ISYE 7406 Homework 5

October 24, 2021

Introduction

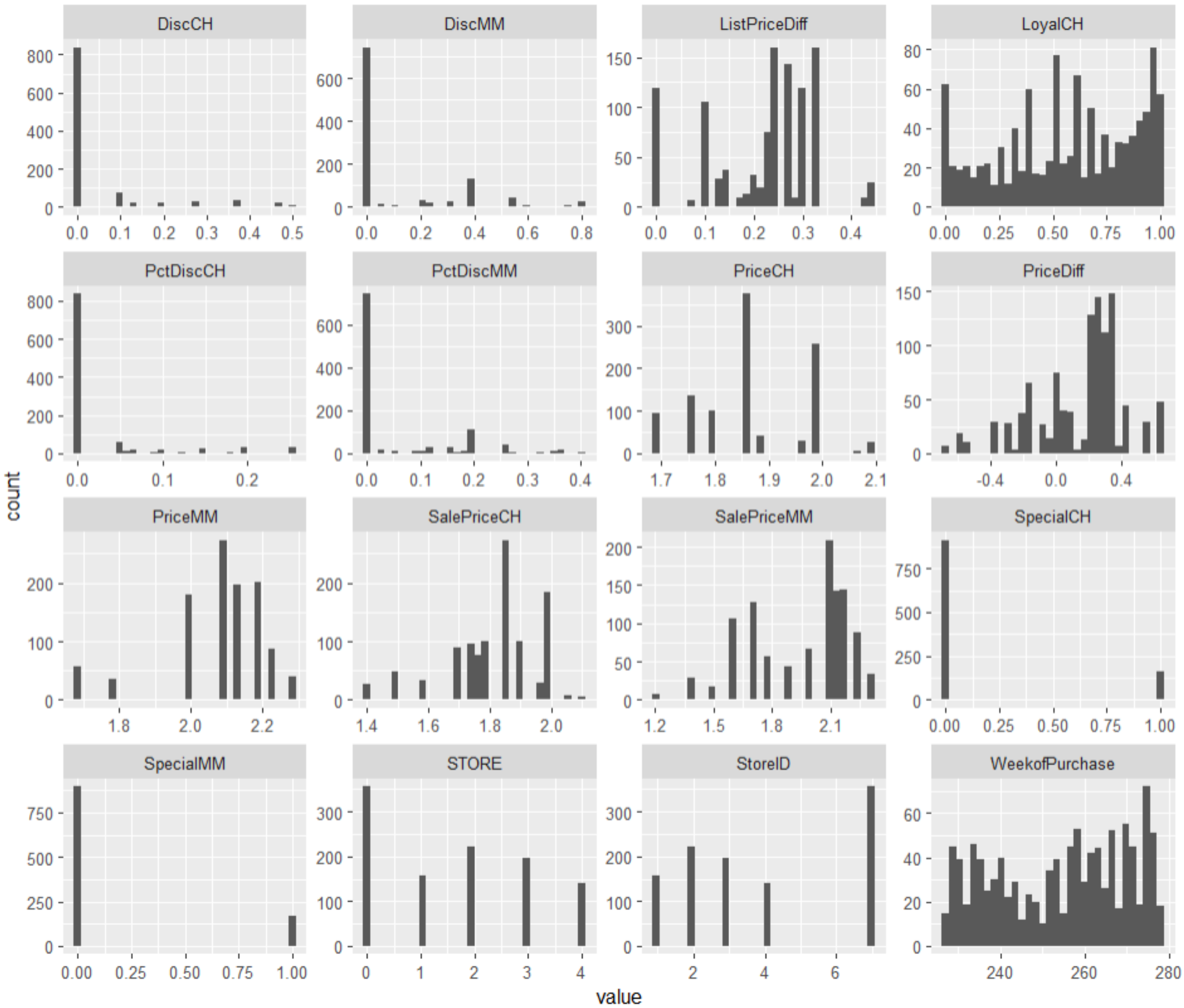
This report will review the orange juice (OJ) dataset which includes the *Purchase* response (y) variable and predictor (x) variables (WeekofPurchase, StoreID, PriceCH, PriceMM, DiscCH, DiscMM) and other variables with their descriptions that can be found here. The purpose of analyzing the dataset is to fit a classification tree with “gini” criterion to the training and test sets to identify the Training & Test errors while also evaluating the results in a confusion matrix. Then we determine the optimal tree size that corresponds to the lowest cross-validation classification error rate and produce a few pruned trees corresponding to the optimal tree size obtained using cross validation and evaluated their performance on the test error and confusion matrix. The purpose of this report is to also identify other findings and visually explore the dataset for important/unusual patterns.

Exploratory Data Analysis

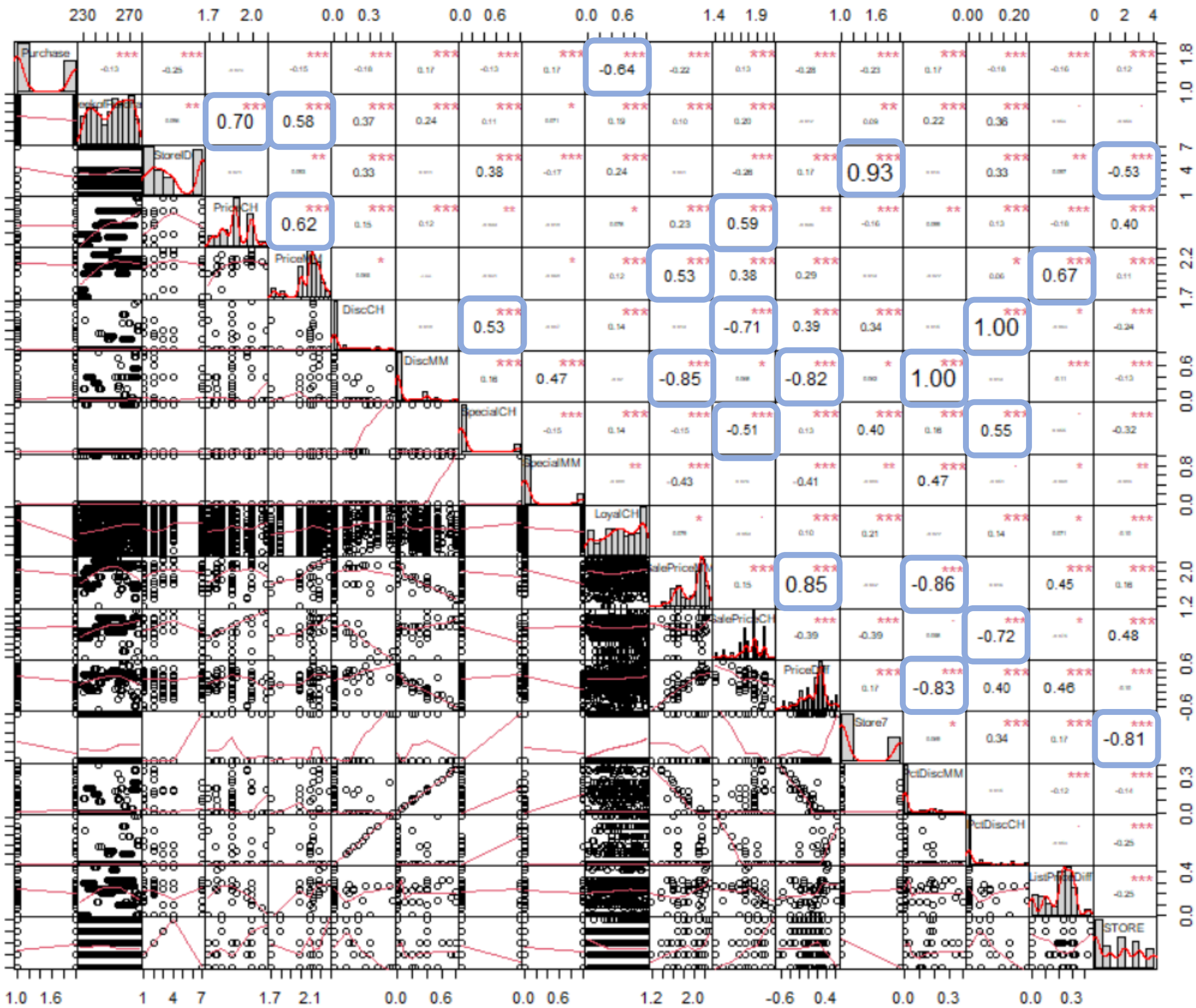
The exploratory data analysis will check for important/unusual patterns in the data through plots and summary statistics. According to the summary statistics shown below, it was interesting to see how PriceMM (price of Minute Maid orange juice) is generally pricier than PriceCH (Citrus Hill orange juice) suggesting that most of the purchases (~64%) are Citrus Hill (CH). It’s also worth noting that StoreID, SpecialCH, SpecialMM, and STORE are categorical variables so I made sure to change the types to “factors” shown below but I changed them back to numeric for modeling purposes. Note: The 0 and 1’s in Purchase represent CH and MM respectively.

```
Purchase WeekofPurchase     StoreID        PriceCH         PriceMM          DiscCH
0:653    Min.   :227.0   Min.   :1.00   Min.   :1.690   Min.   :1.690   Min.   :0.00000
1:417    1st Qu.:240.0   1st Qu.:2.00   1st Qu.:1.790   1st Qu.:1.990   1st Qu.:0.00000
         Median :257.0   Median :3.00   Median :1.860   Median :2.090   Median :0.00000
         Mean   :254.4   Mean   :3.96   Mean   :1.867   Mean   :2.085   Mean   :0.05186
         3rd Qu.:268.0   3rd Qu.:7.00   3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000
         Max.   :278.0   Max.   :7.00   Max.   :2.090   Max.   :2.290   Max.   :0.50000
    DiscMM          SpecialCH        SpecialMM         LoyalCH          SalePriceMM
Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000011   Min.   :1.190
1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.325257   1st Qu.:1.690
Median :0.0000   Median :0.0000   Median :0.0000   Median :0.600000   Median :2.090
Mean   :0.1234   Mean   :0.1477   Mean   :0.1617   Mean   :0.565782   Mean   :1.962
3rd Qu.:0.2300   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.850873   3rd Qu.:2.130
Max.   :0.8000   Max.   :1.0000   Max.   :1.0000   Max.   :0.999947   Max.   :2.290
 SalePriceCH       PriceDiff       Store7     PctDiscMM         PctDiscCH        ListPriceDiff
Min.   :1.390   Min.   :-0.6700   0:714   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
1st Qu.:1.750   1st Qu.: 0.0000   1:356   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.140
Median :1.860   Median : 0.2300           Median :0.0000   Median :0.00000   Median :0.240
Mean   :1.816   Mean   : 0.1465           Mean   :0.0593   Mean   :0.02731   Mean   :0.218
3rd Qu.:1.890   3rd Qu.: 0.3200           3rd Qu.:0.1127   3rd Qu.:0.00000   3rd Qu.:0.300
Max.   :2.090   Max.   : 0.6400           Max.   :0.4020   Max.   :0.25269   Max.   :0.440
    STORE
Min.   :0.000
1st Qu.:0.000
Median :2.000
Mean   :1.631
3rd Qu.:3.000
Max.   :4.000
```

Based on the distributions of the data (shown below), we can see that the discounts are somewhat similar but we can see that Minute Maid (MM) provided higher ranges of discounts compared to Citrus Hill (CH). You can also see from the StoreID distribution that most of the sales are from Store7 which is probably why it has its own predictor variable (Store7). We can also see that the sale prices are generally higher for (MM) compared to (CH) as we previously stated but there were instances where (MM) is cheaper than (CH) when on sale. But again, as we see from the PriceDiff distribution (Sale price of MM less sale price of CH), the distribution is left tailed meaning that (MM) are more expensive.



I decided to run a correlation test as well to display the correlation matrix and provide an easier birds-eye view visual analysis. This was my first time using the “Performance Analytics” package in R to run the correlation and I’m glad I did it because it made looking at the data very easy (although the chart gets cluttered when having more than 10 variables). You can see how Purchase only has one strong correlation with LoyalCH but there are also strong correlations between the predictor “x” variables which I didn’t know until running this plot. Based off research, the correlation values >= +/- 0.50 are considered the significant values associated with response and predictor variables (blue outline in the diagram). You can also see the histograms and how the distribution is shaped which I thought was also a great nice-to-have.



Methodology – Modeling for Training and Test errors

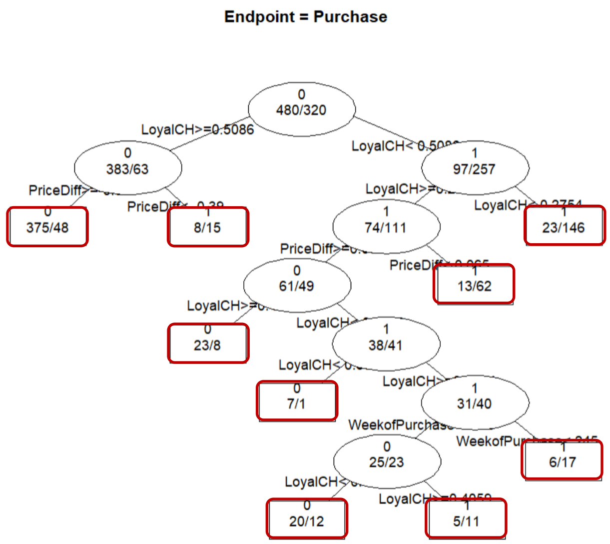
I performed the classification decision tree w/ "gini" criterion to retrieve training and test errors when predicting Purchase (CH or MM). The testing data is used to evaluate the model's true performance since the training data may have excluded/included certain data that may have a significant positive/negative affect on the training error. For this exercise, I performed a 100-run cross validation for the model to obtain the test errors and observed which one had the best test error. Running a Monte Carlo cross-validation helps reduce the variance compared to just running one test run and depicts which model is truly the most accurate on a large scale.

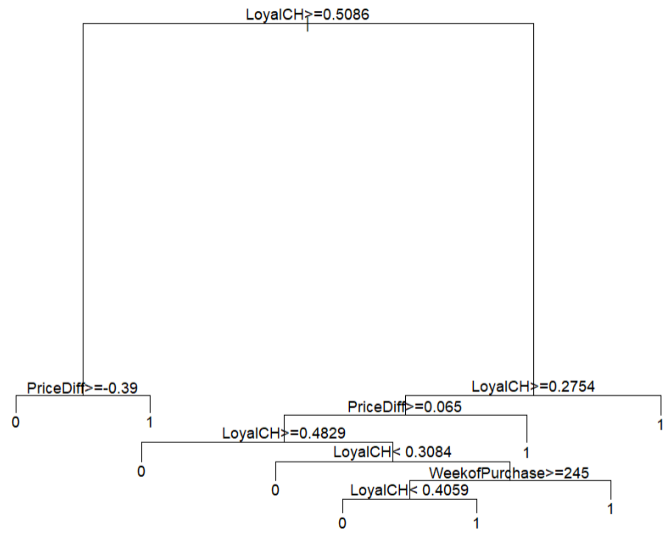
Results & Findings

<u>Classification tree w/ gini criterion</u> – The "Gini" criterion helps calculate the probability of attributes that are classified incorrectly when randomly selected with the degree of Gini index varying between 0 and 1. Based on the summary statistics, we can see *complexity table* (shown below) that provides information about all of the trees of the model. The table displays the complexity parameter (CP), number of splits (np), resolution error rate (rel error), cross-validated error rate (xerror) and associated standard error (xstd). The complexity parameter (CP) determines whether a split is needed based on increasing the model's fit and it's main role is to save computation time by pruning off splits that don't benefit the model. The cross-validated error rate (xerror) shows which tree is optimal based on the lowest xerror which as shown below is tree #5 with xerror 0.5209003 with 7 splits. It's also worth noting that the default cross validation is a 10-fold.

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

         CP nsplit rel error  xerror       xstd
1 0.5000000      0  1.000000 1.00000 0.04330127
2 0.0218750      1  0.500000 0.51250 0.03568252
3 0.0187500      2  0.478125 0.52500 0.03600130
4 0.0140625      4  0.440625 0.50625 0.03551999
5 0.0125000      6  0.412500 0.47500 0.03467483
6 0.0100000      8  0.387500 0.47500 0.03467483
```

When running the classification tree (shown below) the 0 and 1's in the circles represent CH and MM respectively. It's not surprising to see that the first branches came from the LoyalCH variable since we knew that the attribute was the only one that had a high correlation with Purchase. We can also see that there are _9 terminal nodes_ on the tree (outlined in red). Terminal nodes contain small subsets of the data where splitting is no longer valuable. The tree below shows that most of the CH values can be classified by LoyalCH >= .5086 and PriceDiff>=-0.39 which accounted for a count of 375/480 (~78%) but MM as well as the remaining CH values had to be split several times to get to their terminal nodes. This ultimately led to a training error result at 0.155 which is quite high compared to past models we've created.



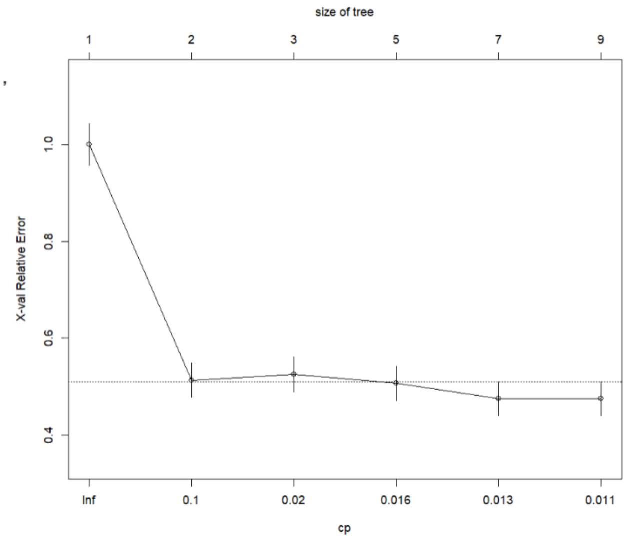



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 150  23
         1  23  74

               Accuracy : 0.8296
                 95% CI : (0.7794, 0.8725)
    No Information Rate : 0.6407
    P-Value [Acc > NIR] : 5.945e-12

                  Kappa : 0.6299

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.7629
            Specificity : 0.8671
         Pos Pred Value : 0.7629
         Neg Pred Value : 0.8671
             Prevalence : 0.3593
         Detection Rate : 0.2741
   Detection Prevalence : 0.3593
      Balanced Accuracy : 0.8150

       'Positive' Class : 1
```

We can also see based on the confusion matrix (shown on the left) how the classification model performed. Accuracy resulted at 82.96% with a Sensitivity of 76.29% and Specificity of 86.71%. A low Sensitivity means that there are more FALSE negative results meaning more cases of Purchase being misclassified as 1. Specificity being higher shows that the model tends give less FALSE positive results meaning less cases of the model stating 1 when the true value is 0. The test set resulted in a test error of 0.1703704 which was higher than the training set.

Going back to the summary output from the training set, we can see that the optimal tree size that corresponds to the lowest cross-validation is tree #5 with 6 splits based on the smallest xerror (shown below).

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

         CP nsplit rel error  xerror       xstd
1 0.5000000      0  1.000000 1.00000 0.04330127
2 0.0218750      1  0.500000 0.51250 0.03568252
3 0.0187500      2  0.478125 0.52500 0.03600130
4 0.0140625      4  0.440625 0.50625 0.03551999
5 0.0125000      6  0.412500 0.47500 0.03467483
6 0.0100000      8  0.387500 0.47500 0.03467483
```



Therefore, it makes sense to prune the tree based on the corresponding optimal tree size obtained above to achieve a better test error. After computing the pruned tree that corresponds to the optimal tree size model with cp = 0.0125, the test error resulted with a lower test error of 0.1666667 and had only *7 terminal nodes*. Just in case, I attempted to run models with each indicated cp point on the above graph to identify all the results (shown below) and it showed that cp =0.014063 and 0.01875 had the best test error of 0.155556 which was a better score than not only the unpruned tree but also the pruned tree with cp =0.0125. It was interesting to see how the best cp values were extremely close to the horizontal line drawn 1 SE above the minimum of the curve. This makes sense since the best choice is often the leftmost value that lies below the horizontal line. It's also interesting to see once the cp value is a little above the line, the test error performs a lot worse.

cp	0.01	0.0125	0.014063	0.01875	0.021875	0.5
Test Error	0.17037	0.166667	0.155556	0.155556	0.2	0.359259

Conclusion

When analyzing all the cross-validated xerrors, it is clear that the pruned trees with cp equaling 0.014063 and 0.01875 performed better than the unpruned & optimized cross-validated prune tree. However, we can also see that test errors for the pruned trees are slightly higher than past classification models we've ran so it's worth investigating a model like Logistic Regression to see how it compares. The test error for the Logistic Regression resulted at 0.1444444 which is better than the pruned tree model. Although decision trees are able to be utilized in both regression and classification and are easier to interpret compared to other machine learning models, models like Logistic Regression can provide a better predictive accuracy depending on the data set. I also noticed that the results were extremely close amongst most of the models so I decided to run different seeds to get a better understanding of the average performance. However, I doubled checked with 3 different seeds (shown below) and it was interesting to see how the Logistic Regression outperformed in each instance and most likely more if additional seeds were run. This leads me to believe that the Logistic Regression model would provide consistent accurate results compared to the pruned decision trees but the decision trees are a lot easier to interpret for audiences. In the future, I plan on re-attempting this type of problem set for my own work and hopefully will be able to better understand each classification model's strengths in a personal work-based application.

Seed	666	999	777
Logistic Regression	0.151852	0.174074	0.174074
Tree model	0.166667	0.185185	0.2