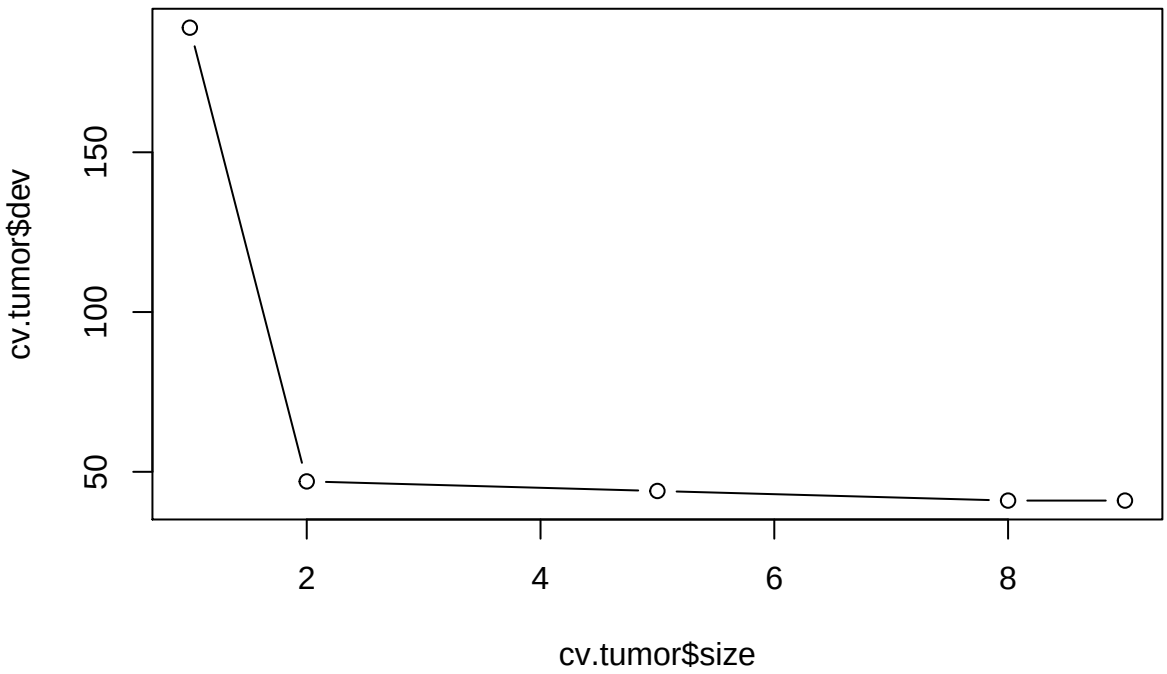
```
cv.tumor = cv.tree(tumor.tree, FUN = prune.misclass)
cv.tumor
```

```
## $size
## [1] 9 8 5 2 1
##
## $dev
## [1]  41  41  44  47 189
##
## $k
## [1]       -Inf   0.000000   3.000000   3.333333 150.000000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

Our best option is 8 terminal nodes.

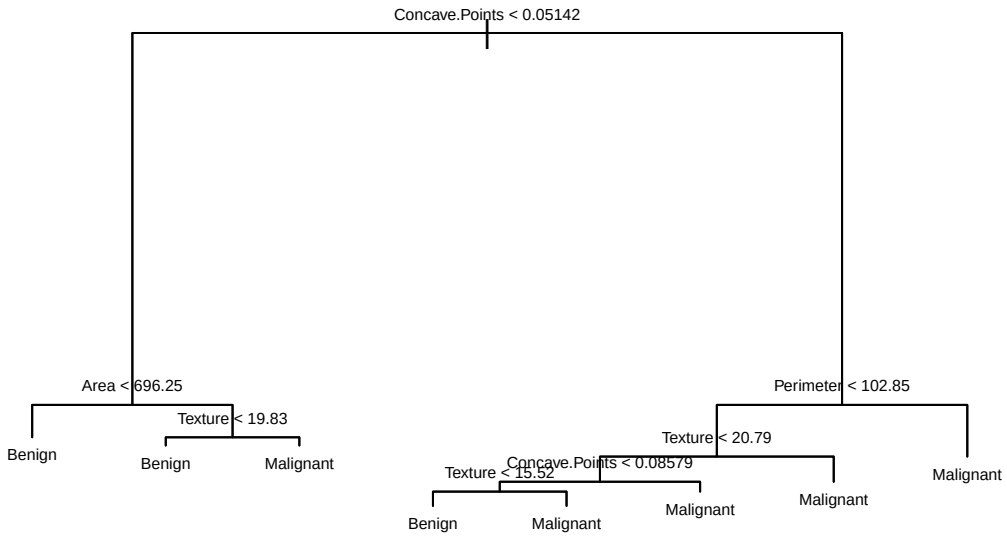
g)

```
plot(cv.tumor$size, cv.tumor$dev, type='b')
```



h)

```
prune.tumor = prune.misclass(tumor.tree, best = 8)
plot(prune.tumor)
text(prune.tumor, cex= .5)
```



i)

```
summary(prune.tumor)
```

```
##
## Classification tree:
## snip.tree(tree = tumor.tree, nodes = 4L)
## Variables actually used in tree construction:
## [1] "Concave.Points" "Area"           "Texture"        "Perimeter"
## Number of terminal nodes:  8
## Residual mean deviance:  0.2258 = 113.8 / 504
## Misclassification error rate: 0.03906 = 20 / 512
```

The training error rate is 3.91%, the same as the unpruned tree.

j)

```
prune.pred = predict(prune.tumor, test, type = 'class')
mean(prune.pred != test$Diagnosis)
```

```
## [1] 0.1052632
```

The test error rate is 10.53%, the same as the pruned tree.

k)

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
bag.tumor = randomForest(Diagnosis~., data= training, mtry= 10,
                         importance= T)
bag.tumor
```

```
##
## Call:
##  randomForest(formula = Diagnosis ~ ., data = training, mtry = 10,      importance = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 5.66%
## Confusion matrix:
##           Benign Malignant class.error
## Benign       308        15  0.04643963
## Malignant     14       175  0.07407407
```

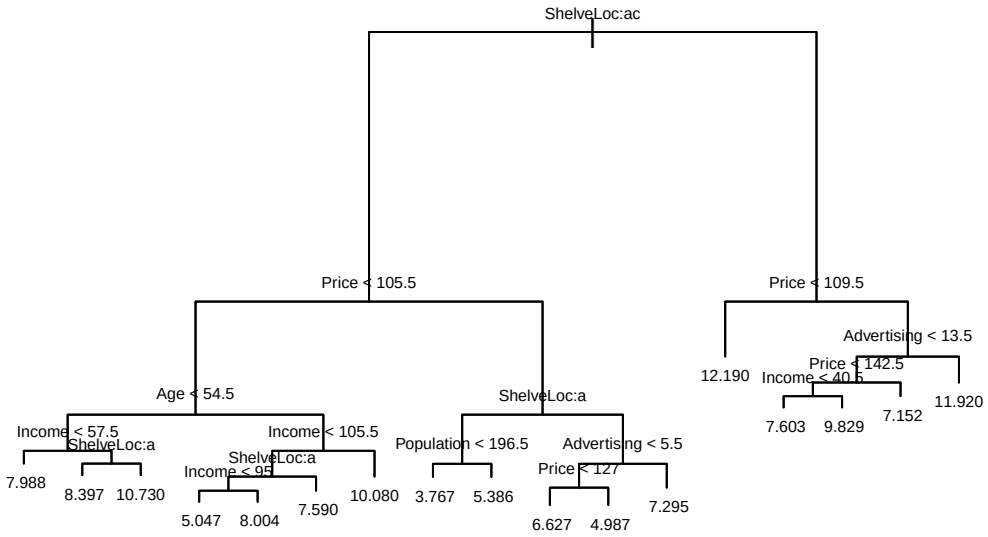
The misclassification rate is 5.66%

##Problem 2 a)

```
library(ISLR)
set.seed(1)
seats= Carseats
train = sample(1:nrow(seats), .5*nrow(seats))
test = seats[-train, ]
training = seats[train, ]
```

b)

```
seats.tree = tree(Sales~., data= Carseats)
plot(seats.tree)
text(seats.tree, cex = .5)
```




```
seats.pred = predict(seats.tree, newdata= test, type = 'vector')
mean((seats.pred-test$Sales)^2)
```

```
## [1] 2.326385
```

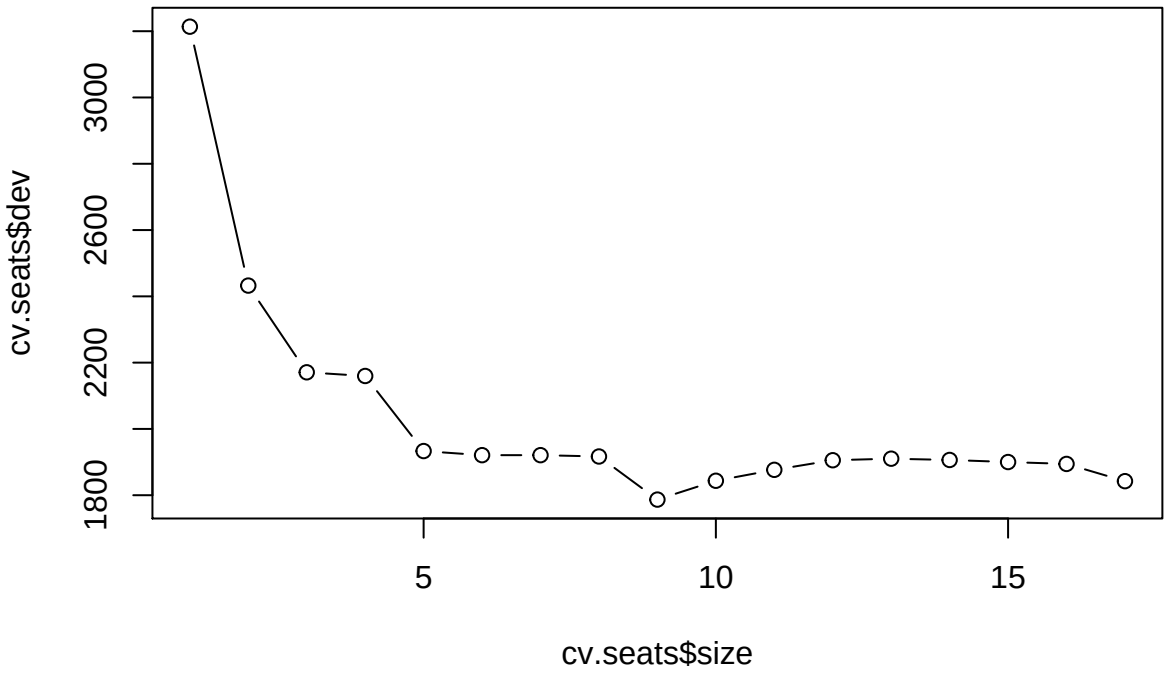
Test MSE is 2.449

c)

```
cv.seats = cv.tree(seats.tree)
cv.seats
```

```
## $size
##  [1] 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
##
## $dev
##  [1] 1842.353 1894.387 1900.112 1906.374 1910.206 1905.532 1876.558 1843.592
##  [9] 1786.814 1916.840 1920.823 1920.823 1933.182 2159.678 2170.816 2432.477
## [17] 3213.613
##
## $k
##  [1]      -Inf  32.78204  33.43341  34.30000  37.83019  38.65535  40.44960
##  [8]  41.83218  51.05171  70.52963  76.20847  76.57441 106.90014 145.33849
## [15] 162.67977 334.36974 797.19286
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.seats$size, cv.seats$dev, type='b')
```



Pruning the tree to 10 terminal nodes improves MSE.