Orange Juice Analytics Project

MKTG 6620 PROJECT 1

ADESH VALECHA, SUSHRUTI ACHARYA

BUSINESS PROBLEM

As per the discussion with Brand manager and Sales manager of the grocery store, they want to know how to make the Orange Juice category perform better than what is does currently. Particularly, the Brand manager wants to find out what variables influences the probability of purchasing MM and what he can do to increase the sales. Our objective is to determine the variables that influence the person's probability of purchasing MM and how those variables can be adjusted to increase this probability. The Sales Manager wants to have a predictive model to predict the probability of a customer purchasing MM. We need to build a predictive model that will predict the probability of purchasing MM with a high accuracy.

The problems of the Brand Manager and Sales Manager seems to be different. But when we analyze the problem, we observe that both are interested to make the orange juice perform better than what it does currently. We can accomplish this task by building the model and determining the factors which most influences the sale of MM. Further, using these important factors we can develop a model to predict the probability of customer buying the MM orange juice.

ANALYSIS METHODS USED

Data Preparation and Variable Selection

On exploring the data, we found that variable 'STORE' and 'StoreID' are exact bijections. So, we removed the variable 'STORE'. Later, we also observed that there several variables are highly correlated. To validate this, we used correlation matrix and removed the highly correlated variables from our dataset. Following variables were removed

- PriceDiff
- SalePriceMM
- SalePriceCH
- PctDiscMM
- DiscCH
- weekofPurchase

When we use all remaining variables in a prediction model, the model does not perform well, as measured by a higher AIC. We are also dropping the below variables because their impact was not significant (as measured by p-values greater than .05) in predicting a customer's purchase.

- StoreID
- SpecialCH
- SpecialMM
- Store7
- ListPriceDiff
-

In order to reduce overfitting in our 1070 observation data set and to ensure that there was no sampling bias, we created a random 70-30 train/test split, and we are setting seed at (100) to allow replication.

Brand Manager:

To determine the variables that influence sales of MM, we used a logistic regression model with the below five independent variables. Each of these variables proved to be significant in determining MM sales, as measured by their low p-values.

- PriceCH
- PriceMM
- DiscMM
- LoyalCH
- PctDiscCH

We have scaled these variables to see how large of an effect size they each have. Doing this helped us identify which variables had the largest impact on the odds of Purchasing MM.

Logistic Model Performance on Test Data

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  X0  X1
##         X0 171  30
##         X1  24  95
##
##               Accuracy : 0.8312
##                 95% CI : (0.7856, 0.8706)
##    No Information Rate : 0.6094
##    P-Value [Acc > NIR] : <2e-16
```

Sales Manager:

To develop a predictive model to determine probability of buying MM, we used a radial kernel. The reason we used this instead of the logistic model developed for the brand manager is because the predictive ability was greater (AUC = 0.8912, Accuracy = 0.8312 for logistic and AUC = 0.8919, Accuracy = 0.8406 for svm). We also found the svmRadial model provides best

predictability over any polynomial or linear kernels. Since the sales manager is not interested in variables impacting the sales, we do not need to use a regression model for this case.

Similar to the logistic model, we have used scaled and standardized variables to identify which to include in the model as it provided greater predictive accuracy. This is evident in the accuracy score of the confusion matrix below.

SVM Model Performance on Test Data

```
Confusion Matrix and Statistics
##
##          Reference
## Prediction  X0  X1
##         X0 173  29
##         X1  22  96
##
##               Accuracy : 0.8406
##                 95% CI : (0.7958, 0.879)
##    No Information Rate : 0.6094
##    P-Value [Acc > NIR] : <2e-16
```

RESULTS AND CONCLUSIONS

Brand Manager:

Our model answers the question of what variables influence the sales of MM. The five variables below are most effective. From this output we know that customer Loyalty to CH has a negative impact on the probability of purchasing MM. We can also see that increasing the price of MM, or increasing the percentage discount on CH strongly reduces the probability of purchasing MM.

5 Strong Predictors

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8186     0.1130  -7.247 4.25e-13 ***
## PriceCH       0.4471     0.1402   3.188 0.001432 **
## PriceMM      -0.4526     0.1288  -3.513 0.000443 ***
## DiscMM        0.5448     0.1066   5.110 3.22e-07 ***
## LoyalCH      -2.0036     0.1451 -13.804  < 2e-16 ***
## PctDiscCH    -0.5652     0.1315  -4.298 1.72e-05 ***
## ---
## 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:  577.04  on 744  degrees of freedom
## AIC: 589.04
```

We are 95% confident that the variables used in this model will accurately determine what the customer purchases for 78-88% of orange juice purchases, as evident in the output from our confusion matrix.

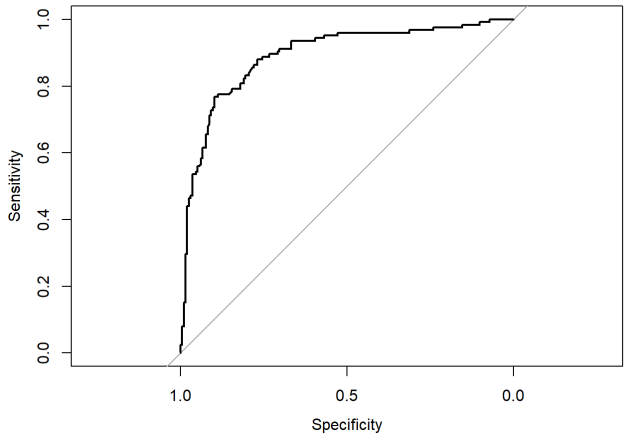
```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  X0  X1
##         X0 171  30
##         X1  24  95
##
##               Accuracy : 0.8312
##                 95% CI : (0.7856, 0.8706)
##    No Information Rate : 0.6094
##    P-Value [Acc > NIR] : <2e-16
##
```

Sales Manager:

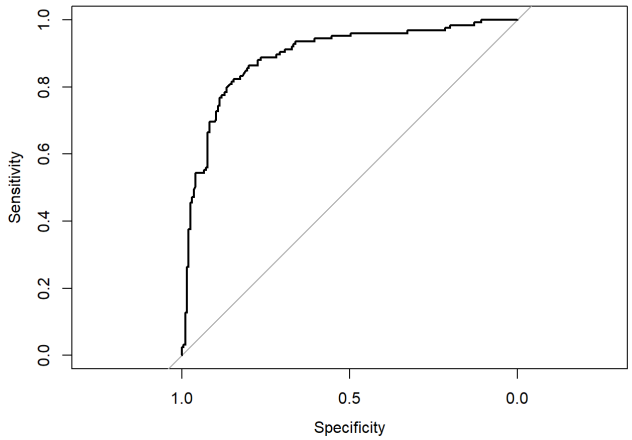
Our model answers the question: "What is the probability consumers will purchase Minute Maid?" with on average 84% accuracy. We are 95% confident that our model will accurately predict the Purchase on 80-88% of all orange juice purchases in the grocery store, as shown in the confusion matrix output below.

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  X0  X1
##         X0 173  29
##         X1  22  96
##
##               Accuracy : 0.8406
##                 95% CI : (0.7958, 0.879)
##    No Information Rate : 0.6094
##    P-Value [Acc > NIR] : <2e-16
##
```

1) **Logistic ROC Curve:** AUC: 0.8912



2) **SVM ROC Curve:** AUC: 0.8919



Recommendation

On the basis of the 5 variables used for the Regression model and SVM we would like to recommend following points to the managers:

1. Increase the price of CH if you want to sell more Minute Maid.
2. Reducing the price of MM will also increase the probability of purchasing MM.
3. Percentage of discount on CH is having a negative impact on the probability of purchasing MM. So, if there is an increase in the percentage of discount of CH then the price of MM should be reduced or the discount on MM should be increase to minimize the negative impact on the purchase of MM.
4. Customer loyalty for CH is the significant variable which negatively impacts the sale of MM. So, the managers should come with ways to increase the loyalty for MM.

Detailed analysis:

We used a logistic regression model with the five independent variables to determine the variables that influence sales of MM.

The model provided the significance of the each variable for the model:

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8186     0.1130  -7.247 4.25e-13 ***
## PriceCH       0.4471     0.1402   3.188 0.001432 **
```

```
## PriceMM      -0.4526     0.1288  -3.513 0.000443 ***
## DiscMM        0.5448     0.1066   5.110 3.22e-07 ***
## LoyalCH      -2.0036     0.1451 -13.804  < 2e-16 ***
## PctDiscCH    -0.5652     0.1315  -4.298 1.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

We are 95% confident that the variables used in this model will accurately determine what the customer purchases for 78-87% of orange juice purchases, as evident in the output from our confusion matrix.

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  X0  X1
##         X0 171  30
##         X1  24  95
##
##               Accuracy : 0.8312
##                 95% CI : (0.7856, 0.8706)
##    No Information Rate : 0.6094
##    P-Value [Acc > NIR] : <2e-16
##
```

We used a radial kernel to provide a predictive model to determine probability of buying MM. Since the sales manager is not interested in the particular variables that influence the purchase, we do not need to use a regression model for this case.

We are 95% confident that our model will accurately predict the Purchase on 79-88% of all orange juice purchases in the grocery store, as shown in the confusion matrix output below.

```
## Confusion Matrix and Statistics
##
```

```
##          Reference
## Prediction  X0  X1
##         X0 173  29
##         X1  22  96
##
##               Accuracy : 0.8406
##                 95% CI : (0.7958, 0.879)
##    No Information Rate : 0.6094
##    P-Value [Acc > NIR] : <2e-16
```

Parameter significance of Logistic model

1. Customer loyalty for CH is the significant variable which negatively impacts the purchase of MM.
2. Increasing the price of CH will increase the probability of purchasing MM.
3. Increasing the price of MM will reduce the probability of purchasing MM.
4. The discount on MM has positive effect on the probability of purchasing MM.
5. The percentage discount offered on CH has negative effect on the probability of purchasing MM.

APPENDIX

```
##Package loading
library(dplyr)
library(mlbench)
library(caret)
library(ROCR)
library(e1071)
library(dataPreparation)
library(ROCR)
library(ggplot2)
library(plotROC)
```

```
library(pROC)

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

summary(data)
str(data)

##No Contants
constant_cols <- whichAreConstant(data)
constant_cols

##No Double Columns
double_cols <- whichAreInDouble(data)
double_cols

##1 Bijection Column - ID 18 [Delete the column]
bijections_cols <- whichAreBijection(data)
bijections_cols

#Deleting column STORE
data <- subset( data, select = -STORE )

str(data)

#Correlation between dependent variables
correlationMatrix <- cor(data[,c(2,3,4,5,6,7,8,9,10,11,12,13,15,16,17)])

#Summarize the correlation matrix
```

```
print(correlationMatrix)

#Attributes that are highly corrected  (>0.7)
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.7,names = T)

#printing highly correlated attributes
print(highlyCorrelated)

#Removing higly correlated variabes
data <- subset( data, select = -c(2,6,11,12,13,15))

str(data)

#Checking for MM [1=M,0=CH]
data$Purchase <- ifelse(data$Purchase == "MM", 1, 0)
data$Purchase <- factor(data$Purchase, levels = c(0, 1))

str(data)

#Creating Train & Test Data Sets
set.seed(100)
trainDataIndex <- createDataPartition(data$Purchase, p=0.7, list = F)  # 70% training data
trainData <- data[trainDataIndex, ]
testData <- data[-trainDataIndex, ]

str(trainData)

scales <- build_scales(dataSet = trainData, cols = c("PriceCH", "PriceMM", "DiscMM",
"LoyalCH", "PctDiscCH", "ListPriceDiff" ), verbose = TRUE)
```

```
trainData <- fastScale(dataSet = trainData, scales = scales, verbose = TRUE)
testData <- fastScale(dataSet = testData, scales = scales, verbose = TRUE)

str(trainData)

#Cross Vaidation
fitControl <- trainControl(## 10-fold CV
  method = "repeatedcv",
  number = 2,
  ## repeated ten times
  repeats = 3,
  summaryFunction=twoClassSummary,
  classProbs = TRUE)

logitmod <- glm(Purchase ~ ., family = "binomial", data=trainData)
summary(logitmod)

#Removing Variabes with high P-Values (>0.05)
trainData <- subset( trainData, select = -c(2,6,7,9,11))
testData <- subset( testData, select = -c(2,6,7,9,11))
str(trainData)

levels(trainData$Purchase) <- make.names(levels(factor(trainData$Purchase)))
levels(testData$Purchase) <- make.names(levels(factor(testData$Purchase)))
str(trainData)
str(testData)
```

```
logitmod_1 <- glm(Purchase ~ ., family = "binomial", data=trainData)
summary(logitmod_1)

##### Logistic
logFit <- train(Purchase ~ .,  data=trainData, method="glm", family="binomial",trControl =
fitControl,metric = "ROC")
logFit
logPred <- predict(logFit, newdata = testData)

#Confusion Matrix on Test Data
confusionMatrix(data = logPred, testData$Purchase)

#Plotting ROC curve for logit model
roc(testData$Purchase, predict(logFit, newdata=testData, type="prob" )[,2],plot=T,
   auc=T)

grid_radial <- expand.grid(sigma = c(.01,.02),
                           C = c(.75,1,1.5))

###### SVM - Radial
svmFit1 <- train(Purchase ~ ., data = trainData,
                 method='svmRadial',
                 trControl = fitControl,
                 #preProc = c("center","scale"),
                 metric = "ROC",
                 verbose = FALSE,
                 probability = TRUE,
                 tuneGrid = grid_radial
```

```
)
svmFit1
svmPred1 <- predict(svmFit1, newdata = testData)

#Confusion Matrix on Test Data
confusionMatrix(data = svmPred1, testData$Purchase)

#Plotting ROC curve for SVM - Radial model
roc(testData$Purchase, predict(svmFit1, newdata=testData, type="prob" )[,2],plot = T, auc = T)

#Finding best value of Cost for linear kernel
grid_linear <- expand.grid(C = c(.75,1,1.5))

svmFit_linear <- train(Purchase ~ ., data = trainData,
                       method='svmLinear',
                       trControl = fitControl,
                       #preProc = c("center","scale"),
                       metric = "ROC",
                       verbose = FALSE,
                       probability = TRUE,
                       tuneGrid = grid_linear
)
svmFit_linear
svmPred_linear <- predict(svmFit_linear, newdata = testData)

#Confusion Matrix on Test Data
confusionMatrix(data = svmPred_linear, testData$Purchase)
```

```
#Plotting ROC curve for SVM model
roc(testData$Purchase, predict(svmFit_linear, newdata=testData, type="prob" )[,2],plot = T, auc = T)


#####SVM - Polynomial
svm_p <- tune.svm(Purchase ~ ., data = trainData,
                  degree = (3:6),
                  gamma = (0.1:0.4),
                  coef0 = 1,
                  kernel = "polynomial")
svm_p


svm_poly <- svm(Purchase ~ ., data=trainData, type='C-classification', kernel='polynomial', degree=3, gamma=0.1, coef0=1, scale=FALSE, probability = TRUE)
summary(svm_poly)


svmPred_poly <- predict(svm_poly, newdata = testData)


#Confusion Matrix on Test Data
confusionMatrix(data = svmPred_poly, testData$Purchase)
```