Homework Week 6

Kai Li

2/15/2017

## (a) Created training and test sets

First is a summary of the OJ dataset. And because of the fact that the three variables, namely STORE, Store7, and StoreID, are repetitive, only STORE is kept in the dataset that is used for analysis.

OJ <- OJ[,-c(3, 14)]  
OJ$Purchase <- as.factor(OJ$Purchase)  
OJ$STORE <- as.factor(OJ$STORE)  
  
describeBy(OJ)

## Warning in describeBy(OJ): no grouping variable requested

## vars n mean sd median trimmed mad min max  
## Purchase\* 1 1070 1.39 0.49 1.00 1.36 0.00 1.00 2.00  
## WeekofPurchase 2 1070 254.38 15.56 257.00 254.79 20.76 227.00 278.00  
## PriceCH 3 1070 1.87 0.10 1.86 1.87 0.15 1.69 2.09  
## PriceMM 4 1070 2.09 0.13 2.09 2.10 0.13 1.69 2.29  
## DiscCH 5 1070 0.05 0.12 0.00 0.02 0.00 0.00 0.50  
## DiscMM 6 1070 0.12 0.21 0.00 0.08 0.00 0.00 0.80  
## SpecialCH 7 1070 0.15 0.35 0.00 0.06 0.00 0.00 1.00  
## SpecialMM 8 1070 0.16 0.37 0.00 0.08 0.00 0.00 1.00  
## LoyalCH 9 1070 0.57 0.31 0.60 0.58 0.39 0.00 1.00  
## SalePriceMM 10 1070 1.96 0.25 2.09 1.99 0.15 1.19 2.29  
## SalePriceCH 11 1070 1.82 0.14 1.86 1.83 0.15 1.39 2.09  
## PriceDiff 12 1070 0.15 0.27 0.23 0.17 0.15 -0.67 0.64  
## PctDiscMM 13 1070 0.06 0.10 0.00 0.04 0.00 0.00 0.40  
## PctDiscCH 14 1070 0.03 0.06 0.00 0.01 0.00 0.00 0.25  
## ListPriceDiff 15 1070 0.22 0.11 0.24 0.23 0.09 0.00 0.44  
## STORE\* 16 1070 2.63 1.43 3.00 2.54 1.48 1.00 5.00  
## range skew kurtosis se  
## Purchase\* 1.00 0.45 -1.80 0.01  
## WeekofPurchase 51.00 -0.21 -1.28 0.48  
## PriceCH 0.40 0.06 -0.78 0.00  
## PriceMM 0.60 -1.46 2.16 0.00  
## DiscCH 0.50 2.41 4.89 0.00  
## DiscMM 0.80 1.59 1.47 0.01  
## SpecialCH 1.00 1.98 1.94 0.01  
## SpecialMM 1.00 1.84 1.37 0.01  
## LoyalCH 1.00 -0.28 -1.06 0.01  
## SalePriceMM 1.10 -0.79 -0.53 0.01  
## SalePriceCH 0.70 -0.95 0.94 0.00  
## PriceDiff 1.31 -0.76 0.45 0.01  
## PctDiscMM 0.40 1.54 1.25 0.00  
## PctDiscCH 0.25 2.42 4.83 0.00  
## ListPriceDiff 0.44 -0.64 -0.24 0.00  
## STORE\* 4.00 0.25 -1.30 0.04

## Define train and test sets  
  
set.seed(1)  
train <- sample(nrow(OJ), 800)  
OJ.train <- OJ[train,]  
OJ.test <- OJ[-train,]

## (b) Grow and interpret the tree

## Growing the tree  
  
tree.OJ <- tree(Purchase ~ .,  
 data = OJ.train)  
summary(tree.OJ)

##   
## Classification tree:  
## tree(formula = Purchase ~ ., data = OJ.train)  
## Variables actually used in tree construction:  
## [1] "LoyalCH" "PriceDiff" "SpecialCH" "ListPriceDiff"  
## Number of terminal nodes: 8   
## Residual mean deviance: 0.7305 = 578.6 / 792   
## Misclassification error rate: 0.165 = 132 / 800

Based on the summary of the model, there are 8 terminal nodes in the model. The error rate is 16.5%. Moreover, out of the 15 predicting variables, only four are used in the tree.

## (c) Interpret nodes

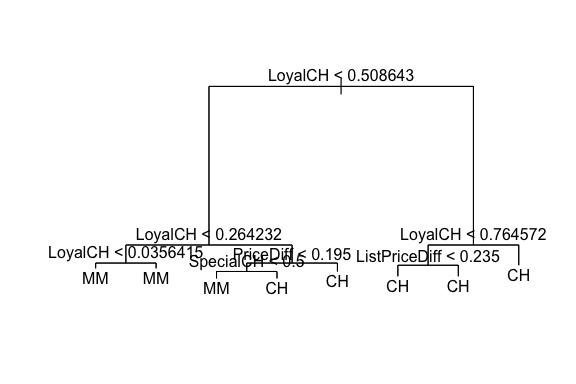
tree.OJ

## node), split, n, deviance, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 800 1064.00 CH ( 0.61750 0.38250 )   
## 2) LoyalCH < 0.508643 350 409.30 MM ( 0.27143 0.72857 )   
## 4) LoyalCH < 0.264232 166 122.10 MM ( 0.12048 0.87952 )   
## 8) LoyalCH < 0.0356415 57 10.07 MM ( 0.01754 0.98246 ) \*  
## 9) LoyalCH > 0.0356415 109 100.90 MM ( 0.17431 0.82569 ) \*  
## 5) LoyalCH > 0.264232 184 248.80 MM ( 0.40761 0.59239 )   
## 10) PriceDiff < 0.195 83 91.66 MM ( 0.24096 0.75904 )   
## 20) SpecialCH < 0.5 70 60.89 MM ( 0.15714 0.84286 ) \*  
## 21) SpecialCH > 0.5 13 16.05 CH ( 0.69231 0.30769 ) \*  
## 11) PriceDiff > 0.195 101 139.20 CH ( 0.54455 0.45545 ) \*  
## 3) LoyalCH > 0.508643 450 318.10 CH ( 0.88667 0.11333 )   
## 6) LoyalCH < 0.764572 172 188.90 CH ( 0.76163 0.23837 )   
## 12) ListPriceDiff < 0.235 70 95.61 CH ( 0.57143 0.42857 ) \*  
## 13) ListPriceDiff > 0.235 102 69.76 CH ( 0.89216 0.10784 ) \*  
## 7) LoyalCH > 0.764572 278 86.14 CH ( 0.96403 0.03597 ) \*

According to the results displayed above, the eight node with an asterisk are terminal nodes. The first of them, #8, is the cluster whose LoyalCH smaller than both 0.508, 0.264, and 0.035. There are 57 observations falling into this category. And the expected classification for this category is MM, which is normal.

## (d) Ploting the tree

plot(tree.OJ)  
text(tree.OJ, pretty=0)



The plot shows, in a direct way, that all the customers whose Loyalty to CH under 0.264 resort to MM. And all of them whose loyalty to CH above 0.508 go to CH. For those in between, price difference and the availability of spcial offer of CH are the two most important factors.

## (e) Predict the response

OJ.pred <- predict(tree.OJ,   
 OJ.test,  
 type="class")  
  
table(Prediction = OJ.pred,  
 Origin = OJ.test$Purchase)

## Origin  
## Prediction CH MM  
## CH 147 49  
## MM 12 62

accuracy\_rate\_1 <- round(mean(OJ.pred == OJ.test$Purchase), digits = 3) \* 100

Below is the confusion matrix of applying the tree model to the test set. The overall accuracy rate is 77.4%.

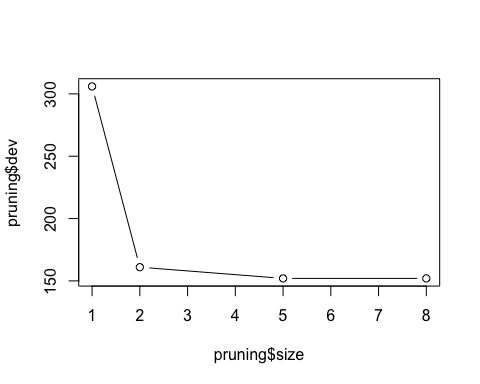
## (f) Prune the tree

Function cv.tree is applied to prune the tree.

set.seed(1)  
pruning <- cv.tree(tree.OJ,  
 FUN = prune.misclass)

## (g) & (h) Plot the results and the most optimal number

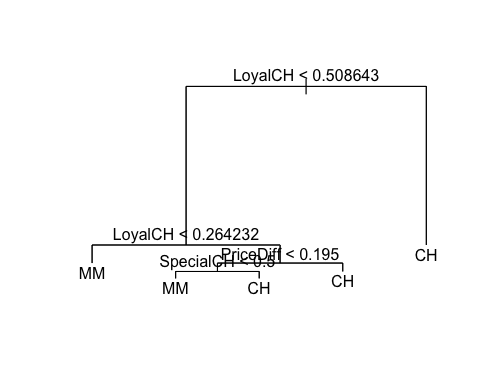
plot(pruning$size,  
 pruning$dev,  
 type="b")



Based on the plot, size = 5 might be the most optimal number of terminal node.

## (i) Pruned tree

tree.OJ.prune <- prune.misclass(tree.OJ,  
 best = 5)  
plot(tree.OJ.prune)  
text(tree.OJ.prune,  
 pretty=0)



## (j) Pruned accuracy rate

Based on the summary of the models per se, the two models have the same error rate.

OJ.pred.prune <- predict(tree.OJ.prune,   
 OJ.test,  
 type="class")  
  
summary(tree.OJ.prune)

##   
## Classification tree:  
## snip.tree(tree = tree.OJ, nodes = 3:4)  
## Variables actually used in tree construction:  
## [1] "LoyalCH" "PriceDiff" "SpecialCH"  
## Number of terminal nodes: 5   
## Residual mean deviance: 0.8256 = 656.4 / 795   
## Misclassification error rate: 0.165 = 132 / 800

## (K) test accuracy rate

table(Prediction = OJ.pred.prune,  
 Origin = OJ.test$Purchase)

## Origin  
## Prediction CH MM  
## CH 147 49  
## MM 12 62

accuracy\_rate\_2 <- round(mean(OJ.pred.prune == OJ.test$Purchase), digits = 3) \* 100

As shown in the confusion table below, the two tables reach the exact same accuracy rate on the test set.