|  |
| --- |
| Statistical Data Mining |
| Final Project:  Fabric Softener |
|  |

|  |
| --- |
| By: Gaurav Jetley |

**Introduction**

The study was conducted to build the most appropriate predictive models for predicting sales/purchases of SKUs at household and weekly levels and also to get some interesting insights from the dataset itself. These insights are listed in the “Dataset Analysis” and “Variables and Importance Analysis” sections of this report.

The first analysis was to predict the sales/purchases of different SKU’s at a weekly level. The goal was to predict the amount of sales of different SKU’s at a weekly level. This is important because companies need to be able to predict the sales of their products in the coming weeks and months and also to be able to add promotions and price cuts at the most optimum times to increase their sales. The second analysis was to predict the sales/purchases of different SKU’s at a household level. The goal was to predict the amount of sales of different SKU’s at a household level in 1992. This is important to predict customer churn and taking appropriate steps to gain more customer base.

The goal was to predict the amount of sales of different SKU’s at a household level in 1992. This is important to predict customer churn and taking appropriate steps to gain more customer base.

Tree based models in both the predictive analysis turned out to be the better models. The predictive models that were built for the prediction of SKU sales/Purchases at both weekly and household level showed the same predictor variables as the top variables in the analysis. They were SKU, APP, ARP, APC and brand. This shows that consumers at highly influenced by price of the product, the promotional discount and the brand of the product as well when choosing to buy a fabric softener (or maybe any other product).

Another interesting find was that the household predictive model had higher accuracy rates than the weekly model. This may be due to the fact that households have a greater chance of repeating their purchase history from the previous year in the next year as well. This is due to their loyalty towards a particular brand or a particular brand in a particular season. Further studies should be conducted to research these findings, especially the loyalty factor as it may be very useful in predicting customer churn.

**Data Sources**

**Fabric Softener Dataset**

The data come from an IRI panel in Philadelphia and cover the period from January 1990 to June 1992. The only criterion for inclusion in our data set is that the household must have made at least one purchase in 1991. This gives us 594 qualifying households that made a total of 9781 purchases over the 2.5 year period (Fader, 1996).

The files specific to the data were split into 4 files. The files were as follows

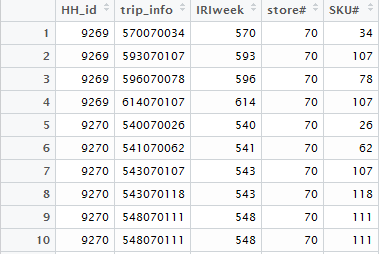
1. D1PUR: The data set contains household purchase history data. It contained two fields: HH\_id and trip\_info
2. MERCH: Contains information regarding store environment. Has five fields: SKU#, store# IRIweek, price\_paid and merchandising
3. ARSP: Contains the average regular selling price of each SKU in each store. Contains three fields: SKU#, store# and ARSP
4. BRSINFO: Contains the attribute information for each SKU. Contains eleven fields: SKU#, description\_of\_SKU, brand, form, formula1, formula2, size, brand#, form#, formula2# and size#
5. IRIWEEK: Time is week of purchase recorded as IRI week; this is a measure used by IRI where week 1 corresponds to the week ending 09/09/79. This spreadsheet provides a mapping of IRI weeks to dates.

**Data Preparation**

The data was prepared according to the “read\_me.txt” file provided in the data set folder. Detailed description of the processes is as follows:

1. **D1PUR:**

HH\_id variable was kept as is. The format of the trip\_info variable is AAABBBCCC where AAA = IRI week, BBB = store# and CCC = SKU# purchased. It was split into its 3 subparts to get the prepared dataset.



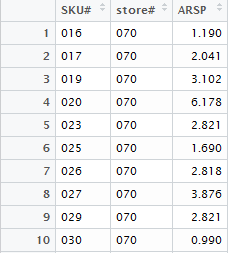
1. **MERCH:**

The format of the merchandising variable is AAABCD where AAA = depromoted price, B = ignore, C = display, D = feature. It was split into its 4 subparts. Dummies were created for display and feature where D\_DISP = 1 if C >= 1; 0 otherwise and D\_FEAT = 1 if D >= 1; 0 otherwise. Depromoted price was renamed as regular\_price and was split into regular\_price and price\_cut where regular\_price = AAA and price\_cut = AAA - price\_paid (if the result is < 0, price\_cut = 0). All the price variables were put into a #.## format.



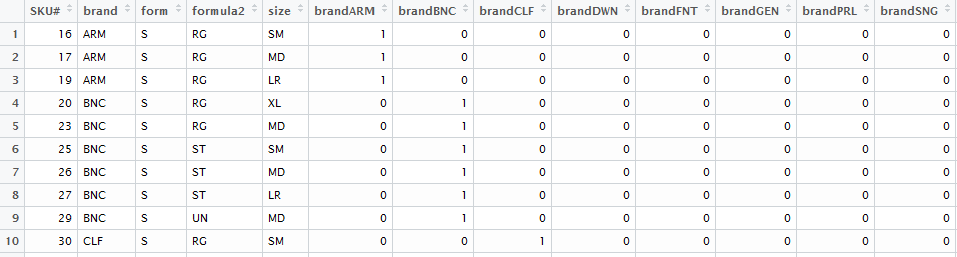
1. **ARSP:**

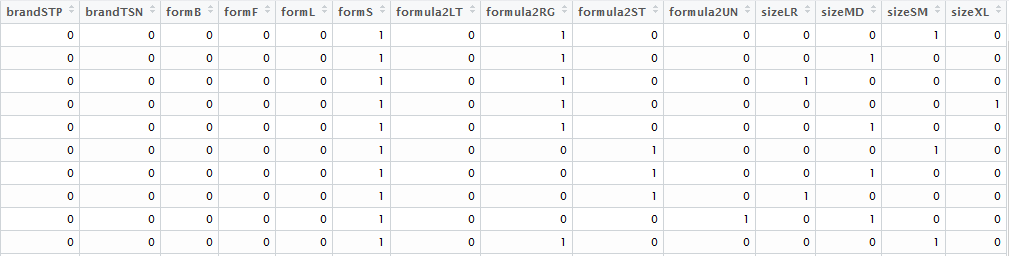
The formats of SKU# and Store# variables were changes to ### format for consistency with other data sets



1. **BRSINFO:**

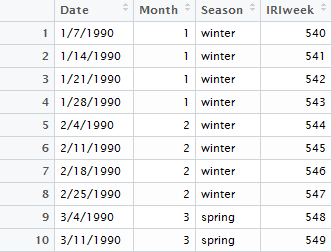
According to the instructions, the formula1 variable was dropped. Dummy variables were created for Brand, Size, Form and Formula2 variables





1. **IRIWEEK:**

The excel file had data regarding weeks and dates. Two more variables were created in this dataset for Seasons and Months the particular Week fell into.



**Data Cleaning**

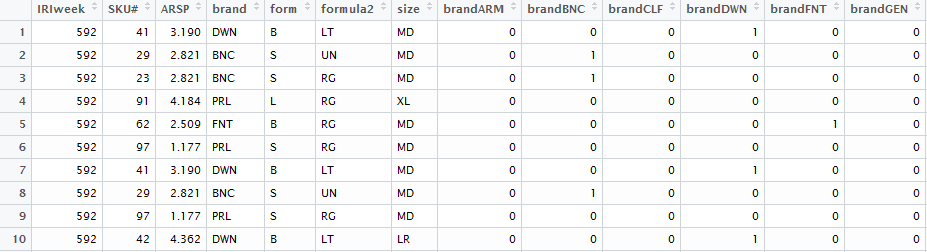
Three datasets were created as the final datasets to be used for the analysis. Each dataset was made for a particular type of analysis.

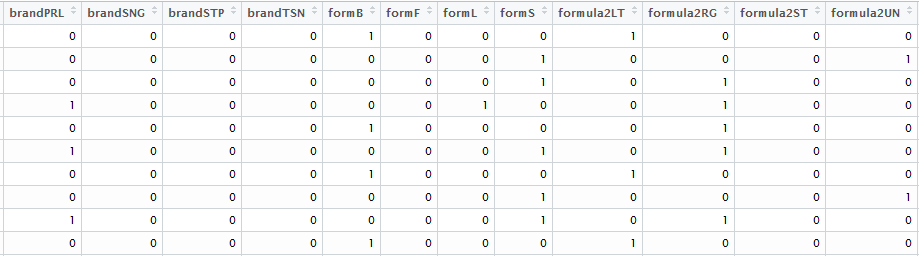
1. **Final Data Set**

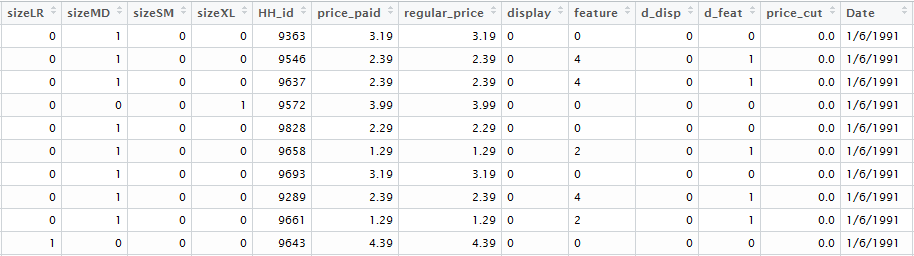
This was to be used for initial exploratory analysis, visualizations and SKU attribute importance analysis.

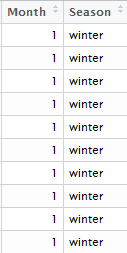
* 1. **Cleaning process:**

First, the ARSP and BRSIFO data sets were combined according to the SKU numbers. Second, D1PUR and MERCH datasets were merged together by the IRIweek, Store# and SKU# variables. Third, the two new datasets were merged together by the SKU# variable and all redundant variables were removed. Finally, the resulting dataset was merged with the IRIWEEK dataset by the IRIweek variable to get the final dataset. The dataset was named “finaldatav3” with 6,554 observations.









* 1. **Train dataset:**

The dataset used for training the predictive models was the data from the year 1991 with 4,117 observations.

* 1. **Test dataset:**

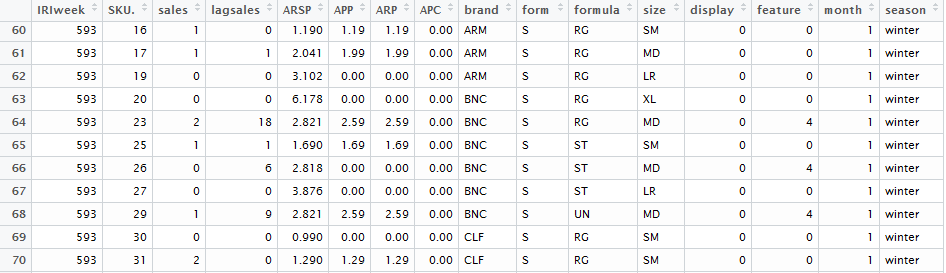
The dataset used for testing the fitted predictive models for accuracy and predictive power is the data from the year 1992 with 2,137 observations.

1. **Weekly Data Set (Temporal)**

The weekly dataset was made to be used in predictive models that explain the total purchases per SKU at a weekly level. The Final Data Set was modified to obtain the weekly dataset.

* 1. **Cleaning:**

The dataset provides the total purchases per SKU at a weekly level. For this, first, a list of weeks with each SKU was made. The total number of weeks were 77 and the total number of individual SKUs were 59 so the final dataset had 77 x 59 observations. The list was populated with more variables necessary for the analysis. I.e. Average price paid is the average price paid for a particular SKU in particular week. The same goes for all other pricing variables. Sales variable was created to show the number of sales of a particular SKU in a particular week. Another variable was created for Lag Sales. The dataset was named “weeklydatav2” with total observations of 4,602.



* 1. **Test:**

The test or the holdout dataset was created by taking observations from 1992 with 1,534 observations.

* 1. **Train:**

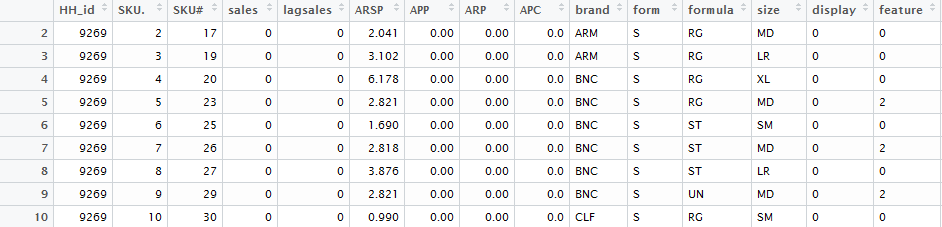
The training dataset was created from observations from 1992 with 3,068 observations.

1. **Household Data Set**
   1. **Cleaning**

The cleaning process followed the exact same flow as the weekly dataset with exception of the following: First, the dataset was modeled from the training and testing datasets created from the final dataset as it would be impossible to divide the dataset into training and test after creation. This is because the dataset doesn’t contain the IRIweek variable. The second difference was that the dataset didn’t contain the variables for season, date, month and IRIweek and included the variable HH\_id

* 1. **Test**

The test set contained 33264 observations and 15 variables



* 1. **Train**

The training set contained 35046 observations and 15 variables



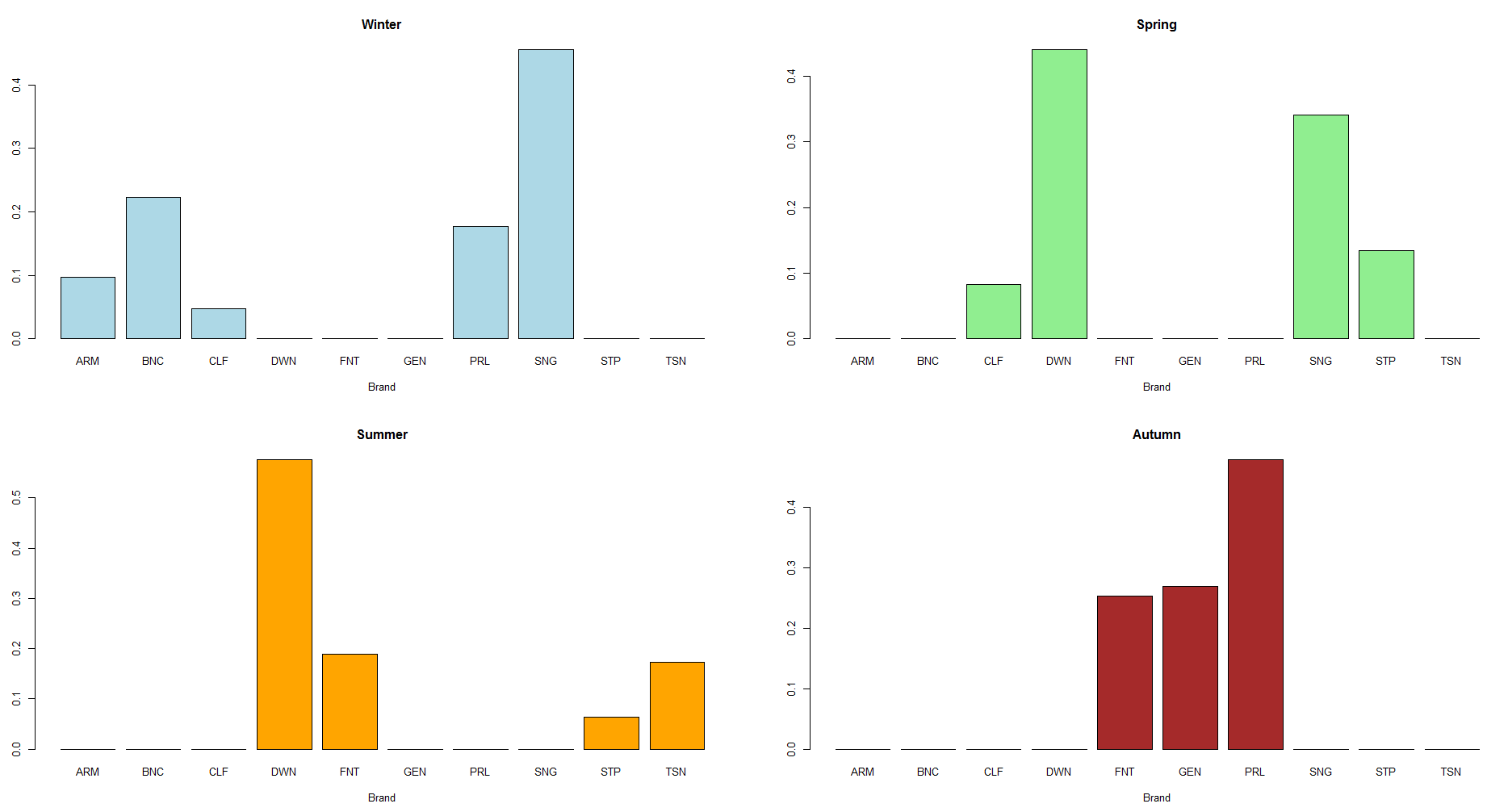
**Dataset Analysis**

**Variations in SKU choices**

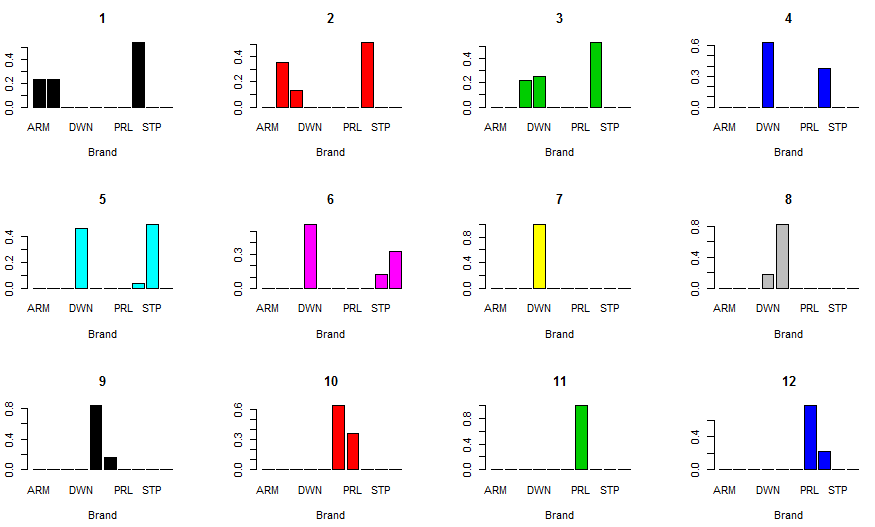
The variation in SKU choices can happen at a both at a weekly level and at a household level. This section highlights the main interesting findings that help us understand the market dynamics, consumer choices and the effect of pricing, promotions, product attributes features in fabric softeners.

1. **Variations at a Weekly Level**

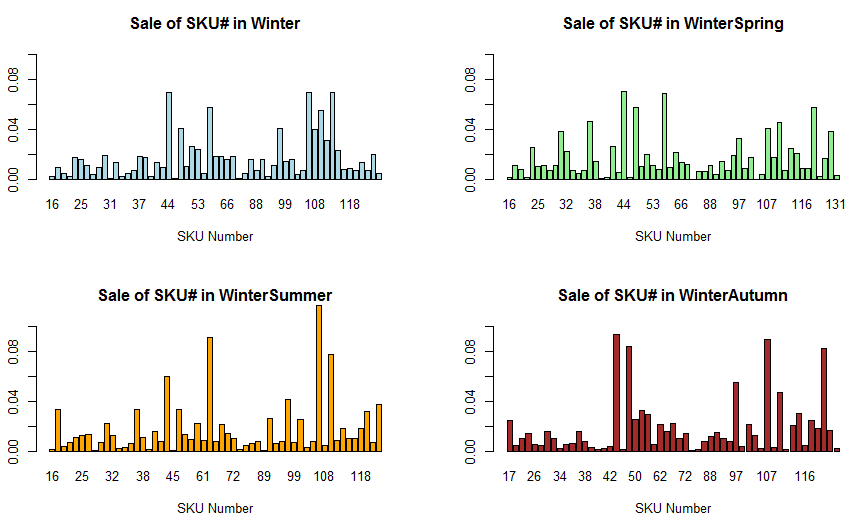
There is clear preference of different brands of fabric softeners in different seasons. For example the sale of brands FNT and GEN are mostly happen in autumn. The Brands DWN and STP are mostly sold in spring and summer and the Brands ARM and BNC are only sold in winter months. This may be because of preference of a brands’ attribute (i.e. size, form, etc.) in a particular time of the year or maybe due to promotions and price cuts in different seasons.



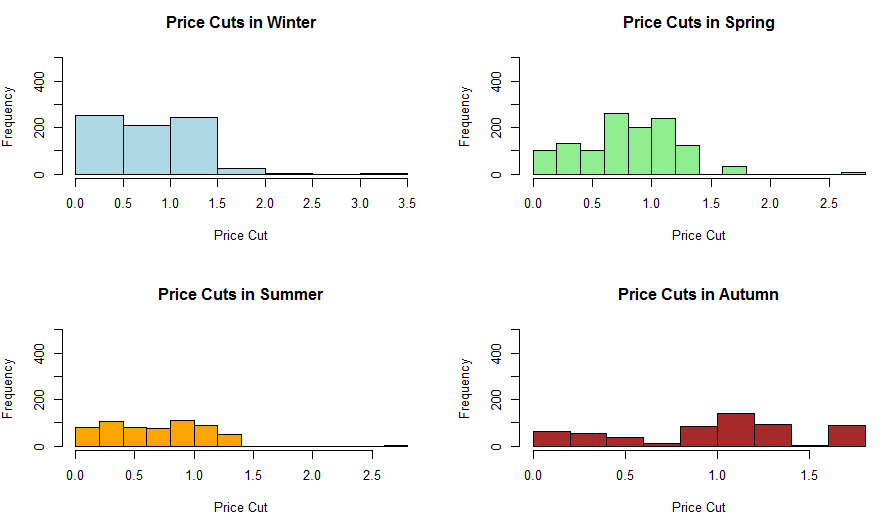
This finding is further emphasized when we look at the same distribution at a monthly level.



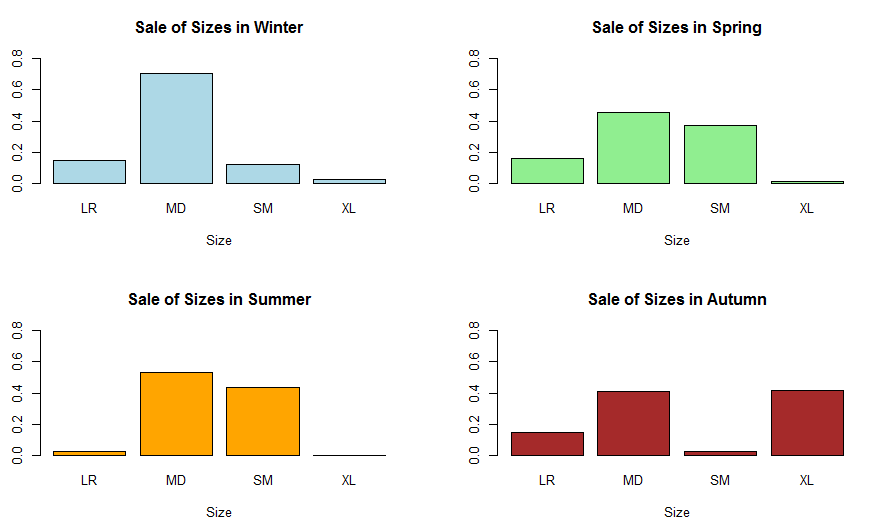
Sale of different SKUs in different seasons also show that the preference of certain SKUs are different according to different seasons.



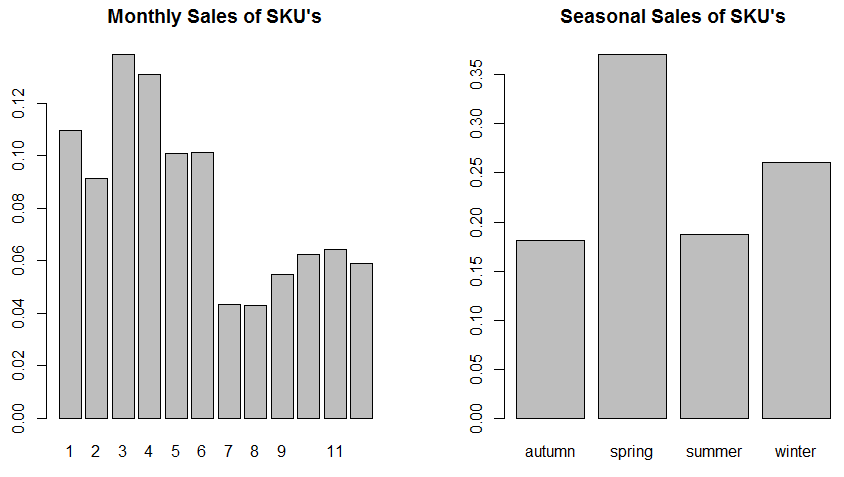
Price cuts show a temporal pattern as most of the price cuts and promotions are in winter or spring and summer sees little price cuts. Autumn shows heavier (larger $ value) price cuts than other seasons. These heavier price cuts may be for the extra-large sized products which are also much more expensive than other sized products.



The sale of different sizes of fabric softeners also show a seasonal patterns. The sale of Extra Large fabric softeners are the highest in autumn and negligible in other months. This concurs with the above finding and may be due to promotions and heavier price cuts on XL sized fabric softeners in autumn as opposed to other months. Medium is the most widely sold size no matter the season and summer and spring see high sales of small fabric softeners. This could be due to the fact that summer has little to no promotions and the consumer would rather buy small quantities of it until promotions and price cuts arrive in autumn.

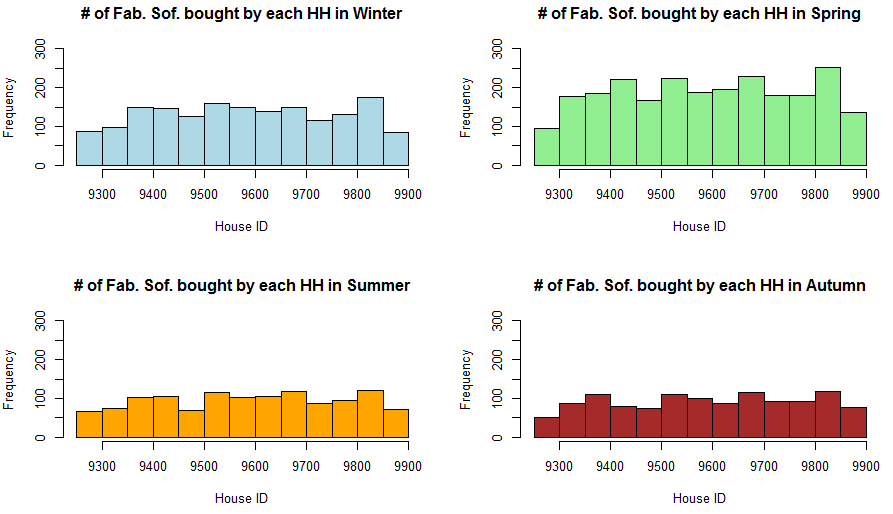


The total monthly and seasonal sales of fabric softeners also show a high variation from season to season and in different months. The highest sales of fabric softeners are in the first half of the year and in the spring. This may again due the effect of promotions and price cuts on total fabric softener sales in different seasons and months.

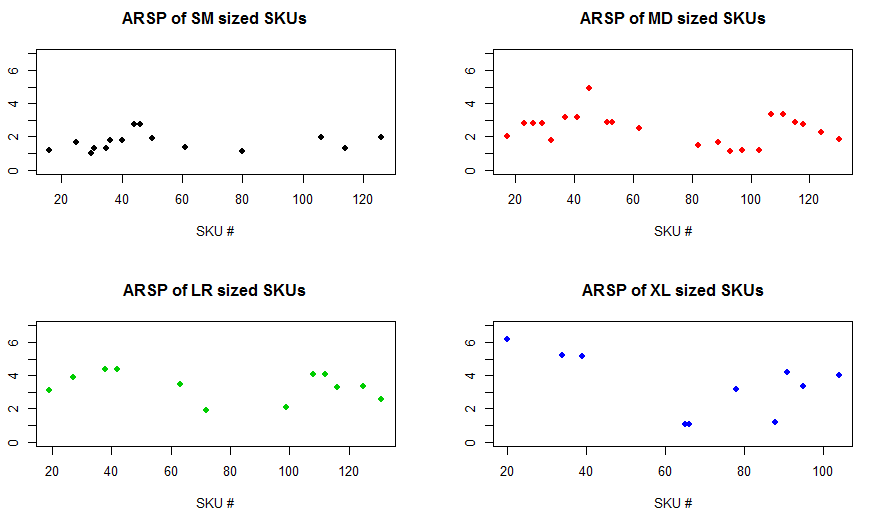


1. **Variations at Household Level**

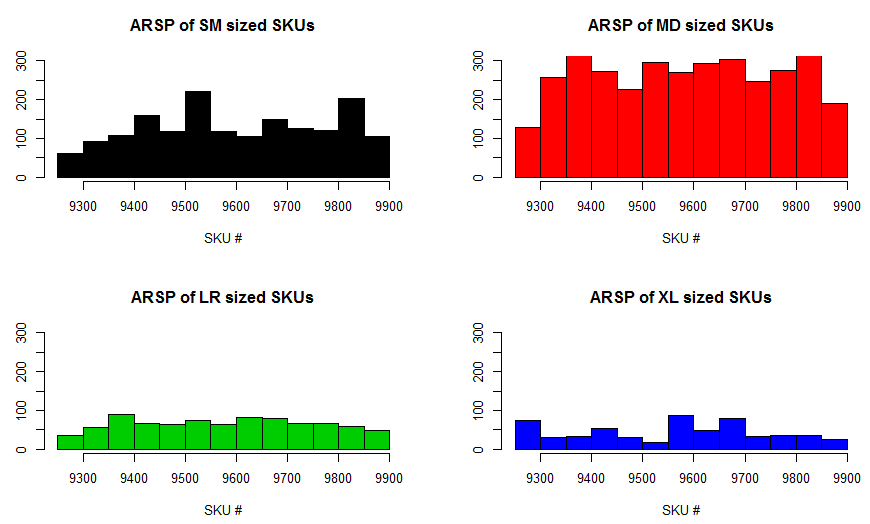
The sale of fabric softeners by individual households also change with seasons. The most sales happen in spring and winter. These are the two months which see the highest amount of price cuts and promotions as well so it makes sense that the sales will also be higher.



The average retail selling prices of different sizes of fabric softeners also reveal interesting information. They show that the XL sized products are the most high priced followed by large and medium and makes sense to by XL in the when there are promotions available and small when no promotions are available.



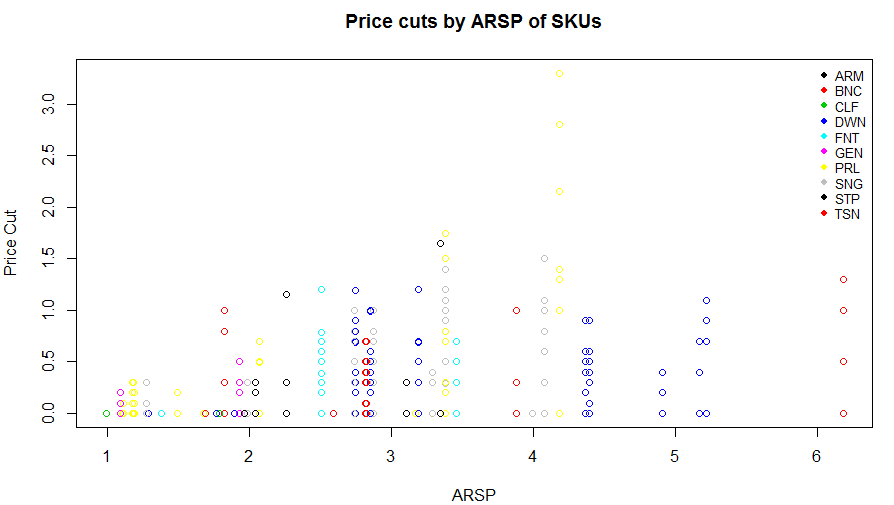
The sale of different sized fabric softeners by different households also show a pattern and concur with the previous findings of medium being the preferred choice of size with small being the next choice. XL shows just a few sales because the quantity of XL id very large and may be used over time by a household and also may be due to promotions only taking place once a year for XL.



**Variables & Importance Analysis**

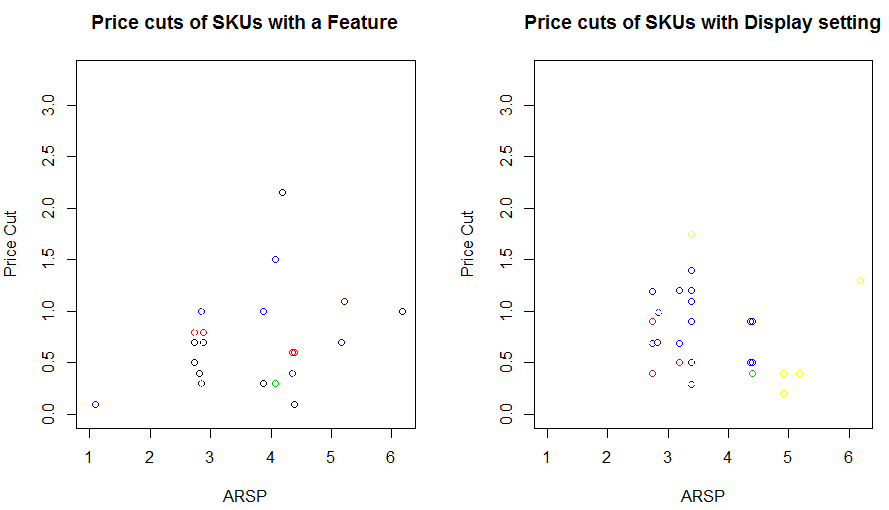
1. **Pricing and Promotions**

Pricing and promotions seem to play a very important part in the sales of different fabric softeners. Further analysis on pricing and promotions reveal that the price cuts increase as ARSP increases and reach the peak when ARSP is little more than $4. After that the price cuts sharply decline. This may be due to lower price cuts on higher priced fabric softeners which are also don’t have high sales. The analysis also reveals that some brands have higher price cuts than others.

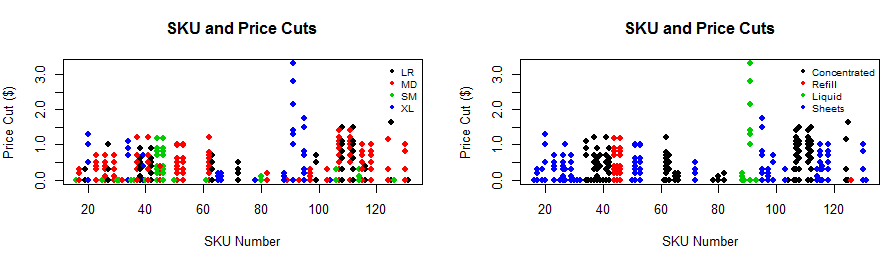


The analysis reveals that fabric softeners which have the presence of a feature are generally higher priced. Fabric softeners which have the presence of a display variable are generally medium priced. This may be due to advertising costs and display costs. Lower priced softeners cannot have these features and display settings and still keep their low price. These features and display settings acting together with price cuts can drive the sales up of the fabric softener. The presence of display variable also have higher price cuts.

This makes since the items which have a price cut are also displayed in a better way than other softeners to increase sales. With the assumption being that better the display, the higher the value of display variable, the analysis reveals even more insight. The better the display the more price cut the SKU has which is in concordance with the assumption that if there is a price cut, the item will be displayed better at stores where the customer sees it clearly and is somewhat attracted to it.



The analysis also reveals that XL (blue) has the highest price cuts. The SKU's which come in liquid form have the highest price cuts and Private Label (Yellow) has the highest as well as the lowest promotions. The lower promoted SKU's in that brand may be funding the higher price cuts in that brand SKU's.

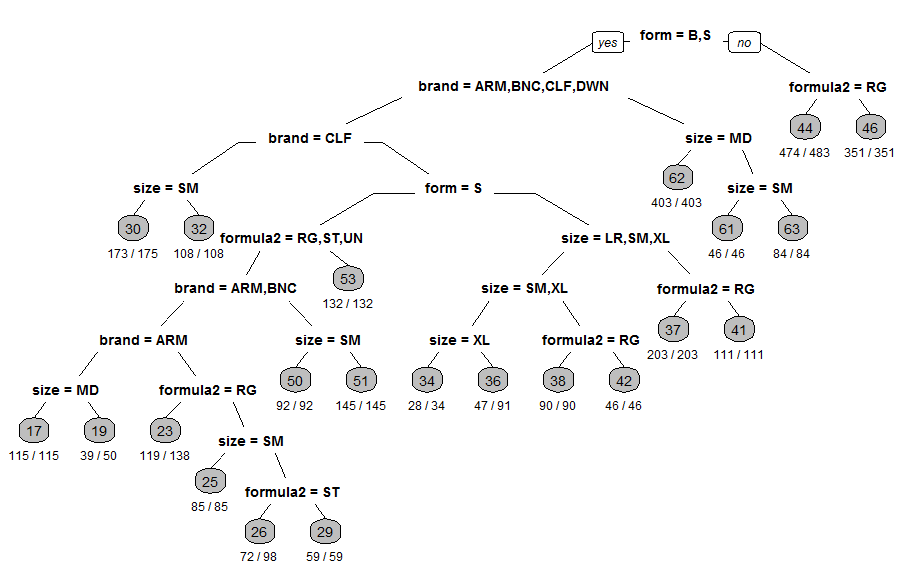




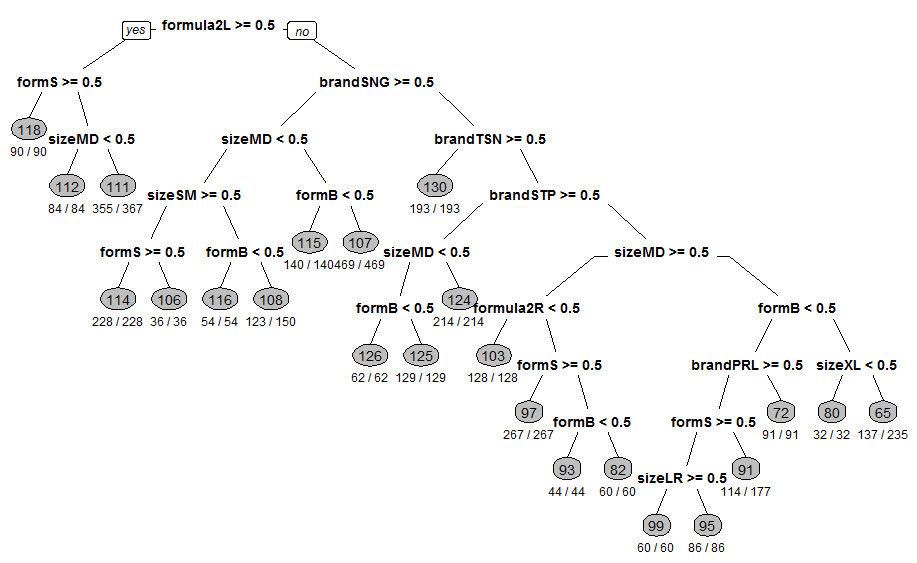
1. **SKU Attributes**

Decision Trees (Classification Trees) were used to evaluate the importance of different attributes of SKUs. The analysis of the decision trees revealed that the most important variable that predicts an SKU was the Size followed by Formula, Brand and the Form of the fabric softener. An interesting finding was that some SKU numbers had identical attributed as others and thus were being misclassified. This may be due to some unrecorded attribute of the SKUs that was not present the dataset which would have created a distinction between the two. To model the decision tree, the data had to be split into parts on the SKU number. This was because the decision trees can only handle 32 levels but the SKU variable had 59 levels. So 2 decision trees were constructed with 1st having 1 to 63 and 2nd having 64 to 131 numbered SKUs.

Tree 1:



Tree 2:



**Predictive Analysis**

1. **SKU Sales/Purchases at a weekly level**

The first analysis was to predict the sales/purchases of different SKU’s at a weekly level. For this analysis, the “weeklydatav2” was used which is described in detail in the data cleaning section.

The goal was to predict the amount of sales of different SKU’s at a weekly level. This is important because companies need to be able to predict the sales of their products in the coming weeks and months and also to be able to add promotions and price cuts at the most optimum times to increase their sales. For this the dataset was split into two parts namely Training and Holdout (test) datasets which are also described in the data cleaning section.

Because the dependent variable, Sales, is a continuous variable, the best choices of predictive models would be the ones which would be able to work with continuous dependent and a combination of continuous and categorical independent variables.

For the predictive checks The Root Node Error is useful in computing the 2 measures of in sample predictive checks which are re-substitution error rate (error rate computed on training sample) and the cross-validated error rate which computes the 10-fold cross validation error rate from the training sample. But these measures will give inflated error rates as they are designed for classification trees and not regression trees.

The best method to check and compare the accuracy of in and out of sample prediction error rates and accuracy will be to compute the Root Mean Squared Error rates of all models and compare them side by side.

The following section describes the different predictive models used, their results, in and out of sample predictive checks and analysis of results.

* 1. **Regression Tree**

Formula:

rpart (formula = as.numeric(sales) ~ as.factor(month) + as.factor(season) + as.factor(brand) + as.factor(form) + as.factor(formula) + as.factor(size) + as.factor(display) + as.factor(feature) + as.numeric(APP) + as.numeric(ARP) + as.numeric(ARSP) + as.numeric(APC) + as.numeric(IRIweek) + as.numeric(lagsales) + as.factor(SKU.), data = weeklytrain)

Results:

Root Node Error: 38008/3068 = 12.388

Optimum Tree Depth: 8

Variable Importance: (High to Low)

APC, SKU., ARP, form, APP, lagsales, size, brand, ARSP and month

In Sample Accuracy Rate:

RMSE = 2.101212

Out of Sample Accuracy Rate:

RMSE = 2.612268

* 1. **Linear Regression**

Formula:

*Lm (formula = as.numeric(sales) ~ as.factor(month) + as.factor(season) + as.factor(brand) + as.factor(form) + as.factor(formula) + as.factor(size) + as.factor(display) + as.factor(feature) + as.numeric(APP) + as.numeric(ARP) + as.numeric(ARSP) + as.numeric(APC) + as.numeric(IRIweek) + as.numeric(lagsales) + as.factor(SKU.), data = weeklytrain)*

Results:

Multiple R-squared: 0.4376

Adjusted R-squared: 0.4223

AIC: 14828.39

BIC: 15328.78

In Sample Accuracy Rate:

RMSE = 2.639605

Out of Sample Accuracy Rate:

RMSE = 3.430332

* 1. **Random Forest**

Formula:

randomForest ( as.numeric(sales) ~ month+season + brand + form + formula + size + display + feature + APP + ARP +ARSP +APC + IRIweek + lagsales+ SKU., data = weeklytrain)

Results:

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 5

Mean of squared residuals: 4.330278

% Var explained: 65.05

Variable Importance: (High to Low)

APC, APP, ARP, APP, lagsales, IRIweek, brand, form and month

In Sample Accuracy Rate:

RMSE = 1.108776

Out of Sample Accuracy Rate:

RMSE = 2.612268

* 1. **Support Vector Regression**

Formula:

svm(formula = as.numeric(sales) ~ month + season + brand + form + formula + size + display + feature + APP + ARP + ARSP + APC + IRIweek + lagsales + SKU., data = weeklytrain)

Results:

SVM-Type: eps-regression

SVM-Kernel: radial

Cost: 1

Gamma: 0.03125

Epsilon: 0.1

Number of Support Vectors: 1205

In Sample Accuracy Rate:

RMSE = 2.432848

Out of Sample Accuracy Rate:

RMSE = 2.038893

* 1. **Predictive Power and Accuracy Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RMSE** | **Decision Tree** | **Linear Regression** | **Random Forest** | **Support Vector Regression** |
| **Training Dataset** | 2.101212 | 2.639605 | **1.108776** | 2.432848 |
| **Holdout Dataset** | 2.612268 | 3.430332 | **1.958551** | 2.038893 |

The results of the predictive power of each model is provided in the table above. The model which was the most accurate in predicting the sales of SKU’s at weekly level was the Random Forest model. It had the lowest Root Mean Squared Error rate of all models in both the in and out of sample predictive checks.

This wasn’t a surprise because Random Forests are generally considered the top performing out of the box predictive models in the industry. The second best performer by its predictive power was the Support Vector Regression model which another top out of the box predictive models being used in the industry today.

The second best model which closely matched the training dataset and which can also be used as an inferential model was the Decision Tree. An interesting finding was that the Support Vector Machine was actually better at predicting than it was at modeling itself to the training dataset. This needs more in-depth analysis.

Linear Regression was the worst performer in both cases which is expected from such a simple model which wasn’t particularly made for predictions.

* 1. **Final Model**

The final suggested model that is selected from the analysis for predicting the sales of SKU’s on a weekly level is the Random Forest model because of its predictive accuracy and because it’s highly accepted in the industry as well for predictive purposes.

1. **Household SKU Sales/Purchases**

The second analysis was to predict the sales/purchases of different SKU’s at a household level. For this analysis, the “hhdatatrainv2” and “hhdatatestv2” datasets were used which are described in detail in the data cleaning section.

The goal was to predict the amount of sales of different SKU’s at a household level in 1992. This is important to predict customer churn and taking appropriate steps to gain more customer base. For this the dataset was split into two parts namely Training and Holdout (test) datasets which are also described in the data cleaning section.

Because the dependent variable, Sales, is again a continuous variable, the best choices of predictive models would be the ones which would be able to work with continuous dependent and a combination of continuous and categorical independent variables.

The best method to check and compare the accuracy of in and out of sample prediction error rates and accuracy will be to compute the Root Mean Squared Error rates of all models and compare them side by side.

The following section describes the different predictive models used, their results, in and out of sample predictive checks and analysis of results.

* 1. **Regression Tree**

Formula:

rpart(formula = as.numeric(sales) ~ as.factor(brand) + as.factor(form) +

as.factor(formula) + as.factor(size) + as.factor(feature) + as.numeric(APP) + as.numeric(ARP) + as.numeric(ARSP) + as.numeric(APC) + as.numeric(HH\_id) +

as.numeric(lagsales) + as.factor(`SKU#`), data = hhdatatrainv2)

Results:

Root Node Error: 19402/35046 = 0.55362

Optimum Tree Depth: 7

Variable Importance: (High to Low)

SKU, APP, ARP, APC and brand

In Sample Accuracy Rate:

RMSE = 0.5000581

Out of Sample Accuracy Rate:

RMSE = 0.2707326

* 1. **Linear Regression**

Formula:

Lm (formula = as.numeric(sales) ~ as.factor(month) + as.factor(season) + as.factor(brand) + as.factor(form) + as.factor(formula) + as.factor(size) + as.factor(display) + as.factor(feature) + as.numeric(APP) + as.numeric(ARP) + as.numeric(ARSP) + as.numeric(APC) + as.numeric(IRIweek) + as.numeric(lagsales) + as.factor(SKU.), data = weeklytrain)

Results:

Multiple R-squared: 0.3402

Adjusted R-squared: 0.3391

AIC: 64284.71

BIC: 64809.51

In Sample Accuracy Rate:

RMSE = 0.6043752

Out of Sample Accuracy Rate:

RMSE = 0.3271541

* 1. **Random Forest**

Random forest ran into trouble when the computing the model fit. This error was unresolvable even after multiple tries.

* 1. **Support Vector Regression**

Formula:

svm(as.numeric(sales) ~ brand+

form + formula +size+

feature + APP +

ARP +ARSP +APC + HH\_id +

lagsales+ `SKU#`, data = hhdatatrainv2))

Results:

SVM-Type: eps-regression

SVM-Kernel: radial

Cost: 1

Gamma: 0.03448276

Epsilon: 0.1

Number of Support Vectors: 1729

In Sample Accuracy Rate:

RMSE = 0.58807

Out of Sample Accuracy Rate:

RMSE = 0.3271541

* 1. **Predictive Power and Accuracy Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RMSE** | **Decision Tree** | **Linear Regression** | **Random Forest** | **Support Vector Regression** |
| **Training Dataset** | **0.5000581** | 0.6043752 | N.A. | 0.58807 |
| **Holdout Dataset** | **0.2707326** | 0.3271541 | N.A. | 0.3271541 |

The results of the predictive power of each model is provided in the table above. The model which was the most accurate in predicting the sales of SKU’s at weekly level was the Decision Tree model. It had the lowest Root Mean Squared Error rate of all models in both the in and out of sample predictive checks.

Decision Trees are generally considered to be one of the top performing out of the box predictive models in the industry after Random Forests and SVM but in this case the Random Forest results could not be obtained. There was a tie between the second best performer by its predictive power. The Support Vector Regression model and the linear regression models both scored equal in their predictive power in the outside of sample dataset but SVM score marginally better than LR in the in sample predictive accuracy. This may be because the SVM model was exactly the same as the LR model with no Kernels. SVM could be modified to get an even more precise predictions.

* 1. **Final Model**

The final suggested model that is selected from the analysis for predicting the sales of SKU’s on a household level is the Decision Tree model because of its predictive accuracy and because it’s is also highly accepted in the industry as well for predictive purposes.

**Conclusion**

The study was conducted to build the most appropriate predictive models for predicting sales/purchases of SKUs at household and weekly levels.

The first analysis was to predict the sales/purchases of different SKU’s at a weekly level. The goal was to predict the amount of sales of different SKU’s at a weekly level. This is important because companies need to be able to predict the sales of their products in the coming weeks and months and also to be able to add promotions and price cuts at the most optimum times to increase their sales.

The second analysis was to predict the sales/purchases of different SKU’s at a household level. The goal was to predict the amount of sales of different SKU’s at a household level in 1992. This is important to predict customer churn and taking appropriate steps to gain more customer base.

The goal was to predict the amount of sales of different SKU’s at a household level in 1992. This is important to predict customer churn and taking appropriate steps to gain more customer base.

Tree based models in both the predictive analysis turned out to be the better models. The predictive models that were built for the prediction of SKU sales/Purchases at both weekly and household level showed the same predictor variables as the top variables in the analysis. They were SKU, APP, ARP, APC and brand. This shows that consumers at highly influenced by price of the product, the promotional discount and the brand of the product as well when choosing to buy a fabric softener (or maybe any other product).

Another interesting find was that the household predictive model had higher accuracy rates than the weekly model. This may be due to the fact that households have a greater chance of repeating their purchase history from the previous year in the next year as well. This is due to their loyalty towards a particular brand or a particular brand in a particular season. Further studies should be conducted to research these findings, especially the loyalty factor as it may be very useful in predicting customer churn.

**Appendix**

**Code:**

#########################

## Loading all data files

d1pur <- read.table("D1PUR.dat")

colnames(d1pur) <- c("HH\_id","trip\_info")

merch <- read.table("MERCH.dat")

colnames(merch) <- c("SKU#", "store#", "IRIweek", "price\_paid", "merchandising")

arsp <- read.table("ARSP.dat")

colnames(arsp) <- c("SKU#", "store#", "ARSP")

brsinfo <- read.table("BRSINFO.dat")

colnames(brsinfo) <- c("SKU#", "description\_of\_SKU", "brand", "form", "formula1", "formula2",

"size", "brand#", "form#", "formula2#", "size#")

iridates <- read.table("IRIdates.csv", skip = 540, sep=",")

iridates <- iridates[,1:2]

colnames(iridates) <- c("week#", "ending\_date")

################################

## Cleaning Individual Dataframes

View(d1pur)

# variable Trip\_Info is in format AAABBBCCC

# AAA = IRI week

# BBB = store#

# CCC = SKU# purchased

d1pur[,3]<- substr(d1pur[,2],0,3)

d1pur[,4]<- substr(d1pur[,2],4,6)

d1pur[,5]<- substr(d1pur[,2],7,9)

colnames(d1pur) <- c("HH\_id", "trip\_info", "IRIweek", "store#", "SKU#")

View(merch)

# Converting SKU# to 3 digit format

merch[,1] <- sprintf("%03d",merch[,1])

# Converting store# to 3 digit format

merch[,2] <- sprintf("%03d",merch[,2])

# Converting all merchandising obs to 6 digit format

merch[,5] <- sprintf("%06d",merch[,5])

# The format of the merchandising variable is AAABCD where

# AAA = depromoted price

# B = ignore

# C = display

# D = feature

merch[,6] <- substr(merch[,5],0,3)

merch[,7] <- substr(merch[,5],4,4)

merch[,8] <- substr(merch[,5],5,5)

merch[,9] <- substr(merch[,5],6,6)

colnames(merch) <- c("SKU#", "store#", "IRIweek", "price\_paid", "merchandising",

"regular\_price", "ignore", "display", "feature")

# Creating dummies for feature and display

merch[,10] <- ifelse(as.numeric(merch[,8])>=1,1,0)

merch[,11] <- ifelse(as.numeric(merch[,9])>=1,1,0)

colnames(merch) <- c("SKU#", "store#", "IRIweek", "price\_paid", "merchandising"

,"regular\_price", "ignore", "display", "feature", "d\_disp", "d\_feat")

# Converting regular\_price into "0.00" format

merch[,6] <- paste(substr(merch[,6],0,1),substr(merch[,6],2,3),sep = ".")

merch[,6] <- as.numeric(merch[,6])

# Creating new variable price\_cut

# price\_cut = regular\_price - price\_paid (if the result is < 0, price\_cut = 0)

merch[,12] <- merch[,6] - merch[,4]

merch[,12] <- ifelse(merch[,12]<0,0,merch[,12])

colnames(merch) <- c("SKU#", "store#", "IRIweek", "price\_paid", "merchandising"

,"regular\_price", "ignore", "display", "feature", "d\_disp", "d\_feat", "price\_cut")

View(arsp)

# Converting SKU# to 3 digit format

arsp[,1] <- sprintf("%03d",arsp[,1])

# Converting store# to 3 digit format

arsp[,2] <- sprintf("%03d",arsp[,2])

View(brsinfo)

# Getting rid of Formula1 col

brsinfo <- brsinfo[,c(1,2,3,4,6,7,8,9,10,11)]

brsinfo <- brsinfo[,c(1,3,4,5,6)]

# Creating Dummy Variables

brsinfo <- cbind2(brsinfo, model.matrix(~brand -1,brsinfo))

brsinfo <- cbind2(brsinfo, model.matrix(~form -1,brsinfo))

brsinfo <- cbind2(brsinfo, model.matrix(~formula2 -1,brsinfo))

brsinfo <- cbind2(brsinfo, model.matrix(~size -1,brsinfo))

######################

# Making Final Dataset

# Combining brsinfo and arsp

arsp\_brs <- cbind2(arsp,brsinfo)

arsp\_brs\_1 <- arsp\_brs[,-1]

## \*\*\* USED Dr. Zantedeschi's function in this part.

## It's a much better and easier way of merging than what I had in mind\*\*\*\*

d1pur[,3] <- as.numeric(d1pur[,3])

d1pur[,4] <- as.numeric(d1pur[,4])

d1pur[,5] <- as.numeric(d1pur[,5])

d1pur1 <- d1pur[,c(5,4,3,1,2)]

merch[,1] <- as.numeric(merch[,1])

merch[,2] <- 70

merch[,3] <- as.numeric(merch[,3])

merch\_d1 <- merge(d1pur1, merch, by=c("IRIweek", "store#","SKU#"))

# Combining all data into final dataset

finaldata <- merge(arsp\_brs\_1,merch\_d1,by=c("SKU#"))

# Removing Redundent variables

finaldatav1 <- finaldata[,c(-2,-31,-33,-35,-37)]

write.csv(finaldatav1,"finaldatav1.csv",row.names = F)

#Importing date/season/month file created in excel from IRIdates.csv

IRIdates\_months\_seasons <- read.csv("IRIdates\_months\_seasons.csv")

finaldatav2 <- merge(finaldatav1,IRIdates\_months\_seasons,by=("IRIweek"))

finaldatav3 <- finaldatav2

write.csv(finaldatav3,"finaldatav3.csv",row.names = F)

########################################

## Making Datasets for Predictive Models

#######################

## Weekly Model Dataset

sku <- sort(unique(finaldatav3$`SKU#`))

iriweek <- as.integer(sort(unique(finaldatav3$IRIweek)))

weeklydata <- data.frame(IRIweek=1:4602,`SKU#`=1:4602)

row <- 1

for (i in iriweek) {

for (j in sku) {

weeklydata$IRIweek[row] <- i

weeklydata$`SKU#`[row] <- j

row <- row+1

}

}

## Adding more variables

# Sales variable (number of sold in paticular week)

row <- 1

for (i in iriweek) {

for (j in sku) {

if (length(table(finaldatav3[finaldatav3$IRIweek==i & finaldatav3$`SKU#`==j,"SKU#"]))==0) {

weeklydata$sales[row]<-0

}

if (length(table(finaldatav3[finaldatav3$IRIweek==i & finaldatav3$`SKU#`==j,"SKU#"]))!=0) {

weeklydata$sales[row]<-table(finaldatav3[finaldatav3$IRIweek==i & finaldatav3$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

#check for consistency

sum(weeklydata$sales) #should be equal to 6554

# Creating lag variable for Sales

row <- 1

for (i in iriweek-1) {

for (j in sku) {

if (length(table(finaldatav3[finaldatav3$IRIweek==i & finaldatav3$`SKU#`==j,"SKU#"]))==0) {

weeklydata$lagsales[row]<-0

}

if (length(table(finaldatav3[finaldatav3$IRIweek==i & finaldatav3$`SKU#`==j,"SKU#"]))!=0) {

weeklydata$lagsales[row]<-table(finaldatav3[finaldatav3$IRIweek==i & finaldatav3$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

sum(weeklydata$lagsales)

# Creating variable for Avg. Price

row<-1

for (i in iriweek) {

for (i in sku) {

weeklydata$ARSP[row] <- finaldatav3$ARSP[finaldatav3$`SKU#`==i][1]

row<-row+1

}}

# Creating variable for Avg. Price Paid (average of price paid in that week; if NA then ARSP)

row<-1

for (i in iriweek) {

for (j in sku) {

if (is.na(sum(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i]))==F)

{

weeklydata$APP[row] <- sum(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])

}

if (is.na(sum(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i]))==T) {

weeklydata$APP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Regular Price (average of price paid in that week; if NA then ARSP)

row<-1

for (i in iriweek) {

for (j in sku) {

if (is.na(sum(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i]))==F)

{

weeklydata$ARP[row] <- sum(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])

}

if (is.na(sum(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i]))==T) {

weeklydata$ARP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Price Cut (average of price paid in that week; if NA then ARSP)

row<-1

for (i in iriweek) {

for (j in sku) {

if (is.na(sum(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i]))==F)

{

weeklydata$APC[row] <- sum(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])

}

if (is.na(sum(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i])/

length(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$IRIweek==i]))==T) {

weeklydata$APC[row] <- 0

}

row<-row+1

}}

weeklydatav1 <- weeklydata

# Adding brand, form, formula, size, display, feature

row <- 1

for (i in iriweek) {

for (j in sku) {

weeklydatav1$brand[weeklydatav1$`SKU#`==j] <- as.character(finaldatav3$brand[finaldatav3$`SKU#`==j])

weeklydatav1$form[weeklydatav1$`SKU#`==j] <- as.character(finaldatav3$form[finaldatav3$`SKU#`==j])

weeklydatav1$formula[weeklydatav1$`SKU#`==j] <- as.character(finaldatav3$formula2[finaldatav3$`SKU#`==j])

weeklydatav1$size[weeklydatav1$`SKU#`==j] <- as.character(finaldatav3$size[finaldatav3$`SKU#`==j])

weeklydatav1$display[weeklydatav1$`SKU#`==j] <- as.character(finaldatav3$display[finaldatav3$`SKU#`==j])

weeklydatav1$feature[weeklydatav1$`SKU#`==j] <- as.character(finaldatav3$feature[finaldatav3$`SKU#`==j])

row=row+1

}

}

# Creating Seasonal and Monthly variables

write.csv(weeklydatav1,"weeklydatav1.csv",row.names = F)

# Manuplated the date time variables in excel and saved as weeklydatav1

weeklydatav2 <- read.csv("weeklydatav2.csv")

# lm1 <- lm(sales ~ IRIweek+as.factor(`SKU#`)+lagsales+ARSP+APP+ARP+APC+as.factor(season)+as.factor(month),data=weeklydatav2)

# summary(lm1)

# plot(lm1)

## Making Train and Test datasets

weeklytrain <- weeklydatav2[1:3068,]

weeklytest <- weeklydatav2[3069:4602,]

# lm1 <- lm(sales ~ IRIweek+as.factor(SKU.)+lagsales+ARSP+APP+ARP+APC+as.factor(brand)+as.factor(form)+

# as.factor(formula)+as.factor(size)+as.factor(display)+as.factor(feature)+as.factor(month)+

# as.factor(season),data=weeklytrain)

# summary(lm1)

# plot(lm1)

# weeklytest[,17] <- predict(lm1,weeklytest[])

# predictions <- weeklytest[,c(3,17)]

# predictions[,3] <- predictions[,2] - predictions[,1]

# mean(predictions[,3]^2) #MSE

###########################

## Household Model Dataset

## Using Test and Train datasets which were created form finaldatav3 to create 2 datasets for houhehold

## It will be impossible to seperate the data according to year after the data ghas been created

## Train Set

## Train Set

sku <- sort(unique(train$`SKU#`))

hh <- as.integer(sort(unique(train$HH\_id)))

hhdatatrain <- data.frame(HH\_id=1:35046,`SKU#`=1:35046)

row <- 1

for (i in hh) {

for (j in sku) {

hhdatatrain$HH\_id[row] <- i

hhdatatrain$`SKU#`[row] <- j

row <- row+1

}

}

## Adding more variables

# Sales variable (number of sold in paticular week)

row <- 1

for (i in hh) {

for (j in sku) {

if (length(table(train[train$HH\_id==i & train$`SKU#`==j,"SKU#"]))==0) {

hhdatatrain$sales[row]<-0

}

if (length(table(train[train$HH\_id==i & train$`SKU#`==j,"SKU#"]))!=0) {

hhdatatrain$sales[row]<-table(train[train$HH\_id==i & train$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

##Check for consistency

nrow(train)

sum(hhdatatrain$sales) #4417

# Creating lag variable for Sales

row <- 1

for (i in hh-1) {

for (j in sku) {

if (length(table(train[train$HH\_id==i & train$`SKU#`==j,"SKU#"]))==0) {

hhdatatrain$lagsales[row]<-0

}

if (length(table(train[train$HH\_id==i & train$`SKU#`==j,"SKU#"]))!=0) {

hhdatatrain$lagsales[row]<-table(train[train$HH\_id==i & train$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

# Creating variable for Avg. Price

row<-1

for (i in hh) {

for (i in sku) {

hhdatatrain$ARSP[row] <- train$ARSP[train$`SKU#`==i][1]

row<-row+1

}}

# Creating variable for Avg. Price Paid (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(train$price\_paid[train$`SKU#`==j & train$HH\_id==i])/

length(train$price\_paid[train$`SKU#`==j & train$HH\_id==i]))==F)

{

hhdatatrain$APP[row] <- sum(train$price\_paid[train$`SKU#`==j & train$HH\_id==i])/

length(train$price\_paid[train$`SKU#`==j & train$HH\_id==i])

}

if (is.na(sum(train$price\_paid[train$`SKU#`==j & train$HH\_id==i])/

length(train$price\_paid[train$`SKU#`==j & train$HH\_id==i]))==T) {

hhdatatrain$APP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Regular Price (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(train$regular\_price[train$`SKU#`==j & train$HH\_id==i])/

length(train$regular\_price[train$`SKU#`==j & train$HH\_id==i]))==F)

{

hhdatatrain$ARP[row] <- sum(train$regular\_price[train$`SKU#`==j & train$HH\_id==i])/

length(train$regular\_price[train$`SKU#`==j & train$HH\_id==i])

}

if (is.na(sum(train$regular\_price[train$`SKU#`==j & train$HH\_id==i])/

length(train$regular\_price[train$`SKU#`==j & train$HH\_id==i]))==T) {

hhdatatrain$ARP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Price Cut (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(train$price\_cut[train$`SKU#`==j & train$HH\_id==i])/

length(train$price\_cut[train$`SKU#`==j & train$HH\_id==i]))==F)

{

hhdatatrain$APC[row] <- sum(train$price\_cut[train$`SKU#`==j & train$HH\_id==i])/

length(train$price\_cut[train$`SKU#`==j & train$HH\_id==i])

}

if (is.na(sum(train$price\_cut[train$`SKU#`==j & train$HH\_id==i])/

length(train$price\_cut[train$`SKU#`==j & train$HH\_id==i]))==T) {

hhdatatrain$APC[row] <- 0

}

row<-row+1

}}

hhdatatrainv1 <- hhdatatrain

# Adding brand, form, formula, size, display, feature

row <- 1

for (i in hh) {

for (j in sku) {

hhdatatrainv1$brand[hhdatatrainv1$`SKU#`==j] <- as.character(train$brand[train$`SKU#`==j & !duplicated(train$`SKU#`)])

hhdatatrainv1$form[hhdatatrainv1$`SKU#`==j] <- as.character(train$form[train$`SKU#`==j & !duplicated(train$`SKU#`)])

hhdatatrainv1$formula[hhdatatrainv1$`SKU#`==j] <- as.character(train$formula2[train$`SKU#`==j & !duplicated(train$`SKU#`)])

hhdatatrainv1$size[hhdatatrainv1$`SKU#`==j] <- as.character(train$size[train$`SKU#`==j & !duplicated(train$`SKU#`)])

hhdatatrainv1$display[hhdatatrainv1$`SKU#`==j] <- as.character(train$display[train$`SKU#`==j & !duplicated(train$`SKU#`)])

hhdatatrainv1$feature[hhdatatrainv1$`SKU#`==j] <- as.character(train$feature[train$`SKU#`==j & !duplicated(train$`SKU#`)])

row=row+1

}

}

hhdatatrainv2 <- hhdatatrainv1

## Testing Dataset

sku <- sort(unique(test$`SKU#`))

hh <- as.integer(sort(unique(test$HH\_id)))

hhdatatest <- data.frame(HH\_id=1:35046,`SKU#`=1:35046)

row <- 1

for (i in hh) {

for (j in sku) {

hhdatatest$HH\_id[row] <- i

hhdatatest$`SKU#`[row] <- j

row <- row+1

}

}

## Adding more variables

# Sales variable (number of sold in paticular week)

row <- 1

for (i in hh) {

for (j in sku) {

if (length(table(test[test$HH\_id==i & test$`SKU#`==j,"SKU#"]))==0) {

hhdatatest$sales[row]<-0

}

if (length(table(test[test$HH\_id==i & test$`SKU#`==j,"SKU#"]))!=0) {

hhdatatest$sales[row]<-table(test[test$HH\_id==i & test$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

##Check for consistency

nrow(test)

sum(hhdatatest$sales) #4417

# Creating lag variable for Sales

row <- 1

for (i in hh-1) {

for (j in sku) {

if (length(table(test[test$HH\_id==i & test$`SKU#`==j,"SKU#"]))==0) {

hhdatatest$lagsales[row]<-0

}

if (length(table(test[test$HH\_id==i & test$`SKU#`==j,"SKU#"]))!=0) {

hhdatatest$lagsales[row]<-table(test[test$HH\_id==i & test$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

# Creating variable for Avg. Price

row<-1

for (i in hh) {

for (i in sku) {

hhdatatest$ARSP[row] <- test$ARSP[test$`SKU#`==i][1]

row<-row+1

}}

# Creating variable for Avg. Price Paid (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(test$price\_paid[test$`SKU#`==j & test$HH\_id==i])/

length(test$price\_paid[test$`SKU#`==j & test$HH\_id==i]))==F)

{

hhdatatest$APP[row] <- sum(test$price\_paid[test$`SKU#`==j & test$HH\_id==i])/

length(test$price\_paid[test$`SKU#`==j & test$HH\_id==i])

}

if (is.na(sum(test$price\_paid[test$`SKU#`==j & test$HH\_id==i])/

length(test$price\_paid[test$`SKU#`==j & test$HH\_id==i]))==T) {

hhdatatest$APP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Regular Price (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(test$regular\_price[test$`SKU#`==j & test$HH\_id==i])/

length(test$regular\_price[test$`SKU#`==j & test$HH\_id==i]))==F)

{

hhdatatest$ARP[row] <- sum(test$regular\_price[test$`SKU#`==j & test$HH\_id==i])/

length(test$regular\_price[test$`SKU#`==j & test$HH\_id==i])

}

if (is.na(sum(test$regular\_price[test$`SKU#`==j & test$HH\_id==i])/

length(test$regular\_price[test$`SKU#`==j & test$HH\_id==i]))==T) {

hhdatatest$ARP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Price Cut (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(test$price\_cut[test$`SKU#`==j & test$HH\_id==i])/

length(test$price\_cut[test$`SKU#`==j & test$HH\_id==i]))==F)

{

hhdatatest$APC[row] <- sum(test$price\_cut[test$`SKU#`==j & test$HH\_id==i])/

length(test$price\_cut[test$`SKU#`==j & test$HH\_id==i])

}

if (is.na(sum(test$price\_cut[test$`SKU#`==j & test$HH\_id==i])/

length(test$price\_cut[test$`SKU#`==j & test$HH\_id==i]))==T) {

hhdatatest$APC[row] <- 0

}

row<-row+1

}}

hhdatatestv1 <- hhdatatest

# Adding brand, form, formula, size, display, feature

row <- 1

for (i in hh) {

for (j in sku) {

hhdatatestv1$brand[hhdatatestv1$`SKU#`==j] <- as.character(test$brand[test$`SKU#`==j & !duplicated(test$`SKU#`)])

hhdatatestv1$form[hhdatatestv1$`SKU#`==j] <- as.character(test$form[test$`SKU#`==j & !duplicated(test$`SKU#`)])

hhdatatestv1$formula[hhdatatestv1$`SKU#`==j] <- as.character(test$formula2[test$`SKU#`==j & !duplicated(test$`SKU#`)])

hhdatatestv1$size[hhdatatestv1$`SKU#`==j] <- as.character(test$size[test$`SKU#`==j & !duplicated(test$`SKU#`)])

hhdatatestv1$display[hhdatatestv1$`SKU#`==j] <- as.character(test$display[test$`SKU#`==j & !duplicated(test$`SKU#`)])

hhdatatestv1$feature[hhdatatestv1$`SKU#`==j] <- as.character(test$feature[test$`SKU#`==j & !duplicated(test$`SKU#`)])

row=row+1

}

}

# Removing SKU 16 from dataset as test set doesnt have SKU #16

hhdatatestv2 <- hhdatatestv1[hhdatatestv1$`SKU#`!=16,]

# Full Dataset

sku <- sort(unique(finaldatav3$`SKU#`))

hh <- as.integer(sort(unique(finaldatav3$HH\_id)))

hhdata <- data.frame(HH\_id=1:35046,`SKU#`=1:35046)

row <- 1

for (i in hh) {

for (j in sku) {

hhdata$HH\_id[row] <- i

hhdata$`SKU#`[row] <- j

row <- row+1

}

}

## Adding more variables

# Sales variable (number of sold in paticular week)

row <- 1

for (i in hh) {

for (j in sku) {

if (length(table(finaldatav3[finaldatav3$HH\_id==i & finaldatav3$`SKU#`==j,"SKU#"]))==0) {

hhdata$sales[row]<-0

}

if (length(table(finaldatav3[finaldatav3$HH\_id==i & finaldatav3$`SKU#`==j,"SKU#"]))!=0) {

hhdata$sales[row]<-table(finaldatav3[finaldatav3$HH\_id==i & finaldatav3$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

##Check for consistency

sum(hhdata$sales) #6554

# Creating lag variable for Sales

row <- 1

for (i in hh-1) {

for (j in sku) {

if (length(table(finaldatav3[finaldatav3$HH\_id==i & finaldatav3$`SKU#`==j,"SKU#"]))==0) {

hhdata$lagsales[row]<-0

}

if (length(table(finaldatav3[finaldatav3$HH\_id==i & finaldatav3$`SKU#`==j,"SKU#"]))!=0) {

hhdata$lagsales[row]<-table(finaldatav3[finaldatav3$HH\_id==i & finaldatav3$`SKU#`==j,"SKU#"])

}

row=row+1

}

}

# Creating variable for Avg. Price

row<-1

for (i in hh) {

for (i in sku) {

hhdata$ARSP[row] <- finaldatav3$ARSP[finaldatav3$`SKU#`==i][1]

row<-row+1

}}

# Creating variable for Avg. Price Paid (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i]))==F)

{

hhdata$APP[row] <- sum(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])

}

if (is.na(sum(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$price\_paid[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i]))==T) {

hhdata$APP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Regular Price (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i]))==F)

{

hhdata$ARP[row] <- sum(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])

}

if (is.na(sum(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$regular\_price[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i]))==T) {

hhdata$ARP[row] <- 0

}

row<-row+1

}}

# Creating variable for Avg. Price Cut (average of price paid in that week; if NA then ARSP)

row<-1

for (i in hh) {

for (j in sku) {

if (is.na(sum(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i]))==F)

{

hhdata$APC[row] <- sum(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])

}

if (is.na(sum(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i])/

length(finaldatav3$price\_cut[finaldatav3$`SKU#`==j & finaldatav3$HH\_id==i]))==T) {

hhdata$APC[row] <- 0

}

row<-row+1

}}

hhdatav1 <- hhdata

# Adding brand, form, formula, size, display, feature

row <- 1

for (i in hh) {

for (j in sku) {

hhdatav1$brand[hhdatav1$`SKU#`==j] <- as.character(finaldatav3$brand[finaldatav3$`SKU#`==j & !duplicated(finaldatav3$`SKU#`)])

hhdatav1$form[hhdatav1$`SKU#`==j] <- as.character(finaldatav3$form[finaldatav3$`SKU#`==j & !duplicated(finaldatav3$`SKU#`)])

hhdatav1$formula[hhdatav1$`SKU#`==j] <- as.character(finaldatav3$formula2[finaldatav3$`SKU#`==j & !duplicated(finaldatav3$`SKU#`)])

hhdatav1$size[hhdatav1$`SKU#`==j] <- as.character(finaldatav3$size[finaldatav3$`SKU#`==j & !duplicated(finaldatav3$`SKU#`)])

hhdatav1$display[hhdatav1$`SKU#`==j] <- as.character(finaldatav3$display[finaldatav3$`SKU#`==j & !duplicated(finaldatav3$`SKU#`)])

hhdatav1$feature[hhdatav1$`SKU#`==j] <- as.character(finaldatav3$feature[finaldatav3$`SKU#`==j & !duplicated(finaldatav3$`SKU#`)])

row=row+1

}

}

## Making Train and Test datasets

weeklytrain <- weeklydatav2[1:3068,]

weeklytest <- weeklydatav2[3069:4602,]

###########################

# Exploratory Data Analysis

## with Fianl Dataset

par(mfrow=c(3,2))

hist(finaldatav3$ARSP, xlab = "ARSP", main= "Distribution of ARSP")

hist(finaldatav3$price\_paid, xlab = "Price Paid", main= "Distribution of Price Paid")

hist(finaldatav3$price\_cut, xlab = "Price Cut", main= "Distribution of Price Cut")

hist(finaldatav3$`SKU#`, breaks=59,xlab = "SKU number", main= "Distribution of ammount bought of each SKU")

hist(finaldatav3$HH\_id,breaks = 622, xlab = "Household ID number", main= "Distribution of total purchases by each Household" )

par(mfrow=c(1,1))

par(mfrow=c(2,2))

barplot(prop.table(table(finaldatav3$brand)), main="Distribution of total purchases by Fabric Softener Brands")

barplot(prop.table(table(finaldatav3$form)), main="Distribution of total purchases by Fabric Softener Froms")

barplot(prop.table(table(finaldatav3$formula2)), main="Distribution of total purchases by Fabric Softener Fromulas")

barplot(prop.table(table(finaldatav3$size)), main="Distribution of total purchases by Fabric Softener Size")

par(mfrow=c(1,1))

par(mfrow=c(1,2))

barplot(prop.table(table(finaldatav3$display)), main = "Distribution of SKUs Display categories")

barplot(prop.table(table(finaldatav3$feature)), main = "Distribution of SKUs Feature categories")

par(mfrow=c(1,1))

par(mfrow=c(2,2))

barplot(prop.table(table(finaldatav3$ARSP)),ylim=c(0,0.2), main = "Distribution of ARSP")

barplot(prop.table(table(finaldatav3$price\_paid)),ylim=c(0,0.2), main = "Distribution of Price Paid")

barplot(prop.table(table(finaldatav3$regular\_price)),ylim=c(0,0.2), main = "Distribution of Regular Price")

barplot(prop.table(table(finaldatav3$price\_cut)),ylim=c(0,0.2), main = "Distribution of Price Cut")

summary(finaldatav3)

plot(finaldatav3[,c(1:20)])

plot(finaldatav3[,c(21:40)])

#####################################

## Temporal Variations in SKU choices

# Size of SKU's bought in IRI Weeks

par(mfrow=c(2,2))

barplot(prop.table(table(finaldatav2$IRIweek[finaldatav2$sizeSM==1 & finaldatav2$Season=="summer"])),ylim=c(0,0.1), main = "Small")

barplot(prop.table(table(finaldatav2$IRIweek[finaldatav2$sizeMD==1 & finaldatav2$Season=="summer"])),ylim=c(0,0.1), main = "Medium")

barplot(prop.table(table(finaldatav2$IRIweek[finaldatav2$sizeLR==1 & finaldatav2$Season=="summer"])),ylim=c(0,0.1), main = "Large")

barplot(prop.table(table(finaldatav2$IRIweek[finaldatav2$sizeXL==1 & finaldatav2$Season=="summer"])),ylim=c(0,0.1), main = "X-Large")

par(mfrow=c(1,1))

# No clear pattern

table(finaldatav2$Season,finaldata$form)

table(finaldatav2$Season,as.factor(finaldata$`SKU#`))

table(finaldatav2$Season,finaldatav2$d\_feat)

table(finaldatav2$Season,finaldatav2$d\_disp)

# Sale of Brands by Season

table(finaldatav2$Season,finaldata$brand)

par(mfrow=c(2,2))

barplot(prop.table(table(finaldata$brand[finaldatav2$Season=="winter"])), main="Winter", xlab = "Brand", col = "Light Blue")

barplot(prop.table(table(finaldata$brand[finaldatav2$Season=="spring"])), main="Spring", xlab = "Brand", col = "light green")

barplot(prop.table(table(finaldata$brand[finaldatav2$Season=="summer"])), main="Summer", xlab = "Brand", col = "orange")

barplot(prop.table(table(finaldata$brand[finaldatav2$Season=="autumn"])), main="Autumn", xlab = "Brand", col = "brown")

par(mfrow=c(1,1))

#Clear distinction of choice of brand in different seasons

# There could be someting in a particular brand that causes it to be prefered in a season

# Sale of Brands by Month

table(finaldatav2$Season,finaldata$brand)

par(mfrow=c(3,4))

for (i in 1:12) {

barplot(prop.table(table(finaldata$brand[finaldatav2$Month==i])), main=i, xlab = "Brand", col = i)

}

par(mfrow=c(1,1))

# Some brands are preferred in colder months than in warmer months

# Sale of Size by Season

table(finaldatav3$Season,finaldatav3$size)

par(mfrow=c(2,2))

barplot(prop.table(table(finaldata$size[finaldatav2$Season=="winter"])), main="Sale of Sizes in Winter", xlab = "Size", col = "Light Blue", ylim = c(0,0.8))

barplot(prop.table(table(finaldata$size[finaldatav2$Season=="spring"])), main="Sale of Sizes in Spring", xlab = "Size", col = "light green", ylim = c(0,0.8))

barplot(prop.table(table(finaldata$size[finaldatav2$Season=="summer"])), main="Sale of Sizes in Summer", xlab = "Size", col = "orange", ylim = c(0,0.8))

barplot(prop.table(table(finaldata$size[finaldatav2$Season=="autumn"])), main="Sale of Sizes in Autumn", xlab = "Size", col = "brown", ylim = c(0,0.8))

par(mfrow=c(1,1))

# Medium is the most widely sold size across seasons

# XL is mostly sold in Autumn

# Summer and Spring see high sales of small

# Price cut by Season

par(mfrow=c(2,2))

hist(finaldatav3$price\_cut[finaldatav3$Season=="winter" & finaldatav3$price\_cut != 0], ylim = c(0,500), main="Price Cuts in Winter", xlab = "Price Cut", col = "Light Blue")

hist(finaldatav3$price\_cut[finaldatav3$Season=="spring" & finaldatav3$price\_cut != 0], ylim = c(0,500), main="Price Cuts in Spring", xlab = "Price Cut", col = "light green")

hist(finaldatav3$price\_cut[finaldatav3$Season=="summer" & finaldatav3$price\_cut != 0], ylim = c(0,500),main="Price Cuts in Summer", xlab = "Price Cut", col = "orange")

hist(finaldatav3$price\_cut[finaldatav3$Season=="autumn" & finaldatav3$price\_cut != 0], ylim = c(0,500), main="Price Cuts in Autumn", xlab = "Price Cut", col = "brown")

par(mfrow=c(1,1))

# The higher price cuts took place in the Autumn

# The most mid level price cuts took place in Spring

# The most low level price custs took place in Winter

# Summer had the least price cuts

# Sale of SKU by Season

table(finaldatav3$Season,finaldatav3$`SKU#`)

par(mfrow=c(2,2))

barplot(prop.table(table(finaldatav3$`SKU#`[finaldatav3$Season=="winter"])), main="Sale of SKU# in Winter", xlab = "SKU Number", col = "Light Blue", ylim = c(0,0.1))

barplot(prop.table(table(finaldatav3$`SKU#`[finaldatav3$Season=="spring"])), main="Sale of SKU# in WinterSpring", xlab = "SKU Number", col = "light green", ylim = c(0,0.1))

barplot(prop.table(table(finaldatav3$`SKU#`[finaldatav3$Season=="summer"])), main="Sale of SKU# in WinterSummer", xlab = "SKU Number", col = "orange", ylim = c(0,0.1))

barplot(prop.table(table(finaldatav3$`SKU#`[finaldatav3$Season=="autumn"])), main="Sale of SKU# in WinterAutumn", xlab = "SKU Number", col = "brown", ylim = c(0,0.1))

par(mfrow=c(1,1))

# Some SKU choices are more popular in some seasons than others

# Monthly/Seasonal Sales of SKU's

par(mfrow=c(1,2))

barplot(prop.table(table(finaldatav3$Month)), main="Monthly Sales of SKU's")

# First Half of the year the sales are high as compared to rest of the year

barplot(prop.table(table(finaldatav3$Season)), main="Seasonal Sales of SKU's")

# Spring sees the highest sales followed by winter

# Highest price cuts in autum maybe because it sees the lowest sales

par(mfrow=c(1,1))

######################################

## Household Variations in SKU choices

par(mfrow=c(2,2))

hist(finaldatav2$HH\_id[finaldatav2$Season=="winter"], ylim = c(0,300), main="# of Fab. Sof. bought by each HH in Winter", xlab = "House ID", col = "Light Blue")

hist(finaldatav2$HH\_id[finaldatav2$Season=="spring"], ylim = c(0,300), main="# of Fab. Sof. bought by each HH in Spring", xlab = "House ID", col = "light green")

hist(finaldatav2$HH\_id[finaldatav2$Season=="summer"], ylim = c(0,300), main="# of Fab. Sof. bought by each HH in Summer", xlab = "House ID", col = "orange")

hist(finaldatav2$HH\_id[finaldatav2$Season=="autumn"], ylim = c(0,300), main="# of Fab. Sof. bought by each HH in Autumn", xlab = "House ID", col = "brown")

par(mfrow=c(1,1))

#Sales of fabric softner go up in winter and spring months. Highest in Spring

# ARSP per SKU by SIZE

par(mfrow=c(2,2))

plot(finaldatav3$`SKU#`[finaldatav3$size=="SM"],finaldatav3$ARSP[finaldatav3$size=="SM"], col = 1, pch =19, ylim=c(0,7), xlab="SKU #",ylab="" ,main="ARSP of SM sized SKUs")

plot(finaldatav3$`SKU#`[finaldatav3$size=="MD"],finaldatav3$ARSP[finaldatav3$size=="MD"], col = 2, pch =19, ylim=c(0,7), xlab="SKU #",ylab="" , main="ARSP of MD sized SKUs")

plot(finaldatav3$`SKU#`[finaldatav3$size=="LR"],finaldatav3$ARSP[finaldatav3$size=="LR"], col = 3, pch =19, ylim=c(0,7), xlab="SKU #",ylab="" , main="ARSP of LR sized SKUs")

plot(finaldatav3$`SKU#`[finaldatav3$size=="XL"],finaldatav3$ARSP[finaldatav3$size=="XL"], col = 4, pch =19, ylim=c(0,7), xlab="SKU #",ylab="" , main="ARSP of XL sized SKUs")

par(mfrow=c(1,1))

#Small size SKU's have a lower price and lease variation.

# XL has the most ammount of variation in price

#SAle of differrent sized sku's by hh's

par(mfrow=c(2,2))

hist(finaldatav3$HH\_id[finaldatav3$size=="SM"],col = 1, pch =19,ylim=c(0,300), xlab="SKU #",ylab="" ,main="ARSP of SM sized SKUs")

hist(finaldatav3$HH\_id[finaldatav3$size=="MD"],col = 2, pch =19,ylim=c(0,300), xlab="SKU #",ylab="" , main="ARSP of MD sized SKUs")

hist(finaldatav3$HH\_id[finaldatav3$size=="LR"],col = 3, pch =19, ylim=c(0,300),xlab="SKU #",ylab="" , main="ARSP of LR sized SKUs")

hist(finaldatav3$HH\_id[finaldatav3$size=="XL"],col = 4, pch =19,ylim=c(0,300), xlab="SKU #",ylab="" , main="ARSP of XL sized SKUs")

par(mfrow=c(1,1))

# Total number of sales in each week (Sales)

# Total ammount ($) of sales in each week (sales \* ARSP)

# Total price cut ($) in each week (Sales \* APC)

# Total average price paid in each week (sum of APP)

###############################################

## Analysis of Variables (Pricing & Promotions)

# Relationship between Price\_cut and average regular selling price

cor(finaldatav3$price\_cut,finaldatav3$ARSP) # +0.5 correlation

#The price cuts increase as ARSP increases and reach the peak when ARSP is little more than $4.

# After that the price cuts sharply decline. This may be due to lower price cuts on higher priced

# fabric softners which are also dont have high sales.

plot(finaldatav3$ARSP,finaldatav3$price\_cut, col=finaldatav3$brand,xlab = "ARSP", ylab = "Price Cut", main = "Price cuts by ARSP of SKUs")

legend('topright', c("ARM", "BNC", "CLF", "DWN", "FNT", "GEN", "PRL", "SNG", "STP", "TSN"),col=(1:10), bty='n',pch = 19, cex=0.8)

#The price cuts increase as ARSP increases and reach the peak when ARSP is little more than $4.

# After that the price cuts sharply decline. This may be due to lower price cuts on higher priced

# fabric softners which are also dont have high sales.

# Some brands have higher price cuts than others

par(mfrow=c(1,2))

plot(finaldatav3$ARSP,finaldatav3$price\_cut, col=finaldatav3$feature,xlab = "ARSP", ylab = "Price Cut", main = "Price cuts of SKUs with a Feature")

# Fabric softners which have the presence of a feature are genrally higher priced

plot(finaldatav3$ARSP,finaldatav3$price\_cut, col=finaldatav3$display,xlab = "ARSP", ylab = "Price Cut", main = "Price cuts of SKUs with Display setting")

# Fabric softners which have the presence of a display variable are genrally medium priced

# this may be due to advertising costs and display costs. Lower priced softners cannot

# have these features and display settings and still keep their low price

# The presence of display variable also have higher price cuts. This makes sence since the items

# which have a price cut are also displayed in a better way than other softners to increase sales

par(mfrow=c(1,1))

# this plot reveals even more insight. The better the display (assumption is better the display,

par(mfrow=c(2,2))

# Price cut and SKU in different sizes(Do some SKU's have more price cuts than others?)

plot(finaldatav3$`SKU#`,finaldatav3$price\_cut, col=finaldatav3$size,

ylab = "Price Cut ($)", xlab = "SKU Number", pch=19,

main = "SKU and Price Cuts")

legend('topright', c("LR","MD","SM","XL"),pch = 19,col=c(1,2,3,4), cex=.75,bty='n')

# XL (blue) has the highest price cuts

# Price cut and SKU in different Forms

plot(finaldatav3$`SKU#`,finaldatav3$price\_cut, col=finaldatav3$form,

ylab = "Price Cut ($)", xlab = "SKU Number", pch=19,

main = "SKU and Price Cuts")

legend('topright', c("Concentrated","Refill","Liquid","Sheets"),col=c(1,2,3,4), cex=.75,bty='n',pch = 19)

# The SKU's which come in liquid form have the highest price cuts

par(mfrow=c(1,1))

# Price Cuts and SKU's with different brands

plot(finaldatav3$`SKU#`,finaldatav3$price\_cut, col=finaldatav3$brand,

ylab = "Price Cut ($)", xlab = "SKU Number", pch=19,

main = "SKU and Price Cuts")

legend('topright', c("ARM", "BNC", "CLF", "DWN", "FNT", "GEN", "PRL", "SNG", "STP", "TSN"),col=(1:10), bty='n',pch = 19, cex=0.8)

# Private Label has the highest as well as the lowest promotions.

# The lower promoted SKU's in that brand may be funding the higher promoted SKU's

par(mfrow=c(1,1))

#################################

## Attribute analysis of Each SKU

# Tree can be drawn showing the formulation of skus by different brands, sizes, froms and formulas

library(rpart) # Popular decision tree algorithm

library(rattle) # Fancy tree plot

library(rpart.plot) # Enhanced tree plots

library(RColorBrewer) # Color selection for fancy tree plot

library(party) # Alternative decision tree algorithm

library(partykit) # Convert rpart object to BinaryTree

library(caret)

skutree <- rpart(as.factor(`SKU#`) ~ brandARM + brandBNC + brandCLF + brandDWN + brandFNT + brandGEN + brandPRL +

brandSNG + brandSTP + brandTSN+ formB+formF+formL+formS+formula2LT+formula2RG+formula2ST+

formula2UN+sizeLR+sizeMD+sizeSM+sizeXL,data=finaldatav3, method= "class")

plot(tree1, uniform=TRUE, main="Classification Tree for Brand")

text(tree1, use.n=TRUE, all=TRUE, cex=.8)

summary(skutree)

prp(skutree,cex = .9) # Will plot the tree

prp(skutree,varlen=3,cex = .9) #shorten variable names

fancyRpartPlot(skutree)

# Quite accurate but SKU# has 59 levels and the alsorithm can handle only 32 levels

# Work around could be to divide the skus into 2 equal groups

# 1 to 63 and 65 to 131

skutree1 <- rpart(as.factor(`SKU#`) ~ brandARM + brandBNC + brandCLF + brandDWN + brandFNT + brandGEN + brandPRL +

brandSNG + brandSTP + brandTSN+ formB+formF+formL+formS+formula2LT+formula2RG+formula2ST+

formula2UN+sizeLR+sizeMD+sizeSM+sizeXL,data=finaldatav3[finaldatav3$`SKU#` > 1 & finaldatav3$`SKU#` < 64,], method= "class")

prp(skutree1,cex = .9,extra = 2, under = T, branch = .5,xflip = F, box.col = "grey")

summary(skutree1)

skutree2 <- rpart(as.factor(`SKU#`) ~ brandARM + brandBNC + brandCLF + brandDWN + brandFNT + brandGEN + brandPRL +

brandSNG + brandSTP + brandTSN+ formB+formF+formL+formS+formula2LT+formula2RG+formula2ST+

formula2UN+sizeLR+sizeMD+sizeSM+sizeXL,data=finaldatav3[finaldatav3$`SKU#` > 64 & finaldatav3$`SKU#` < 131,], method= "class")

prp(skutree2,cex = .9,extra = 2, under = T, branch = .5,xflip = T, box.col = "grey")

summary(skutree2)

skutree.varimp1 <- rpart(as.factor(`SKU#`) ~ brand+ form +formula2 +size,

data=finaldatav3[finaldatav3$`SKU#` > 1 & finaldatav3$`SKU#` < 64,], method= "class")

prp(skutree.varimp1,cex = .9,extra = 2, under = T, branch = .5,xflip = F, box.col = "grey")

summary(skutree.varimp1)

# Variable importance

# size formula2 brand form

# 32 31 22 15

skutree.varimp2 <- rpart(as.factor(`SKU#`) ~ brand+ form +formula2 +size,

data=finaldatav3[finaldatav3$`SKU#` > 64 & finaldatav3$`SKU#` < 131,], method= "class")

prp(skutree.varimp2,cex = .9,extra = 2, under = T, branch = .5,xflip = T, box.col = "grey")

summary(skutree.varimp2)

# Variable importance

# form size brand formula2

# 34 31 23 12

# A few SKU's have the exact same characteristics as others

# Using Caret Package to get variable importance

####################

## Predictive Models

## Weekly Dataset

# Decision Tree

library("rpart")

week.tree1 <- rpart( as.numeric(sales) ~ as.factor(month) + as.factor(season) + as.factor(brand) +

as.factor(form) + as.factor(formula) + as.factor(size) +

as.factor(display) + as.factor(feature) + as.numeric(APP) +

as.numeric(ARP) +as.numeric(ARSP) +as.numeric(APC) + as.numeric(IRIweek) +

as.numeric(lagsales)+ as.factor(SKU.), data = weeklytrain)

summary(week.tree1$splits)

# Variable importance

# as.numeric(APC) as.factor(SKU.) as.numeric(ARP) as.factor(form) as.numeric(APP)

# 22 19 14 10 8

# as.numeric(lagsales) as.factor(size) as.factor(brand) as.numeric(ARSP) as.factor(month)

# 8 6 5 3 2

# In sample predictive check

weeklytrain[,17] <- predict(week.tree1,weeklytrain)

in.weektree1 <- data.frame(Actual=weeklytrain[,3],Predictedin=weeklytrain[,17])

install.packages("hydroGOF")

library(hydroGOF)

rmse(in.weektree1$Predictedin,in.weektree1$Actual)

# In sample predictive check

weeklytest[,17] <- predict(week.tree1,weeklytest)

out.weektree1 <- data.frame(Actual=weeklytest[,3],Predictedout=weeklytest[,17])

rmse(out.weektree1$Predictedout,out.weektree1$Actual)

# Linear Regression

week.lm1 <- lm(as.numeric(sales) ~ as.factor(month) + as.factor(season) + as.factor(brand) +

as.factor(form) + as.factor(formula) + as.factor(size) +

as.factor(display) + as.factor(feature) + as.numeric(APP) +

as.numeric(ARP) +as.numeric(ARSP) +as.numeric(APC) + as.numeric(IRIweek) +

as.numeric(lagsales)+ as.factor(SKU.) , data = weeklytrain)

summary(week.lm1)

AIC(week.lm1)

BIC(week.lm1)

# In sample predictive check

weeklytrain[,17] <- predict(week.lm1,weeklytrain)

in.weeklm1 <- data.frame(Actual=weeklytrain[,3],Predictedin=weeklytrain[,17])

rmse(in.weeklm1$Predictedin,in.weeklm1$Actual)

# Out of sample predictive check

weeklytest[,17] <- predict(week.lm1,weeklytest)

out.weeklm1 <- data.frame(Actual=weeklytest[,3],Predictedout=weeklytest[,17])

rmse(out.weeklm1$Predictedout,out.weeklm1$Actual)

abline(week.lm1)

# Random Forest

library(randomForest)

week.rtree1 <- randomForest(as.numeric(sales) ~ month+season+brand+

form + formula +size+

display+ feature + APP +

ARP +ARSP +APC + IRIweek +

lagsales+ SKU., data = weeklytrain)

summary(week.rtree1)

week.rtree1

weeklytrain[,17] <- predict(week.rtree1,weeklytrain)

in.weekrtree1 <- data.frame(Actual=weeklytrain[,3],Predictedin=weeklytrain[,17])

rmse(in.weekrtree1$Predictedin,in.weekrtree1$Actual)

# In sample predictive check

weeklytest[,17] <- predict(week.rtree1,weeklytest)

out.weekrtree1 <- data.frame(Actual=weeklytest[,3],Predictedout=weeklytest[,17])

rmse(out.weekrtree1$Predictedout,out.weekrtree1$Actual)

install.packages("graphics")

require(stats)

points(out.weektree1$Actual,out.weektree1$Predictedout)

#SVM

library(e1071)

week.svr1 <- svm(as.numeric(sales) ~ month+season+brand+

form + formula +size+

display+ feature + APP +

ARP +ARSP +APC + IRIweek +

lagsales+ SKU., data = weeklytrain)

summary(week.svr1)

# In sample predictive check

weeklytrain[,17] <- predict(week.svr1,weeklytrain)

in.weeksvr1 <- data.frame(Actual=weeklytrain[,3],Predictedin=weeklytrain[,17])

rmse(in.weeksvr1$Predictedin,in.weeksvr1$Actual)

# Out of sample predictive check

weeklytest[,17] <- predict(week.svr1,weeklytest)

out.weeksvr1 <- data.frame(Actual=weeklytest[,3],Predictedout=weeklytest[,17])

rmse(out.weeksvr1$Predictedout,out.weeksvr1$Actual)

points(out.weeksvr1$Predictedout,out.weeksvr1$Actual, col = "red", pch=4)

####################

## Predictive Models

## HouseHold Dataset

# Decision Tree

library("rpart")

hh.tree1 <- rpart( as.numeric(sales) ~ as.factor(brand) +

as.factor(form) + as.factor(formula) + as.factor(size) +

as.factor(feature) + as.numeric(APP) +

as.numeric(ARP) +as.numeric(ARSP) +as.numeric(APC) + as.numeric(HH\_id) +

as.numeric(lagsales)+ as.factor(`SKU#`), data = hhdatatrainv2)

summary(hh.tree1)

printcp(hh.tree1)

# Variable importance

# as.numeric(APC) as.factor(SKU.) as.numeric(ARP) as.factor(form) as.numeric(APP)

# 22 19 14 10 8

# as.numeric(lagsales) as.factor(size) as.factor(brand) as.numeric(ARSP) as.factor(month)

# 8 6 5 3 2

# In sample predictive check

library(hydroGOF)

hhdatatrainv2[,16] <- predict(hh.tree1,hhdatatrainv2)

in.hhtree1 <- data.frame(Actual=hhdatatrainv2[,4],Predictedin=hhdatatrainv2[,16])

rmse(in.hhtree1$Predictedin,in.hhtree1$Actual)

# In sample predictive check

hhdatatestv2[,16] <- predict(hh.tree1,hhdatatestv2)

out.hhtree1 <- data.frame(Actual=hhdatatestv2[,4],Predictedout=hhdatatestv2[,16])

rmse(out.hhtree1$Predictedout,out.hhtree1$Actual)

# Linear Regression

hh.lm1 <- lm(as.numeric(sales) ~ as.factor(brand) +

as.factor(form) + as.factor(formula) + as.factor(size) +

as.factor(feature) + as.numeric(APP) +

as.numeric(ARP) +as.numeric(ARSP) +as.numeric(APC) + as.numeric(HH\_id) +

as.numeric(lagsales)+ as.factor(`SKU#`), data = hhdatatrainv2)

summary(hh.lm1)

AIC(hh.lm1)

BIC(hh.lm1)

# In sample predictive check

hhdatatrainv2[,16] <- predict(hh.lm1,hhdatatrainv2)

in.hhlm1 <- data.frame(Actual=hhdatatrainv2[,4],Predictedin=hhdatatrainv2[,16])

rmse(in.hhlm1$Predictedin,in.hhlm1$Actual)

# Out of sample predictive check

hhdatatestv2[,16] <- predict(hh.lm1,hhdatatestv2)

out.hhlm1 <- data.frame(Actual=hhdatatestv2[,4],Predictedout=hhdatatestv2[,16])

rmse(out.hhlm1$Predictedout,out.hhlm1$Actual)

# Random Forest

library(randomForest)

hh.rtree1 <- randomForest(sales~brand+form + formula +size+feature + APP + ARP +ARSP +APC + HH\_id +lagsales, data = hhdatatrainv2)

summary(hh.rtree1)

hh.rtree1

# In sample predictive check

hhdatatrainv2[,16] <- predict(hh.rtree1,hhdatatrainv2)

in.hhrtree1 <- data.frame(Actual=hhdatatrainv2[,4],Predictedin=hhdatatrainv2[,16])

rmse(in.hhrtree1$Predictedin,in.hhrtree1$Actual)

# Out of sample predictive check

hhdatatestv2[,16] <- predict(hh.rtree1,hhdatatestv2)

out.hhrtree1 <- data.frame(Actual=hhdatatestv2[,4],Predictedout=hhdatatestv2[,16])

rmse(out.hhrtree1$Predictedout,out.hhrtree1$Actual)

#SVM

library(e1071)

hh.svr1 <- svm(as.numeric(sales) ~ brand+

form + formula +size+

feature + APP +

ARP +ARSP +APC + HH\_id +

lagsales+ `SKU#`, data = hhdatatrainv2)

summary(hh.svr1)

# In sample predictive check

hhdatatrainv2[,16] <- predict(hh.svr1,hhdatatrainv2)

in.hhsvr1 <- data.frame(Actual=hhdatatrainv2[,4],Predictedin=hhdatatrainv2[,16])

rmse(in.hhsvr1$Predictedin,in.hhsvr1$Actual)

# Out of sample predictive check

hhdatatestv2[,16] <- predict(hh.svr1,hhdatatestv2)

out.hhsvr1 <- data.frame(Actual=hhdatatestv2[,4],Predictedout=hhdatatestv2[,16])

rmse(out.hhsvr1$Predictedout,out.hhsvr1$Actual)

points(out.weeksvr1$Predictedout,out.weeksvr1$Actual, col = "red", pch=4)