STAT 2450 Assignment 6 (40 points)

Alice Liu

Banner: B00783546

1. ISLR, chapter 8, problem 9.

- YOU HAVE TO TREAT ALL QUESTIONS (a. to k.)
- USE the ISLR BOOK : http://faculty.marshall.usc.edu/gareth-james/ISL/ISLR%20Seventh%20Printing.pdf

a)

```
set.seed(666)
library(ISLR)
index=sample(1:nrow(OJ),800,replace=F)
OJtrain=OJ[index,]
OJtest=OJ[-index,]

PredictTrain = OJ$Purchase[index]
PredictTest = OJ$Purchase[-index]
```

b)

```
library(tree)
# OJtraintree=tree( enter your model specification here ,data=OJtrain)
OJtraintree=tree(Purchase~., data=OJtrain)
OJsummary=summary(OJtraintree)
OJsummary
```

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = OJtrain)
## Variables actually used in tree construction:
## [1] "LoyalCH"       "SalePriceMM"   "PriceDiff"     "ListPriceDiff"
## Number of terminal nodes:  8
## Residual mean deviance:  0.7507 = 594.5 / 792
## Misclassification error rate: 0.1538 = 123 / 800
```

(4 points for fitting tree, reporting training error rate and number terminal nodes. Seed wasn't specified prior to "sample" command, so different individuals are likely to have different trees.)

We can know that the tree has 8 terminal and the training error rate is 0.1538.

(c) Type in the name of the tree object in order to get a detailed text output. Pick one of the terminal nodes, and interpret the information displayed.

```
OJtraintree
```

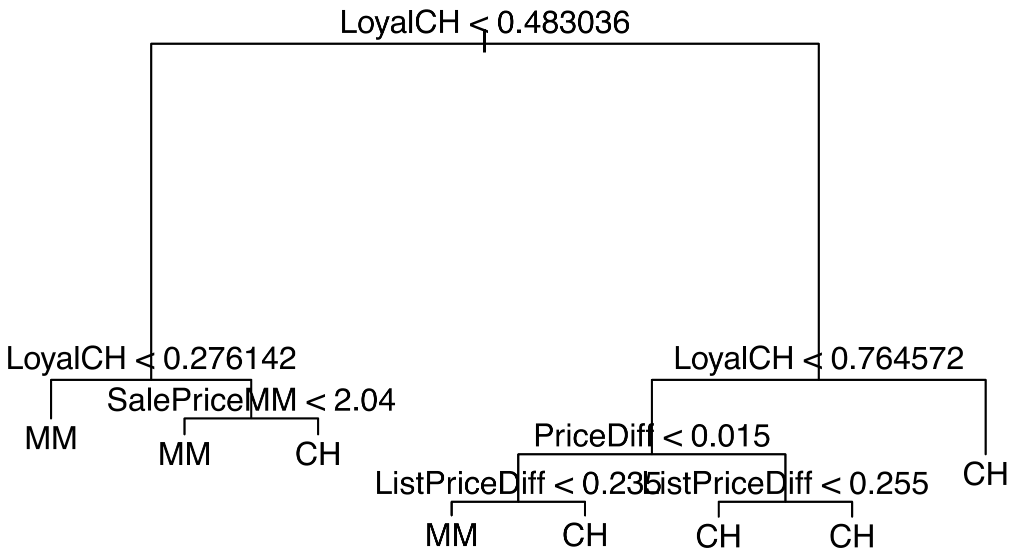
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 1076.00 CH ( 0.60125 0.39875 )
##    2) LoyalCH < 0.483036 304  325.60 MM ( 0.22697 0.77303 )
##      4) LoyalCH < 0.276142 163  104.70 MM ( 0.09816 0.90184 ) *
##      5) LoyalCH > 0.276142 141  186.70 MM ( 0.37589 0.62411 )
##       10) SalePriceMM < 2.04 79   87.16 MM ( 0.24051 0.75949 ) *
##       11) SalePriceMM > 2.04 62   85.37 CH ( 0.54839 0.45161 ) *
##    3) LoyalCH > 0.483036 496  451.20 CH ( 0.83065 0.16935 )
##      6) LoyalCH < 0.764572 232  287.40 CH ( 0.68966 0.31034 )
##       12) PriceDiff < 0.015 75  101.00 MM ( 0.40000 0.60000 )
##         24) ListPriceDiff < 0.235 52   60.58 MM ( 0.26923 0.73077 ) *
##         25) ListPriceDiff > 0.235 23   28.27 CH ( 0.69565 0.30435 ) *
##       13) PriceDiff > 0.015 157  144.10 CH ( 0.82803 0.17197 )
##         26) ListPriceDiff < 0.255 67   81.69 CH ( 0.70149 0.29851 ) *
##         27) ListPriceDiff > 0.255 90   49.20 CH ( 0.92222 0.07778 ) *
##      7) LoyalCH > 0.764572 264   97.63 CH ( 0.95455 0.04545 ) *
```

For the terminal node: "24) ListPriceDiff $<$ 0.235 52 60.58 MM (0.26923 0.73077)*" This node contains 52 observations. The prediction for this node is MM. 73.077% of the 52 observations are, in truth, MM.

(3 points for sensible interpretation for the node chosen.)

(d) Create a plot of the tree, and interpret the results.

```
plot(OJtraintree)
text(OJtraintree)
```



When (LoyalCH<0.483036 AND LoyalCH<0.276142), then the predicton is MM. When (LoyalCH<0.483036 AND LoyalCH>=0.276142 AND SalePriceMM<2.04), then the predicton is MM. When (LoyalCH<0.483036 AND LoyalCH>=0.276142 AND SalePriceMM>=2.04), then the predicton is CH. WHen (LoyalCH>=0.483036 AND LoyalCH<0.764572 AND PriceDiff<0.015 AND ListPriceDiff<0.235),then the predicton is MM. WHen (LoyalCH>=0.483036 AND LoyalCH<0.764572 AND PriceDiff<0.015 AND ListPriceDiff>=0.235),then the predicton is CH. WHen (LoyalCH>=0.483036 AND LoyalCH<0.764572 AND PriceDiff>=0.015 AND ListPriceDiff<0.255),then the predicton is CH. WHen (LoyalCH>=0.483036 AND LoyalCH<0.764572 AND PriceDiff>=0.015 AND ListPriceDiff>=0.255),then the predicton is CH. WHen (LoyalCH>=0.483036 AND LoyalCH>=0.764572),then the predicton is CH.

(3 points for plot of tree with text.)

(e) Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

```
predictTest = predict(OJtraintree,newdata = OJtest, type = "class")
conftable=table(predictTest, PredictTest)
print(conftable)
```

```
##            PredictTest
## predictTest  CH  MM
##          CH 150  24
##          MM  22  74
```

```
testErrorRate = (sum(conftable)-sum(diag(conftable)))/sum(conftable)
testErrorRate
```

```
## [1] 0.1703704
```

The test error rate is 0.1703704. (4 points: 3 for the confusion matrix, 1 for reporting the correct test error rate)

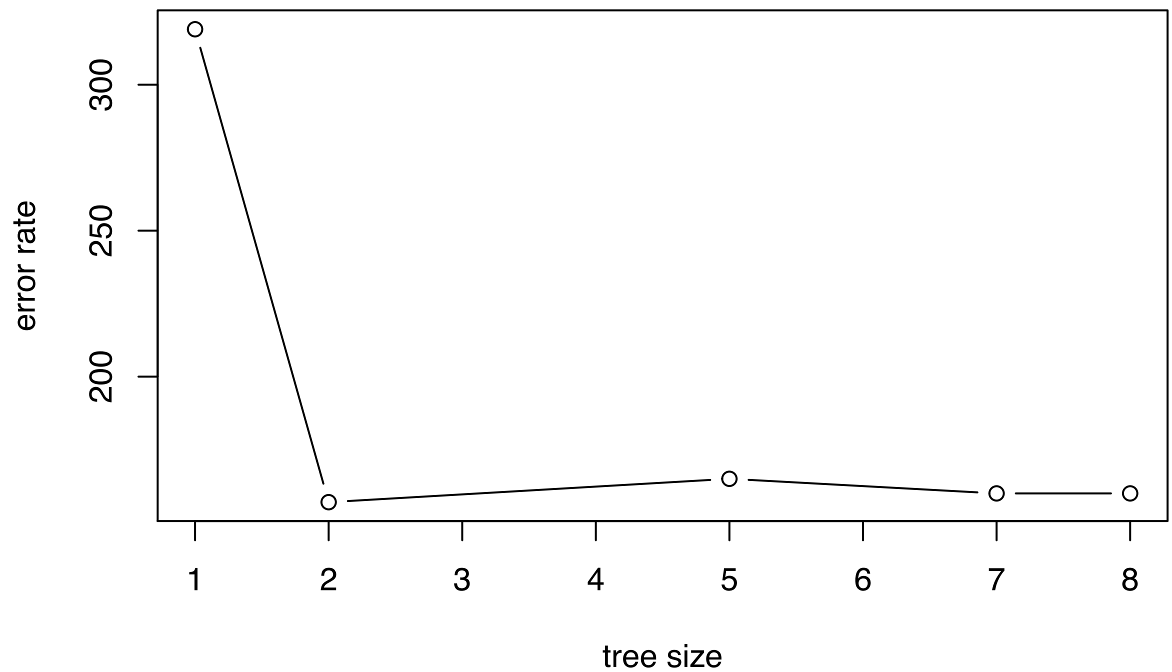
(f) Apply the cv.tree() function to the training set in order to determine the optimal tree size.

```r
#cv is cross validation
cvtree = cv.tree(OJtraintree, FUN = prune.misclass)
cvtree
```

```
## $size
## [1] 8 7 5 2 1
##
## $dev
## [1] 160 160 165 157 319
##
## $k
## [1] -Inf    0    3    8  166
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

(g) Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.

```r
plot(cvtree$size, cvtree$dev,type = 'b', xlab = 'tree size', ylab = 'error rate')
```



(5 points for the plot)

(h) Which tree size corresponds to the lowest cross-validated classification error rate?

```
besttreesize = cvtree$size[cvtree$dev == min(cvtree$dev)]
print("The best tree size is: ")
```

```
## [1] "The best tree size is: "
```

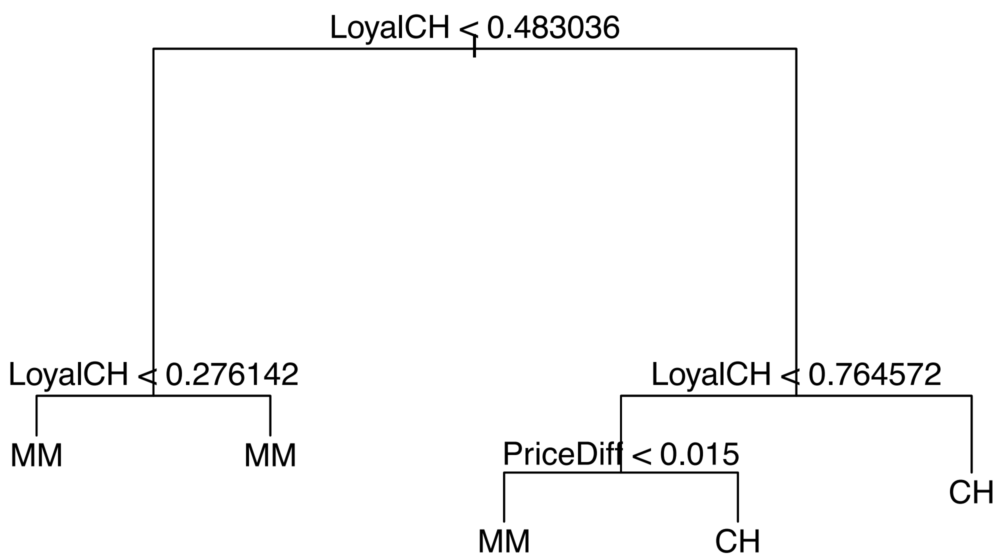
```
besttreesize
```

```
## [1] 2
```

The tree with 2 terminal nodes has lowest cross-validated classification error rate. (3 points for reporting the correct error rate)

(i) Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.

```
cvprunetree = prune.tree(OJtraintree, best = 5)
plot(cvprunetree)
text(cvprunetree)
```



The tree has 5 terminal nodes. (3 points for the plot, with text)

j) Compare training error rates between pruned and unpruned trees. Which is higher?

```
predictunprune = predict(OJtraintree, newdata = data.frame(OJtrain), type = "class")
predictprune = predict(cvprunetree, newdata = data.frame(OJtrain), type = "class")

tableJ1 = table(PredictTrain,predictunprune)
print("training error: confusion matrix for unpruned tree")
```

```
## [1] "training error: confusion matrix for unpruned tree"
```

```
print(tableJ1)
```

```
##              predictunprune
## PredictTrain  CH  MM
##           CH 432  49
##           MM  74 245
```

```
unprunnetrainrate = (sum(tableJ1)-sum(diag(tableJ1)))/sum(tableJ1)
unprunnetrainrate
```

```
## [1] 0.15375
```

```r
tableJ2 = table(PredictTrain,predictprune)
print("training error: confusion matrix for pruned tree")
```

```
## [1] "training error: confusion matrix for pruned tree"
```

```r
print(tableJ2)
```

```
##              predictprune
## PredictTrain  CH  MM
##           CH 382  99
##           MM  39 280
```

```r
prunnetrainrate = (sum(tableJ2)-sum(diag(tableJ2)))/sum(tableJ2)
prunnetrainrate
```

```
## [1] 0.1725
```

0.1725 > 0.15375. So, the training error rate of pruned tree is higher than unpruned tree. (3 points) Conclusion may differ if a different initial seed was used.

k) Compare test error rates between pruned and unpruned trees. Which is higher?

```r
predicunprunetest = predict(OJtraintree, newdata = data.frame(OJtest), type = "class")
predictprunetest = predict(cvprunetree, newdata = data.frame(OJtest), type = "class")

tableK1 = table(PredictTest, predicunprunetest)
print("testing error: confusion matrix for unpruned tree")
```

```
## [1] "testing error: confusion matrix for unpruned tree"
```

```r
print(tableK1)
```

```
##            predicunprunetest
## PredictTest  CH  MM
##          CH 150  22
##          MM  24  74
```

```r
testunprunerate = (sum(tableK1)-sum(diag(tableK1)))/sum(tableK1)
testunprunerate
```

```
## [1] 0.1703704
```

```r
tableK2 = table(PredictTest, predictprunetest)
print("testing error: confusion matrix for pruned tree")
```

```
## [1] "testing error: confusion matrix for pruned tree"
```

```r
print(tableK2)
```

```
##            predictprunetest
## PredictTest  CH  MM
##          CH 131  41
##          MM  14  84
```

```r
testprunerate = (sum(tableK2)-sum(diag(tableK2)))/sum(tableK2)
testprunerate
```

```
## [1] 0.2037037
```

$0.1703704 < 0.2037037$, so the training error rate of unpruned tree is higher than pruned tree. (2 points) Conclusion may differ if a different initial seed was used.

2. ISLR, chapter 8, problem 4.

Sketch the tree (part a), and the partition of the predictor space (part b), by hand.

(10 points, 5 for part a, 5 for part b)

You can draw your tree by hand on paper, take a picture and embed it in your Rmd with the image tag