lab8quiz

Will Scheib

11/5/2021

9

a

```
set.seed(11)
OJ2 <- OJ %>% select(-STORE)

sample.data<-sample.int(nrow(OJ), 800, replace = F)
train<-OJ2[sample.data, ]
test<-OJ2[-sample.data, ]
```

b

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

```
##
## Classification tree:
## tree::tree(formula = Purchase ~ ., data = train)
## Variables actually used in tree construction:
## [1] "LoyalCH"       "SalePriceMM"   "SpecialCH"     "PriceDiff"
## [5] "ListPriceDiff"
## Number of terminal nodes:  10
## Residual mean deviance:  0.6981 = 551.5 / 790
## Misclassification error rate: 0.1438 = 115 / 800
```

Training error rate: 0.1438 Terminal nodes: 10

c

```
tree.class.train
```

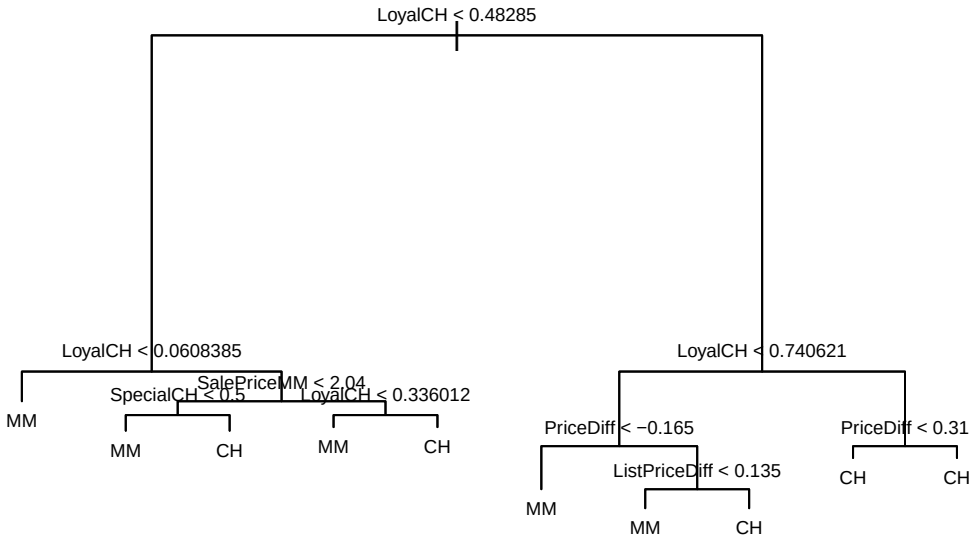
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
```

```
##  1) root 800 1070.00 CH ( 0.61000 0.39000 )
##    2) LoyalCH < 0.48285 294  308.10 MM ( 0.21769 0.78231 )
##      4) LoyalCH < 0.0608385 64   10.30 MM ( 0.01562 0.98438 ) *
##      5) LoyalCH > 0.0608385 230  270.10 MM ( 0.27391 0.72609 )
##       10) SalePriceMM < 2.04 128  120.60 MM ( 0.17969 0.82031 )
##         20) SpecialCH < 0.5 109   79.67 MM ( 0.11927 0.88073 ) *
##         21) SpecialCH > 0.5 19   26.29 CH ( 0.52632 0.47368 ) *
##       11) SalePriceMM > 2.04 102  136.60 MM ( 0.39216 0.60784 )
##         22) LoyalCH < 0.336012 59   66.90 MM ( 0.25424 0.74576 ) *
##         23) LoyalCH > 0.336012 43   58.47 CH ( 0.58140 0.41860 ) *
##    3) LoyalCH > 0.48285 506  448.40 CH ( 0.83794 0.16206 )
##      6) LoyalCH < 0.740621 219  272.90 CH ( 0.68493 0.31507 )
##       12) PriceDiff < -0.165 34   34.57 MM ( 0.20588 0.79412 ) *
##       13) PriceDiff > -0.165 185  198.20 CH ( 0.77297 0.22703 )
##         26) ListPriceDiff < 0.135 29   39.89 MM ( 0.44828 0.55172 ) *
##         27) ListPriceDiff > 0.135 156  140.60 CH ( 0.83333 0.16667 ) *
##      7) LoyalCH > 0.740621 287  105.90 CH ( 0.95470 0.04530 )
##       14) PriceDiff < 0.31 190   94.82 CH ( 0.93158 0.06842 ) *
##       15) PriceDiff > 0.31 97    0.00 CH ( 1.00000 0.00000 ) *
```

64 observations reached this node. 0.98437500 purchased MM.

d

```
plot(tree.class.train)
text(tree.class.train, cex=0.6, pretty=0)
```



e

```
tree.pred.test<-predict(tree.class.train, newdata=test, type="class")
head(tree.pred.test)
```

```
## [1] CH CH CH CH CH CH
## Levels: CH MM
```

```
pred.probs<-predict(tree.class.train, newdata=test)
head(pred.probs)
```

```
##           CH         MM
## 3  0.8333333 0.16666667
## 9  0.9315789 0.06842105
## 10 0.9315789 0.06842105
## 14 0.8333333 0.16666667
## 27 0.9315789 0.06842105
## 28 0.9315789 0.06842105
```

```
tree.conf.mat <- table(test$Purchase, tree.pred.test)

c(test.error.rate=(tree.conf.mat["CH", "MM"]+tree.conf.mat["MM", "CH"]) / sum(tree.conf.mat))
```

```
## test.error.rate
##       0.2333333
```

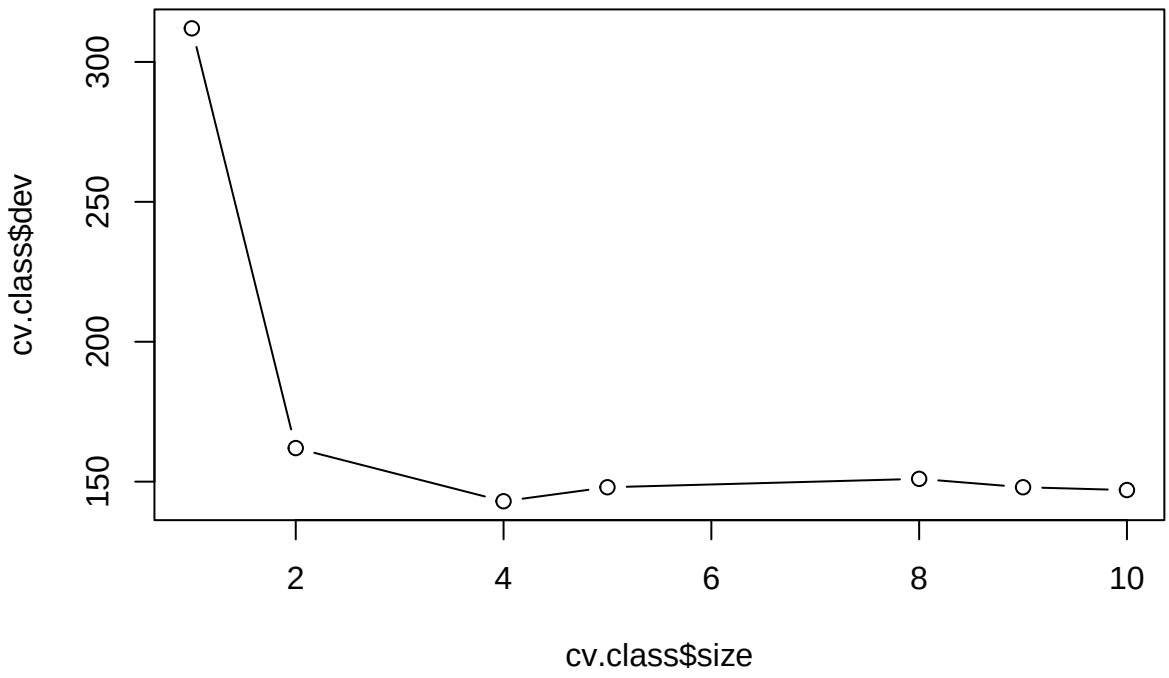
f

```
set.seed(22)
cv.class<-tree::cv.tree(tree.class.train, K=10, FUN=prune.misclass)
cv.class
```

```
## $size
## [1] 10  9  8  5  4  2  1
##
## $dev
## [1] 147 148 151 148 143 162 312
##
## $k
## [1]       -Inf   0.000000   1.000000   2.333333   3.000000  10.000000 166.000000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

g

```
plot(cv.class$size, cv.class$dev,type='b')
```



h

4 corresponds to the lowest cross-validated classification error rate

i

```
trees.num.class<-cv.class$size[which.min(cv.class$dev)]
trees.num.class
```

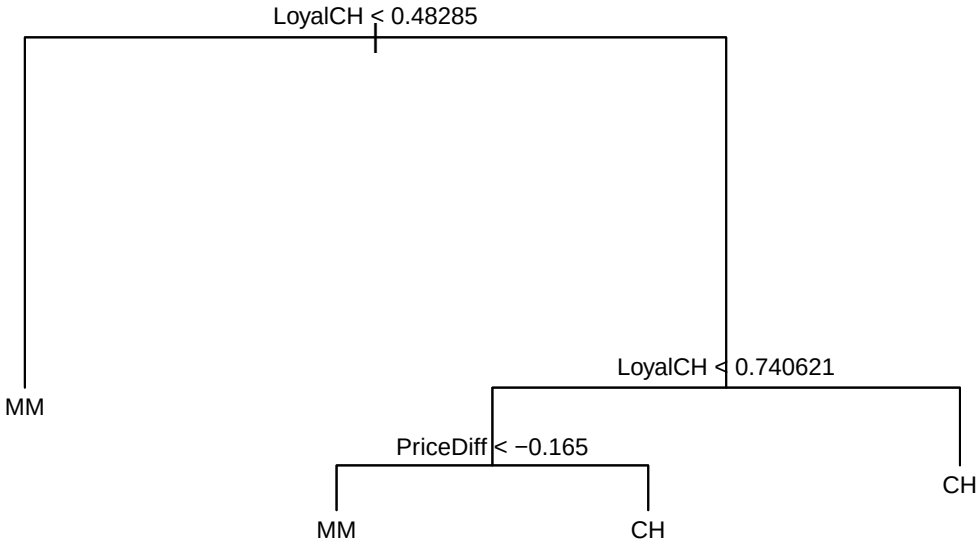
```
## [1] 4
```

```
prune.class<-tree::prune.misclass(tree.class.train, best=trees.num.class)
prune.class
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 1070.00 CH ( 0.6100 0.3900 )
##    2) LoyalCH < 0.48285 294  308.10 MM ( 0.2177 0.7823 ) *
##    3) LoyalCH > 0.48285 506  448.40 CH ( 0.8379 0.1621 )
##      6) LoyalCH < 0.740621 219  272.90 CH ( 0.6849 0.3151 )
##       12) PriceDiff < -0.165 34   34.57 MM ( 0.2059 0.7941 ) *
```

```
##       13) PriceDiff > -0.165 185  198.20 CH ( 0.7730 0.2270 ) *
##      7) LoyalCH > 0.740621 287  105.90 CH ( 0.9547 0.0453 ) *
```

```
plot(prune.class)
text(prune.class, cex=0.75, pretty=0)
```



j

```
tree.misclass <- summary(tree.class.train)$misclass
pruned.misclass <- summary(prune.class)$misclass
c(
  tree.train.error=tree.misclass[1]/tree.misclass[2],
  pruned.train.error=pruned.misclass[1]/pruned.misclass[2]
)
```

```
##   tree.train.error pruned.train.error
##            0.14375            0.15750
```

k

```
prune.pred.test<-predict(prune.class, newdata=test, type="class")
head(prune.pred.test)
```

```
## [1] CH CH CH CH CH CH
## Levels: CH MM
```

```
prune.conf.mat <- table(test$Purchase, prune.pred.test)

c(
  tree.test.error=(tree.conf.mat["CH", "MM"]+tree.conf.mat["MM", "CH"]) / sum(tree.conf.mat),
  prune.test.error=(prune.conf.mat["CH", "MM"]+prune.conf.mat["MM", "CH"]) / sum(prune.conf.mat)
)
```

```
##  tree.test.error prune.test.error
##        0.2333333        0.2111111
```

Interesting how the prune train error is higher but the prune test error is lower.