MKTG 6620: Final Project

Sam Erickson

11/22/2020

Define the Problem

The business problem we are facing is how to increase performance of our orange juice sales. Currently, Minute Maid is outperforming Citrus Hill in overall margins. Both the brand manager and the sales manager would like to understand how to best increase sales of Minute Maid orange juice in order to best capitalize on the higher margins the brand provides.

In the last year, Minute Maid made up only on average 39% of the orange juice purchases at the grocery store chain. Since Minute Maid has shown to have much higher margins, the company would profit more from an increase in sales of Minute Maid orange juice. Thus, developing a model to understand the influence of other variables will enable the brand and sales manager to form an educated opinion on how to best proceed in improving performance of orange juice sales at the company.

Although both managers have the same higher goal of improving the orange juice category, each have their own specific questions and tasks that they would like answered and accomplished. The following two subsections discuss the specifics of each of their expectations.

Problem Specifics and Expectations: Brand Manager

While speaking with the Brand Manager, it is clear that he is most interested in finding out what variables influence a person's probability of buying Minute Maid orange juice. The expecations set from the brand manager are the following:

- What predictor variables influence the purchase of Minute Maid?
- Are all the variables in the dataset effective or are some more effective than others?
- How confident am I in my recommendations?
- Based on my analysis, what are some specific recommendations I have for the brand manager?

Problem Specifics and Expectations: Sales Manager

In addition to knowing the variables of importance, the sales manager is also interested in having a predictive model where he can simply predict the probability of a customer purchasing Minute Maid orange juice. His specific expectations are the following:

- Provide him a predictive model that can tell him the probability of customers buying Minute Maid orange juice.
- Insight into how good the model is in its predictions.
- An understanding of how confident I am in my recommendations.

Define Method(s):

While working on this data I first tried testing both a logistic regression and a SVM model. After analyzing the two models I came to the conclusion that the logistic regression model was sufficient to answer the needs of both the brand and sales manager. Through the logistic regression I was able to compute an accuracy score as well as a kappa coefficient to help in determining predictive capability to meet the needs of the sales manager; and through this same model I was able to view which variables had greatest significance on the purchase outcome in order to best respond to the needs of the brand manager.

The following points of information explain further into the methodology I took to complete this report:

I first prepared the data by using the lapply function to understand what class each variable was assigned. Using this I converted Purchase, SpecialCH, SpecialMM, Store7, and STORE to factor variables - per information provided from the data dictionary.

```
OJ<-read.csv(url("http://data.mishra.us/files/OJ.csv"))
lapply(OJ, class)
```

```
## $Purchase
## [1] "character"
##
## $WeekofPurchase
## [1] "integer"
##
## $StoreID
## [1] "integer"
##
## $PriceCH
## [1] "numeric"
##
## $PriceMM
## [1] "numeric"
##
## $DiscCH
## [1] "numeric"
##
## $DiscMM
## [1] "numeric"
##
## $SpecialCH
## [1] "integer"
##
## $SpecialMM
## [1] "integer"
##
## $LoyalCH
## [1] "numeric"
##
## $SalePriceMM
## [1] "numeric"
##
## $SalePriceCH
## [1] "numeric"
##
```

```
## $PriceDiff
## [1] "numeric"
##
## $Store7
## [1] "character"
##
## $PctDiscMM
## [1] "numeric"
##
## $PctDiscCH
## [1] "numeric"
##
## $ListPriceDiff
## [1] "numeric"
##
## $STORE
## [1] "integer"
```

```
OJ$Purchase <- as.factor(OJ$Purchase)
OJ$SpecialCH <- as.factor((OJ$SpecialCH))
OJ$SpecialMM <- as.factor(OJ$SpecialMM)
OJ$Store7 <- as.factor(OJ$Store7)
OJ$STORE <- as.factor(OJ$STORE)
```

I then ran an original logistic regression of Purchase against all variables.

Original Logistic Regression Model looking at Purchase against all variables

```
Model1 <- glm(Purchase ~., data = OJ, family = binomial(link = 'logit'))

summary(Model1)#$coefficients
```

```
##
## Call:
## glm(formula = Purchase ~ ., family = binomial(link = "logit"),
##     data = OJ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7518  -0.5413  -0.2306   0.5265   2.8005
##
## Coefficients: (6 not defined because of singularities)
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.65324    2.45280   2.305  0.02118 *
## WeekofPurchase  -0.01123    0.01137  -0.988  0.32320
## StoreID         -0.31883    0.35198  -0.906  0.36503
## PriceCH          4.56495    1.88740   2.419  0.01558 *
## PriceMM         -3.68490    0.91548  -4.025 5.69e-05 ***
## DiscCH          11.27171   18.82388   0.599  0.54931
## DiscMM          25.44305    9.31290   2.732  0.00629 **
## SpecialCH1       0.26336    0.34304   0.768  0.44265
## SpecialMM1       0.31350    0.27586   1.136  0.25576
## LoyalCH         -6.24874    0.41040 -15.226  < 2e-16 ***
## SalePriceMM           NA         NA      NA       NA
```

```
## SalePriceCH           NA         NA      NA       NA
## PriceDiff             NA         NA      NA       NA
## Store7Yes        0.84026    1.31186   0.641  0.52184
## PctDiscMM      -48.55687   19.50068  -2.490  0.01277 *
## PctDiscCH      -28.26083   35.57517  -0.794  0.42696
## ListPriceDiff         NA         NA      NA       NA
## STORE1          -0.42484    0.92197  -0.461  0.64494
## STORE2          -0.15315    0.58645  -0.261  0.79398
## STORE3                NA         NA      NA       NA
## STORE4                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.85  on 1069  degrees of freedom
## Residual deviance:  816.25  on 1055  degrees of freedom
## AIC: 846.25
##
## Number of Fisher Scoring iterations: 5
```

When I called the summary function I found that multicollinearity did indeed exist and decided to build a correlation matrix and plot to understand more clearly what was going on.

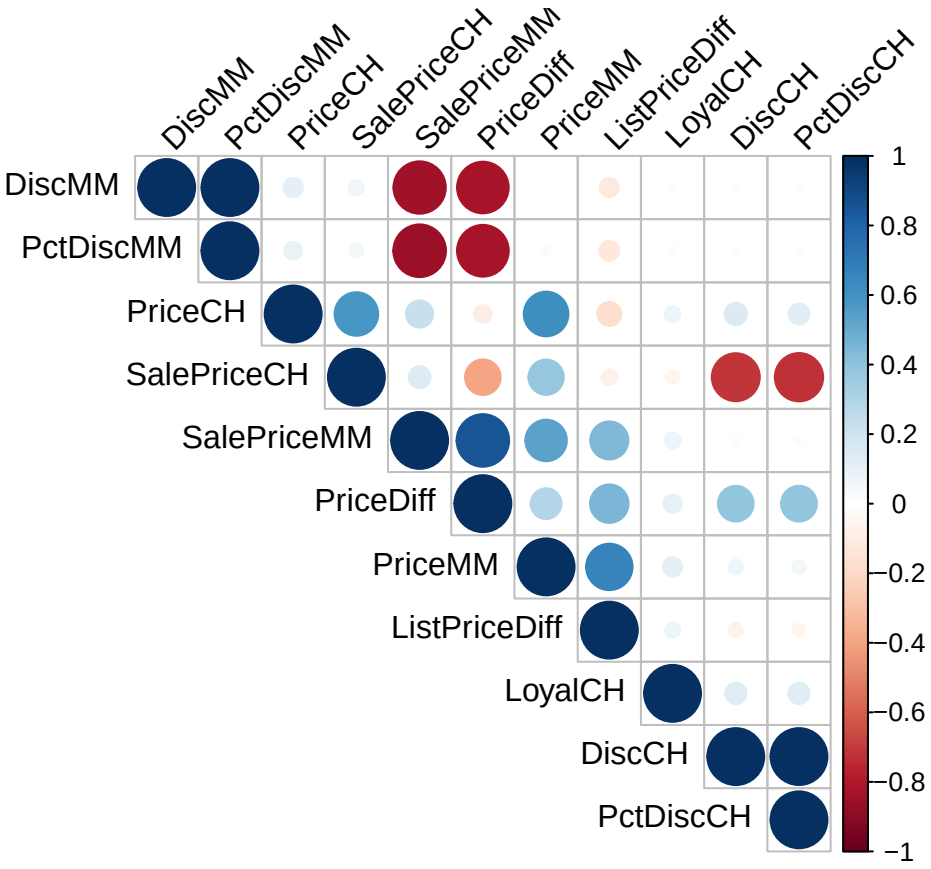
```
#code to select numeric variables to run correlations
oj <- OJ[, c(4,5,6,7,10,11,12,13,15,16,17)]
#res is the variable assigned to the cor() function called on the oj variable
res <- cor(oj)

library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.3
```

```
## corrplot 0.84 loaded
```

```
#plot to visually see correlation
corrplot(res,type = "upper", order="hclust",tl.col = "black", tl.srt = 45)
```




```
#
alias(Model1)
```

```
## Model :
## Purchase ~ WeekofPurchase + StoreID + PriceCH + PriceMM + DiscCH +
##     DiscMM + SpecialCH + SpecialMM + LoyalCH + SalePriceMM +
##     SalePriceCH + PriceDiff + Store7 + PctDiscMM + PctDiscCH +
##     ListPriceDiff + STORE
##
## Complete :
##               (Intercept) WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM
## SalePriceMM    0           0              0       0       1       0     -1
## SalePriceCH    0           0              0       1       0      -1      0
## PriceDiff      0           0              0      -1       1       1     -1
## ListPriceDiff  0           0              0      -1       1       0      0
## STORE3         4           0             -1       0       0       0      0
## STORE4        -3           0              1       0       0       0      0
##               SpecialCH1 SpecialMM1 LoyalCH Store7Yes PctDiscMM PctDiscCH
## SalePriceMM    0          0          0       0         0         0
## SalePriceCH    0          0          0       0         0         0
## PriceDiff      0          0          0       0         0         0
## ListPriceDiff  0          0          0       0         0         0
## STORE3         0          0          0       3         0         0
## STORE4         0          0          0      -4         0         0
##               STORE1 STORE2
## SalePriceMM    0      0
```

```
## SalePriceCH    0      0
## PriceDiff      0      0
## ListPriceDiff  0      0
## STORE3        -3     -2
## STORE4         2      1
```

I found that not only did mulitcollinearity exist, but there were several instances of perfect multicollinearity which I discovered by running the alias() function on the original model.

With this information I decided to remove the variables of StoreID, SalePriceMM, SalePriceCH, PriceDiff, ListPriceDiff, STORE, PctDiscMM, and PctDiscCH to improve model accuracy and to avoid overfitting. The below model is the logistic regression now without multicollinearity.

```
Model1 <- glm(Purchase ~. -StoreID -SalePriceMM - SalePriceCH -PriceDiff -ListPriceDiff -STORE -PctDiscl

summary(Model1)#$coefficients
```

```
##
## Call:
## glm(formula = Purchase ~ . - StoreID - SalePriceMM - SalePriceCH -
##     PriceDiff - ListPriceDiff - STORE - PctDiscMM - PctDiscCH,
##     family = binomial(link = "logit"), data = OJ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7864  -0.5522  -0.2419   0.5354   2.7782
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.262340   1.727140   1.889 0.058909 .
## WeekofPurchase -0.001722   0.009347  -0.184 0.853848
## PriceCH         3.452054   1.379886   2.502 0.012360 *
## PriceMM        -3.112024   0.858444  -3.625 0.000289 ***
## DiscCH         -3.565320   1.105089  -3.226 0.001254 **
## DiscMM          2.330014   0.531466   4.384 1.16e-05 ***
## SpecialCH1      0.098777   0.329731   0.300 0.764506
## SpecialMM1      0.311618   0.270866   1.150 0.249958
## LoyalCH        -6.288362   0.394216 -15.952  < 2e-16 ***
## Store7Yes      -0.609153   0.215778  -2.823 0.004757 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.85  on 1069  degrees of freedom
## Residual deviance:  825.01  on 1060  degrees of freedom
## AIC: 845.01
##
## Number of Fisher Scoring iterations: 5
```

Running the function vif() I am able to access the variance inflation factor which explains the amount of variance that a regression coefficient is inflated by due to existing multicollinearity in the model. A low VIF score is always best and preferred to be below 10.

```
car::vif(Model1)
```

```
## WeekofPurchase        PriceCH        PriceMM         DiscCH         DiscMM
##       2.959493       2.860345       2.085889       1.515556       1.702604
##      SpecialCH      SpecialMM        LoyalCH         Store7
##       1.607303       1.419025       1.093839       1.219943
```

```
#create new data set with highly correlated variables removed.
OJ_clean <- OJ[-c(3,11,12,13,15,16,17,18)]
```

In relation to overfitting, I used cross validation in efforts to decrease overfitting and improve my model which in turn resulted in a lower AIC score than the original model.

Although I would normally be inclined to include all variables in an effort to avoid bias and overfitting, the dataset was so entrenched in multicollinearity that excluding those variables actually resulted in the more accurate and least biased model. This was due to singularity issues and redundancy among variables.

I would like to also include this note in here that in the following section you will see that I did in fact run a SVM model on the data to compute RMSE. The RMSE was indeed a low score indicating that the model was a good fit, however the logistic regression model was sufficient to respond to the needs of the team.

Results and Conclusion

Sales Manager

Below is the depiction of the Prediction Model created by using cross-validation on logistic regression. Through the process of cross-validation, I was able to achieve a model that performs at an accuracy rate of 81% - which is very significant - enabling us to be confident in the predictions that the model can perform.

Another value that is important to my recommendation of the usage of this model is the AIC metric which is a value of 608.89. In statistics, the AIC is an estimate to help us understand the overall fit of the model and accuracy. When comparing models, the lower the AIC the better the model is at explaining and predicting the data. Therefore, since this model's AIC metric is lower than that of our original logistic regression, we can conclude that this model is a more excellent fit in providing accurate predictions for future data.

```
set.seed(1234)

inTrain <- createDataPartition(OJ_clean$Purchase, p = .70, list = FALSE)

purchaseTrain <- OJ_clean[inTrain, ]
purchaseTest <- OJ_clean[-inTrain, ]







train_control <- trainControl(method = "cv", number = 10)

#train model on training set
predictionModel <- train(Purchase~.,
                         data = purchaseTrain,
                         trControl = train_control,
                         method = "glm",
                         family = binomial(link = 'logit'))


summary(predictionModel)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7381  -0.5465  -0.2559   0.5468   2.6966
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.010098   2.063384   1.459 0.144616
## WeekofPurchase -0.006855   0.011198  -0.612 0.540440
## PriceCH         3.852753   1.626927   2.368 0.017879 *
## PriceMM        -2.789533   1.039237  -2.684 0.007270 **
## DiscCH         -3.717263   1.278735  -2.907 0.003649 **
## DiscMM          2.193323   0.644117   3.405 0.000661 ***
```

```
## SpecialCH1      0.149567   0.389249   0.384 0.700796
## SpecialMM1      0.305343   0.317609   0.961 0.336361
## LoyalCH        -5.918644   0.446076 -13.268  < 2e-16 ***
## Store7Yes      -0.667847   0.257510  -2.593 0.009501 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1002.67  on 749  degrees of freedom
## Residual deviance:  588.89  on 740  degrees of freedom
## AIC: 608.89
##
## Number of Fisher Scoring iterations: 5
```

```
predictionModel$results
```

```
##   parameter  Accuracy     Kappa AccuracySD    KappaSD
## 1      none 0.8117724 0.6013127 0.03890442 0.07883016
```

Brand Manager:

As seen in the following table, the most significant variables to the purchase outcome were first, if the customer was a Loyal Citrus Hill customer meaning that their historical purchase behavior resulted in them continuing to purchase Citrus Hill brands. Although the first variable pertains only to Citrus Hill, the second most significant variable was DiscMM which represented discounts offered on Minute Maid orange juice. This means that for every unit increase in Discounts to Minute Maid, the log odds of purchasing Minute Maid orange juice increase by 2.19 units.

Another interesting takeaway from this model is that the Price of Minute Maid is also statistically significant, meaning that for every unit increase in the price of Minute Maid orange juice, the log odds of a customer buying that brand decreases by 2.79 units.

```
importance <- varImp(predictionModel)

importance
```

```
## glm variable importance
##
##                Overall
## LoyalCH        100.000
## DiscMM          23.447
## DiscCH          19.580
## PriceMM         17.851
## Store7Yes       17.147
## PriceCH         15.398
## SpecialMM1       4.479
## WeekofPurchase   1.769
## SpecialCH1       0.000
```

Having this knowledge, I can be on average 81% confident that as a brand manager you should focus your efforts on the Minute Maid discounts as well as the list prices for the brand. Through discounting and/or

lowering the list price of Minute Maid products, we are 81% confident that we can begin to capture more of the orange juice customer market thus further enabling us to capitalize on the greater margins that come from selling Minute Maid products to our customers.

A final suggestion is derived from the principle of substitution, meaning that if you have two similar products and one products price increases, demand for that of the lower price but similar product increases while demand for the now higher priced product decreases. Therefore, increasing the price of Citrus Hill or decreasing the price of Minute Maid will result in more sales to Minute Maid as these two products are able to be substituted.

In conclusion, having accounted for multicollinearity and with the model accuracy of 81% I am confident in my recommendations to use this model as a predictive model to improve the overall performance of the orange juice category at our company. As we look to significant variables to monitor I would advise to focus on Discounts of Minute Maid orange juice, the list price of Minute Maid, and the price of its competitor Citrus Hill. Focusing our attention on these variables will enable us to shift more of the customer base towards Minute Maid and further enable us to capitalize on its higher margins.

Appendix

The following code is showing the SVM model and RMSE result that I had performed while deciding between which model to use for my analysis.

```
set.seed(1234)
# one-hot encoding for the entire data set
onehot_data1 <- as.data.frame(model.matrix( ~ .-1, OJ_clean) )
onehot_data1 <- onehot_data1 %>% mutate(id = row_number())
onehot_data1 <- onehot_data1[,-1]
#Create training set
trainset <- onehot_data1 %>% sample_frac(.70)
#Create test set by selecting id values that are not in trainset
testset <- anti_join(onehot_data1, trainset, by = 'id')
trainset=subset(trainset, select =-c(id))
testset=subset(testset, select =-c(id))
predictor_x <- trainset[, c(1:10)] # picking predictors
outcome_y <- trainset,"PurchaseMM"]


model3 <- tune(svm,PurchaseMM~., data=trainset,
ranges = list(epsilon = seq(0,1,0.5), cost = 2^(2:3)))
summary(model3)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  epsilon cost
##      0.5    4
##
## - best performance: 0.1388408
##
## - Detailed performance results:
##   epsilon cost     error  dispersion
## 1     0.0    4 0.1390333 0.022444878
## 2     0.5    4 0.1388408 0.015913336
## 3     1.0    4 0.2373705 0.002299201
## 4     0.0    8 0.1442692 0.022898103
## 5     0.5    8 0.1449411 0.017552177
## 6     1.0    8 0.2360148 0.002938530
```

```
# Model fitting
# try other values of epsilon and cost by chnaging the range in the tuning
# code. This is how you would reduce SVR RMSE
model3<- svm(PurchaseMM~., data=trainset, kernel="radial",
cost=4, epsilon=0.5)

test_x <- testset[,c(1:10)]
test_y <- testset[,"PurchaseMM"]
```

```r
predictedY <- predict(model3, test_x)

error1 <- test_y - predictedY
rmse <- function(error1)
{sqrt(mean(error1^2))}
# calculate the RMSE error for the SVM
svrPredictionRMSE <- rmse(error1)
svrPredictionRMSE
```

```
## [1] 0.3810256
```

References

Himanshu, Mishra. Understanding your Customers: Logistic Regression. 2019

Himansu, Mishra. Support Vector Machines. 2019

Shaikh, Raheel. "Cross Validation Explained: Evaluating estimator performance." Towards Data Science, Nov 26 2018, https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

Kassambara. "Multicollinearity Essentials and VIF in R." STHDA. Nov 3 2018. http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/#:~:text=For%20a%20given%20predictor%20(p,to%20multicollinearity%20in%20the%20model.

Alboukadel."Kappa Coefficient Interpretation." Nov 14. https://www.datanovia.com/en/blog/kappa-coefficient-interpretation/