Sales Analysis of Minute Maid Orange Juice

MKTG 6620 -Machine Learning For Business Applications

Huzefa Saifee-u1274086

Date: 12/03/2019

Problem Definition:

Basically, both the Brand Manager and the Sales Manager, are interested in increasing the sales of Minute Maid Orange Juice. However, the Brand Manager wants to know which variables are influencing the customer's probability of buying Minute Maid while the Sales manager wants to know the probability of a customer purchasing Minute Maid.
For our model analysis, we will focus on two areas. Firstly, we will determine which variables influence the purchase of Minute Maid Orange Juice and how much they impact Minute Maid's purchase. Then we will focus on building a model that predicts whether a customer will purchase the Minute Maid Orange Juice or not. Also, we will consider the accuracy of our model to achieve optimum results.

Objectives:

1. Increase Minute Maid Orange Juice's Sales
2. Determine influential variables responsible for Minute Maid's Sale
3. Identify best suitable model to predict Minute Maid's sales
4. Provide answers to the queries of Brand & Sales Manager
5. Generate recommendations & provide support

```
# PACKAGES UTILIZED
library("dataPreparation")
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
## Loading required package: stringr
```

```
## Loading required package: Matrix
```

```
## Loading required package: progress
```

```
## dataPreparation 0.4.2
```

```
## Type dataPrepNews() to see new features/changes/bug fixes.

library("mlbench")
library("e1071")
library("caret")

## Loading required package: lattice

## Loading required package: ggplot2

library("ROCR")

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

library("kernlab")

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':
##
##     alpha

library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library("corrplot")

## corrplot 0.84 loaded

library("plotROC")
library("ggplot2")
```

Methods Used:

1. Exploratory Data Analysis
2. Data Preparation for modeling
3. Split data in Train and Test Data
4. Determining important variables based on P-Value
5. Apply Logistic & SVM Models

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

Exploratory Data Analysis:

1. Check for Outliers
2. Check for NULL / NA Values
3. Check for Mis-Classified variables

Why Exploratory Data Analysis?

It is common that in real-world data, there might be some errors because of which outliers are generated. Sometimes, because of noise, the data gets corrupted and we get NA values. Also, there are cases when the data has inconsistency in variable names because of which they are misinterpreted or "misclassified." So performing exploratory data analysis, in other words preparing the data is important.

```
# CHECK VARIABLES TYPE
lapply(OJ, class)
```

```
## $Purchase
## [1] "factor"
##
## $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] "factor"
##
## $PctDiscMM
## [1] "numeric"
##
## $PctDiscCH
## [1] "numeric"
##
## $ListPriceDiff
## [1] "numeric"
##
## $STORE
## [1] "integer"
```

```
## DATA CLEANING ##
# RecodING MM/CH as Y/N IN PURCHASE VARIABLE
# ALSO, FACTORIZING CATEGORICAL VARIABLES
OJ <- OJ %>%
  mutate( Purchase = recode_factor(Purchase, "MM" = "Y", "CH" = "N"),
          StoreID = factor(StoreID),
          SpecialCH = factor(SpecialCH),
          SpecialMM = factor(SpecialMM),
          Purchase = factor(Purchase))

# CHECK VARIABLES TYPE
lapply(OJ, class)
```

```
## $Purchase
## [1] "factor"
##
## $WeekofPurchase
## [1] "integer"
##
## $StoreID
## [1] "factor"
```

```
##
## $PriceCH
## [1] "numeric"
##
## $PriceMM
## [1] "numeric"
##
## $DiscCH
## [1] "numeric"
##
## $DiscMM
## [1] "numeric"
##
## $SpecialCH
## [1] "factor"
##
## $SpecialMM
## [1] "factor"
##
## $LoyalCH
## [1] "numeric"
##
## $SalePriceMM
## [1] "numeric"
##
## $SalePriceCH
## [1] "numeric"
##
## $PriceDiff
## [1] "numeric"
##
## $Store7
## [1] "factor"
##
## $PctDiscMM
## [1] "numeric"
##
## $PctDiscCH
## [1] "numeric"
##
## $ListPriceDiff
## [1] "numeric"
##
## $STORE
## [1] "integer"
```

Data Preparation:

Remove All:

1. Constant Variables
2. Double Variables
3. Bijection Variables:
   a. STORE of StoreID
4. Included Variables:
   a. Store7 in StoreID
   b. DiscCH in PctDiscCH
   c. DiscMM in PctDiscMM

```
# IDENTIFY AND LIST VARIABLES THAT ARE CONSTANTS
constant_cols <- whichAreConstant(OJ)

## [1] "whichAreConstant: it took me 0.06s to identify 0 constant column(s)"

# IDENTIFY AND LIST VARIABLES THAT ARE DOUBLES
double_cols <- whichAreInDouble(OJ)

## [1] "whichAreInDouble: it took me 0.03s to identify 0 column(s) to drop."

# IDENTIFY AND LIST VARIABLES THAT ARE EXACT BIJECTIONS
bijections_cols <- whichAreBijection(OJ)

## [1] "whichAreBijection: STORE is a bijection of StoreID. I put it in drop
list."
## [1] "whichAreBijection: it took me 0.77s to identify 1 column(s) to drop."

# REMOVE ALL BIJECTIONS
OJ <- OJ[,-18]

# IDENTIFY AND LIST VARIABLES THAT ARE INCLUDED IN OTHER VARIABLES
included_cols <- whichAreIncluded(OJ)

## [1] "whichAreIncluded: Store7 is included in column StoreID."
## [1] "whichAreIncluded: DiscCH is included in column PctDiscCH."
## [1] "whichAreIncluded: DiscMM is included in column PctDiscMM."

# REMOVE ALL INCLUDED VARIABLES
OJ <- OJ[,-14]
OJ <- OJ[,-7]
OJ <- OJ[,-6]
```

Apply Correlation on the Data, and remove following highly correlated variables:

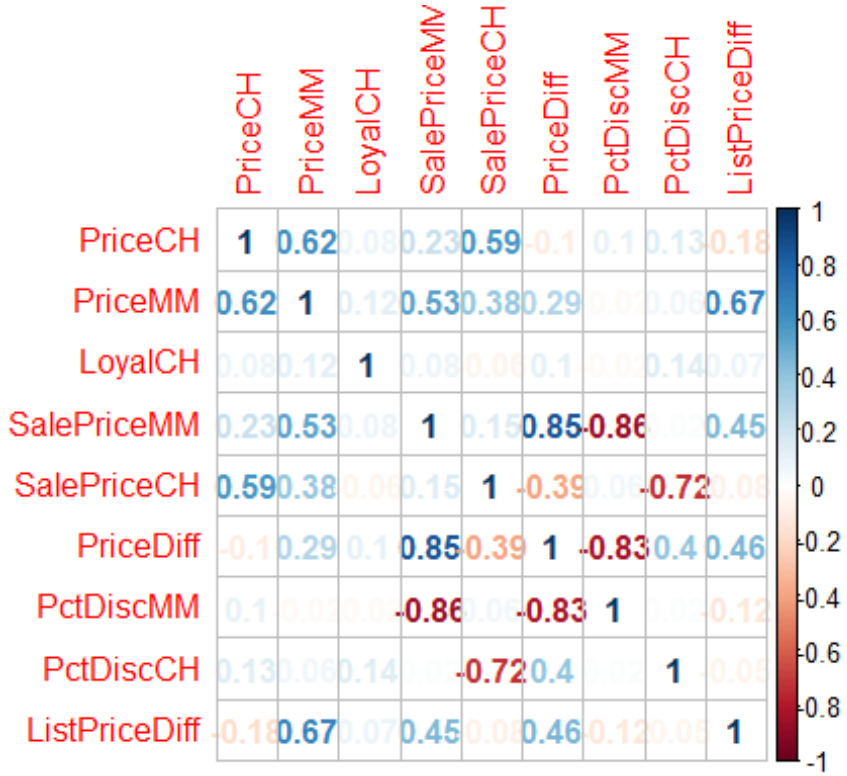
1. PriceDiff
2. SalePriceMM
3. SalePriceCH
4. ListPriceDiff

```
# CHECK CORRELATION AMONGST NUMERIC VARIABLES
OJ_numeric <- OJ[, c(4,5,8,9,10,11,12,13,14)]
res <- cor(OJ_numeric)
round(res, 2)
```

```
##                PriceCH PriceMM LoyalCH SalePriceMM SalePriceCH PriceDiff
## PriceCH           1.00    0.62    0.08        0.23        0.59     -0.10
## PriceMM           0.62    1.00    0.12        0.53        0.38      0.29
## LoyalCH           0.08    0.12    1.00        0.08       -0.06      0.10
## SalePriceMM       0.23    0.53    0.08        1.00        0.15      0.85
## SalePriceCH       0.59    0.38   -0.06        0.15        1.00     -0.39
## PriceDiff        -0.10    0.29    0.10        0.85       -0.39      1.00
## PctDiscMM         0.10   -0.02   -0.02       -0.86        0.06     -0.83
## PctDiscCH         0.13    0.06    0.14        0.02       -0.72      0.40
## ListPriceDiff    -0.18    0.67    0.07        0.45       -0.08      0.46
##                PctDiscMM PctDiscCH ListPriceDiff
## PriceCH             0.10      0.13         -0.18
## PriceMM            -0.02      0.06          0.67
## LoyalCH            -0.02      0.14          0.07
## SalePriceMM        -0.86      0.02          0.45
## SalePriceCH         0.06     -0.72         -0.08
## PriceDiff          -0.83      0.40          0.46
## PctDiscMM           1.00      0.02         -0.12
## PctDiscCH           0.02      1.00         -0.05
## ListPriceDiff      -0.12     -0.05          1.00
```

Corrplot() on the Data shows highly correlated variables:

```
corrplot(res, method="number")
```



```
# REMOVE ALL CORRELATED VARIABLES
OJ <- OJ[,-14]
OJ <- OJ[,-11]
OJ <- OJ[,-10]
OJ <- OJ[,-9]

# REMOVE MIS-CLASSIFIED VARIABLES
OJ <- OJ[,-2]
########################
```

Split in Train & Test Data:

In order to reduce overfitting in the data, we randomly split a specific percentage of data into train data and then using the test data for cross-validation

```
## TRAIN & TEST DATA ##
# SPECIFY PROPORTION OF DATA TO TEST (I SET AT 80% TRAIN) AND SEED FOR REPLIC
ATION
split = .8
set.seed(99894)

## DATA IS SPLIT INTO TRAIN / TEST(HOLDOUT) ##
train_index <- sample(1:nrow(OJ), split * nrow(OJ)) ## 80% of data randomly s
elected for train
test_index <- setdiff(1:nrow(OJ), train_index) ## the remaining 20% of the da
ta is used for holdout testing

X_train_unscaled <- OJ[train_index,-1]
y_train <- OJ[train_index, 1]

X_test_unscaled <- OJ[test_index, -1]
y_test <- OJ[test_index, 1]

# DATA IS STANDARDIZED AND ENCODED
# Standardizing continuous variables..
scales <- build_scales(dataSet = X_train_unscaled, cols = "auto", verbose = F
ALSE)

X_train <- fastScale(dataSet = X_train_unscaled, scales = scales, verbose = F
ALSE)
X_test <- fastScale(dataSet = X_test_unscaled, scales = scales, verbose = FAL
SE)

# EncodING categorical variables..
encoding <- build_encoding(dataSet = X_train, cols = "auto", verbose = FALSE)
X_train <- one_hot_encoder(dataSet = X_train, encoding = encoding, drop = TRU
E, verbose = FALSE)
X_test <- one_hot_encoder(dataSet = X_test, encoding = encoding, drop = TRUE,
verbose = FALSE)
```

```
# Create one data frame using both Outcome and Predictor Variables
train_Data <- cbind(y_train,X_train)
test_Data <- cbind(y_test,X_test)
#######################
```

Influential Variables:

```
## DETERMINING INFLUENTIAL PREDICTOR VARIABLES ##
scale <- build_scales(dataSet = OJ, verbose = TRUE)

## [1] "build_scales: I will compute scale on  5 numeric columns."
## [1] "build_scales: it took me: 0.01s to compute scale for 5 numeric column
s."

OJ_2 <- fastScale(dataSet = OJ, scales = scale, verbose = TRUE)

## [1] "fastScale: I will scale 5 numeric columns."
## [1] "fastScale: it took me: 0s to scale 5 numeric columns."

predictionModel <- glm(Purchase ~ ., data = OJ_2,family = binomial(link = 'lo
git'))
summary(predictionModel)$coefficients

##                Estimate Std. Error     z value     Pr(>|z|)
## (Intercept)  0.67513331  0.2353324  2.86884988 4.119673e-03
## StoreID2    -0.15155900  0.2760642 -0.54899904 5.830061e-01
## StoreID3    -0.02238713  0.3350029 -0.06682667 9.467197e-01
## StoreID4     0.31170847  0.3770611  0.82667895 4.084191e-01
## StoreID7     0.59361756  0.2805804  2.11567690 3.437230e-02
## PriceCH     -0.39926132  0.1398212 -2.85551348 4.296730e-03
## PriceMM      0.45064454  0.1130219  3.98723375 6.684816e-05
## SpecialCH1  -0.14898892  0.3327831 -0.44770580 6.543655e-01
## SpecialMM1  -0.28402007  0.2750592 -1.03257779 3.018015e-01
## LoyalCH      1.90437239  0.1249061 15.24643646 1.738667e-52
## PctDiscMM   -0.46736828  0.1116401 -4.18638292 2.834350e-05
## PctDiscCH    0.44340591  0.1243206  3.56663136 3.615996e-04
```

Determining Influential variables using P-Value helps us understand that variables such as SpecialCH, SpecialMM are not influencing the purchase of MM

Using AIC value to corroborate our intuition of selecting only a few variables:

```
# AIC MODEL ON INFLUENTIAL VARIABLES
Model1 <- glm(Purchase ~ ., data = OJ_2, family = binomial(link = "logit"))
Model2 <- glm(Purchase ~ StoreID + PriceCH + PriceMM + LoyalCH + PctDiscMM +
PctDiscCH, data = OJ_2, family = binomial(link = "logit"))
print(paste("Model 1:", AIC(Model1), "Model 2:", AIC(Model2)))

## [1] "Model 1: 848.336185372214 Model 2: 845.427598958795"

#######################
```

If we use all the variables, as we did in Model1, the AIC value will be more and if consider only influential variables, as we did in Model2, AIC value will be less, which implies our Model2 is performing better

Logistic Model:

Applying Logistic Model on train data with the selected influential variables After applying glm(), we use the predict to get the result in the form of probability Then Converting Probabilities to "Y" and "N" format and factorizing the variable to match it with the reference variable's (Purchase) data format

```
#################
## LOGIT MODEL ##
#################
predictionModel <- glm(Purchase ~ PriceCH + PriceMM + LoyalCH + PctDiscMM + P
ctDiscCH + StoreID.1 + StoreID.2 + StoreID.3 + StoreID.4, data = train_Data,
family = binomial(link = 'logit'))

# Predict
X_test$prediction <- predict(predictionModel, newdata = X_test, type ="respon
se")

# Converting Probilities into Categorical Predictions
X_test$binary_prediction<-ifelse(X_test$prediction < 0.55,"Y","N")
X_test$binary_prediction<-as.factor(X_test$binary_prediction)

# CONFUSION MATRIX
confusionMatrix(data = X_test$binary_prediction, as.factor(y_test$Purchase))

## Warning in confusionMatrix.default(data = X_test$binary_prediction,
## as.factor(y_test$Purchase)): Levels are not in the same order for referenc
e and
## data. Refactoring data to match.

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   Y   N
##          Y  75  12
##          N  18 109
##
##                Accuracy : 0.8598
##                  95% CI : (0.806, 0.9034)
##     No Information Rate : 0.5654
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7126
##
##  Mcnemar's Test P-Value : 0.3613
##
```

```
##             Sensitivity : 0.8065
##             Specificity : 0.9008
##          Pos Pred Value : 0.8621
##          Neg Pred Value : 0.8583
##              Prevalence : 0.4346
##          Detection Rate : 0.3505
##    Detection Prevalence : 0.4065
##       Balanced Accuracy : 0.8536
##
##        'Positive' Class : Y
##

###################################
```

The Above Confusion Matrix displays the accuracy and the confidence interval of the Logit Model which are required for the comparison with other models and to answer the questions asked by the Brand Manager and Sales Manager

Linear SVM Model:

Applying Linear SVM Model to the train data with different values of C to get the optimum Accuracy Predicting the model with the optimum C value and generating the Confusion Matrix with the predicted values and the test data (Purchase)

```
######################
## LINEAR SVM MODEL ##
######################
cctrl <- trainControl(method = "cv", number = 3, returnResamp = "all", classP
robs = TRUE)

grid2 <- data.frame(C = c(0.1,1,20,50,100))

# FIND OPTIMAL TUNING PARAMETER (C)
svmFit2 <- train(Purchase ~ ., data = train_Data, method='svmLinear', trContr
ol = cctrl, tuneGrid = grid2, preProc = c("center", "scale"))

# Predict
svmPred2 <- predict(svmFit2, newdata = X_test, probability = TRUE)

# CONFUSION MATRIX
confusionMatrix(data = svmPred2, as.factor(y_test$Purchase))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   Y   N
##          Y  71  12
##          N  22 109
##
##                Accuracy : 0.8411
```

```
##                  95% CI : (0.7851, 0.8874)
##     No Information Rate : 0.5654
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6726
##
##  Mcnemar's Test P-Value : 0.1227
##
##             Sensitivity : 0.7634
##             Specificity : 0.9008
##          Pos Pred Value : 0.8554
##          Neg Pred Value : 0.8321
##              Prevalence : 0.4346
##          Detection Rate : 0.3318
##    Detection Prevalence : 0.3879
##       Balanced Accuracy : 0.8321
##
##        'Positive' Class : Y
##

##############################################
```

The Above Confusion Matrix displays the accuracy and the confidence interval of the Linear SVM Model which are required for the comparison with other models and to answer the questions asked by the Brand Manager and Sales Manager

Radial SVM Model:

Applying Radial SVM Model to the train data with different values of C and Sigma to get the optimum Accuracy Predicting the model with the optimum C & Sigma value pair and generating the Confusion Matrix with the predicted values and the test data (Purchase)

```
######################
## RADIAL SVM MODEL ##
######################
fitControl <- trainControl(## 4-fold CV
  method = "repeatedcv",
  number = 4,
  ## repeated two times
  repeats = 2,
  summaryFunction=twoClassSummary,
  classProbs = TRUE)

grid <- expand.grid(sigma = c(.01, .02),
                    C = c(.69, .75, 0.70, 0.72, 1))

# FIND OPTIMAL TUNING PARAMETERS (C and SIGMA)
svmFit1 <- train(Purchase ~ ., data = train_Data,
                 method='svmRadial',
                 trControl = fitControl,
```

```
                metric = "ROC",
                verbose = FALSE,
                probability = TRUE,
                tuneGrid = grid
)

# Predict
svmPred <- predict(svmFit1, newdata = X_test, probability = TRUE)

# CONFUSION MATRIX
confusionMatrix(data = svmPred, as.factor(y_test$Purchase))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   Y   N
##          Y  70  11
##          N  23 110
##
##                Accuracy : 0.8411
##                  95% CI : (0.7851, 0.8874)
##     No Information Rate : 0.5654
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6718
##
##  Mcnemar's Test P-Value : 0.05923
##
##             Sensitivity : 0.7527
##             Specificity : 0.9091
##          Pos Pred Value : 0.8642
##          Neg Pred Value : 0.8271
##              Prevalence : 0.4346
##          Detection Rate : 0.3271
##    Detection Prevalence : 0.3785
##       Balanced Accuracy : 0.8309
##
##        'Positive' Class : Y
##

###############################################
```

The Above Confusion Matrix displays the accuracy and the confidence interval of the Radial SVM Model which are required for the comparison with other models and to answer the questions asked by the Brand Manager and Sales Manager

Brand Manager's Questions:

1. What predictor variables influence the purchase of MM? The Predictor variables influencing the purchase of MM are:
    a. PriceCH
    b. PriceMM
    c. LoyalCH
    d. PctDiscMM
    e. PctDiscCH
    f. StoreID
2. Are all the variables in the dataset effective, or are some more effective than others?
    Not all variables have the same influence on the Purchase variable. The more significant variables are:
    a. LoyalCH
    b. PctDiscMM
    c. PctDiscCH
    d. PriceMM
3. How confident are you in your recommendations?
    While determining the influential predictor variables, we saw that all the four variables are having very low P-Value inferring that all four variables are Statistically Significant and are effective variables.

Sales Manager's Questions:

1. Can you provide a predictive model that can tell the probability of customers buying MM?
    Logistic Regression will be the best suitable model to predict a customer buying an MM.
2. How good is the model in its predictions?
    The model is 85.98% Accurate.
3. How confident are you in your recommendations?
    The model has a Confidence Interval of 95%. Also, it is visible in the Confusion Matrix of Logit Model as:
    `95% CI : (0.806, 0.9034)`

Recommendations:

1. Give Discounts on MM to attract more customers, as Discounts do play important role
2. Just like the Loyalty towards CH is there, Promoting Customer's Loyalty towards MM might help in increase in Sales
3. Different stores have different locations, which indirectly play a huge role in product sales; keeping that in mind for creating marketing strategy will help in the promotion of an MM at different StoreIDs, individually, as well as it will help in planning a better budget distribution.