HW # 7

Bilal Gilani

11/13/2020

1.

Please see attached:

2.

```
majority <- c(0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75) ## 10 values given

table(majority > 0.5)
```

```
##
## FALSE  TRUE
##     4     6
```

```
## 6 for Red and 4 for Green
mean(majority)
```

```
## [1] 0.45
```

```
## average of the 10 values is 0.45
```

In the case of a majority vote, X would belong in the Red class. However, in the case of taking the average probability, X does **NOT** belong in the Red class.

3.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.3
```

```
data("OJ")
attach(OJ)
head(OJ)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            237       1    1.75    1.99   0.00    0.0         0
## 2       CH            239       1    1.75    1.99   0.00    0.3         0
## 3       CH            245       1    1.86    2.09   0.17    0.0         0
## 4       MM            227       1    1.69    1.69   0.00    0.0         0
## 5       CH            228       7    1.69    1.69   0.00    0.0         0
## 6       CH            230       7    1.69    1.99   0.00    0.0         0
##   SpecialMM  LoyalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0 0.500000        1.99        1.75      0.24     No  0.000000
## 2         1 0.600000        1.69        1.75     -0.06     No  0.150754
## 3         0 0.680000        2.09        1.69      0.40     No  0.000000
## 4         0 0.400000        1.69        1.69      0.00     No  0.000000
## 5         0 0.956535        1.69        1.69      0.00    Yes  0.000000
## 6         1 0.965228        1.99        1.69      0.30    Yes  0.000000
##   PctDiscCH ListPriceDiff STORE
## 1  0.000000          0.24     1
## 2  0.000000          0.24     1
## 3  0.091398          0.23     1
## 4  0.000000          0.00     1
## 5  0.000000          0.00     0
## 6  0.000000          0.30     0
```

a.

```
set.seed(1)

Z <- sample(1:nrow(OJ), 800)
train <- OJ[Z, ]
test <- OJ[-Z, ]
```

b.

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.6.3
```

```
tree.fit <- tree(Purchase ~ ., data = train)
summary(tree.fit)
```

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = train)
## Variables actually used in tree construction:
## [1] "LoyalCH"       "PriceDiff"     "SpecialCH"     "ListPriceDiff"
## [5] "PctDiscMM"
## Number of terminal nodes:  9
## Residual mean deviance:  0.7432 = 587.8 / 791
## Misclassification error rate: 0.1588 = 127 / 800
```

There are 9 terminal nodes, with the training error rate being 0.1588.

c.

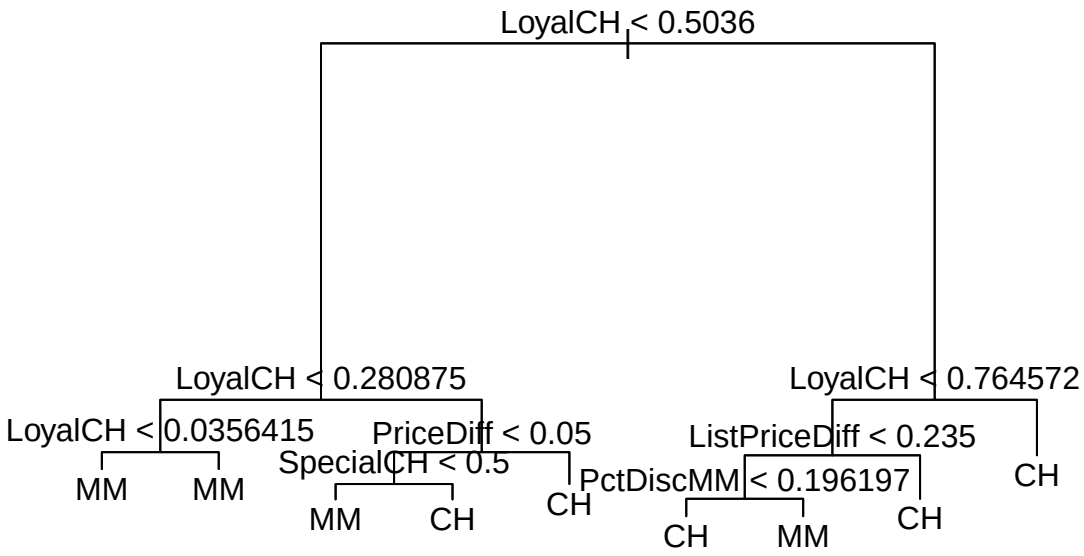
```
tree.fit
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 1073.00 CH ( 0.60625 0.39375 )
##    2) LoyalCH < 0.5036 365  441.60 MM ( 0.29315 0.70685 )
##      4) LoyalCH < 0.280875 177  140.50 MM ( 0.13559 0.86441 )
##        8) LoyalCH < 0.0356415 59   10.14 MM ( 0.01695 0.98305 ) *
##        9) LoyalCH > 0.0356415 118  116.40 MM ( 0.19492 0.80508 ) *
##      5) LoyalCH > 0.280875 188  258.00 MM ( 0.44149 0.55851 )
##       10) PriceDiff < 0.05 79   84.79 MM ( 0.22785 0.77215 )
##         20) SpecialCH < 0.5 64   51.98 MM ( 0.14062 0.85938 ) *
##         21) SpecialCH > 0.5 15   20.19 CH ( 0.60000 0.40000 ) *
##       11) PriceDiff > 0.05 109  147.00 CH ( 0.59633 0.40367 ) *
##    3) LoyalCH > 0.5036 435  337.90 CH ( 0.86897 0.13103 )
##      6) LoyalCH < 0.764572 174  201.00 CH ( 0.73563 0.26437 )
##       12) ListPriceDiff < 0.235 72   99.81 MM ( 0.50000 0.50000 )
##         24) PctDiscMM < 0.196197 55   73.14 CH ( 0.61818 0.38182 ) *
##         25) PctDiscMM > 0.196197 17   12.32 MM ( 0.11765 0.88235 ) *
##       13) ListPriceDiff > 0.235 102   65.43 CH ( 0.90196 0.09804 ) *
##      7) LoyalCH > 0.764572 261   91.20 CH ( 0.95785 0.04215 ) *
```

The first terminal node that pops up is 8 so I will use that one. Number 8 has a split criterion of **LoyalCH** < 0.036. This terminal node has 59 observations, has a deviance of 10.14, and selects Minute Maid as it's final selection. About 1.7% of the observations take on the value of Citrus Hill while the remaining 98.3% take on the value of Minute Maid.

d.

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



The plot shows that when **LoyalCH** is below 0.5036, the majority of the preferred choices are Minute Maid. Citrus Hill is only selected if the **Price Difference** is above 0.05 OR if in the case that Price Difference is less than 0.05, **SpecialCH** is greater than 0.5. On the other side, Citrus Hill is the preferred choice except in the case where **LoyalCH** is less than 0.765, the **ListPriceDiff** is less than 0.235 and finally when **PctDiscMM** is greater than 0.196. ### e.

```
preds.tree <- predict(tree.fit, test, type = "class")

table(preds.tree, test$Purchase) ## create confusion matrix
```

```
##
## preds.tree  CH  MM
##         CH 160  38
##         MM   8  64
```

```
1 - mean(preds.tree == test$Purchase) ## test error rate
```

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

The test error rate is about 17%.

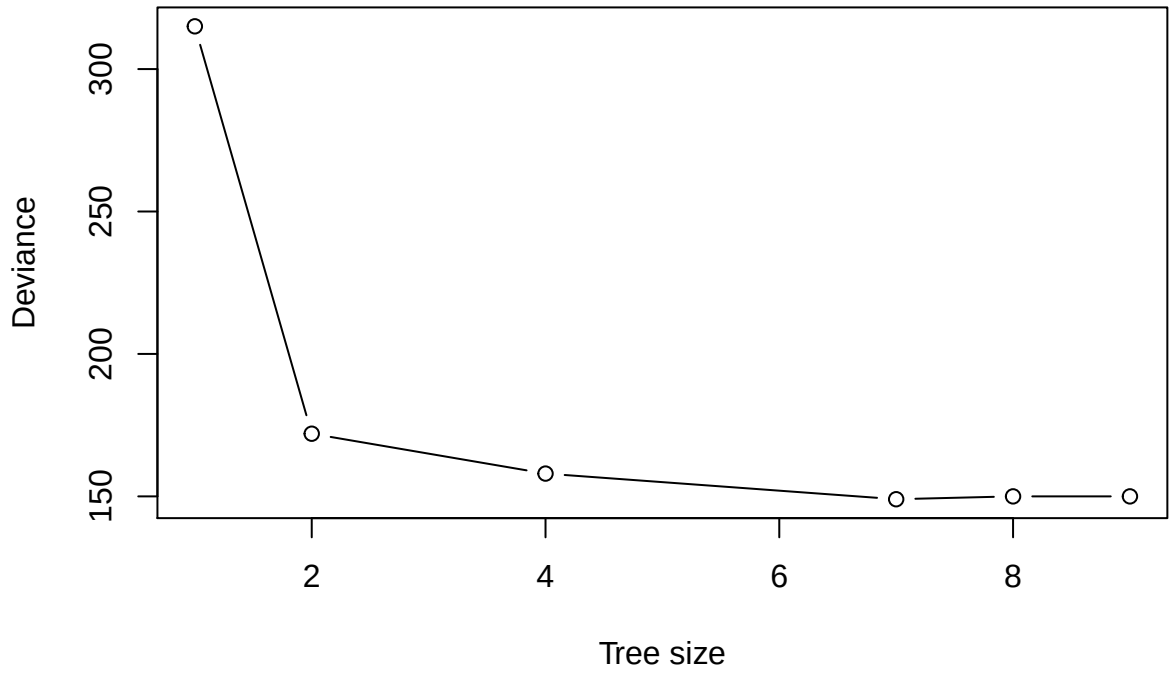
f.

```
cv.tree.fit <- cv.tree(tree.fit, FUN = prune.misclass) # doing cv on model that was set on training set
cv.tree.fit
```

```
## $size
## [1] 9 8 7 4 2 1
##
## $dev
## [1] 150 150 149 158 172 315
##
## $k
## [1]       -Inf   0.000000   3.000000   4.333333  10.500000 151.000000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

g.

```
plot(cv.tree.fit$size, cv.tree.fit$dev, type = "b",
     xlab = "Tree size", ylab = "Deviance") ## cv tree error plot
```



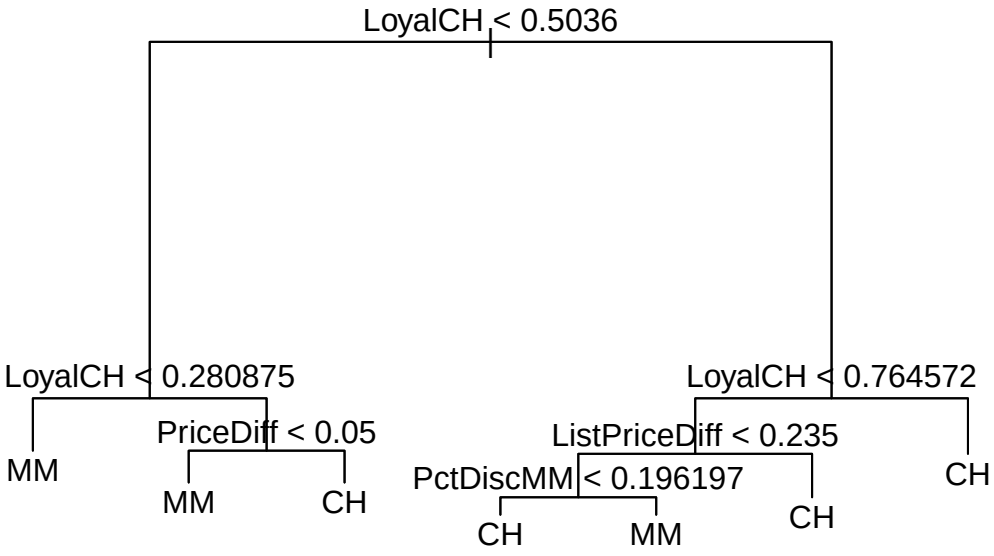
h.

7 terminal nodes appears to have the lowest level of deviance, which differs from the 9 terminal nodes in the previous case.

i.

```
prune.tree.fit <- prune.misclass(tree.fit, best = 7) ## create pruned tree object

plot(prune.tree.fit)
text(prune.tree.fit, pretty = 0)
```



j.

```
summary(tree.fit)
```

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = train)
## Variables actually used in tree construction:
## [1] "LoyalCH"       "PriceDiff"     "SpecialCH"     "ListPriceDiff"
```

```
## [5] "PctDiscMM"
## Number of terminal nodes:  9
## Residual mean deviance:  0.7432 = 587.8 / 791
## Misclassification error rate: 0.1588 = 127 / 800
```

```
summary(prune.tree.fit)
```

```
##
## Classification tree:
## snip.tree(tree = tree.fit, nodes = c(4L, 10L))
## Variables actually used in tree construction:
## [1] "LoyalCH"       "PriceDiff"     "ListPriceDiff" "PctDiscMM"
## Number of terminal nodes:  7
## Residual mean deviance:  0.7748 = 614.4 / 793
## Misclassification error rate: 0.1625 = 130 / 800
```

```
## comparing error rates
```

The error rate of the pruned tree (0.1625) is higher than the error rate of our non-pruned treet's error rate (0.1588).

k.

```
preds.pruned <- predict(prune.tree.fit, test, type = "class")

table(preds.pruned, test$Purchase) ## again, create a confusion matrix
```

```
##
## preds.pruned  CH  MM
##           CH 160  36
##           MM   8  66
```

```
1 - mean(preds.pruned == test$Purchase) ## test error rate
```

```
## [1] 0.162963
```

The test error rate of the pruned tree is lower than the test error rate of the original (pre-pruned) tree by 1% (16% vs. 17%).