STA 545 Statistical Data Mining I, Fall 2020

Homework 9

Stella Liao

November 18, 2020

Problem 1

D_P denote the parent node; D_{left} denote the left child node; D_{right} denote the left child node.

1) Entropy as impurity measure, calculate Information Gain(IG_H)

a. Split 1

$$IG_H(D_P) = -\frac{7}{10} \times \log_2(\frac{7}{10}) - \frac{3}{10} \times \log_2(\frac{3}{10})$$

```
dp = -7/10*log2(7/10)-3/10*log2(3/10)
dp
```

```
## [1] 0.8812909
```

$$IG_H(D_{Left}) = -1 \times \log_2(1) - 0 = 0$$

$$IG_H(D_{right}) = -\frac{4}{7} \times \log_2(\frac{4}{7}) - \frac{3}{7} \times \log_2(\frac{3}{7})$$

```
dr_1 = -4/7*log2(4/7)-3/7*log2(3/7)
dr_1
```

```
## [1] 0.9852281
```

$$IG_H = IG_H(D_P) - \frac{3}{10} \times IG_H(D_{Left}) - \frac{7}{10} \times IG_H(D_{right})$$

```
print(dp - 0 - 7/10 * dr_1)
```

```
## [1] 0.1916312
```

b. Split 2

$$IG_H(D_{Left}) = -\frac{2}{3} \times \log_2(\frac{2}{3}) - \frac{1}{3} \times \log_2(\frac{1}{3})$$

```
dl_2 = -2/3*log2(2/3)-1/3*log2(1/3)
dl_2
```

```
## [1] 0.9182958
```

$$IG_H(D_{right}) = -\frac{5}{7} \times \log_2(\frac{5}{7}) - \frac{2}{7} \times \log_2(\frac{2}{7})$$

```r
dr_2 = -5/7*log2(5/7)-2/7*log2(2/7)
dr_2
```

```
## [1] 0.8631206
```

$$IG_H = IG_H(D_P) - \frac{3}{10} \times IG_H(D_{Left}) - \frac{7}{10} \times IG_H(D_{right})$$

```r
print(dp - 3/10 * dl_2 - 7/10 * dr_2)
```

```
## [1] 0.001617751
```

2) Missclassification error as impurity measure, calculate Information Gain(IG_E)

a. Split 1

$$IG_E(D_P) = 1 - \frac{7}{10} = \frac{3}{10}$$

$$IG_E(D_{Left}) = 1 - \frac{3}{3} = 0$$

$$IG_E(D_{right}) = 1 - \frac{4}{7} = \frac{3}{7}$$

$$IG_E = IG_E(D_P) - \frac{3}{10} \times IG_E(D_{Left}) - \frac{7}{10} \times IG_E(D_{right}) = \frac{3}{10} - \frac{3}{10} \times 0 - \frac{7}{10} \times \frac{3}{7} = 0$$

b. Split 2

$$IG_E(D_{Left}) = 1 - \frac{2}{3} = \frac{1}{3}$$

$$IG_E(D_{right}) = 1 - \frac{5}{7} = \frac{2}{7}$$

$$IG_E = IG_E(D_P) - \frac{3}{10} \times IG_E(D_{Left}) - \frac{7}{10} \times IG_E(D_{right}) = \frac{3}{10} - \frac{3}{10} \times \frac{1}{3} - \frac{7}{10} \times \frac{2}{7} = 0$$

Therefore, in terms of Entropy, we prefer Split 1, since the information gain of Split 1(0.1916) is greater than that of Split 2(0.0016); in terms of Misclassification error, both of the splits are a good choice, since the information gain of them are equal.

Problem 2

(a)

```r
library(ISLR)

set.seed(2)
train.num <- sample(nrow(OJ), 800)
OJ.train <- OJ[train.num , ]
OJ.test <- OJ[-train.num ,]
```

(b)

```r
library(tree)
tree.OJ <- tree(Purchase ~., OJ.train)
summary(tree.OJ)
```

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = OJ.train)
## Variables actually used in tree construction:
```

```
## [1] "LoyalCH"   "PriceDiff"
## Number of terminal nodes:  9
## Residual mean deviance:  0.7009 = 554.4 / 791
## Misclassification error rate: 0.1588 = 127 / 800
```

The summary above shows that `LoyalCH` and `PriceDiff` are the most important variables to determine whether the customer purchased Citrus Hill or Minute Maid Orange Juice; there are 9 terminal nodes in this tree model and the training error rate is 15.88%.

(c)

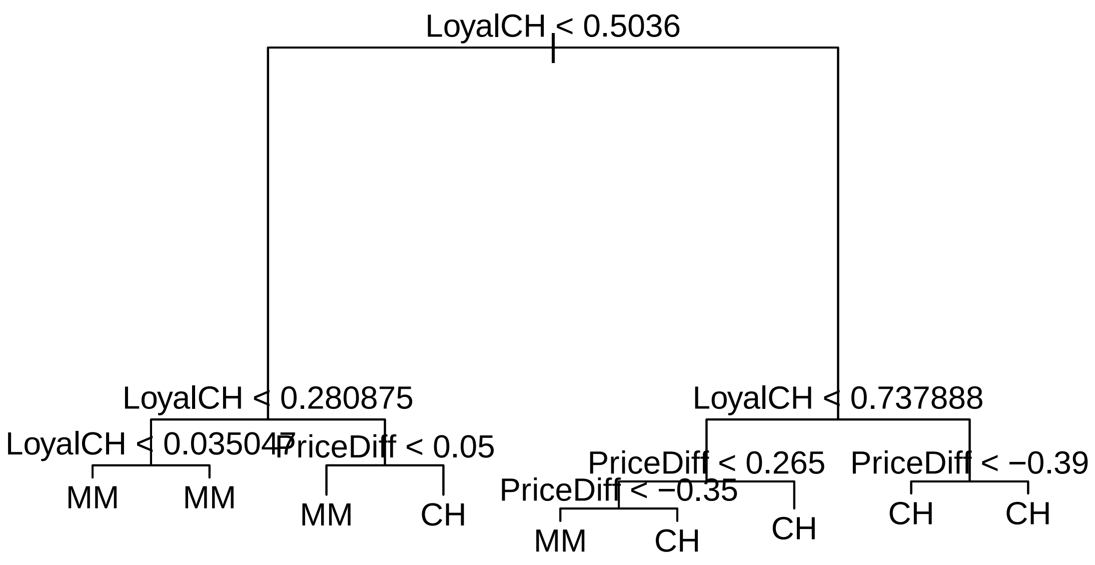
```r
tree.OJ
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 1068.00 CH ( 0.61250 0.38750 )
##    2) LoyalCH < 0.5036 359  422.80 MM ( 0.27577 0.72423 )
##      4) LoyalCH < 0.280875 172  127.60 MM ( 0.12209 0.87791 )
##        8) LoyalCH < 0.035047 56   10.03 MM ( 0.01786 0.98214 ) *
##        9) LoyalCH > 0.035047 116  106.60 MM ( 0.17241 0.82759 ) *
##      5) LoyalCH > 0.280875 187  254.10 MM ( 0.41711 0.58289 )
##       10) PriceDiff < 0.05 73   71.36 MM ( 0.19178 0.80822 ) *
##       11) PriceDiff > 0.05 114  156.30 CH ( 0.56140 0.43860 ) *
##    3) LoyalCH > 0.5036 441  311.80 CH ( 0.88662 0.11338 )
##      6) LoyalCH < 0.737888 168  191.10 CH ( 0.74405 0.25595 )
##       12) PriceDiff < 0.265 93  125.00 CH ( 0.60215 0.39785 )
##         24) PriceDiff < -0.35 12   10.81 MM ( 0.16667 0.83333 ) *
##         25) PriceDiff > -0.35 81  103.10 CH ( 0.66667 0.33333 ) *
##       13) PriceDiff > 0.265 75   41.82 CH ( 0.92000 0.08000 ) *
##      7) LoyalCH > 0.737888 273   65.11 CH ( 0.97436 0.02564 )
##       14) PriceDiff < -0.39 11   12.89 CH ( 0.72727 0.27273 ) *
##       15) PriceDiff > -0.39 262   41.40 CH ( 0.98473 0.01527 ) *
```

According to the last terminal node, we could see that when `LoyalCH` is greater than 0.738, then the customer would be almost always(about 97.44%) purchases Citrus Hill orange juice. In other words, if the loyal customers stay more loyal and it is more likely they will always purchase the juice whose brand they support.

(d)

```r
plot(tree.OJ, main='Decision Tree')
text(tree.OJ, pretty=0)
```



According to the tree, `LoyalCh` is the most important variable in the tree. If `LoyalCh`(customer brand loyalty for Citrus Hill Orange Juice) is smaller than 0.035, then the tree predicts that the customer will choose Minute Maid Orange Juice(MM); while if `LoyalCh` is greater than 0.74, the tree predicts that the customer will choose Citrus Hill Orange Juice(CH); and `PriceDiff` shows its power when `LoyalCh` is between 0.035 and 0.74, for example, when `LoyalCh` is between 0.035 and 0.281, if `PriceDiff` is smaller than 0.05, the tree model will predict that the customer will choose Minute Maid Orange Juice(MM).

(e)

```
pred.OJ <- predict(tree.OJ, OJ.test, type='class')

#confusion matrix
table(pred.OJ, OJ.test$Purchase)
```

```
##
## pred.OJ  CH  MM
##      CH 148  37
##      MM  15  70
```

```
#test error rate
mean(pred.OJ != OJ.test$Purchase)
```

```
## [1] 0.1925926
```

(f)

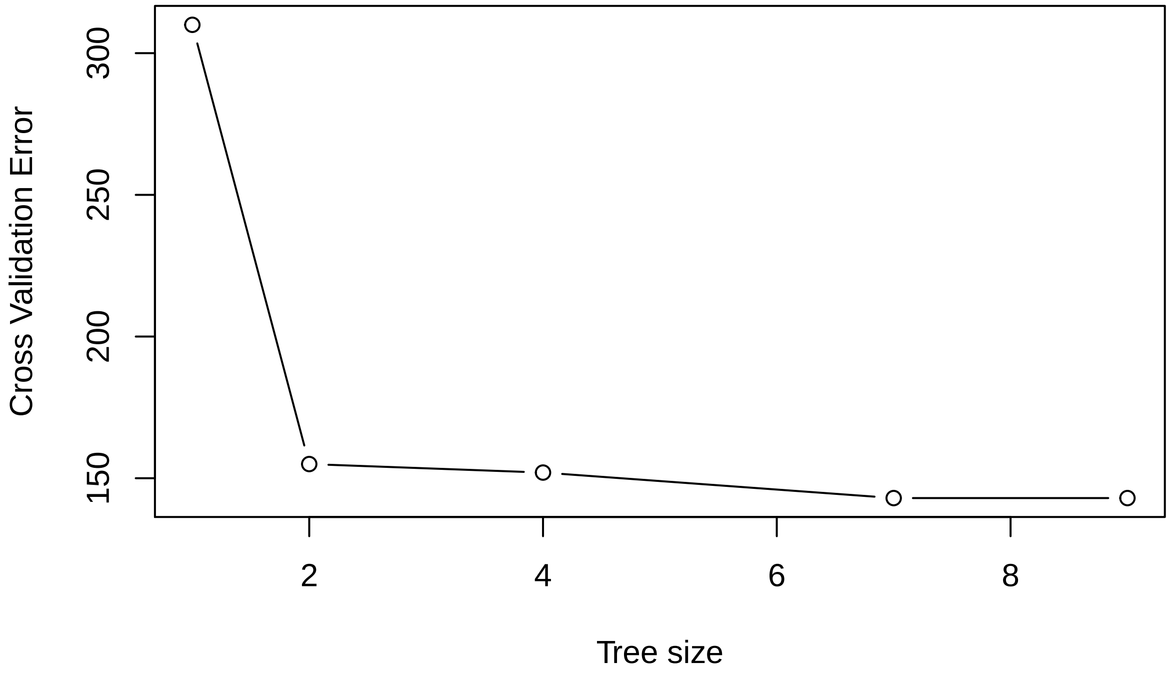
```
cv.OJ <- cv.tree(tree.OJ, FUN = prune.misclass)
cv.OJ
```

```
## $size
## [1] 9 7 4 2 1
##
## $dev
## [1] 143 143 152 155 310
##
## $k
## [1]       -Inf   0.000000   2.666667   7.000000 161.000000
##
## $method
## [1] "misclass"
```

```
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

(g)

```
plot(cv.OJ$size, cv.OJ$dev, type = "b", xlab = "Tree size", ylab = "Cross Validation Error")
```



(h) Based on the result of `cv.oj` and the plot above, we could see the cross validation error is smallest when the size is equal to 9 or 7. And to make the model simpler, we finally chose 7 as the optimal tree size.

(i)

```
prune.tree.OJ <- prune.misclass(tree.OJ, best = 7)
summary(prune.tree.OJ)
```

```
##
## Classification tree:
## snip.tree(tree = tree.OJ, nodes = c(4L, 7L))
## Variables actually used in tree construction:
## [1] "LoyalCH"   "PriceDiff"
## Number of terminal nodes:  7
## Residual mean deviance:  0.7266 = 576.2 / 793
## Misclassification error rate: 0.1588 = 127 / 800
```

(j)

```
pred.prune.train.OJ = predict(prune.tree.OJ, OJ.train, type='class')
prune.train.error <- mean(pred.prune.train.OJ != OJ.train$Purchase)

# training error of pruned tree
prune.train.error
```

```
## [1] 0.15875
```

The training error of the original tree is 15.88% and that of the pruned tree is around 15.88%, so there is no difference between them in this case.

(k)

```
pred.prune.test.OJ= predict(prune.tree.OJ, OJ.test, type='class')
prune.test.error <- mean(pred.prune.test.OJ != OJ.test$Purchase)

# test error of pruned tree
prune.test.error
```

```
## [1] 0.1925926
```

The test error of the original tree is about 19.26% and that of the pruned tree is around 19.26%, so there is no difference between them in this case.

Probelm 3

1) iris dataset

```
library(datasets)
data(iris)
set.seed(50321222)
train.num2=sample(1:nrow(iris),.7*nrow(iris),replace=FALSE)
iris.train = iris[train.num2,]
iris.test = iris[-train.num2,]
```

2)random forest model

```
library(randomForest)
```

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

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

```
rf.iris <- randomForest(Species~.,data=iris.train,ntree=10,proximity=TRUE)
rf.pred.iris<-predict(rf.iris,newdata=iris.test)
table(rf.pred.iris, iris.test$Species)
```

```
##
## rf.pred.iris setosa versicolor virginica
##   setosa         14          0         0
##   versicolor      0         16         2
##   virginica       0          1        12
```

```
rf.accuracy <- sum(rf.pred.iris == iris.test$Species)/length(iris.test$Species)

rf.accuracy
```

```
## [1] 0.9333333
```

3)LDA

```
library(MASS)
lda.iris <- lda(Species ~ ., data = iris.train)
lda.pred.iris <- predict(lda.iris,newdata = iris.test)

table(lda.pred.iris$class, iris.test$Species)
```

```
##
##              setosa versicolor virginica
##   setosa         14          0         0
##   versicolor      0         16         1
##   virginica       0          1        13
```

```r
lda.accuracy <- sum(lda.pred.iris$class == iris.test$Species)/length(iris.test$Species)

lda.accuracy
```

```
## [1] 0.9555556
```

4)Logistic regression

```r
library(nnet)
mulog.iris <- multinom(Species ~., data = iris.train)
```

```
## # weights:  18 (10 variable)
## initial  value 115.354290
## iter  10 value 11.269025
## iter  20 value 3.287536
## iter  30 value 2.745049
## iter  40 value 2.458528
## iter  50 value 2.279643
## iter  60 value 2.179225
## iter  70 value 2.079851
## iter  80 value 2.015347
## iter  90 value 1.844346
## iter 100 value 1.784488
## final  value 1.784488
## stopped after 100 iterations
```

```r
mulog.pred.iris <- predict(mulog.iris, iris.test)
mulog.accuracy <- sum(mulog.pred.iris == iris.test$Species)/length(iris.test$Species)

mulog.accuracy
```

```
## [1] 0.9333333
```

Based on the results above, the accuracy rates of random forest, LDA, logistic regression model are 93.33%, 95.56% and 93.33%, therefore LDA model has the best performance in this case.