Tree-Based Methods

Maureen Renaud

7/30/2020

INTRODUCTION

Orange juice sales in the United States have fallen nearly every year since 1998 (Ferdman, 2014)[1]. This has occured for a variety of reasons, including the rising cost of growing oranges, changes in the ways Americans eat breakfast, and growing concern for the amount of sugar often hidden in juices. In order to combat these challenges, it is up to market analysts to fully understand the purchasing trends in order to understand why, when, and where Americans buy orange juice.

This study employs a decision tree to predict whether a customer will purchase Minute Maid or Citrus Hill orange juice. This data set contains a variety of sales information for the Citrus Hill and Minute Maid brands of orange juice.

According to *An Introduction to Statistical Learning*[2], when using a classification tree, we predict each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. Decision trees are a popular method because they are easy to explain and are graphically very interpretable. However, they are not as accurate as some regression and classification methods. They are also not very robust because small changes in the obseravations can result in a very different tree.

After dividing the data into a test and training set and then pruning the tree, we ended up with a tree with five nodes and based on three variables: “LoyalCH”, “PriceDiff”, and ListPriceDiff". Based on a customers loyalty to Citrus Hill and the difference in both price and list price, it is possible to predict which type of orange juice a customer will buy.

DATA

The data for this study comes from the OJ dataset in the ISLR library which contains 1070 observations on 18 variables related to sales information for the Citrus Hill and Minute Maid brands of orange juice. These variables are described below:

Variable	Description
Purchase	A factor with levels CH and MM indicating whether the customer purchased Citrus Hill or Minute Maid Orange Juice
WeekofPurchase	Week of purchase
StoreID	Store ID
PriceCH	Price charged for CH
PriceMM	Price charged for MM
DiscCH	Discount offered for CH
DiscMM	Discount offered for MM
SpecialCH	Indicator of special on CH
SpecialMM	Indicator of special on MM
LoyalCH	Customer brand loyalty for CH

Variable	Description
SalePriceMM	Sale price for MM
SalePriceCH	Sale price for CH
PriceDiff	Sale price of MM less sale price of CH
Store7	A factor with levels No and Yes indicating whether the sale is at Store 7
PctDiscMM	Percentage discount for MM
PctDiscCH	Percentage discount for CH
ListPriceDiff	List price of MM less list price of CH
STORE	Which of 5 possible stores the sale occured at

Now that we are familiar with the data, we can move on to the analyses. Prior to beginning, I have created a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

ANALYSES

DECISION TREE

I will now fit a tree to the training data, with Purchase as the response and the other variables as predictors.

A summary is provided below:

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = OJ, subset = my.sample)
## Variables actually used in tree construction:
## [1] "LoyalCH"       "PriceDiff"     "ListPriceDiff"
## Number of terminal nodes:  7
## Residual mean deviance:  0.7425 = 588.8 / 793
## Misclassification error rate: 0.1575 = 126 / 800
```

As seen in the output, this tree uses the variables LoyalCH, PriceDiff, and ListPriceDiff. It has 7 terminal nodes and a training error rate of 15.75.

I will now examine a detailed text output.

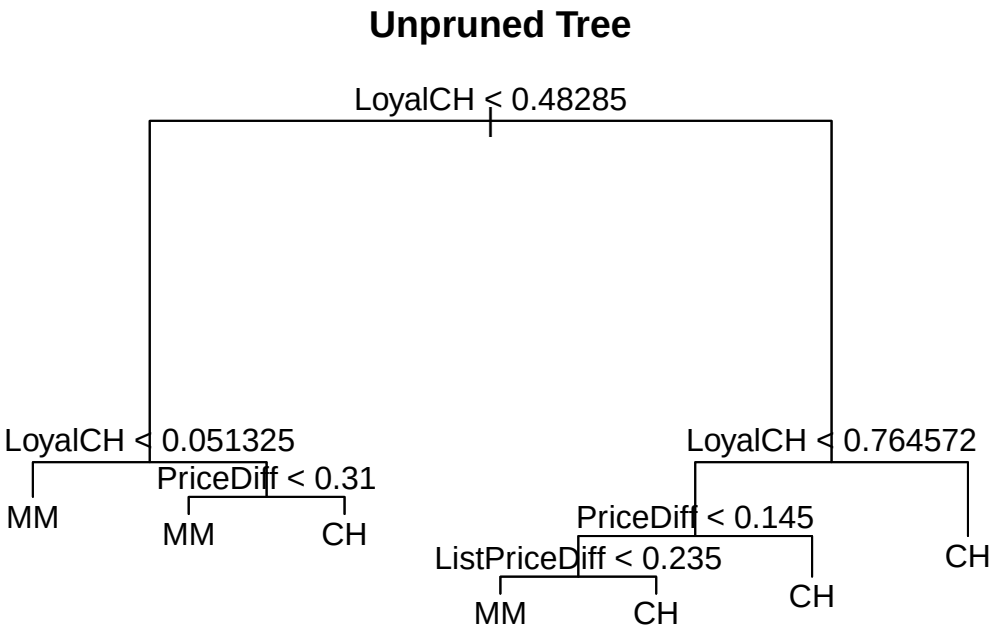
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 1064.00 CH ( 0.61750 0.38250 )
##    2) LoyalCH < 0.48285 290  308.60 MM ( 0.22414 0.77586 )
##      4) LoyalCH < 0.051325 55    0.00 MM ( 0.00000 1.00000 ) *
##      5) LoyalCH > 0.051325 235  277.20 MM ( 0.27660 0.72340 )
##       10) PriceDiff < 0.31 186  193.70 MM ( 0.21505 0.78495 ) *
##       11) PriceDiff > 0.31 49   67.91 CH ( 0.51020 0.48980 ) *
##    3) LoyalCH > 0.48285 510  446.50 CH ( 0.84118 0.15882 )
##      6) LoyalCH < 0.764572 244  294.30 CH ( 0.70902 0.29098 )
##       12) PriceDiff < 0.145 96  133.00 MM ( 0.48958 0.51042 )
##         24) ListPriceDiff < 0.235 65   84.47 MM ( 0.35385 0.64615 ) *
##         25) ListPriceDiff > 0.235 31   33.12 CH ( 0.77419 0.22581 ) *
##       13) PriceDiff > 0.145 148  124.40 CH ( 0.85135 0.14865 ) *
##      7) LoyalCH > 0.764572 266   85.24 CH ( 0.96241 0.03759 ) *
```

This information corresponds to each branch of the tree. The items with stars at the ends are terminal nodes. If we take a closer look at one of the terminal nodes, we can interpret it:

4) LoyalCH < 0.051325 55 0.00 MM (0.00000 1.00000) *

We see that the split criterion is LoyalCH < 0.051325, there are 55 observations in that branch, the deviance is 0.00, the overall prediction is Minute Maid (MM) and the fraction of observations in this branch that take on the value of Citrus Hill(CH) is 0, while the fraction of observations that take on the value MM is 1.00 (all of them).

A plot of the tree is included below:



The first split criterion is based on customer loyalty to Citrus Hill orange juice. Below 0.48285 is sent to the left branch and greater than or equal to 0.48285 is sent to the right branch.

In the left branch, customer loyalty once again is the criterion with customers with a loyalty of less than 0.051325 likely to select Minute Maid orange juice. For customers whose loyalty is rated greater than or equal to that value, price difference plays a role. If the price difference is less than 31 cents, the customer will select Minute Maid. Otherwise, they will select Citrus Hill orange juice.

Examining the right branch, customer loyalty is again the criterion. Those with a loyalty rating equal to or greater than 0.764572 will select Citrus Hill. If the their loyalty rating is less than this value, price difference is the next criterion. If the price difference is greater than or equal to 14.5 cents, the customer will select Citrus Hill. If it is less than 14.5 cents, the final decision will be based on list price difference. If that difference is greater than or equal to 23.5 cents, the customer will select Citrus Hill orange juice. If it is less than 23.5 cents, they will go with Minute Maid.

I will now predict the response on the test data and produce a confusion matrix comparing the test labels to the predicted test labels. The accuracy is 0.80 and the test error rate is 0.20.

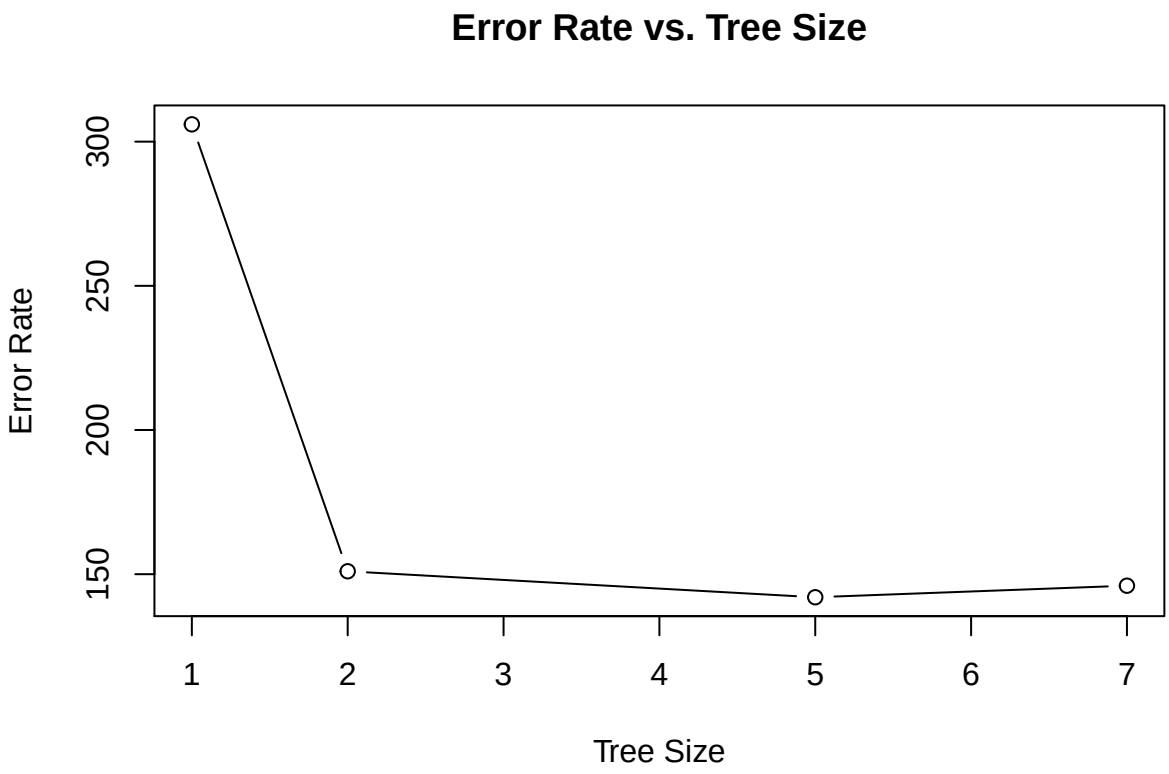
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 132  27
##         MM  27  84
##
##                Accuracy : 0.8
##                  95% CI : (0.7472, 0.846)
##     No Information Rate : 0.5889
##     P-Value [Acc > NIR] : 1.346e-13
##
##                   Kappa : 0.5869
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8302
##             Specificity : 0.7568
##          Pos Pred Value : 0.8302
##          Neg Pred Value : 0.7568
##              Prevalence : 0.5889
##          Detection Rate : 0.4889
##    Detection Prevalence : 0.5889
##       Balanced Accuracy : 0.7935
##
##        'Positive' Class : CH
##
```

The next step is to apply the cv.tree() function to the training set in order to determine the optimal tree size.

```
## $size
## [1] 7 5 2 1
##
## $dev
## [1] 146 142 151 306
##
## $k
## [1]       -Inf   0.500000   6.333333 160.000000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

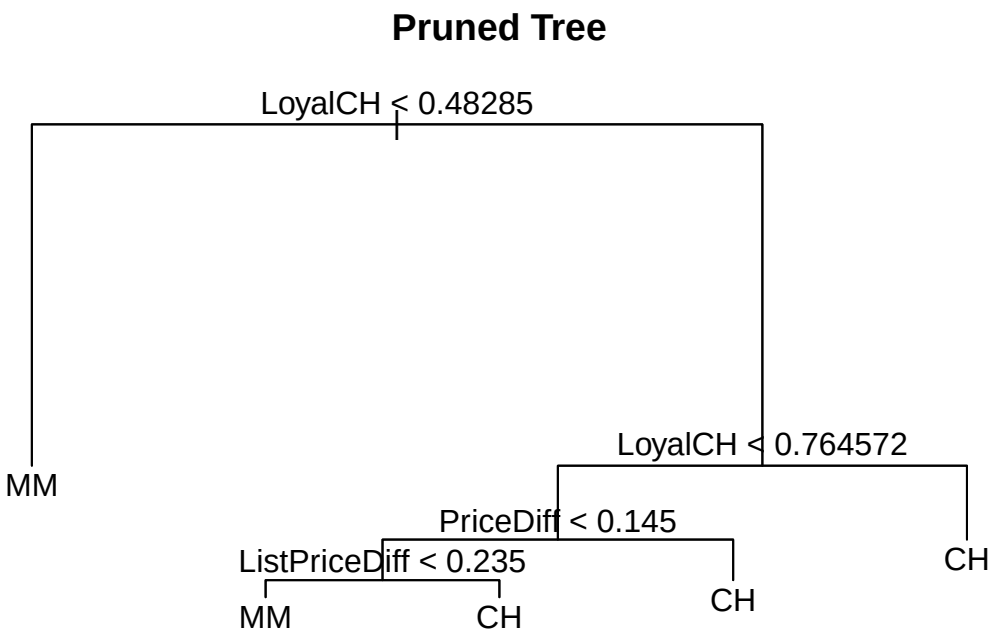
The tree with 5 terminal nodes results in the lowest cross-validation error rate, with 142 cross-validation errors.

Below is a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.



This plot confirms what we already deduced from the output.

Based on this knowledge, below is a pruned tree corresponding to the optimal tree size of 5 that was obtained using cross-validation.



The left side of pruned tree is clearly more simple than the unpruned tree. In this tree, if the customer loyalty is less than 0.48285, the customer is automatically predicted to select Minute Maid orange juice with no further reflection. The right branch is unchanged.

We can also take a closer look at the pruned tree data and examine the error rate on the training set:

```
##
## Classification tree:
## snip.tree(tree = tree.OJ, nodes = 2L)
## Variables actually used in tree construction:
## [1] "LoyalCH"       "PriceDiff"     "ListPriceDiff"
## Number of terminal nodes:  5
## Residual mean deviance:  0.7998 = 635.9 / 795
## Misclassification error rate: 0.1588 = 127 / 800
```

We see that the training error rate is 0.1588 which is very slightly higher than the training error rate for the unpruned tree. We can now examine the test error rate for the pruned tree.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 123  36
##         MM  17  94
##
##                Accuracy : 0.8037
```

```
##                  95% CI : (0.7512, 0.8494)
##     No Information Rate : 0.5185
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6048
##
##  Mcnemar's Test P-Value : 0.01342
##
##             Sensitivity : 0.8786
##             Specificity : 0.7231
##          Pos Pred Value : 0.7736
##          Neg Pred Value : 0.8468
##              Prevalence : 0.5185
##          Detection Rate : 0.4556
##    Detection Prevalence : 0.5889
##       Balanced Accuracy : 0.8008
##
##        'Positive' Class : CH
##
```

The below chart contains the training and test error rates for both the pruned and unpruned trees.

Tree	Training Error	Test Error
Unpruned Tree	0.1575	0.1588
Pruned Tree	0.2000	0.1963

The pruned tree has only a marginally better test error rate than the unpruned rate, but has the added advantage of being simpler and more easy to interpret, making it the better choice when determining which orange juice a customer will choose.

CONCLUSION AND RECOMMENDATIONS

In analyzing this data set with a decision tree, we learned a few important things about the data. The primary variables that influence which orange juice a customer will purchase are Customer Loyalty, Price Difference, and List Price Difference.

Additionally, a pruned tree with five nodes performed marginally better than the unpruned tree with a test error rate of 0.1963 compared to 0.200. This is likely because the reduction in variance from pruning only slightly overcame the increase in bias.

While the difference in test error rates is not large enough to likely be significant, the fact that the pruned tree is also more simple and easier to interpret, make it an ideal choice for a market analyst.

SOURCES

1) Ferdman, R. (2014). *How American Fell Out of Love With Orange Juice* Quartz. https://qz.com/176096/how-america-fell-out-of-love-with-orange-juice/

2) James, G et al. (2017). *An Introduction To Statistical Learning.* New York, NY.

APPENDIX

R Code:

```
RNGkind(sample.kind="Rounding")

#Loading the ISLR library

library(ISLR)

set.seed(3)

#Selecting a random sample of 800 rows from the 1070
my.sample <- sample(1:nrow(OJ), 800)

#Assiging those 800 selected rows to the training set
OJ.train <- OJ[my.sample,]

#Assigning the remaining rows to the test set
OJ.test <- OJ[-my.sample,]

#Fit a tree to the training data

library(tree)
tree.OJ = tree(Purchase ~ ., OJ, subset = my.sample)

#Providing a summary of the tree

summary(tree.OJ)

#Examining a detailed text output.

tree.OJ

#Plotting the tree

plot(tree.OJ)
text(tree.OJ, pretty = 0)
title("Unpruned Tree")
```

```
#Predict the response on the test data

library(caret)
test.tree.OJ = OJ[-my.sample,]

tree.pred = predict(tree.OJ, test.tree.OJ, type = "class")

#Creating the confusion matrix

confusionMatrix(OJ.test$Purchase, tree.pred)

#Determine the optimal tree size

set.seed (3)
```

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

#Plot with tree size on the x-axis and cv classification error rate on the y-axis.

plot(cv.OJ$size, cv.OJ$dev, type="b", xlab="Tree Size", ylab="Error Rate",)
title("Error Rate vs. Tree Size")
```

```
#Plotting the pruned tree

prune.OJ = prune.misclass(tree.OJ, best = 5)
plot(prune.OJ)
text(prune.OJ, pretty =0)
title("Pruned Tree")
```

```
#Summary of the pruned tree data

summary(prune.OJ)

#Examine the test error rate for the pruned tree

tree.pred.2 = predict(prune.OJ, test.tree.OJ, type= "class")

#Creating the confusion matrix
confusionMatrix(OJ.test$Purchase, tree.pred.2)
```