Assignment: Chapter 8 Exercises (Week 6)

Amol Gote

08/16/2020

# Problem 9 This problem involves the OJ data set which is part of the ISLR package.

# 9a Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

training\_oj\_subset <- sample(nrow(OJ), 800)  
training\_oj\_ds = OJ[training\_oj\_subset, ]  
test\_oj\_ds = OJ[-training\_oj\_subset, ]  
nrow(training\_oj\_ds)

## [1] 800

nrow(test\_oj\_ds)

## [1] 270

Training dataset size 800

# 9b Fit a tree to the training data, with Purchase as the response and the other variables as predictors. Use the summary() function to produce summary statistics about the tree, and describe the results obtained. What is the training error rate? How many terminal nodes does the tree have?

tree\_oj <- tree(Purchase∼., OJ ,subset = training\_oj\_subset )  
summary(tree\_oj)

##   
## Classification tree:  
## tree(formula = Purchase ~ ., data = OJ, subset = training\_oj\_subset)  
## Variables actually used in tree construction:  
## [1] "LoyalCH" "PriceDiff" "WeekofPurchase" "ListPriceDiff"   
## Number of terminal nodes: 8   
## Residual mean deviance: 0.7368 = 583.6 / 792   
## Misclassification error rate: 0.16 = 128 / 800

1. Tree has 8 terminal nodes
2. It uses 4 variables LoyalCH, PriceDiff, WeekofPurchase, ListPriceDiff
3. Training error rate is 0.16

# 9c 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.

tree\_oj

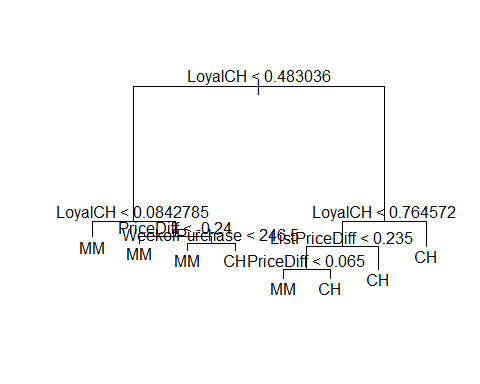
## node), split, n, deviance, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 800 1055.000 CH ( 0.62875 0.37125 )   
## 2) LoyalCH < 0.483036 295 334.500 MM ( 0.25424 0.74576 )   
## 4) LoyalCH < 0.0842785 84 32.160 MM ( 0.04762 0.95238 ) \*  
## 5) LoyalCH > 0.0842785 211 269.500 MM ( 0.33649 0.66351 )   
## 10) PriceDiff < -0.24 25 8.397 MM ( 0.04000 0.96000 ) \*  
## 11) PriceDiff > -0.24 186 246.400 MM ( 0.37634 0.62366 )   
## 22) WeekofPurchase < 246.5 98 111.300 MM ( 0.25510 0.74490 ) \*  
## 23) WeekofPurchase > 246.5 88 121.900 CH ( 0.51136 0.48864 ) \*  
## 3) LoyalCH > 0.483036 505 431.200 CH ( 0.84752 0.15248 )   
## 6) LoyalCH < 0.764572 241 281.000 CH ( 0.73029 0.26971 )   
## 12) ListPriceDiff < 0.235 98 135.800 MM ( 0.48980 0.51020 )   
## 24) PriceDiff < 0.065 62 77.970 MM ( 0.32258 0.67742 ) \*  
## 25) PriceDiff > 0.065 36 38.140 CH ( 0.77778 0.22222 ) \*  
## 13) ListPriceDiff > 0.235 143 96.010 CH ( 0.89510 0.10490 ) \*  
## 7) LoyalCH > 0.764572 264 97.630 CH ( 0.95455 0.04545 ) \*

* + Indicates that it is terminal node.

1. Terminal node 24,
   1. Variable used to split us PriceDiff, split is on PriceDiff < 0.065
   2. There are 62 observations or records or points in this terminal node.
   3. Deviance is 77.97
   4. Prediction is for “MM”
   5. Fraction of observations in this branch that takes on values of “CH” and “MM” = (0.32258, 0.67742)

# 9d Create a plot of the tree, and interpret the results.

plot(tree\_oj)  
text(tree\_oj, pretty=0)



1. The prime indicator/factor for Purchase appears to LoyalCH
2. LoyalCH Values less than 0.483036 is been primarily classified as MM
3. LoyalCH Values greater than than 0.764572 is been primarily classified as CH
4. LoyalCH Values less than 0.764572 will be classified based on 2 factors ListPriceDiff and then PriceDiff
5. LoyalCH Values less than 0.764572 will be primarily calssified as CH with exception of ListPriceDiff < 0.235 and PriceDiff < 0.065

# 9e. 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?

prediction <- predict(tree\_oj, test\_oj\_ds, type="class")  
table(prediction, test\_oj\_ds$Purchase)

##   
## prediction CH MM  
## CH 139 45  
## MM 11 75

test\_error\_rate <- mean(prediction != test\_oj\_ds$Purchase)  
test\_error\_rate\_percentage <- test\_error\_rate \* 100  
test\_error\_rate\_percentage

## [1] 20.74074

test\_accuracy\_rate <- mean(prediction == test\_oj\_ds$Purchase)  
test\_accuracy\_rate\_percentage <- test\_accuracy\_rate \* 100  
test\_accuracy\_rate\_percentage

## [1] 79.25926

Test error rate is 0.2074074 which is 20.7407407%

Test accuracy is 0.7925926 which is 79.2592593%

# 9f Apply the cv.tree() function to the training set in order to determine the optimal tree size.

optimal\_tree <- cv.tree(tree\_oj, FUN = prune.tree)  
optimal\_tree

## $size  
## [1] 8 7 6 5 4 3 2 1  
##   
## $dev  
## [1] 719.9450 703.7707 704.8605 693.3820 722.7332 796.9987 793.6833  
## [8] 1057.7367  
##   
## $k  
## [1] -Inf 13.10430 14.77100 19.70606 32.80981 49.16307 52.62601  
## [8] 289.64352  
##   
## $method  
## [1] "deviance"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

Optimal tree size is 8.