¶ Orange Juice Sales at Wasatch Grocery Chain

Identification of Significant Predictor Variables and Predictive Modelling of Customer Preference in Minute Maid Sales

Chris Gearheart and Chris Porter

2022-11-29

```r
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)

set.seed(1234)
df <- read.csv(url("http://data.mishra.us/files/project/OJ_data.csv"))
df[-1] <- lapply(df[-1],as.numeric)
df$Purchase <- as.factor(df$Purchase)
purchase_testtrain <- initial_split(df, prop = 0.75, strata = Purchase)
train <- training(purchase_testtrain)
test <- testing(purchase_testtrain)
```

Introduction

Wasatch Grocery Chain (WGC) is a regional grocery chain operating in the Intermoutain West of the US. WGC sells two brands of orange juice in its stores, Citrus Hill (CH) and Minute Maid (MM) of which MM is the more profitable to the company. This report will identify **what customer factors** within available data **contribute to purchase of Minute Maid over Citrus Hill**, as well as **to what degree these factors influence customer choice**. In addition, a predictive model is created that will allow the Sales Department to identify other customers within our customer base that are more likely to purchase Minute Maid brand orange juice, thus driving profitability across the company.

Available Data

The data set used in this report contains 13 possible predictor variables as well as 1 outcome variable, Purchase, which records whether or not a customer purchased MM. There are a total of 1070 observations in the data set. The data set was further partitioned into a **training** data set, containing 801 observations, and a validation **testing** data set containing 269 observations.

The code below imports the data set, coverts the binary `Purchase` outcome into a factor, and pulls out 25% of the observations as a hold-out set or test set against which our final model can be tested. Doing so helps us avoid the mistake of training a model that performs well against the sample data, but fails to generalize to a new data set from the same population.

Methods

Logistic Regression:

WGC's management team wants to know **which variables contribute to an customer outcome of "Yes; Purchased Minute Maid**." Their goal matches the strengths of a logistic regression, which can explain the strength and direction of independent variables' effects on a binary classification outcome (often yes/no or is/is not). This algorithm will tell management which variables push customers towards or away from a Minute Maid purchase, plus which variables have no bearing on the outcome. Significant variables proven to have big enough effects can become levers for action or intervention for management.

Pre-processing Logistic regressions work when:

1. Qualitative variable have been turned into quantitative dummy variables.
2. No columns are uniformly filled with one unique value
3. There is no missing data.
4. There is no correlation between the variables.

Fortunately, the first three conditions were already true of our dataset.

```r
# 1. Dummy variables are unnecessary because only `Purchase` is a factor, and it's already expressed us
# 2. No columns are uniformly filled with one unique value - there is spread in each of the 13 independ
summary(train)
```

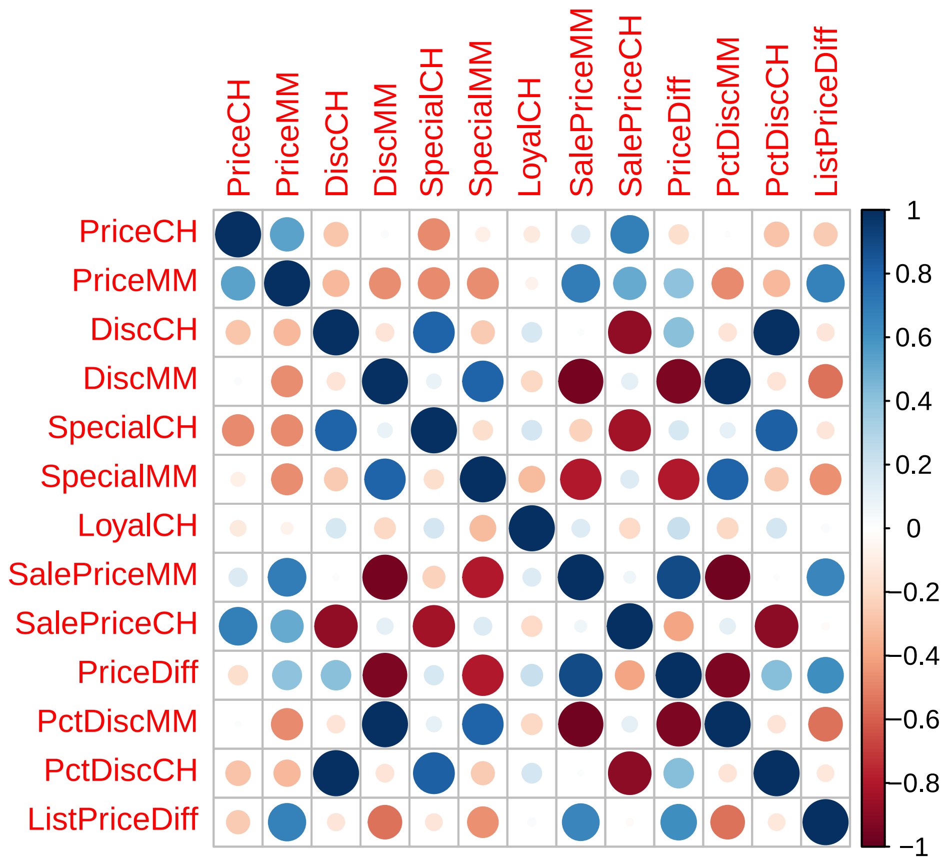
```
##  Purchase    PriceCH         PriceMM          DiscCH            DiscMM
##  0:312    Min.   :1.690   Min.   :1.690   Min.   :0.00000   Min.   :0.0000
##  1:489    1st Qu.:1.790   1st Qu.:1.990   1st Qu.:0.00000   1st Qu.:0.0000
##           Median :1.860   Median :2.090   Median :0.00000   Median :0.0000
##           Mean   :1.865   Mean   :2.088   Mean   :0.05192   Mean   :0.1281
##           3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000   3rd Qu.:0.2400
##           Max.   :2.090   Max.   :2.290   Max.   :0.50000   Max.   :0.8000
##    SpecialCH        SpecialMM         LoyalCH          SalePriceMM
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000014   Min.   :1.19
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.320000   1st Qu.:1.69
##  Median :0.0000   Median :0.0000   Median :0.585435   Median :2.09
##  Mean   :0.1548   Mean   :0.1685   Mean   :0.555908   Mean   :1.96
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.836160   3rd Qu.:2.18
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.999947   Max.   :2.29
##   SalePriceCH      PriceDiff          PctDiscMM         PctDiscCH
##  Min.   :1.390   Min.   :-0.6700   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:1.750   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :1.860   Median : 0.2400   Median :0.00000   Median :0.00000
##  Mean   :1.813   Mean   : 0.1464   Mean   :0.06164   Mean   :0.02739
##  3rd Qu.:1.890   3rd Qu.: 0.3200   3rd Qu.:0.11834   3rd Qu.:0.00000
##  Max.   :2.090   Max.   : 0.6400   Max.   :0.40201   Max.   :0.25269
##  ListPriceDiff
##  Min.   :0.0000
##  1st Qu.:0.1400
##  Median :0.2400
##  Mean   :0.2225
##  3rd Qu.:0.3000
##  Max.   :0.4400
```

```
# 3. There is no missing data - imputation is not necessary
sum(is.na(train))
```

```
## [1] 0
```

A correlogram confirms that there is high correlation between the thirteen variables. Some of them appear to be multicollinear, or not fully independent of one another (for correlation coefficients see Appendix).

```
corr <- cor(df[-1])
corr %>% cor() %>% corrplot()
```



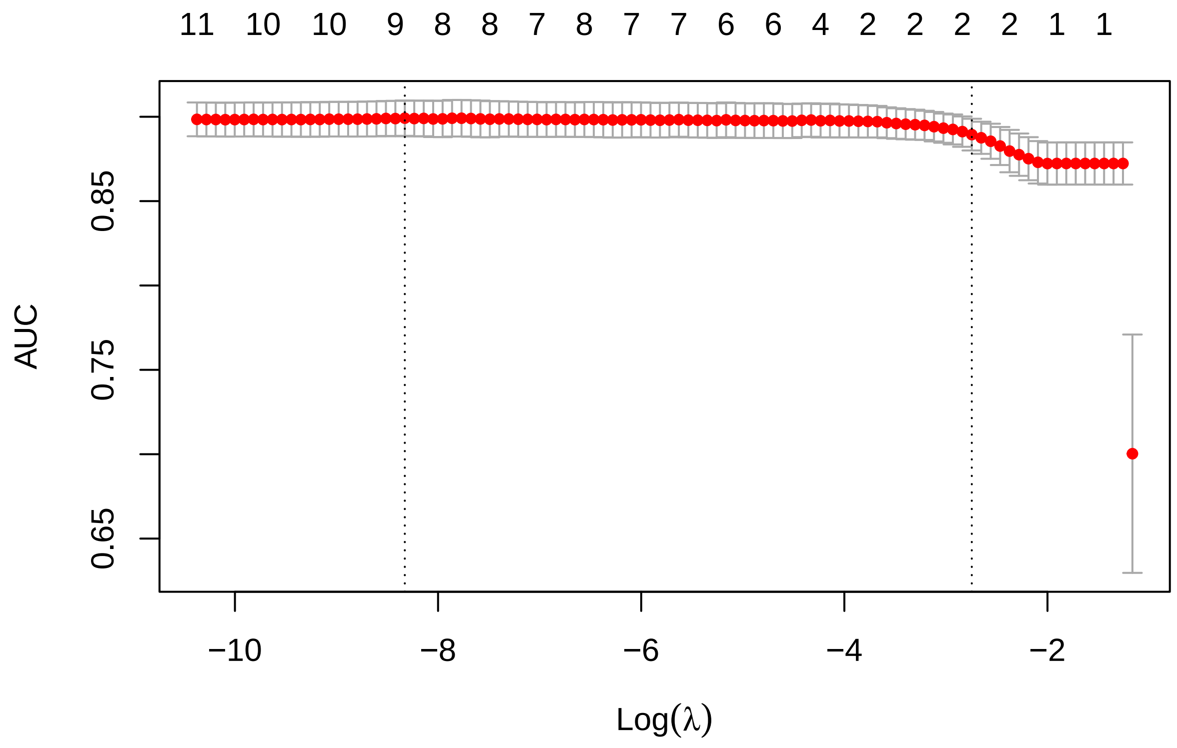
Accordingly, our team decided to use the "Lasso" method of logistic regression that regresses all variables against all other variables, modifying each variable's predictive weight based on its correlation to to other variables by strengthening, weakening, or even nullifying its effect.

Variable selection and model design The `cv.glmnet()` function below applied that method to our training data set, printing out coefficients for each variable that have been penalized or nullified if their relationship to other variables is multicollinear.

Additionally, in a microcosm of the training/test split we set up at the beginning of the project, this method cross-validates the results of the trained regression by testing it against seven different one-seventh chunks of the entire set.

The code below performs a logistic regression, but it uses Lasso (`alpha = 1`), giving us something to say about the magnitude and direction of variables, plus which variables' influences were shrunk to zero when all variables were regressed against each other (`Price MM`, `Disc CH`, `SalePriceCH`, and `PctDiscCH`). **Doing that gives us an AUC of 0.9**, and that's after inline k-fold validation of 7 when training the model.

```
predictors <- train[,c(2:13)] %>%
  mutate(
    SpecialCH = as.factor(SpecialCH),
    SpecialMM = as.factor(SpecialMM)
    )

predictors <- data.matrix(predictors)

set.seed(1234)
cv.binomial <- cv.glmnet(x = predictors, y = train$Purchase,
                         alpha = 1, family = "binomial",
                         nfolds = 7, standardize = TRUE, type.measure = "auc")

plot(cv.binomial)
```



```
(best.lambda <- cv.binomial$lambda.min)
```

```
## [1] 0.0002418264
```

```
y4<- coef(cv.binomial, s="lambda.min", exact=FALSE)
print(y4)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept)  -3.9603048
## PriceCH      -0.3496704
## PriceMM       .
## DiscCH        .
## DiscMM      -13.2136017
## SpecialCH     0.1216139
## SpecialMM    -0.2974192
## LoyalCH       6.6471993
## SalePriceMM   0.6000532
## SalePriceCH   .
## PriceDiff     3.5224475
## PctDiscMM    31.5219099
## PctDiscCH     .
```

Since the Lasso function of the regression has shrunk the effects of `PriceMM`, `DiscCH`, `SalesPriceCH` and `PctDiscCH` to zero or "`.`" in light of multicollinearity, management can be confident that those variables are not meaningful levers for action.

Interpreting the coefficients Since `cv.glmnet()` standardized/scaled the data so that the inconsistently sized ranges of values used by different variables won't accidentally weight different variables as more important, the coefficients above are not interpretable yet.

```
# add 11/28
(varInt <- y4 %>% as.matrix %>% as.data.frame)
```

```
##                        s1
## (Intercept)  -3.9603048
## PriceCH      -0.3496704
## PriceMM       0.0000000
## DiscCH        0.0000000
## DiscMM      -13.2136017
## SpecialCH     0.1216139
## SpecialMM    -0.2974192
## LoyalCH       6.6471993
## SalePriceMM   0.6000532
## SalePriceCH   0.0000000
## PriceDiff     3.5224475
## PctDiscMM    31.5219099
## PctDiscCH     0.0000000
```

```
varInt$varDdevs <- c("NA", sd(train$PriceCH), sd(train$PriceMM), sd(train$DiscCH), sd(train$DiscMM), sd

varInt <- varInt %>%
  mutate(
    s1 = ifelse(s1 == 0, NA, s1),
    logodds = ifelse(s1 == 0.000, NA, exp(varInt[,1])),
    varDdevs = ifelse(s1 == 0, NA, as.numeric(varDdevs)),
    varDdevs = round(varDdevs, 3),
    logodds = round(logodds, 3),
    s1 = round(s1,3)
  )

colnames(varInt) <- c(s1 = 'Coefficient', 'For every variable unit increase of this size...', '... the 

varInt
```

```
##             Coefficient For every variable unit increase of this size...
## (Intercept)      -3.960                                               NA
## PriceCH          -0.350                                            0.101
## PriceMM              NA                                               NA
## DiscCH               NA                                               NA
## DiscMM          -13.214                                            0.217
## SpecialCH         0.122                                            0.362
## SpecialMM        -0.297                                            0.375
## LoyalCH           6.647                                            0.305
## SalePriceMM       0.600                                            0.256
## SalePriceCH          NA                                               NA
## PriceDiff         3.522                                            0.275
## PctDiscMM        31.522                                            0.103
## PctDiscCH            NA                                                 NA
##              ... the odds of purchase increase by this much
## (Intercept)                                  1.900000e-02
## PriceCH                                      7.050000e-01
## PriceMM                                                NA
## DiscCH                                                 NA
## DiscMM                                       0.000000e+00
## SpecialCH                                    1.129000e+00
## SpecialMM                                    7.430000e-01
## LoyalCH                                      7.706230e+02
## SalePriceMM                                  1.822000e+00
## SalePriceCH                                            NA
## PriceDiff                                    3.386700e+01
## PctDiscMM                                    4.895438e+13
## PctDiscCH                                              NA
```

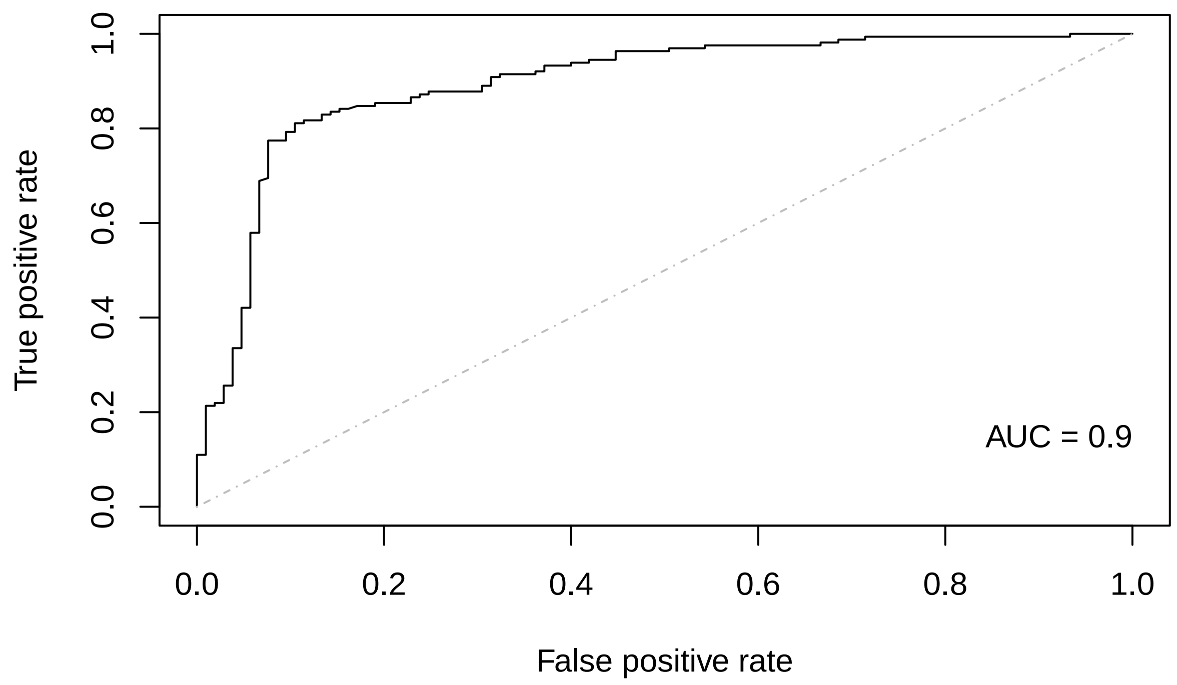
Our model reports that the three variables with the strongest effects are `LoyalCH`, `PriceDiff` and `PctDiscMM`. The odds of a customer purchasing MinuteMaid increase by 722 when the `LoyalCH`, the most influential variable in the model, increases by the variable's standard deviation of 30%. `PriceDiff` increases chance of purchase by 30 with every $0.27 increase in the difference between MinuteMaid and Citrus Hill. The odds of a MinuteMaid purchase increase by 18 for every 10% increase in `PctDiscMM`.

Performance against test data The predictions of this logistic regression performed well against the ground truth outcomes in the test set held in reserve at the beginning of our analysis.

Our regression turned variables into percentage likelihoods, but it is up to the analyst to decide what percentage triggers a label of "Yes; Purchased MinuteMaid", a decision called the "classification threshold." The area-under-the-curve (AUC) metric is a sign of an model's general performance in classification — a higher AUC means a model is good at balancing the risk of true positives to true negatives.

The area-under-curve for this model is 0.90.

```r
test_predictors <- test[,c(2:13)]
test_predictors <- data.matrix(test_predictors)
pred <- predict(cv.binomial, newx = test_predictors,
              type = "response", s ="lambda.min")
pred <- prediction(pred, test$Purchase)
perf <- performance(pred,"tpr","fpr")
auc_ROCR<- performance(pred,measure ="auc")

plot(perf,colorize=FALSE, col="black")
lines(c(0,1),c(0,1),col = "gray", lty = 4 )
text(1,0.15,labels=paste("AUC = ",round(auc_ROCR@y.values[[1]],
                                        digits=2),sep=""),adj=1)
```



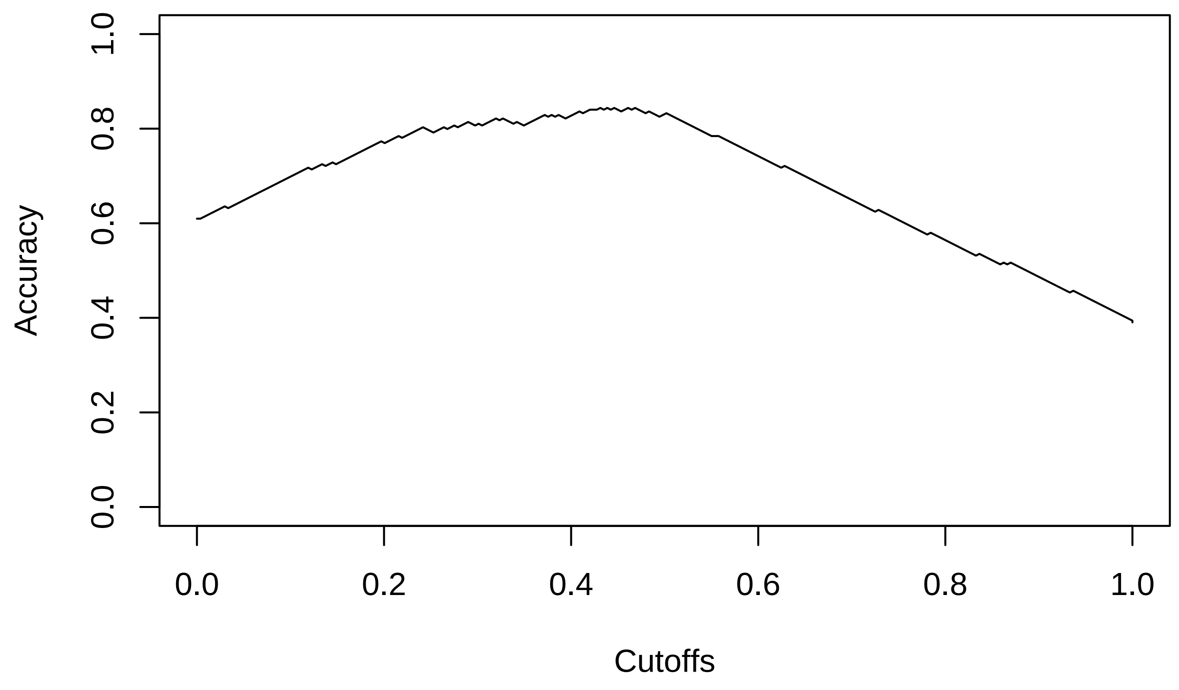
The AUC tells us how well our model can handle the balance between true and false positives, but we will ultimately need to choose the optimal threshold for our model.

The analysis below lets us know that the optimal classification threshold for our model is $P = 0.465$ — any customer with that high or higher a likelihood of purchasing MinuteMaid should be classified as "Yes; Will Purchase MinuteMaid." That probability threshold optimally balances the likelihood of true positives and the risk of false positives.

```r
# Calculate classification threshold of highest accuracy

# Add predictions to test set
test$lass_preds <- predict(cv.binomial, newx = test_predictors,
               type = "response", s ="lambda.min")

# Create data frame of accuracy rates at various thresholds

acc_matrix <- AUC::accuracy(test$lass_preds, test$Purchase)

acc_df <- data.frame(thresh = acc_matrix$cutoffs, acc = acc_matrix$measure)

# Isolate highest threshold and accuracy rate

acc_df %>%
  filter(acc == max(acc)) %>%
  filter(thresh == max(thresh))
```

```
##      thresh       acc
## 1 0.4684015 0.8438662
```

```
# View highest threshold and accuracy rate in context.

plot(acc_matrix, col="black")
```



Gradient Boosted Decision Trees:

Management also wants to be able to predict the likelihood that any given future customer will buy Minute Maid. Knowing how many customers are likely to purchase Minute Maid can help in (1) forecasting cash flow and supply chain demand and (2) targeting marketing to customers who are in the ideal position to buy and ignoring those who are not.

Decision tree modelling models the data and assigns a probabilistic decision path to assign classification, in this case either to a likely Minute Maid purchase or not. However, the way decision trees are assembled can lead to overfitting to the data if the tree is too deep or has too many branches, in addition they are prone to fall prey to data sampling errors, creating trees that reflect the train sample better than they do the ground truth. To overcome this, Gradient Boosted Trees (GBT) are a machine learning algorithm that overcomes the propensity of decision tree algorithms to overfit the data and susceptibility to data sampling errors. GBT overcomes this by building a more accurate complex model iteratively by combining many smaller less predictive models. Each successive round of learning seeks to explain the remaining error left by the previously assembled tree.

```
set.seed(1234)
recipe_oj <- recipe(Purchase ~ ., train)

model_oj_bt <- boost_tree(trees = tune(), tree_depth = tune(), learn_rate = tune()) %>%
  set_engine('xgboost', verbosity = 0) %>%
  set_mode('classification')
```

```r
hyperparameter_grid <- grid_regular(trees(), tree_depth(), learn_rate(), levels = 5)

purchase_folds <- vfold_cv(train, v=4) # 4-fold Cross validation

oj_workflow <- workflow() %>% add_model(model_oj_bt) %>% add_recipe(recipe_oj) #Set Workflow

# Tune Hyper-parameters
oj_tune <- oj_workflow %>% tune_grid(resamples = purchase_folds,
                                     grid = hyperparameter_grid,
                                     metrics = metric_set(accuracy))

best_bt_model <- oj_tune %>% select_best('accuracy') #Select best Hyper-parameters from grid
```

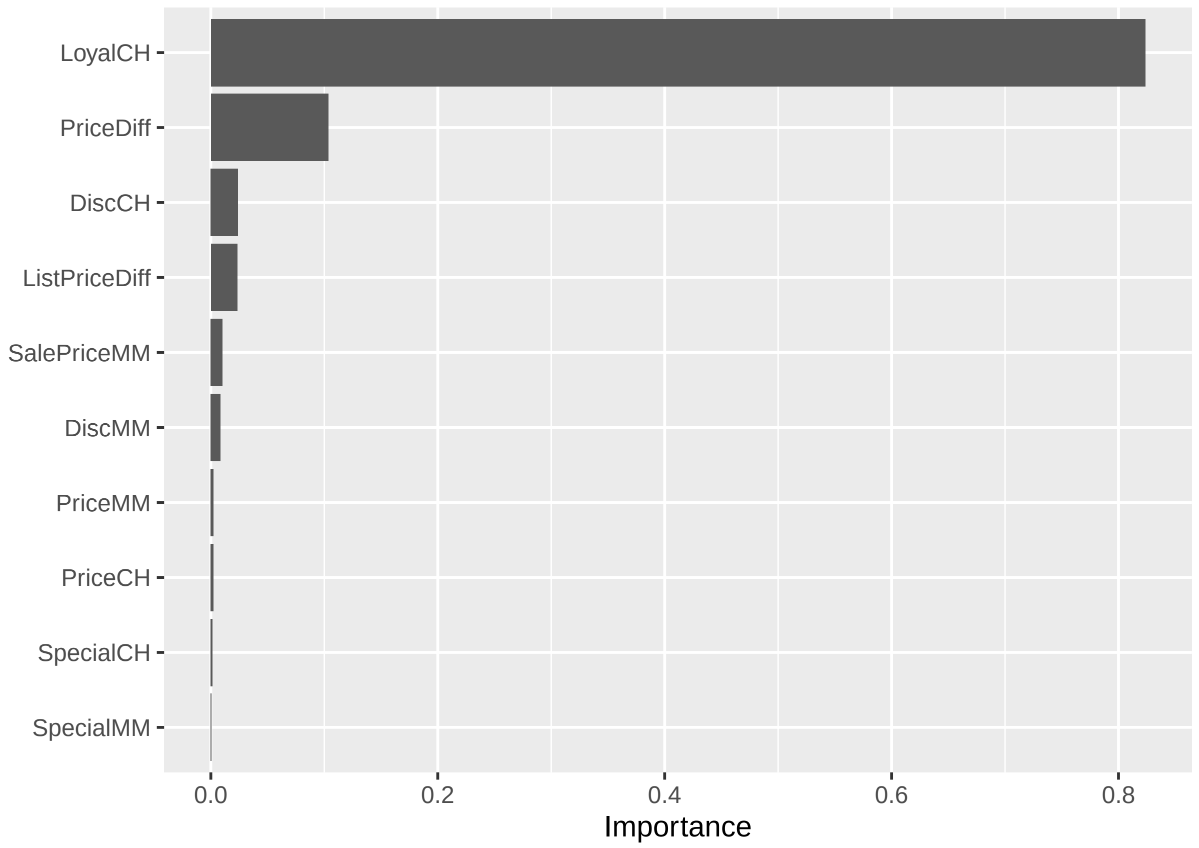
The hyperparameters for number of trees, tree depth, and learn rate for the boosted tree model were tuned using a grid with 5 levels and 4-fold cross validation. Hyperparameter performance was evaluated by overall model accuracy of prediction. The final hyperparameters for the model are number of trees (1000), tree depth (1), and learn rate (0.1).

```r
oj_final_workflow <- oj_workflow %>% finalize_workflow(best_bt_model) # Create Final Workflow based upo

final_fit <- oj_final_workflow %>% last_fit(split = purchase_testtrain) # Final Fit Model

final_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.799 Preprocessor1_Model1
## 2 roc_auc  binary         0.892 Preprocessor1_Model1
```

The finalized model gave an AUC of 0.89, which is comparable, but slightly underperforms the logistic regression model previously discussed.

```r
oj_final_workflow %>% fit(data = train) %>% extract_fit_parsnip() %>% vip(geom = 'col') #Plot most impo
```



```
vi_values <- oj_final_workflow %>% fit(data = train) %>% extract_fit_parsnip() %>% vi()

vi_values
```

```
## # A tibble: 11 x 2
##    Variable      Importance
##    <chr>              <dbl>
##  1 LoyalCH         0.824
##  2 PriceDiff       0.104
##  3 DiscCH          0.0239
##  4 ListPriceDiff   0.0232
##  5 SalePriceMM     0.0104
##  6 DiscMM          0.00852
##  7 PriceMM         0.00237
##  8 PriceCH         0.00237
##  9 SpecialCH       0.00133
## 10 SpecialMM       0.000512
## 11 SalePriceCH     0.000356
```

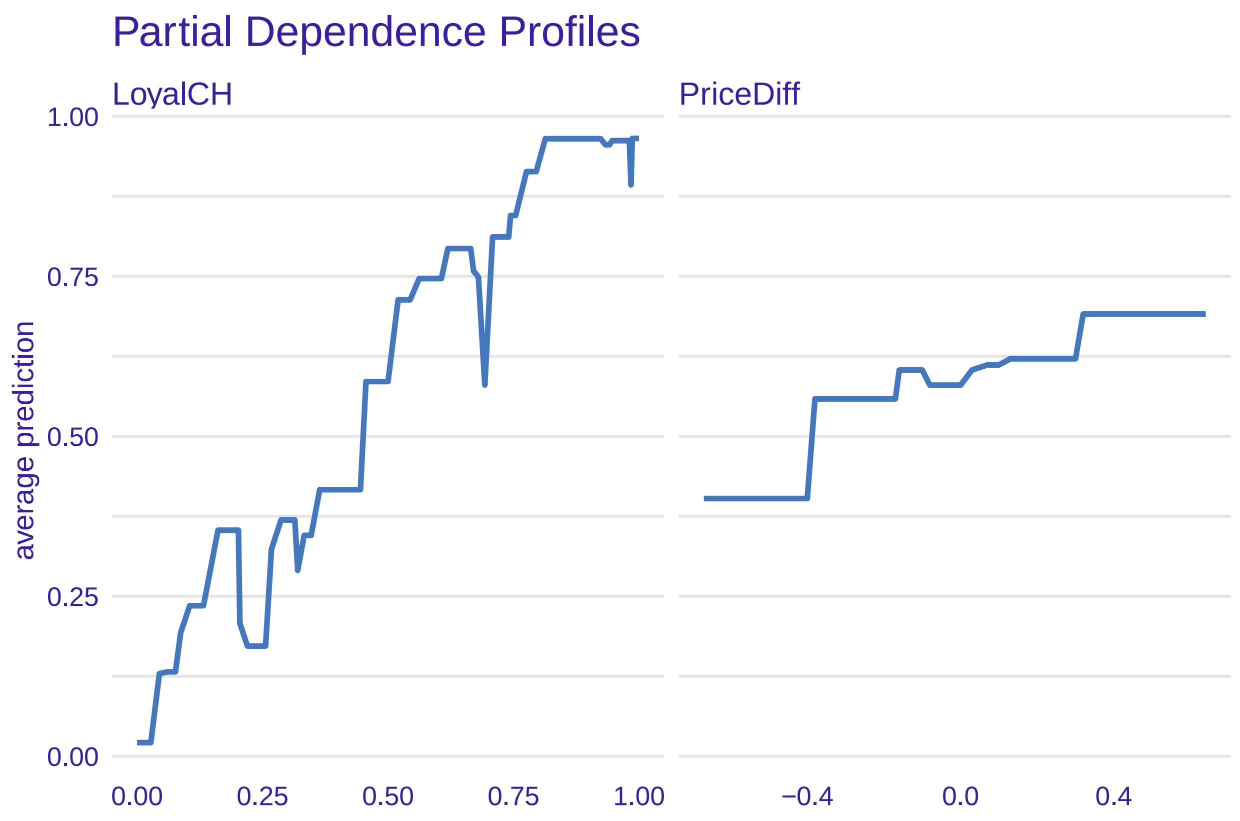
One drawback to using a black-box machine learning algorithm like Gradient Boosted Trees, is that understanding the insights the model provides are not immediately available, and the use of explanatory analysis is required to further understand what actions management can take to increase sales of Minute Maid. One such tool is the use of variable importance to understand which variables the model sees as most important in determining a customer outcome of **"Yes; Purchased Minute Maid"**.

The most important variable according to the Boosted Tree model is Customer Brand Loyalty to Citrus Hill(LoyalCH) with 82.35% importance, followed by Price Difference(PriceDiff) with 10.35% importance. All other independent variables displayed importance of <3%.

```
model_fitted <- oj_final_workflow %>% fit(data = train)

explainer_rf <- explain_tidymodels(model_fitted,
                                   data = train[,-1],
                                   y = train$Purchase,
                                   type = "pdp",verbose = FALSE)

pdp <- model_profile(explainer_rf,
                     variables = c("LoyalCH","PriceDiff"),
                     N=NULL)
```

In addition to understanding which variables are important for management to focus on, it is also important to understand how those variables interact with the prediction for Minute Maid purchases by the customer. It is useful to know that Brand Loyalty is important, but even more useful to know how to use that lever to identify potential crossover customers. Partial Dependence Profiling (PDP) allows some insight what is happening inside a blackbox model such as GBTs. The above plot shows the partial independent portion of a variable's influence on the dependent outcome variable. Comparable to information that can be obtained from linear or logistic regression.

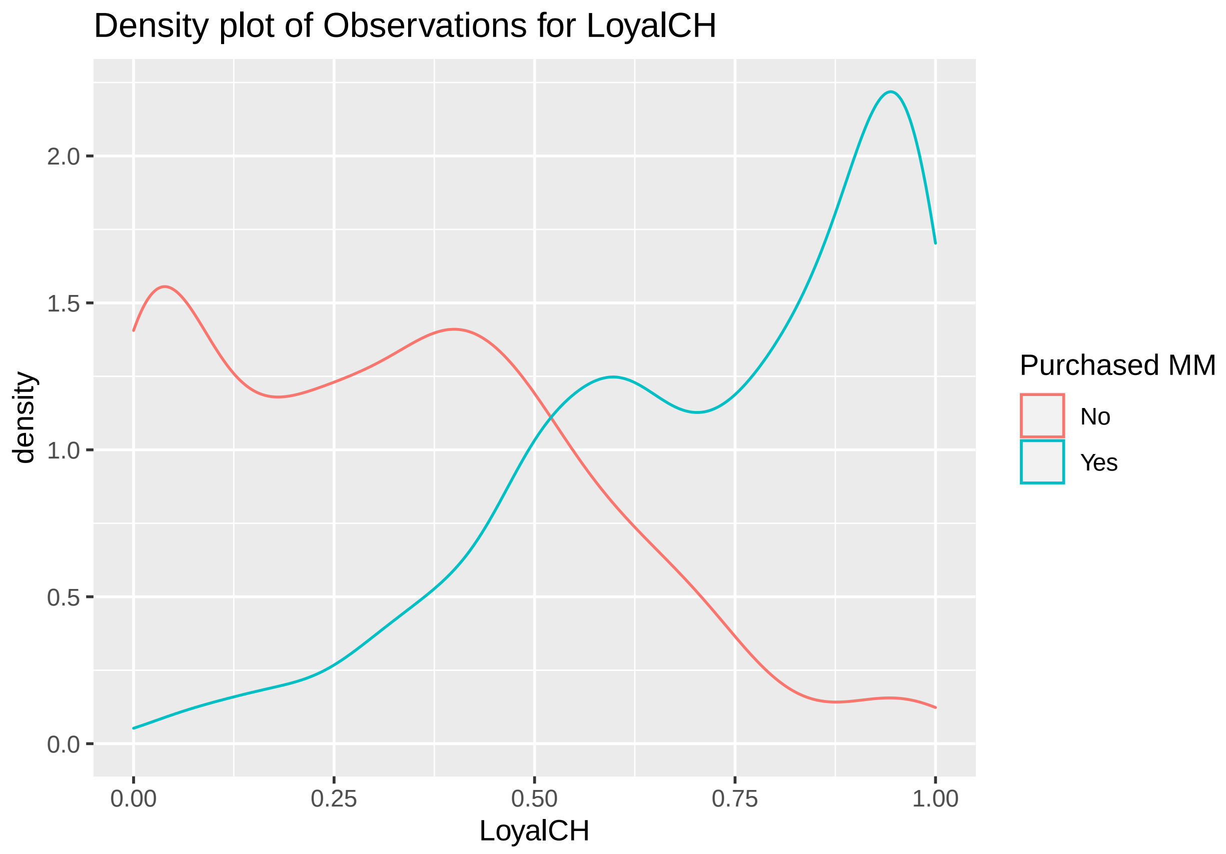
```
plot(pdp, title='Partial Dependence Profiles', subtitle=' ')
```



Both variables display a positive relationship with the purchase of Minute Maid. Meaning, that the more Brand Loyalty a customer displays towards Citrus Hill and the larger the price difference between MM and CH (in Citrus Hill's favor) the more likely the customer was to purchase Minute Maid. This would seem to be counter-intuitive and so it was verified by looking at the original data, where this observation was supported (see below). This would seem to indicate that there is a unique positioning opportunity for Minute

Maid in WGC stores.

```
ggplot2::ggplot(df, aes(LoyalCH, color=Purchase)) + geom_density() + scale_color_discrete(name='Purchase
```



Conclusions and Recommendations

At the beginning of this project we met with stakeholders in the Branding and Sales departments and identified key deliverables to ensure that this project provided actionable information and value to the company. Based upon our work we suggest the following interpretations and courses of action moving forward.

Brand

Both the logistic model and the explanatory analysis supporting the gradient boosted trees model give us insight into the predictor variables which influence the purchase of Minute Maid orange juice by our customers. Both models tell us that `LoyalCH`$(\beta = 6.58, \mathrm{i} = 82.4\%)$ and `PriceDiff`$(3.41, 10.4\%)$ and `PctDiscMM`$(\beta = 2.91)$ are primary contributors to a customers decision to purchase Minute Maid. `PriceMM`, `DiscCH`, `SalesPriceCH` and `PctDiscCH` do not contribute significantly to predicting customer behavior. All other variables are of limited significance, and provide little additional insight into customer behavior.

When examined holistically, it becomes apparent that two major factors are supported by the data. First, that customers that exhibit high levels of Citrus Hill Brand Loyalty are more likely to purchase Minute Maid. Second, that both discounting of Minute Maid and price parity between Minute Maid and Citrus Hill have antagonistic effects on customers choosing to purchase Minute Maid brand orange juice. These factors support the concept that Minute Maid should be positioned as a Premium brand within WGC stores, and that efforts to discount or price match Citrus Hill erode the customers perception of Minute Maid as a premium brand and should be avoided. It also supports the fact that loyal Citrus Hill purchasers can more appropriately be viewed as loyal Orange Juice purchasers and that targeting this customer segment with marketing techniques that enhance the perception of Minute Maid as a premium brand may lead to customer conversion.

Both models showed remarkable accuracy at predicting Minute Maid customer purchases as measured by AUC (LR = 0.90, GBT = 0.89). We can be very confident that these models are accurately capturing customer behavior. Understanding the factors which are making the models so accurate allows us to be equally confident in the recommendations arising from these models. Also of note is the fact that both methodologies independently found similar factors to be at work.

Sales

A key deliverable of this project was to explore the viability of a predictive model that could be used by the Sales Department to provide the probability a customer would purchase Minute Maid. We tested a predictive statistical model as well as a machine learning model. Both models performed well. When compared by AUC (a metric which represents a model's ability to correctly identify Minute Maid purchases balanced against predictions of purchase which do not occur) the logistic regression model slightly outperformed the machine learning model (see above). In terms of overall model accuracy the logistic regression model again outperformed the machine learning model (LR = 84.3%, GBT = 79.9%). The optimal decision threshold to achieve the most accurate results was a probability of ≥ 0.468. An additional benefit of the logistic regression model, it is also significantly more computationally efficient than the machine learning model.

No real world model will be perfect at correctly classifying customers as Minute Maid purchase vs. no purchase, however we propose that correctly classifying customers 84.3% of the time provides sufficient added value to the company that implementation of the model in the Sales Department will positively impact business operations in regards to Minute Maid orange juice sales.

Appendix: Data Characteristics

```
summary(df)
```

```
##  Purchase    PriceCH         PriceMM          DiscCH            DiscMM
##  0:417    Min.   :1.690   Min.   :1.690   Min.   :0.00000   Min.   :0.0000
##  1:653    1st Qu.:1.790   1st Qu.:1.990   1st Qu.:0.00000   1st Qu.:0.0000
##           Median :1.860   Median :2.090   Median :0.00000   Median :0.0000
##           Mean   :1.867   Mean   :2.085   Mean   :0.05186   Mean   :0.1234
##           3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000   3rd Qu.:0.2300
##           Max.   :2.090   Max.   :2.290   Max.   :0.50000   Max.   :0.8000
##    SpecialCH        SpecialMM         LoyalCH          SalePriceMM
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000011   Min.   :1.190
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.325257   1st Qu.:1.690
##  Median :0.0000   Median :0.0000   Median :0.600000   Median :2.090
##  Mean   :0.1477   Mean   :0.1617   Mean   :0.565782   Mean   :1.962
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.850873   3rd Qu.:2.130
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.999947   Max.   :2.290
##   SalePriceCH      PriceDiff          PctDiscMM        PctDiscCH
##  Min.   :1.390   Min.   :-0.6700   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:1.750   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :1.860   Median : 0.2300   Median :0.0000   Median :0.00000
##  Mean   :1.816   Mean   : 0.1465   Mean   :0.0593   Mean   :0.02731
##  3rd Qu.:1.890   3rd Qu.: 0.3200   3rd Qu.:0.1127   3rd Qu.:0.00000
##  Max.   :2.090   Max.   : 0.6400   Max.   :0.4020   Max.   :0.25269
##  ListPriceDiff
##  Min.   :0.000
##  1st Qu.:0.140
##  Median :0.240
##  Mean   :0.218
##  3rd Qu.:0.300
##  Max.   :0.440
```

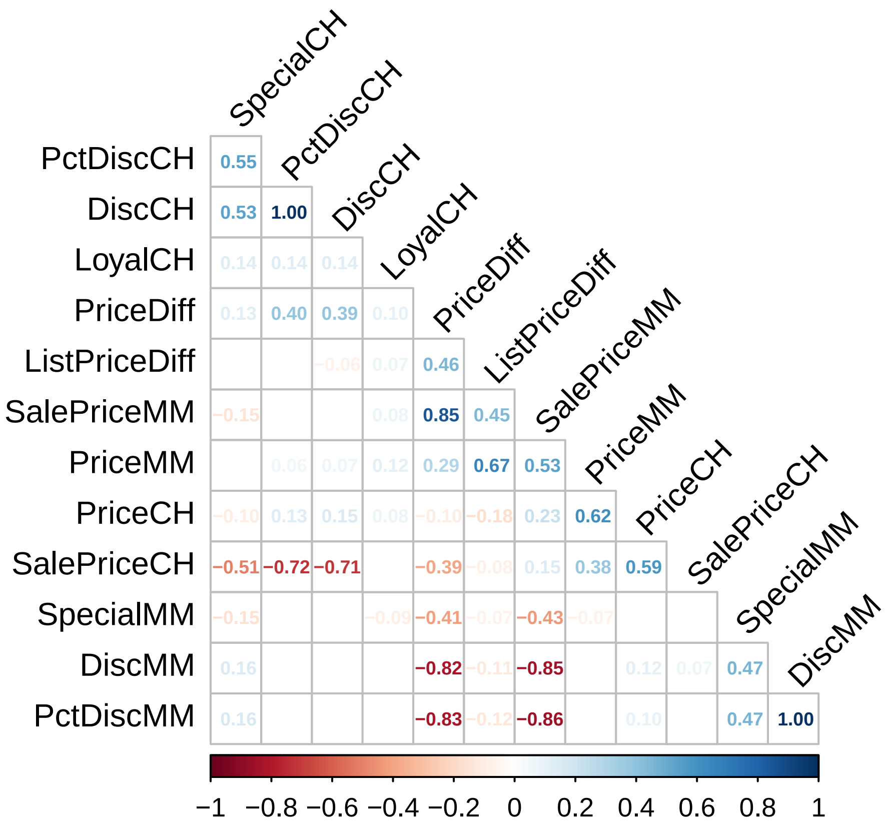
```
summary(test)
```

```
##  Purchase    PriceCH         PriceMM          DiscCH            DiscMM
##  0:105    Min.   :1.690   Min.   :1.690   Min.   :0.00000   Min.   :0.0000
##  1:164    1st Qu.:1.790   1st Qu.:1.990   1st Qu.:0.00000   1st Qu.:0.0000
##           Median :1.860   Median :2.090   Median :0.00000   Median :0.0000
##           Mean   :1.874   Mean   :2.079   Mean   :0.05167   Mean   :0.1094
##           3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000   3rd Qu.:0.2000
##           Max.   :2.090   Max.   :2.290   Max.   :0.50000   Max.   :0.8000
##    SpecialCH        SpecialMM         LoyalCH          SalePriceMM
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000011   Min.   :1.190
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.384000   1st Qu.:1.780
##  Median :0.0000   Median :0.0000   Median :0.635200   Median :2.090
##  Mean   :0.1264   Mean   :0.1413   Mean   :0.595184   Mean   :1.969
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.875808   3rd Qu.:2.130
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.999870   Max.   :2.290
##   SalePriceCH      PriceDiff          PctDiscMM          PctDiscCH
##  Min.   :1.390   Min.   :-0.6700   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:1.750   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :1.860   Median : 0.2300   Median :0.00000   Median :0.00000
##  Mean   :1.823   Mean   : 0.1468   Mean   :0.05231   Mean   :0.02709
##  3rd Qu.:1.890   3rd Qu.: 0.3000   3rd Qu.:0.09569   3rd Qu.:0.00000
## Max.   :2.090   Max.   : 0.6400   Max.   :0.40201   Max.   :0.25269
## ListPriceDiff    lass_preds.lambda.min
## Min.   :0.0000   Min.   :0.0188692
## 1st Qu.:0.1000   1st Qu.:0.3185372
## Median :0.2400   Median :0.8071108
## Mean   :0.2045   Mean   :0.6592625
## 3rd Qu.:0.2900   3rd Qu.:0.9659129
## Max.   :0.4400   Max.   :0.9962001
```

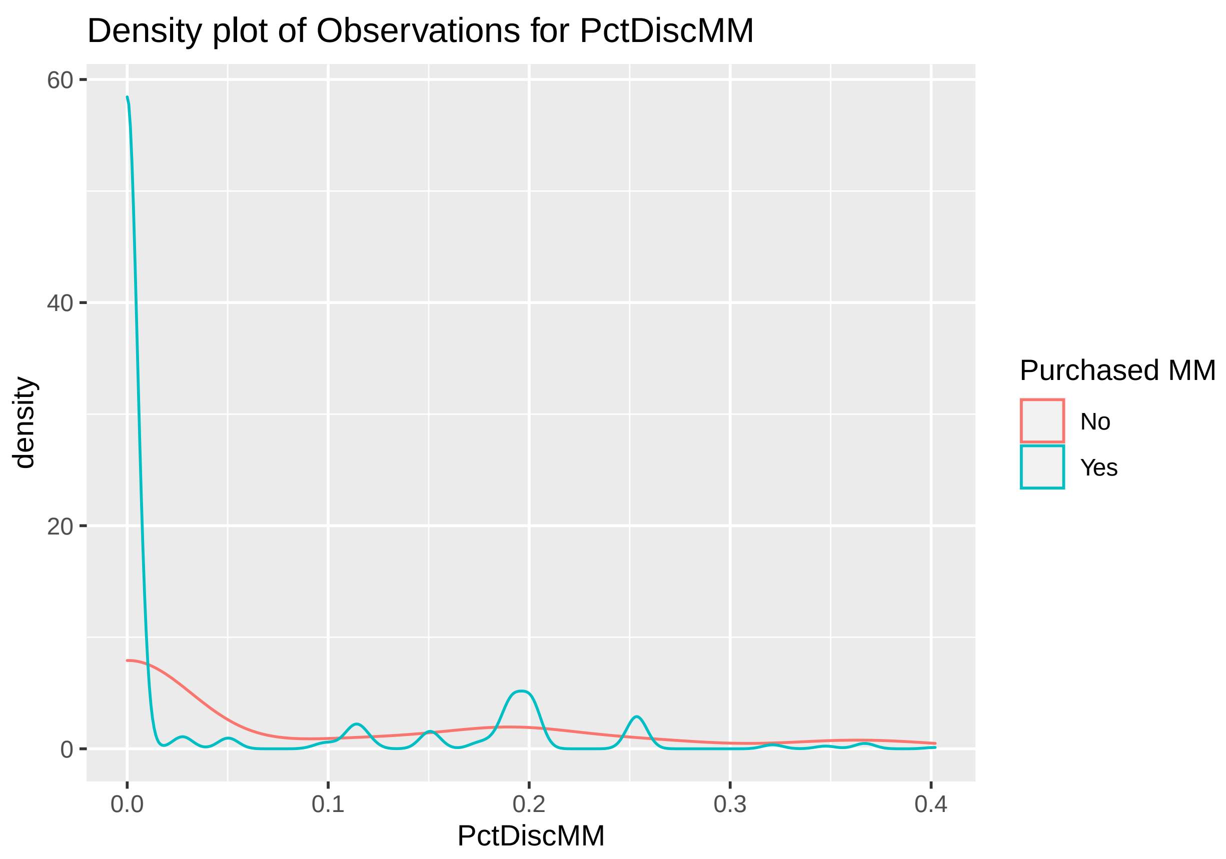
```
summary(train)
```

```
## Purchase    PriceCH         PriceMM          DiscCH            DiscMM
## 0:312     Min.   :1.690   Min.   :1.690   Min.   :0.00000   Min.   :0.0000
## 1:489     1st Qu.:1.790   1st Qu.:1.990   1st Qu.:0.00000   1st Qu.:0.0000
##           Median :1.860   Median :2.090   Median :0.00000   Median :0.0000
##           Mean   :1.865   Mean   :2.088   Mean   :0.05192   Mean   :0.1281
##           3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000   3rd Qu.:0.2400
##           Max.   :2.090   Max.   :2.290   Max.   :0.50000   Max.   :0.8000
##   SpecialCH        SpecialMM         LoyalCH          SalePriceMM
## Min.   :0.0000   Min.   :0.0000   Min.   :0.000014   Min.   :1.19
## 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.320000   1st Qu.:1.69
## Median :0.0000   Median :0.0000   Median :0.585435   Median :2.09
## Mean   :0.1548   Mean   :0.1685   Mean   :0.555908   Mean   :1.96
## 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.836160   3rd Qu.:2.18
## Max.   :1.0000   Max.   :1.0000   Max.   :0.999947   Max.   :2.29
##  SalePriceCH      PriceDiff          PctDiscMM         PctDiscCH
## Min.   :1.390   Min.   :-0.6700   Min.   :0.00000   Min.   :0.00000
## 1st Qu.:1.750   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.00000
## Median :1.860   Median : 0.2400   Median :0.00000   Median :0.00000
## Mean   :1.813   Mean   : 0.1464   Mean   :0.06164   Mean   :0.02739
## 3rd Qu.:1.890   3rd Qu.: 0.3200   3rd Qu.:0.11834   3rd Qu.:0.00000
## Max.   :2.090   Max.   : 0.6400   Max.   :0.40201   Max.   :0.25269
## ListPriceDiff
## Min.   :0.0000
## 1st Qu.:0.1400
## Median :0.2400
## Mean   :0.2225
## 3rd Qu.:0.3000
## Max.   :0.4400
```

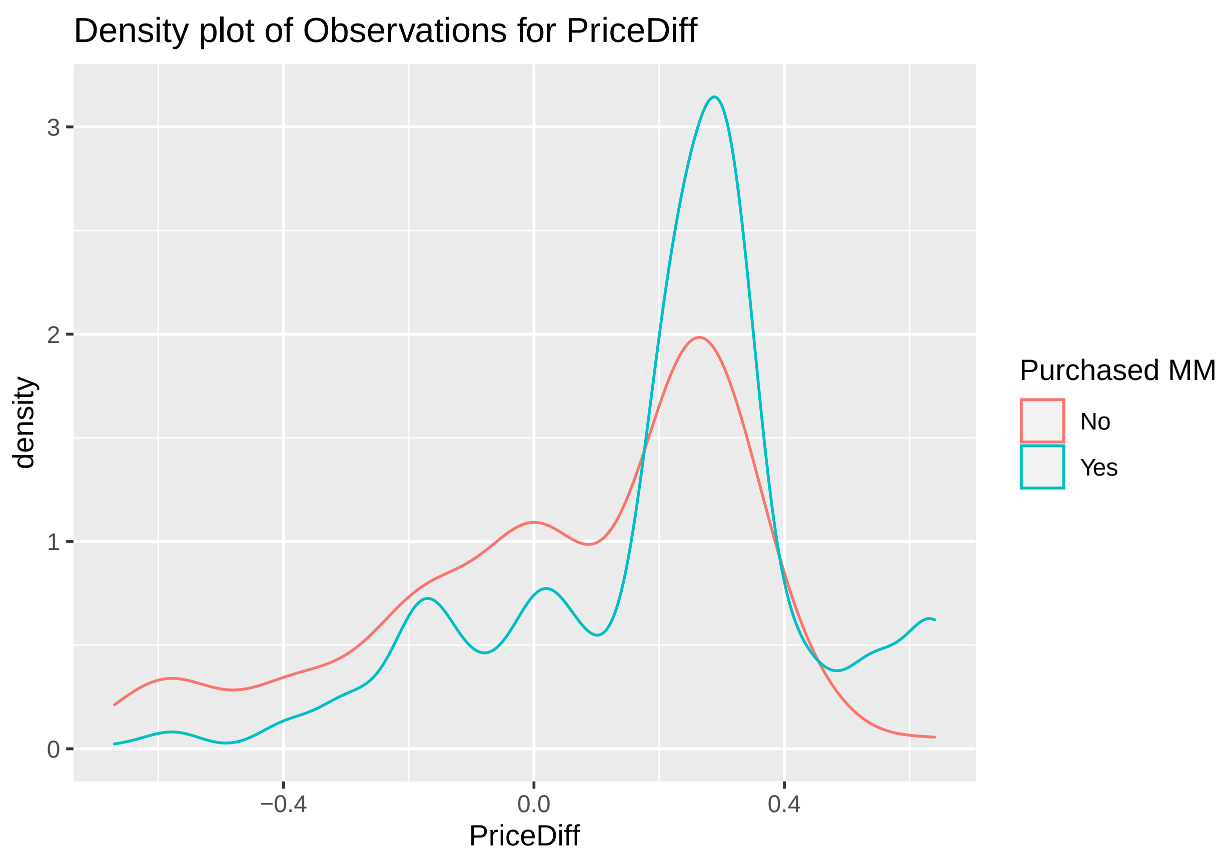
```
corr <- cor(df[-1]) #correlogram of numeric variables, excluding outcome variable
testDf <- cor.mtest(df[-1], conf.level = 0.95) #compute significance of correlation
# Plot correlogram with coefficients
corrplot(corr, p.mat = testDf$p, method = 'number', type = 'lower', insig='blank',
         addCoef.col ='black', number.cex = 0.6, order = 'AOE', diag=FALSE, tl.srt = 45, tl.col = 'blac
```



```
ggplot2::ggplot(df, aes(PctDiscMM, color=Purchase)) + geom_density() + scale_color_discrete(name='Purcha
```



```
#ggplot2::ggplot(df, aes(LoyalCH, color=Purchase)) + geom_density() + scale_color_discrete(name='Purcha

ggplot2::ggplot(df, aes(PriceDiff, color=Purchase)) + geom_density() + scale_color_discrete(name='Purcha
```


Density plot of Observations for PriceDiff
density
3
2
1
0
−0.4
0.0
0.4
PriceDiff
Purchased MM
No
Yes