ORANGE JUICE ANALYSIS

OCTOBER 20, 2021

KO TSZ NGA (VALERIE)
valerie.ktn@gmail.com

Introduction

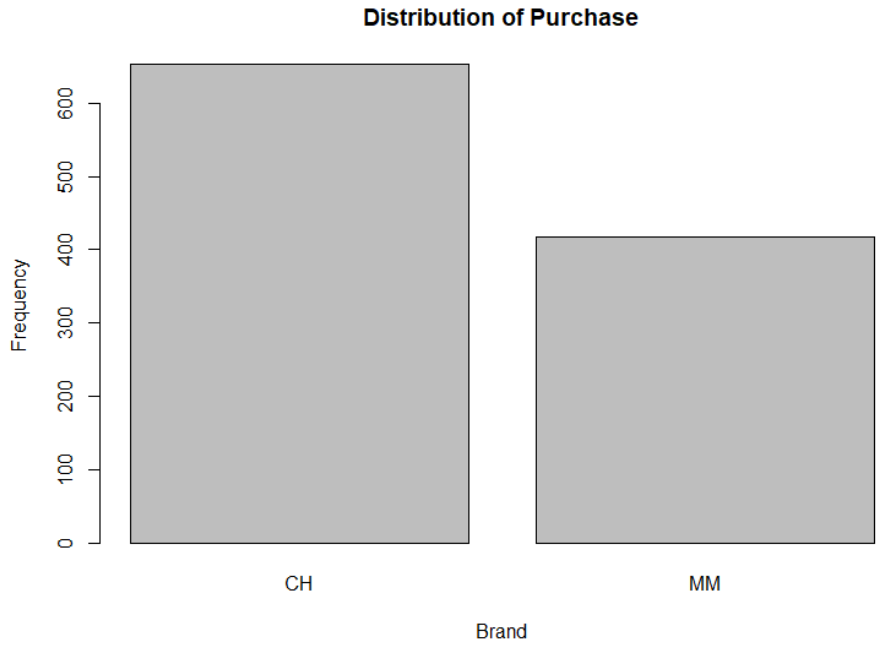
The aim of the analysis is to use decision tree to classify whether customer purchase Citrus Hill or Minute Maid Orange Juice. The model will be optimised with tree pruning methods and their respective performance will be compared.

Background of the dataset

The OJ dataset is part of the ISLR package in R. It consists of 1070 rows of purchase record. There are 17 attributes including the predicting variable (ie. Purchase).

Exploratory data analysis

As tree-based model, unlike other statistical model, that has assumptions for the independent variables. Therefore, I will **skip the variables correlation part** and just discuss the distribution of the dependent variable (ie. Purchase) in this section.



There are **more purchases for Citrus Hill** (ie CH) than Minute Maid (ie. MM).

Methodology

Split the data into training and testing sets

Train test split is performed with 800 rows of data are randomly picked as the training set and the remaining 270 rows are used in the testing set.

Result

(a) Original classification tree

'Gini' criterion is used for training the model. Below is part of the summary output.

```
> summary(rpart.gini)
Call:
rpart(formula = Purchase ~ ., data = oj_train, method = "class",
    parms = list(split = "gini"))
  n= 800

         CP nsplit rel error    xerror       xstd
1 0.49201278      0 1.0000000 1.0000000 0.04410089
2 0.03514377      1 0.5079872 0.5335463 0.03672576
3 0.02555911      2 0.4728435 0.5175719 0.03631437
4 0.01277955      4 0.4217252 0.4664537 0.03490406
5 0.01000000      7 0.3833866 0.4472843 0.03433576

Variable importance
      LoyalCH       StoreID     PriceDiff   SalePriceMM
           45             9             9             6
WeekofPurchase       PriceMM       DiscMM    PctDiscMM
            6             5             5             4
      PriceCH  ListPriceDiff   SalePriceCH         STORE
            4             3             2             1
    SpecialCH
            1
```

In the variable importance section, we can see that **'LoyalCH' is the most important** variable, whereas **'STORE' and 'SpecialCH' are the least important**.

'StoreID' and 'PriceDiff' are the second most important.

The training error is calculated to be **1.**

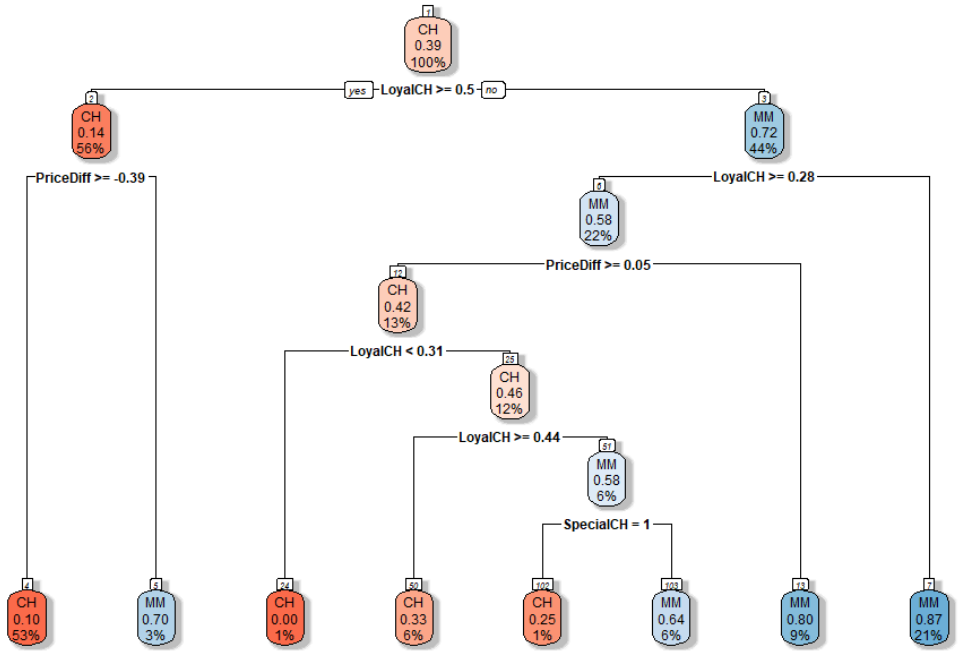
Number of terminal nodes

```
> print(rpart.gini)
n= 800

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 800 313 CH (0.60875000 0.39125000)
    2) LoyalCH>=0.5036 450  61 CH (0.86444444 0.13555556)
      4) PriceDiff>=-0.39 423  42 CH (0.90070922 0.09929078) *
      5) PriceDiff< -0.39 27   8 MM (0.29629630 0.70370370) *
    3) LoyalCH< 0.5036 350  98 MM (0.28000000 0.72000000)
      6) LoyalCH>=0.2761415 180  76 MM (0.42222222 0.57777778)
       12) PriceDiff>=0.05 106  45 CH (0.57547170 0.42452830)
         24) LoyalCH< 0.3084325 8   0 CH (1.00000000 0.00000000) *
         25) LoyalCH>=0.3084325 98  45 CH (0.54081633 0.45918367)
           50) LoyalCH>=0.442144 46  15 CH (0.67391304 0.32608696) *
           51) LoyalCH< 0.442144 52  22 MM (0.42307692 0.57692308)
            102) SpecialCH>=0.5 8   2 CH (0.75000000 0.25000000) *
            103) SpecialCH< 0.5 44  16 MM (0.36363636 0.63636364) *
       13) PriceDiff< 0.05 74  15 MM (0.20270270 0.79729730) *
      7) LoyalCH< 0.2761415 170  22 MM (0.12941176 0.87058824) *
```

The asterisk specifies **the terminal nodes, which have 8 in total.**

Plot of the tree

The graph below displayed the details of the tree model.



There are 8 terminal nodes in total. **Over half (ie. 53%) of the data falls on the node which is on the leftmost**. As described by the graph, they are the points with **'LoyalCH' >=0.5 and 'PriceDiff'>=-0.39.**

Predict the response on test data

(i) Testing error

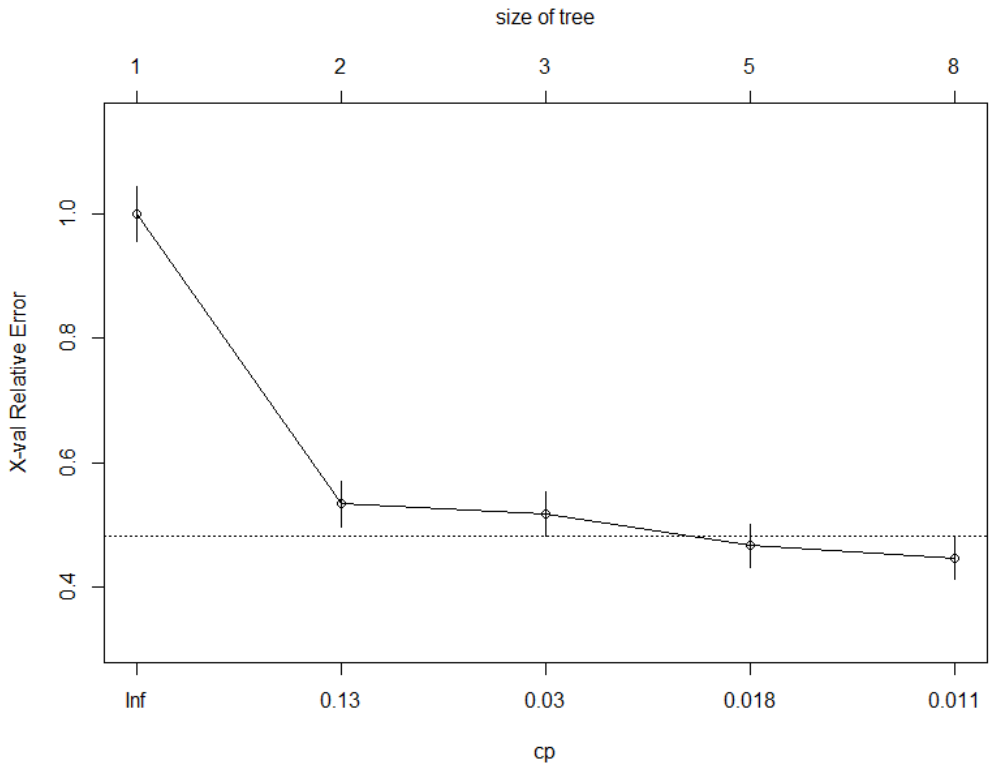
The testing error is calculated to be **0.1814815.**

(ii) Confusion matrix

pred	CH	MM
CH	141	24
MM	25	80

Accuracy is calculated to be **81.9%.**

Optimal tree size



From the graph above, it is shown that the **error drops significantly when the tree size increases from 1 to 2 and then the error keeps dropping steadily onwards**. The **optimal tree size** that corresponds to the lowest cross-validation classification error rate is **5.**

(b) Tree pruning

I want to prune the tree to see if there is further improvement found. I therefore find the **cp value corresponds to the optimal tree size** that I found (ie. 5). The value is calculated as **0.01**, which is **the default cp** in the original tree-based model. **Therefore, if we prune the tree with cp=0.01, it should give us the same result.**

To verify if 0.01 is truly the optimal cp or not, it is suggested to grow out the tree again with negative cp value and check for the optimal cp.

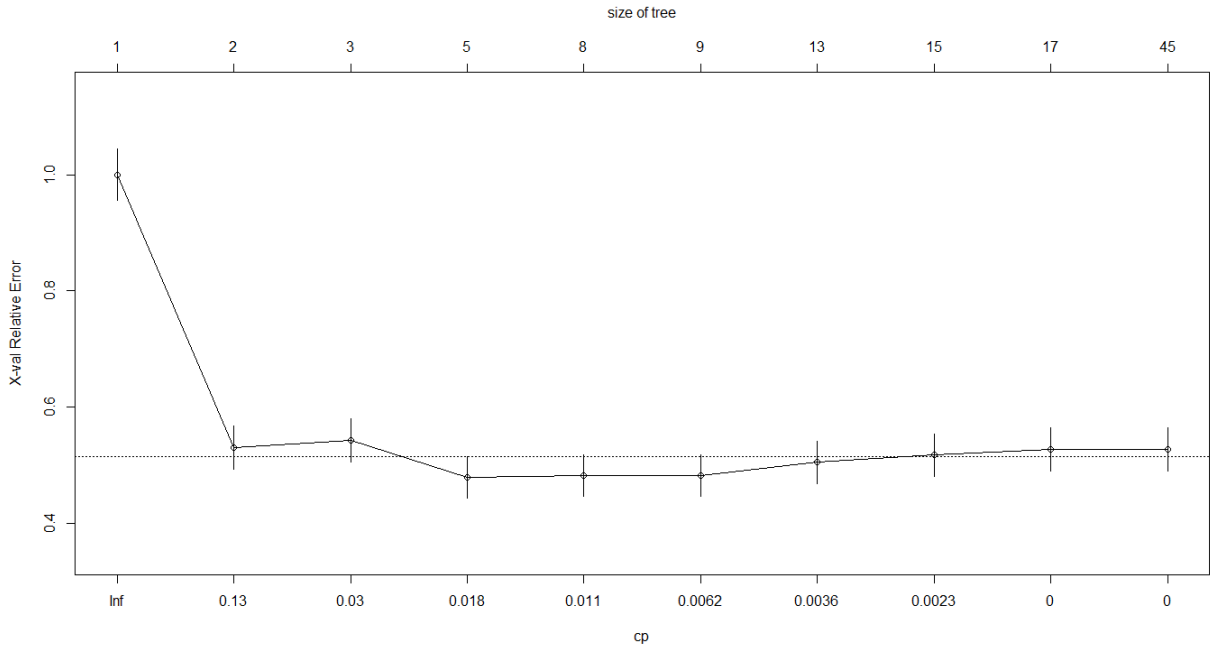
Tree growing with negative cp

I arbitrarily chose **cp=-0.1** to grow the tree. Below shows its summary.

```
Call:
rpart(formula = Purchase ~ ., data = oj_train, method = "class",
    parms = list(split = "gini"), cp = -0.1)
  n= 800

            CP nsplit rel error    xerror       xstd
1   0.492012780      0 1.0000000 1.0000000 0.04410089
2   0.035143770      1 0.5079872 0.5271565 0.03656280
3   0.025559105      2 0.4728435 0.5143770 0.03623048
4   0.012779553      4 0.4217252 0.4664537 0.03490406
5   0.009584665      7 0.3833866 0.4824281 0.03536075
6   0.003993610      8 0.3738019 0.4888179 0.03553927
7   0.003194888     12 0.3578275 0.5175719 0.03631437
8   0.001597444     14 0.3514377 0.5111821 0.03614603
9   0.000000000     16 0.3482428 0.5015974 0.03588938
10 -0.100000000     44 0.3482428 0.5015974 0.03588938

Variable importance
      LoyalCH        StoreID      PriceDiff    SalePriceMM WeekofPurchase        PriceMM        PriceCH
           38              9              8              7              6              6              5
       DiscMM      PctDiscMM  ListPriceDiff    SalePriceCH          STORE      SpecialCH         Store7
            5              4              4              3              3              1              1
       DiscCH      PctDiscCH
            1              1
```

As a side note, **again 'LoyalCH' is the most important** variable.

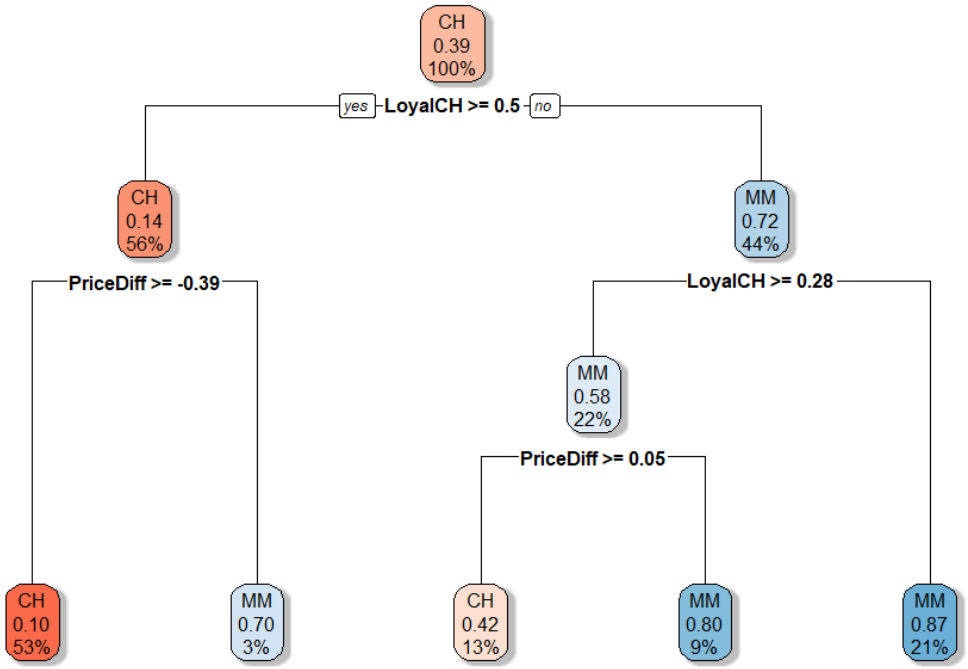
Find optimal tree size and the corresponding cp



The optimal tree size is found to be 4 and the **corresponding cp is 0.01277955**, which is close to what we found in the original tree model. To improve the original tree model, I am going to **prune the tree with the newly found cp value.**

Tree pruning with the cp=0.01277955

The below graph visualises the pruned tree.



There are **5** terminal nodes in total.

Its training error is calculated to be 1 and the **testing error is calculated to be 0.1851852.** Below displays its confusion matrix:

pred2	CH	MM
CH	150	34
MM	16	70

Its corresponding **accuracy is calculated to be 81.48%.**

Findings

Comparison between the pruned and unpruned trees

The below table summarise the findings in both tree models.

	Original (unpruned)	**Pruned**
Training error	1	1
Testing error	0.1814815	0.1851852
Accuracy	81.9%.	81.48%.

The finding is **out of my expectation.** As we can see that the **testing error of the pruned tree is higher, and its accuracy is lower**. Therefore, **the original (unpruned) tree model is better.**