STATISTICAL DATA MINING

FINAL PROJECT

ON

FABRIC SOFTENER DATASET ANALYSIS

Submission By:

Sree Ranga Vasudha Moda

Part -1

Data Merging and Cleaning

The fabric softener dataset contains a set of data files with various parameters that could Would describe the interaction between the **SKU's** and **Amounts / Purchases**.

Before performing the predictive analysis using the data obtained from the folder, the individual datasets need to be cleaned appropriately and merged based on significant columns.

As per the Instructions provided in the **readme** file, the specific data files have been cleaned as follows:

1. **D1PUR.DAT**

    - Contains the **Household Purchase** history data identified by HH_id.
    - There are two columns HH_id and trip_info where trip_info (AAABBBCCC) needs to be splitted into three parts: **IRI Week, Store # and SKU# Purchased.**

2. **MERCH.DAT**

    - This file contains the Store Environment Information.
    - Contains 5 Fields : **SKU# store# IRIweek price_paid merchandising**

3. **ARSP.DAT**

Contains the Avg. Regular Selling Price of each SKU in each store.

4.**Member ship Panel**

R-CODE:

```
//Set the directory
setwd("C:\\SDM_FabricSoftner");

//-----------DATAONEPUR.DAT--------------------------------------------------------------

//Read the File: DIPUR.DAT and assigning column names

dataone = read.table("DATAONEPUR.DAT",header=F, fill=T,
col.names=c("HH_id","trip_info"),strip.white=T)

//Splitting the trip_info Column into 3 components and eliminating the main column

datatwo=read.fwf(file=textConnection(as.character(dataone$trip_info)), widths=rep(3,3), col.names
=c("IRIweek","storeno","SKUno"))
```

```
dataone$trip_info <- NULL

//Binding the two datasets

Data12bind <- cbind(dataone,datatwo)


//-----------MERCH.DAT---------------------------------------------------------------

//Reading the Notepad File : MERCH.DAT and assigning column names

datathree = read.table("MERCH.DAT",header=F, fill=T, col.names=c("SKUno","storeno","IRIweek",
            "price_paid","merchandising"),strip.white=T)
attach(datathree)


//Splitting the merchandising Column into 2 components : Dep.Price and Temp

datafour=read.fwf(file=textConnection(as.character(datathree$merchandising)), widths=rep(3,2),
        col.names=c("depromoted_price","temp"))


//Further Splitting the Temp Column into 3 components : Ignore,Display and Feature

datafive=read.fwf(file=textConnection(as.character(datafour$temp)), widths=rep(1,3),
        col.names=c("ignore","DISP","FEAT"))

//Binding the Cleaned Datasets

mergeddata <- cbind(datathree,datafour,datafive)

//Before incorporating the final part of the MERCH.DAT contents of readme file,....
//...,we have to load the ARSP.DAT Dataset


//-----------ARSP.DAT---------------------------------------------------------------

//Reading the Notepad File : ARSP.DAT and assigning column names
//strip.white=T Ensures to remove the leading white spaces

arspdataa= read.table("ARSP.DAT",header=F, fill=T, col.names=c("SKU","storeno","ARSP"),
          strip.white=T)


//Going Back to the MERCH.DAT file :

//Changing the values to numeric and merging the datasets by common columns(IRIweek,storeno and
SKUno)

mergeddata$IRIweek <- as.numeric(mergeddata$IRIweek)
Merge2 <- merge(Data12bind,mergeddata,by=c("IRIweek","storeno","SKUno"))

//Reading the Configuration for for processing SKU's and eliminating the S.No Column


atribdata <- read.csv("MembershipPaneldata.csv")
atribdata  <- atribdata [,-1]

//Renaming the column Name and merging the datasets
```

```
colnames(Merge2)[3] <- "SKU"
attribandmergeddata <- merge(Merge2,atribdata ,by=c("SKU"))
```

// Merging the arsp and attribandmergeddata files

```
finaldataset <- merge(arspdata,attribandmergeddata, by=c("SKU","storeno"))
colnames(finaldataset)[7] <- "Regprice"
colnames(finaldataset)[3] <- "AveragePrice"
```

//Computing the Price cut and setting the negative pricecut values to 0 if < 0

```
finaldataset$pricecut<-finaldataset$Regprice-finaldataset$price_paid
finaldataset$pricecut<-ifelse(finaldataset$pricecut<0, 0, finaldataset$pricecut)
```

//Sorting the Final Dataset as per the ascending order of the Column : IRIWeek

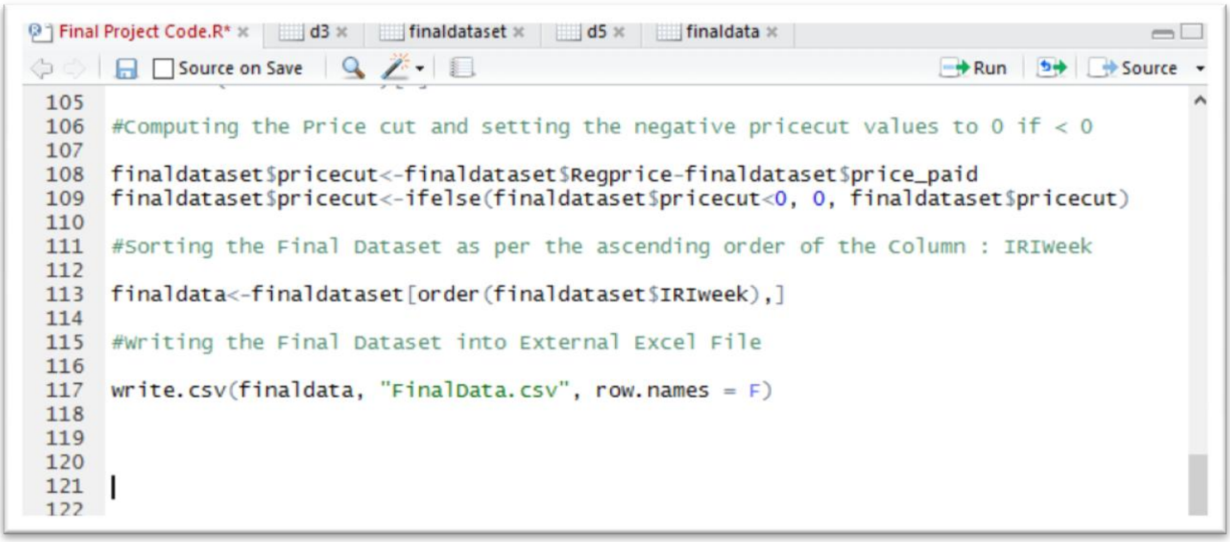
```
finaldata<-finaldataset[order(finaldataset$IRIweek),]
```

//Writing the Final Dataset into External Excel File

```
write.csv(finaldata, "FinalData.csv", row.names = F)
```

Execution in R

```
>
>
> setwd("C:\\SDM_FabricSoftner");
> dataone = read.table("D1PUR.DAT",header=F, fill=T, col.names=c("HH_id","trip_info"),strip.white=T)
> datatwo=read.fwf(file=textConnection(as.character(dataone$trip_info)), widths=rep(3,3),col.names=c("IRIweek","storen","SKUno
"))
> dataone$trip_info <- NULL
> data12binded <- cbind(dataone,datatwo)
> datathree = read.table("MERCH.DAT",header=F, fill=T, col.names=c("SKUno","storeno","IRIweek","price_paid","merchandising"),s
trip.white=T)
> attach(datathree)
> datafour=read.fwf(file=textConnection(as.character(datathree$merchandising)), widths=rep(3,2),
+
+             col.names=c("depromoted_price","temp"))
> datafive=read.fwf(file=textConnection(as.character(datafour$temp)), widths=rep(1,3),col.names=c("ignore","DISP","FEAT"))
> mergeddata <- cbind(datathree,datafour,datafive)
>
```



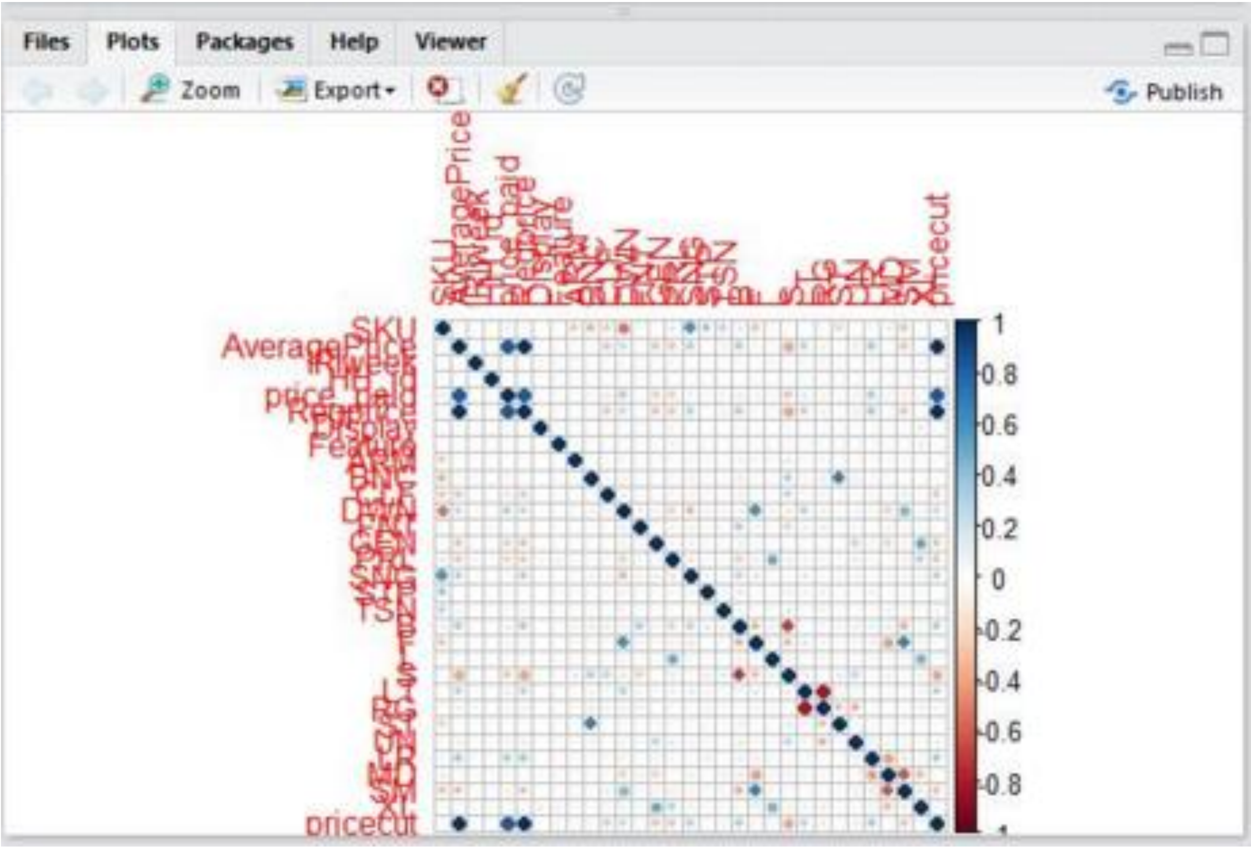

```
#Computing the Price cut and setting the negative pricecut values to 0 if < 0

finaldataset$pricecut<-finaldataset$Regprice-finaldataset$price_paid
finaldataset$pricecut<-ifelse(finaldataset$pricecut<0, 0, finaldataset$pricecut)

#Sorting the Final Dataset as per the ascending order of the Column : IRIweek

finaldata<-finaldataset[order(finaldataset$IRIweek),]

#Writing the Final Dataset into External Excel File

write.csv(finaldata, "FinalData.csv", row.names = F)
```

Analysis & Modelling:

1.)Correlation:

Correlation plots help to give relation between various variables.

```
dataw=cor(FinalFSData,method="Kendall")
corrplot(cor(FinalFSData),method="circle")
```



We have got the final Cleaned DataSet and we can run multiple models on the arrived dataset with combinations of the best suited variables.

Predictive Models:

Let us start the analysis part with the Linear Regression Model.

For IRIWeek>644 ,it is given that the data is forecasted.Hence,for predictive modelling,we are considering the actual data.So,final cleaned data has to be filtered for IRIWeek<=644.

```
> FinalFSData=FinalFSData[IRIweek<=644,]
```

Linear Regression Model:

For CNU being the dependant variable.

Kitchen Sink Model - Including all the variables into the regression

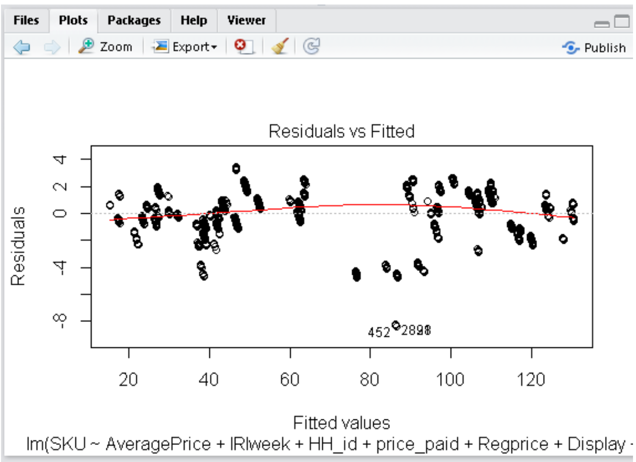
```
kitchenSinkModel2=lm(SKU~AveragePrice+IRIweek+HH_id+price_paid+Regprice+Display+Feature+ARM+BNC+CLF+DWN+FNT+GEN+PRL+SNG+STP+B+F+L+LT+RG+ST+LR+MD+SM)
> summary(kitchenSinkModel2)
```

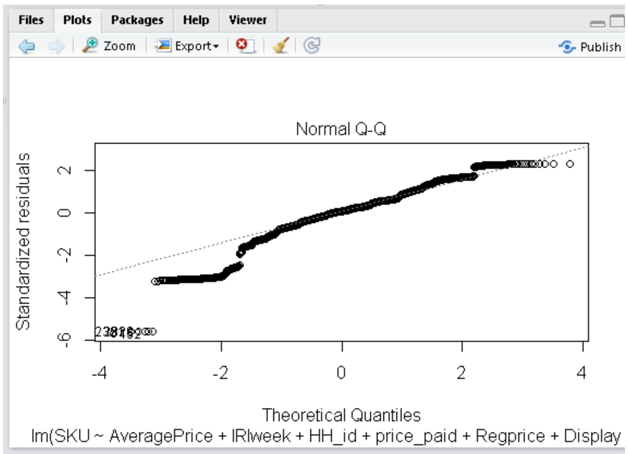
Summaries:

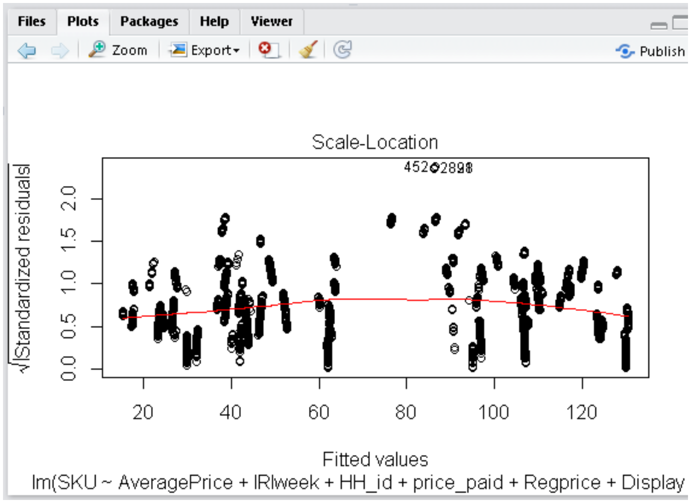
```
Multiple R-squared:  0.9982,   Adjusted R-squared:  0.9982
F-statistic: 1.446e+05 on 25 and 6528 DF,  p-value: < 2.2e-16

> AIC(kitchenSinkModel)
[1] 23810.08
```

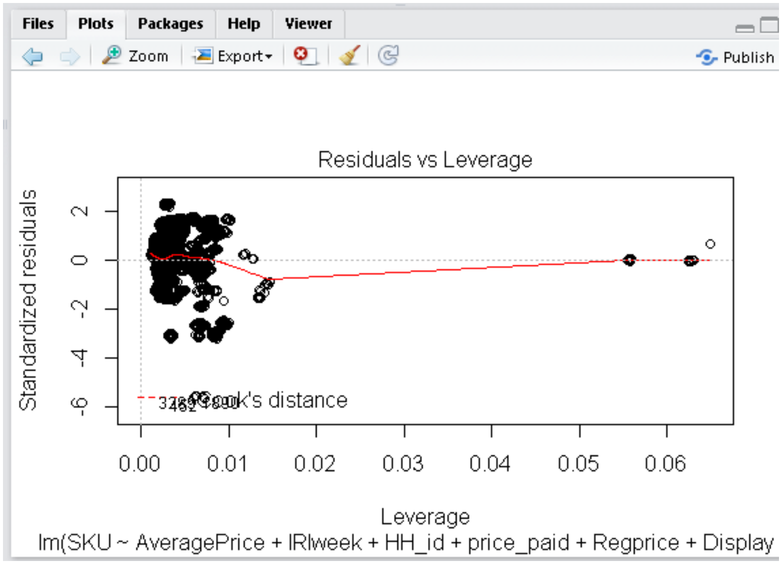
```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1247 on 6529 degrees of freedom
Multiple R-squared:  0.9827,    Adjusted R-squared:  0.9826
F-statistic: 1.546e+04 on 24 and 6529 DF,  p-value: < 2.2e-16

> AIC(kitchenSinkModel)
[1] -8664.94
> |
```

Plots:




Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(SKU ~ AveragePrice + IRIweek + HH_id + price_paid + Regprice + Display

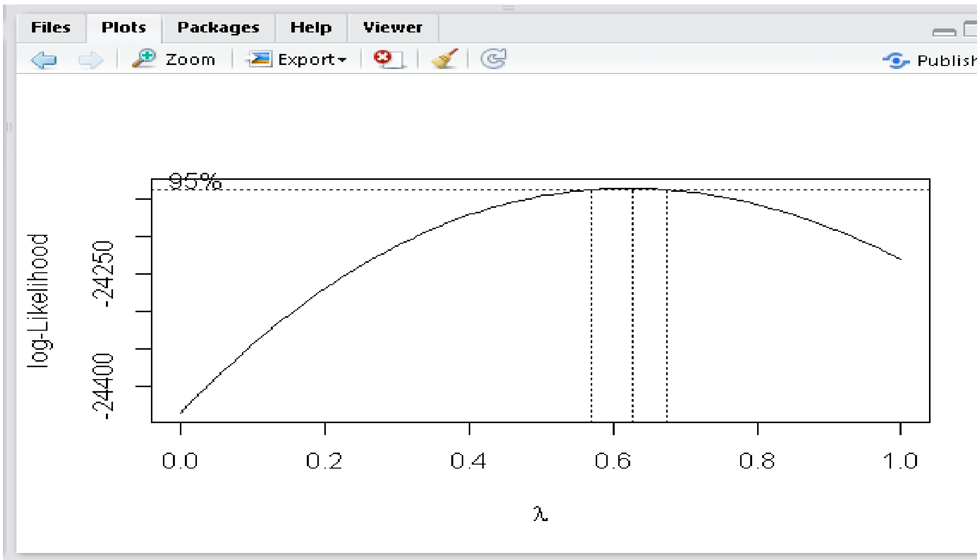

Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Scale-Location
√|Standardized residuals|
Fitted values
lm(SKU ~ AveragePrice + IRIweek + HH_id + price_paid + Regprice + Display


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(SKU ~ AveragePrice + IRIweek + HH_id + price_paid + Regprice + Display

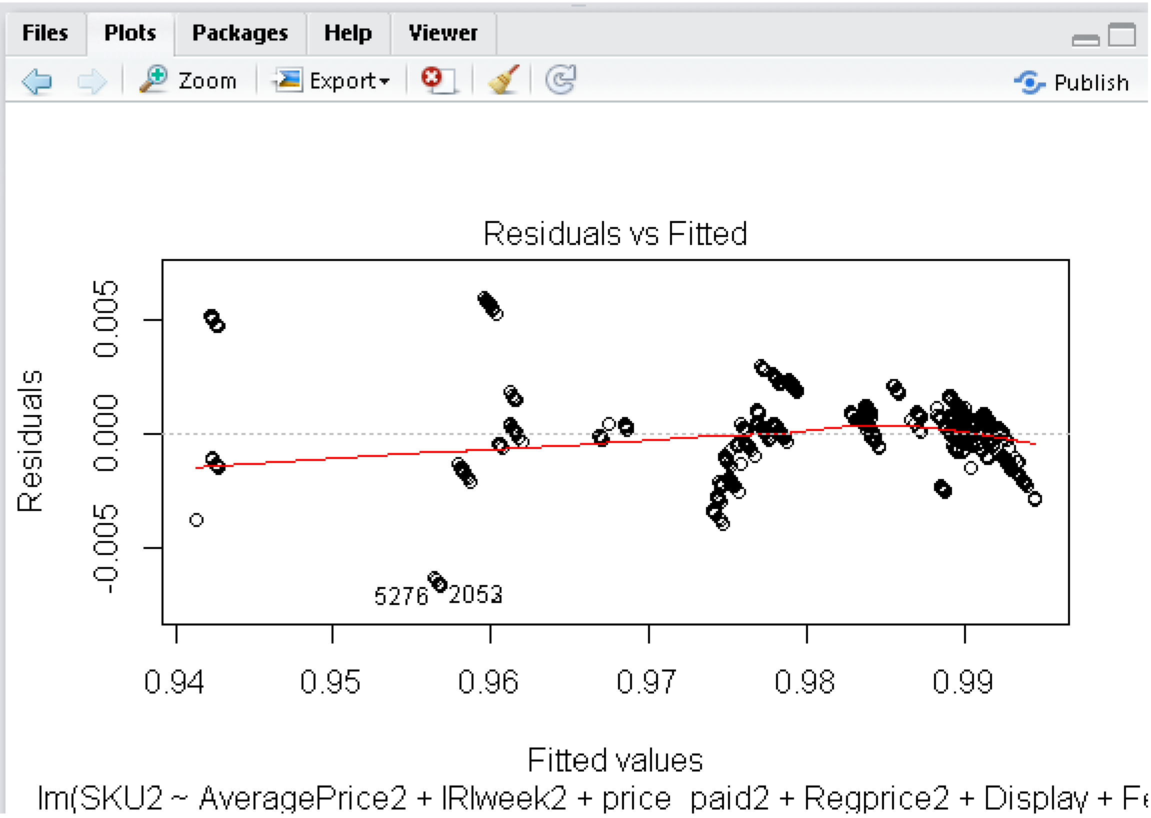
>

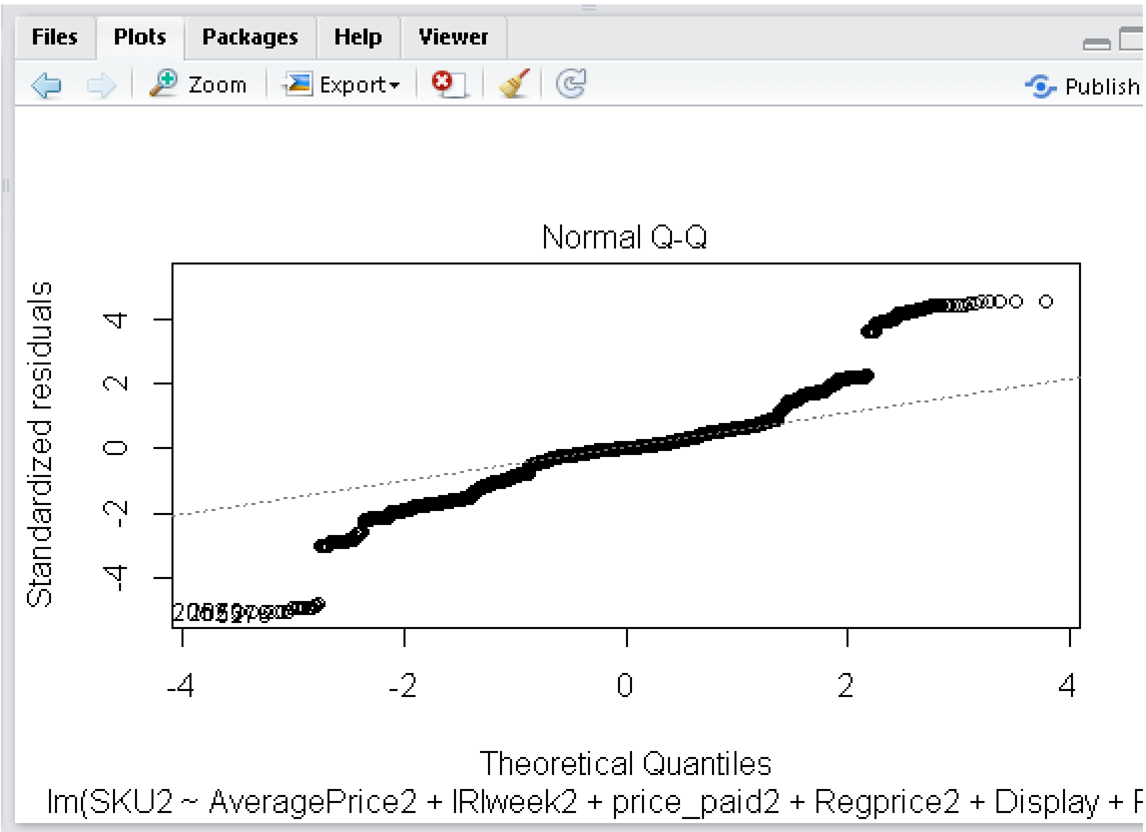
BOX-Cox Model

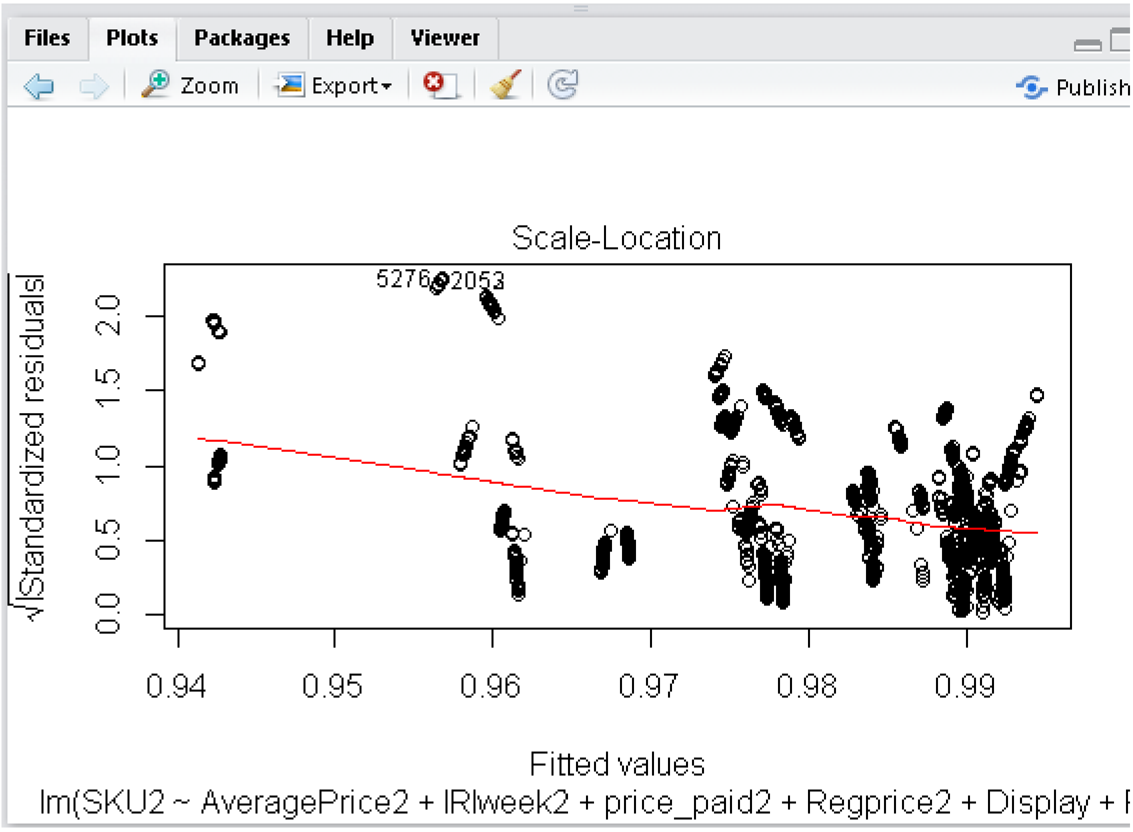
```
> library(forecast)
>
> BoxCox.lambda(AveragePrice)
[1] -0.5772121
> AveragePrice2<- ((AveragePrice^-0.5772121)-1)/-0.5772121
> BoxCox.lambda(SKU)
[1] -0.9999242
> SKU2 <- ((SKU^-0.9999242)-1)/-0.9999242
> BoxCox.lambda(IRIweek)
[1] 1.000001
> IRIweek2 <- ((IRIweek^  1.000001)-1)/  1.000001
> BoxCox.lambda(HH_id)
[1] 1.999924
> HH_id <- ((HH_id^1.999924)-1)/ 1.999924
> BoxCox.lambda(price_paid)
[1] -0.5752739
> price_paid <- ((price_paid^ -0.5752739)-1)/ -0.5752739
>
> >
boxcoxmodel1=lm(SKU2~AveragePrice2+IRIweek2+price_paid2+Regprice2+Display+Feature+ARM+BNC+CLF+DWN+FNT+GEN+PRL+SNG+STP+B+F+L+LT+RG+ST+LR+MD+SM)
> summary(boxcoxmodel1)


Residual standard error: 0.001326 on 6529 degrees of freedom
Multiple R-squared:  0.9859,   Adjusted R-squared:  0.9859
F-statistic: 1.907e+04 on 24 and 6529 DF,  p-value: < 2.2e-16


> AIC(boxcoxmodel1)
[1] -68219.34
```

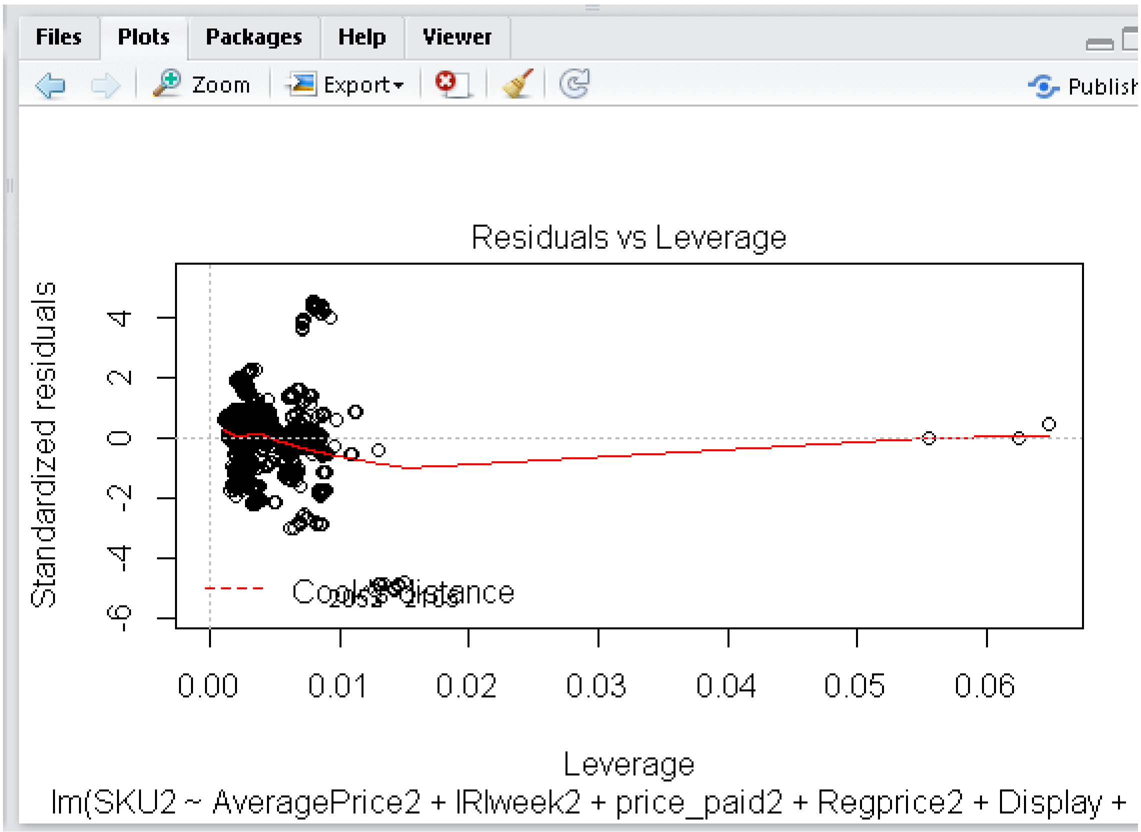


```
Console C:/SDM_FabricSoftner/
> install.packages("zoo")
Installing package into 'C:/Users/VASUDHA/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
Warning in install.packages :
  package 'zoo' is in use and will not be installed
> library(forecast)
>
> BoxCox.lambda(AveragePrice)
[1] -0.5772121
> AveragePrice2<- ((AveragePrice^-0.5772121)-1)/-0.5772121
> BoxCox.lambda(SKU)
[1] -0.9999242
> SKU2 <- ((SKU^-0.9999242)-1)/-0.9999242
> BoxCox.lambda(IRIweek)
[1] 1.000001
> IRIweek2 <- ((IRIweek^ 1.000001)-1)/ 1.000001
> BoxCox.lambda(HH_id)
[1] 1.999924
> HH_id <- ((HH_id^1.999924)-1)/ 1.999924
> BoxCox.lambda(price_paid)
[1] -0.5752739
> price_paid <- ((price_paid^ -0.5752739)-1)/ -0.5752739
> boxcoxmodel1=lm(SKU2~AveragePrice2+IRIweek2+price_paid2+Regprice2+IRIweek2*BNC+IRIweek2*CLF
+IRIweek2*DWN+IRIweek2*FNT+IRIweek2*GEN+IRIweek2*PRL+IRIweek2*SNG+IRIweek2*B+IRIweek2*F)
```




Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Normal Q-Q
Standardized residuals
-4
-2
0
2
4
Theoretical Quantiles
lm(SKU2 ~ AveragePrice2 + IRIweek2 + price_paid2 + Regprice2 + Display + F


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Scale-Location
5276
2053
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.94
0.95
0.96
0.97
0.98
0.99
Fitted values
lm(SKU2 ~ AveragePrice2 + IRIweek2 + price_paid2 + Regprice2 + Display + F



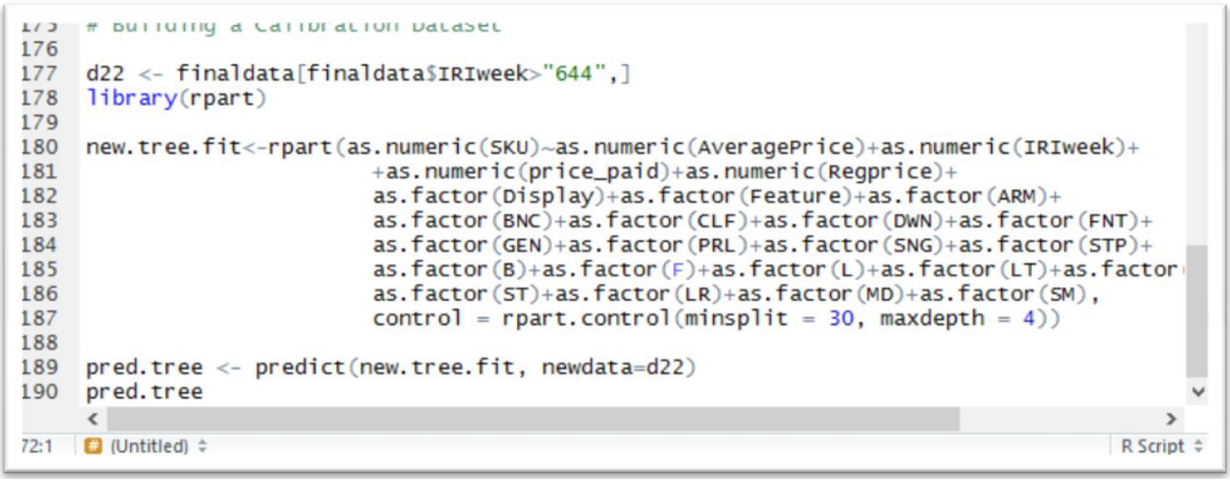
Box Cox model is better than Kitchen sink as AIC value is very less when compared to former one.

Decision Trees for Predictive Models

```
The downloaded binary packages are in
        C:\Users\VASUDHA\AppData\Local\Temp\RtmpyG5idB\downloaded_packages
> library(tree)
>
> summary(tree(as.numeric(SKU)~as.numeric(AveragePrice)+as.numeric(IRIweek)+
+                  +as.numeric(price_paid)+as.numeric(Regprice)+
+                  as.factor(Display)+as.factor(Feature)+as.factor(ARM)+
+                  as.factor(BNC)+as.factor(CLF)+as.factor(DWN)+as.factor(FNT)+
+                  as.factor(GEN)+as.factor(PRL)+as.factor(SNG)+as.factor(STP)+
+                  as.factor(B)+as.factor(F)+as.factor(L)+as.factor(LT)+as.factor(RG)+
+                  as.factor(ST)+as.factor(LR)+as.factor(MD)+as.factor(SM)))

Regression tree:
tree(formula = as.numeric(SKU) ~ as.numeric(AveragePrice) + as.numeric(IRIweek) +
    +as.numeric(price_paid) + as.numeric(Regprice) + as.factor(Display) +
    as.factor(Feature) + as.factor(ARM) + as.factor(BNC) + as.factor(CLF) +
    as.factor(DWN) + as.factor(FNT) + as.factor(GEN) + as.factor(PRL) +
    as.factor(SNG) + as.factor(STP) + as.factor(B) + as.factor(F) +
    as.factor(L) + as.factor(LT) + as.factor(RG) + as.factor(ST) +
    as.factor(LR) + as.factor(MD) + as.factor(SM))
Variables actually used in tree construction:
 [1] "as.factor(SNG)"           "as.factor(STP)"           "as.factor(PRL)"
 [4] "as.numeric(AveragePrice)" "as.factor(SM)"            "as.factor(ARM)"
 [7] "as.factor(CLF)"           "as.factor(B)"             "as.factor(GEN)"
[10] "as.factor(BNC)"
Number of terminal nodes:  11
Residual mean deviance:  33.31 = 218000 / 6543
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-25.49000  -2.76000  -0.08962   0.00000   1.90800  24.50000
>
```

Building a Calibration Dataset

```
library(rpart)

new.tree.fit<-rpart(as.numeric(SKU)~as.numeric(AveragePrice)+as.numeric(IRIweek)+
                    +as.numeric(price_paid)+as.numeric(Regprice)+
                    as.factor(Display)+as.factor(Feature)+as.factor(ARM)+
                    as.factor(BNC)+as.factor(CLF)+as.factor(DWN)+as.factor(FNT)+
                    as.factor(GEN)+as.factor(PRL)+as.factor(SNG)+as.factor(STP)+
                    as.factor(B)+as.factor(F)+as.factor(L)+as.factor(LT)+as.factor(RG)+
                    as.factor(ST)+as.factor(LR)+as.factor(MD)+as.factor(SM),
                    control = rpart.control(minsplit = 30, maxdepth = 4))

pred.tree <- predict(new.tree.fit, newdata=FinalFSData)
pred.tree
```



```
# Building a Calibration Dataset

d22 <- finaldata[finaldata$IRIweek>"644",]
library(rpart)

new.tree.fit<-rpart(as.numeric(SKU)~as.numeric(AveragePrice)+as.numeric(IRIweek)+
                    +as.numeric(price_paid)+as.numeric(Regprice)+
                    as.factor(Display)+as.factor(Feature)+as.factor(ARM)+
                    as.factor(BNC)+as.factor(CLF)+as.factor(DWN)+as.factor(FNT)+
                    as.factor(GEN)+as.factor(PRL)+as.factor(SNG)+as.factor(STP)+
                    as.factor(B)+as.factor(F)+as.factor(L)+as.factor(LT)+as.factor
                    as.factor(ST)+as.factor(LR)+as.factor(MD)+as.factor(SM),
                    control = rpart.control(minsplit = 30, maxdepth = 4))

pred.tree <- predict(new.tree.fit, newdata=d22)
pred.tree
```


Multinomial Regression Analysis

It is the linear regression analysis to conduct if the dependent variable is nominal with more than two levels.

```
d33 <- finaldata[finaldata$IRIweek<=643,]
d33 <- finaldata[finaldata$IRIweek>=600,]

attach(d33)

library(nnet)

multnm <- multinom(SKU ~ AveragePrice+B+LT+SM+MD+LR+ARM+BNC+CLF+
        DWN+FNT+GEN+PRL+SNG+STP, data=d33
```

The Select dataset is taken from **600 – 643**

```r
#Multinomial Regression Analysis

d33 <- finaldata[finaldata$IRIweek<=643,]
d33 <- finaldata[finaldata$IRIweek>=600,]

attach(d33)


library(nnet)

multnm <- multinom(SKU ~ AveragePrice+B+LT+SM+MD+LR+ARM+BNC+CLF+
                   DWN+FNT+GEN+PRL+SNG+STP, data=d33)
```

Output :

```
> multnm <- multinom(SKU ~ AveragePrice+B+LT+SM+MD+LR+ARM+BNC+CLF+
+                    DWN+FNT+GEN+PRL+SNG+STP, data=d33)
# weights:  986 (912 variable)
initial  value 24070.306167
iter  10 value 6072.440840
iter  20 value 3934.689703
iter  30 value 2959.493601
iter  40 value 1715.358942
iter  50 value 1306.170009
iter  60 value 936.254047
iter  70 value 734.846845
iter  80 value 596.086204
iter  90 value 557.715441
iter 100 value 225.047997
final  value 225.047997
stopped after 100 iterations
>
```

Confusion Matrix

```r
#Building the Confusion Matrix

d44 <- finaldata[finaldata$IRIweek>643,]

d44$SKU <- relevel(factor(d44$SKU),ref="50")

confmatrix <- table(predict(multnm),d44$SKU)
```

The ref=50 (The value needs to be set an existing level)