PHASE 3 - PROJECT

Market Basket Analysis

Market basket analysis with Apriori algorithm

The retailer wants to target customers with suggestions on item set that a customer is most likely to purchase. I was given dataset contains data of a retailer; the transaction data provides data around all the transactions that have happened over a period of time. Retailer will use result to grove in his industry and provide for customer suggestions on itemset, we be able increase customer engagement and improve customer experience and identify customer behavior. I will solve this problem with use Association Rules type of unsupervised learning technique that checks for the dependency of one data item on another data item.

Introduction

Association Rule is most used when you are planning to build association in different objects in a set. It works when you are planning to find frequent patterns in a transaction database. It can tell you what items do customers frequently buy together and it allows retailer to identify relationships between the items.

An Example of Association Rules

Assume there are 100 customers, 10 of them bought Computer Mouth, 9 bought Mat for Mouse and 8 bought both of them.

- bought Computer Mouth => bought Mat for Mouse
- support = P(Mouth & Mat) = 8/100 = 0.08
- confidence = support/P(Mat for Mouse) = 0.08/0.09 = 0.89
- lift = confidence/P(Computer Mouth) = 0.89/0.10 = 8.9
  This just simple example. In practice, a rule needs the support of several hundred transactions, before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

Strategy

- Data Import
- Data Understanding and Exploration
- Transformation of the data – so that is ready to be consumed by the association rules algorithm
- Running association rules
- Exploring the rules generated
- Filtering the generated rules
- Visualization of Rule

Dataset Description

- File name: Assignment-1_Data
- List name: retail data
- File format:. xlsx
- Number of Row: 522065

- Number of Attributes: 7
    - Bill No: 6-digit number assigned to each transaction. Nominal.
    - Item name: Product name. Nominal.
    - Quantity: The quantities of each product per transaction. Numeric.
    - Date: The day and time when each transaction was generated. Numeric.
    - Price: Product price. Numeric.
    - Customer ID: 5-digit number assigned to each customer. Nominal.
    - Country: Name of the country where each customer resides. Nominal.

	A	B	C	D	E	F	G
1	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
2	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850	United Kingdom
3	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850	United Kingdom
4	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850	United Kingdom
5	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850	United Kingdom
6	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850	United Kingdom

Libraries in R

First, we need to load required libraries. Shortly I describe all libraries.

- arules - Provides the infrastructure for representing,
  manipulating and analysing transaction data and patterns (frequent itemises and association rules).
- arulesViz - Extends package 'arules' with various visualization.
  techniques for association rules and item-sets. The package also includes several interactive visualizations for rule exploration.
- tidy verse - The tidy verse is an opinionated collection of R packages designed for data science.
- readxl - Read Excel Files in R.
- plyr - Tools for Splitting, Applying and Combining Data.
- ggplot2 - A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.
- knitr - Dynamic Report generation in R.
- magrittr- Provides a mechanism for chaining commands with a new forward-pipe operator, %>%. This operator will forward a value, or the result of an expression, into the next function call/expression. There is flexible support for the type of right-hand side expressions.
- dplyr - A fast, consistent tool for working with data frame like objects, both in memory and out of memory.
- tidyverse - This package is designed to make it easy to install and load multiple 'tidyverse' packages in a single step.

```r
library(arules) #Provides the infrastructure for representing
library(arulesViz) #Extends package 'arules' with various visualization.
library(tidyverse) #The tidyverse is an opinionated collection of  R packages designed for data science.
library(readxl) #Read Excel Files in R.
library(knitr) #Dynamic Report generation in R
library(ggplot2) #A system for 'declaratively' creating graphics,
library(plyr) #Tools for Splitting, Applying and Combining Data.
library(magrittr) #Provides a mechanism for chaining commands with a new forward-pipe operator, %>%.
library(dplyr) #A fast, consistent tool for working with data frame like objects, both in memory and out of memory.
library(tidyverse) #This package is designed to make it easy to install and load multiple 'tidyverse' packages in a single step.
```

Data Pre-processing

Next, we need to upload Assignment-1_Data. xlsx to R to read the dataset.Now we can see our data in R.

```r
#Load excel in R dataframe i named it itemslist
itemslist <- read_excel('/Users/asik/Desktop/Assignment-1_Data.xlsx')
```

Untitled1* | itemslist | Filter

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
1	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
2	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
3	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
4	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
5	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
6	536365	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom
7	536365	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom
8	536366	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
9	536366	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
10	536367	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom
11	536367	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
12	536367	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
13	536367	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	2010-12-01 08:34:00	3.75	13047	United Kingdom
14	536367	IVORY KNITTED MUG COSY	6	2010-12-01 08:34:00	1.65	13047	United Kingdom
15	536367	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	2010-12-01 08:34:00	4.25	13047	United Kingdom
16	536367	BOX OF VINTAGE JIGSAW BLOCKS	3	2010-12-01 08:34:00	4.95	13047	United Kingdom
17	536367	BOX OF VINTAGE ALPHABET BLOCKS	2	2010-12-01 08:34:00	9.95	13047	United Kingdom
18	536367	HOME BUILDING BLOCK WORD	3	2010-12-01 08:34:00	5.95	13047	United Kingdom
19	536367	LOVE BUILDING BLOCK WORD	3	2010-12-01 08:34:00	5.95	13047	United Kingdom
20	536367	RECIPE BOX WITH METAL HEART	4	2010-12-01 08:34:00	7.95	13047	United Kingdom
21	536367	DOORMAT NEW ENGLAND	4	2010-12-01 08:34:00	7.95	13047	United Kingdom
22	536368	JAM MAKING SET WITH JARS	6	2010-12-01 08:34:00	4.25	13047	United Kingdom

After we will clear our data frame, will remove missing values.

```r
#complete.cases(data) removing rows with missing values in any column of data frame
itemslist <- itemslist[complete.cases(itemslist), ]
```

To apply Association Rule mining, we need to convert data frame into transaction data to make all items that are bought together in one invoice will be in one row. Below lines of code will combine all products from one Bill No and Date and combine all products from that Bill No and Date as one row, with each item, separated by (,)

```
#ddply(dataframe, variables_to_split_dataframe, function)
transaxtionData <- ddply(itemslist,c("BillNo","Date"),
                         function(df1)paste(df1$Itemname,
                                            collapse = ","))
```

We don't need Bill No and Date, we will make it as Null.
Next, you have to store this transaction data into .csv

```
transaxtionData$BillNo <- NULL
transaxtionData$Date <- NULL
#will gave the name to column "item"
colnames(transaxtionData) <- c("items")
```

This how should look transaction data before we will go to next step.

```
#quote: If TRUE it will surround character or factor column with double quotes.
#If FALSE nothing will be quoted
#row.names: either a logical value indicating whether the row names of x are to be
#written along with x, or a character vector of row names to be written.
write.csv(transaxtionData, "assigment1_itemslist.csv", quote = FALSE, row.names = FALSE)
```

items			
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	KNITTED UNION FLAG HOT WATER BOTTLE
HAND WARMER UNION JACK	HAND WARMER RED POLKA DOT		
ASSORTED COLOUR BIRD ORNAMENT	POPPY'S PLAYHOUSE BEDROOM	POPPY'S PLAYHOUSE KITCHEN	FELTCRAFT PRINCESS CHARLOTTE DOLL
JAM MAKING SET WITH JARS	RED COAT RACK PARIS FASHION	YELLOW COAT RACK PARIS FASHION	BLUE COAT RACK PARIS FASHION
BATH BUILDING BLOCK WORD			
ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	ALARM CLOCK BAKELIKE GREEN	PANDA AND BUNNIES STICKER SHEET
PAPER CHAIN KIT 50'S CHRISTMAS			
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK		
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	EDWARDIAN PARASOL RED
VICTORIAN SEWING BOX LARGE			
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	EDWARDIAN PARASOL RED
HOT WATER BOTTLE TEA AND SYMPATHY	RED HANGING HEART T-LIGHT HOLDER		
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK		
JUMBO BAG PINK POLKADOT	JUMBO BAG BAROQUE BLACK WHITE	JUMBO BAG CHARLIE AND LOLA TOYS	STRAWBERRY CHARLOTTE BAG
JAM MAKING SET PRINTED			
RETROSPOT TEA SET CERAMIC 11 PC	GIRLY PINK TOOL SET	JUMBO SHOPPER VINTAGE RED PAISLEY	AIRLINE LOUNGE

At this step we already have our transaction dataset, and it shows the matrix of items which bought together. We can't see here any rules and how often it was purchase together. Now let's check how many transactions we have and what they are. We will have to have to load this transaction data into an object of the transaction class. This is done by using the R function read. Transactions of the arules package. Our format of Data frame is basket.

```r
transactions <- read.transactions('/Users/asik/Desktop/assigment1_itemslist.csv',
                                  format = 'basket', sep=',')
```

Let’s have a view our transaction object by summary(transaction)

```r
summary(transactions)
```

We can see 18193 transactions (rows) and 7698 items (columns). 7698 is the product descriptions and 18193 transactions are collections of these items.

```
transactions as itemMatrix in sparse format with
 18193 rows (elements/itemsets/transactions) and
 7698 columns (items) and a density of 0.002291294

most frequent items:
WHITE HANGING HEART T-LIGHT HOLDER           REGENCY CAKESTAND 3 TIER           JUMBO BAG RED RETROSPOT
                              1718                               1468                              1395
                     PARTY BUNTING      ASSORTED COLOUR BIRD ORNAMENT                           (Other)
                              1245                               1226                            313843

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27
1546  860  744  743  743  696  642  633  632  566  598  517  494  520  533  508  460  428  468  406  385  307  306  267  232  246  226
  28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54
 210  213  209  164  153  135  140  131  108  109   88  108   90   86   84   84   63   58   67   59   58   57   48   60   39   39   47
  55   56   57   58   59   60   61   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79   80   81
  41   35   27   37   29   26   27   16   24   25   20   27   24   23   13   20   19   13   16   15   11   15   12    6    7   14   13
  82   83   84   85   86   87   88   89   90   91   92   93   94   95   96   97   98   99  100  101  102  103  104  105  106  107  108
  10    8    8   11   10   13    8    6    5    5   11    5    4    4    3    5    5    2    4    1    4    4    2    2    2    6    3
 109  110  111  112  113  114  116  117  118  120  121  122  123  125  126  127  131  132  133  134  140  141  142  143  145  146  147
   4    3    2    1    3    1    3    3    3    1    2    2    1    3    2    2    1    1    2    1    1    2    2    1    1    2    1
 150  154  157  168  171  177  178  180  182  202  204  228  249  250  285  320  400  419
   1    3    2    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    5.00   13.00   17.64   23.00  419.00

includes extended item information - examples:
                      labels
1                   1 HANGER
2      10 COLOUR SPACEBOY PEN
3 12 COLOURED PARTY BALLOONS
```

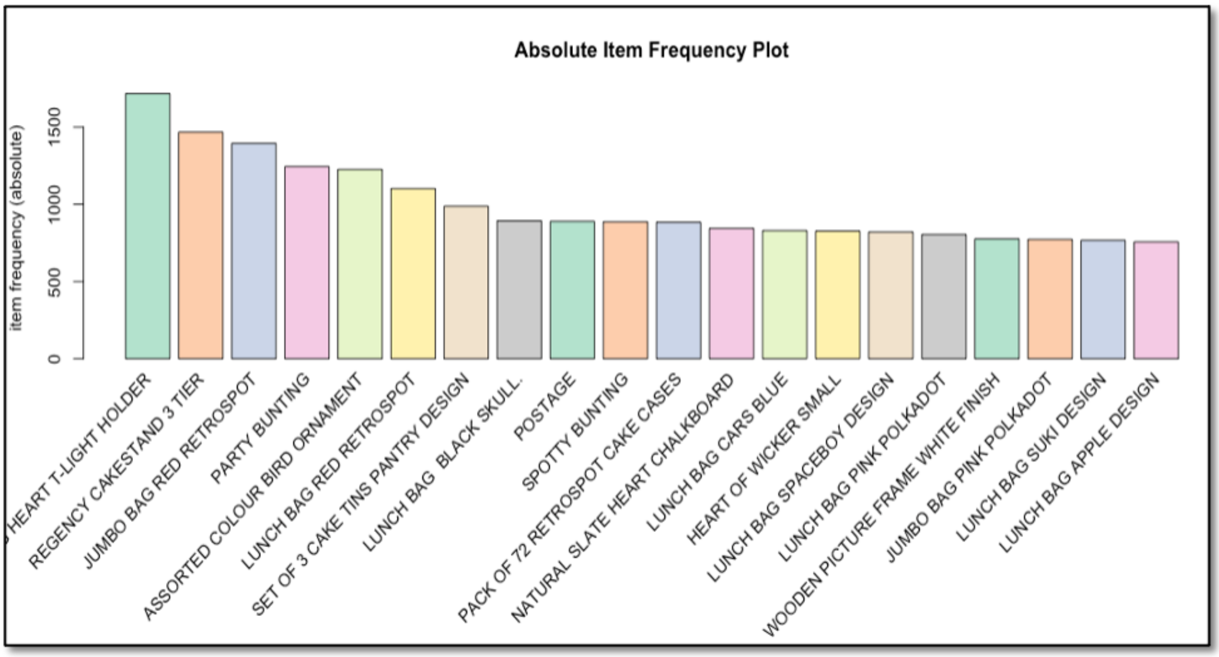
The summary gives us some useful information:

- Density tells the percentage of non-zero cells in a sparse matrix. In other words, total number of items that are purchased divided by a possible number of items in that matrix. You can calculate how many items were purchased by using density: 18193x7698x0.002291294=337445
- Summary will show us most frequent items.
- Element (itemset/transaction) length distribution: It will gave us how many transactions are there for 1-itemset, 2-itemset and so on. The first row is telling you a number of items and the second row is telling you the number of transactions.
  For example, there is only 1546 transaction for one item, 860 transactions for 2 items, and there are 419 items in one transaction which is the longest.

Let’s check item frequency plot, we will generate an itemFrequencyPlot to create an item Frequency Bar Plot to view the distribution of objects based on itemMatrix (e.g., >transactions or items in >itemsets and >rules) which is our case.

```r
itemFrequencyPlot(transactions,topN=20,type="absolute",
                  col=brewer.pal(8,'Pastel2'), main="Absolute Item Frequency Plot")
```

```
if (!require("RColorBrewer")) {install.packages("RColorBrewer")
  library(RColorBrewer)
```



In item Frequency Plot (transaction,topN=20,type="absolute") first argument - our transaction object to be plotted that is tr. Top N is allows us to plot top N highest frequency items. type can be as type="absolute" or type="relative". If we will choose absolute, it will plot numeric frequencies of each item independently. If relative it will plot how many times these items have appeared as compared to others. As well I made it in colure for better visualization.

Generating Rules

Next, we will generate rules using the Priory algorithm. The function apriority() is from package a rules. The algorithm employs level-wise search for frequent item sets. Algorithm will generate frequent item sets and association rules. We pass sup =0.001 and conf =0.8 to return all the rules that have a support of at least 0.1% and confidence of at least 80%. We sort the rules by decreasing confidence and will check summary of the rules.

The apriority will take (transaction) as the transaction object on which mining is to be applied. parameter will allow you to set min_sup and min_confidence. The default values for parameter are minimum support of 0.1, the minimum confidence of 0.8, maximum of 10 items (maxlen).

```
set of 97267 rules

rule length distribution (lhs + rhs):sizes
    2     3     4     5     6     7     8     9    10
  111  3146 10141 27586 33296 17263  4634   933   157

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   5.000   6.000   5.714   6.000  10.000

summary of quality measures:
    support           confidence        coverage             lift             count
 Min.   :0.001044   Min.   :0.8000   Min.   :0.001044   Min.   :  8.472   Min.   : 19.00
 1st Qu.:0.001099   1st Qu.:0.8333   1st Qu.:0.001209   1st Qu.: 18.833   1st Qu.: 20.00
 Median :0.001209   Median :0.8750   Median :0.001374   Median : 24.059   Median : 22.00
 Mean   :0.001378   Mean   :0.8861   Mean   :0.001563   Mean   : 50.882   Mean   : 25.06
 3rd Qu.:0.001484   3rd Qu.:0.9286   3rd Qu.:0.001704   3rd Qu.: 41.754   3rd Qu.: 27.00
 Max.   :0.021492   Max.   :1.0000   Max.   :0.026439   Max.   :673.815   Max.   :391.00

mining info:
 data ntransactions support confidence
   tr         18193   0.001        0.8
```

Summary of rules give us clear information as:

- Number of rules: 97267
- The distribution of rules by length: a length of 6 items has the most 33296 and length of 2 items has lowest number of rules 111
- The summary of quality measures: ranges of support, confidence, and lift.
- The information on data mining: total data mined, and the minimum parameters we set earlier

Now, 97267 it a lot of rules. We will identify only top 10.

```
45  inspect(generated.rules[1:10])
```

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{WOBBLY CHICKEN}	=>	{DECORATION}	0.001484087	1	0.001484087	371.2857	27
[2]	{WOBBLY CHICKEN}	=>	{METAL}	0.001484087	1	0.001484087	371.2857	27
[3]	{BILLBOARD FONTS DESIGN}	=>	{WRAP}	0.001374155	1	0.001374155	673.8148	25
[4]	{DECOUPAGE}	=>	{GREETING CARD}	0.001154290	1	0.001154290	336.9074	21
[5]	{BLACK TEA}	=>	{SUGAR JARS}	0.002088715	1	0.002088715	256.2394	38
[6]	{BLACK TEA}	=>	{COFFEE}	0.002088715	1	0.002088715	65.6787	38
[7]	{WOBBLY RABBIT}	=>	{DECORATION}	0.001868851	1	0.001868851	371.2857	34
[8]	{WOBBLY RABBIT}	=>	{METAL}	0.001868851	1	0.001868851	371.2857	34
[9]	{FUNK MONKEY}	=>	{ART LIGHTS}	0.002033749	1	0.002033749	491.7027	37
[10]	{ART LIGHTS}	=>	{FUNK MONKEY}	0.002033749	1	0.002033749	491.7027	37

Using the above output, you can make analysis such as:

- 100% of the customers who bought 'ART LIGHTS ' also bought 'FUNK MONKEY'.

- 100% of the customers who bought 'BILLBOARD FONTS DESIGN ' also bought 'WRAP'.
  We can limit the size and number of rules generated. we can set parameter in Apriori. If we want stronger rules, we must to increase the value of conf. and for more extended rules give higher value to maxlen.

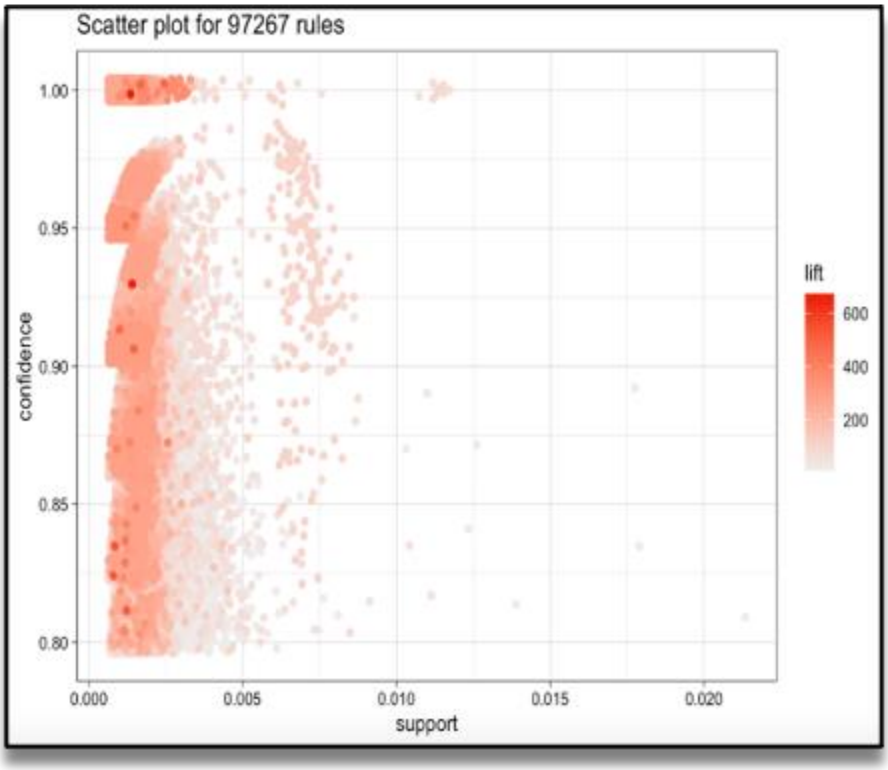
Visualizing Association Rules

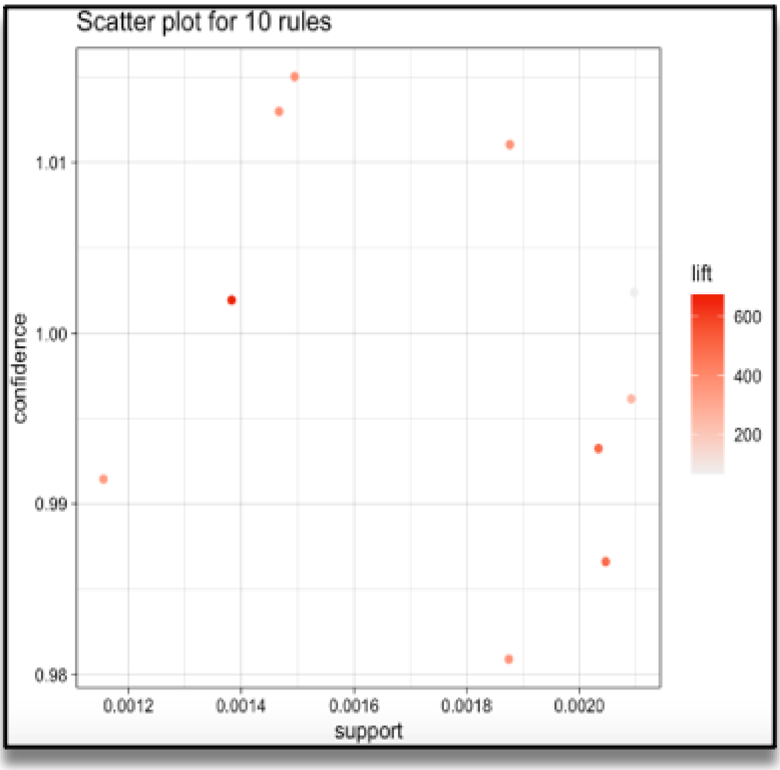
We have thousands of rules generated based on data, we will need a couple of ways to present our findings. We will use ItemFrequencyPlot to visualize association rules.

Scatter-Plot:

```
# Filter rules with confidence greater than 0.6 or 60%
Rules<-generated.rules[quality(generated.rules)$confidence>0.6]
#Plot Rules
plot(Rules)
top10Rules <- head(generated.rules, n = 10, by = "confidence")
plot(top10Rules)
```

A straight-forward visualization of association rules is to use a scatter plot using plot() of the a rules Viz package. It uses Support and Confidence on the axes. In addition, third measure Liftis used by default to colour (grey levels) of the points.

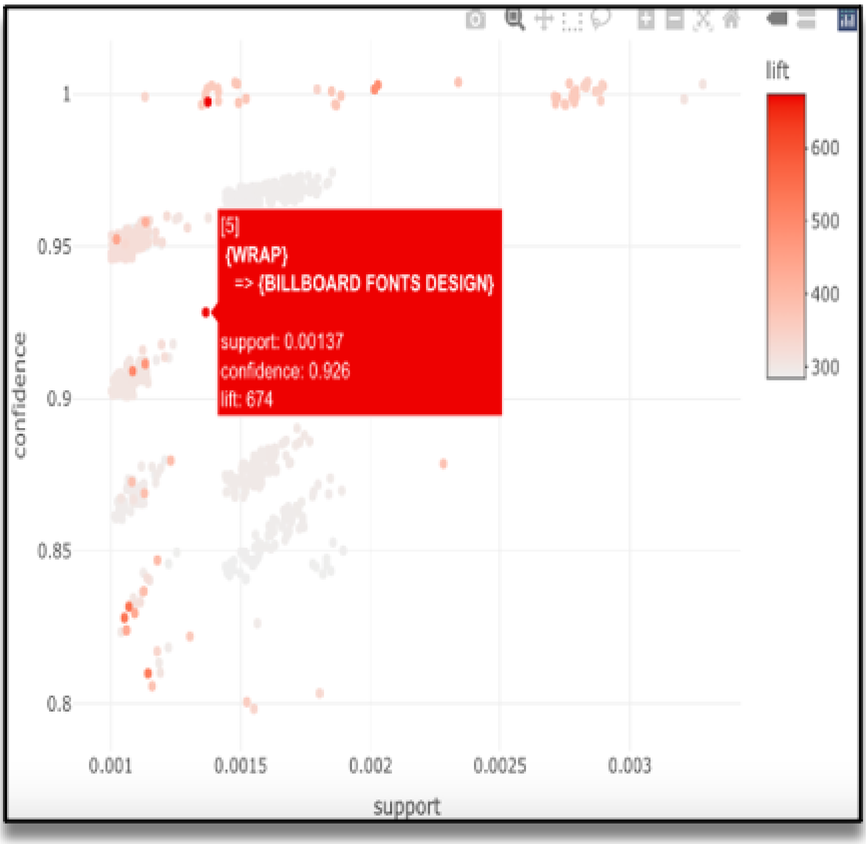


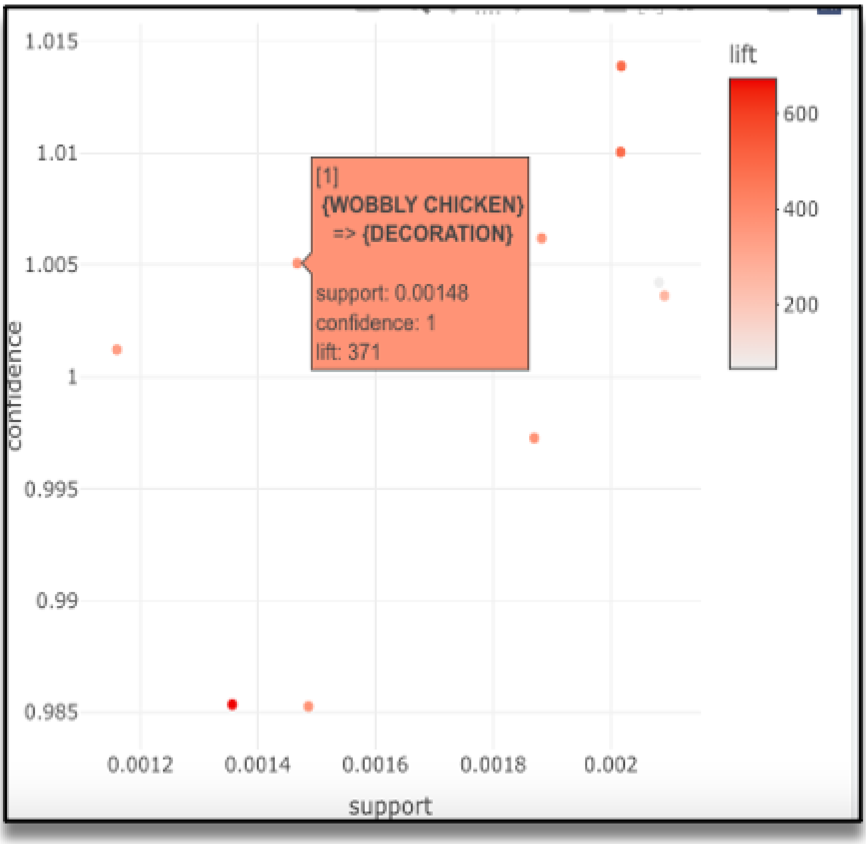


Interactive Scatter-Plot:

We can have a look for each rule (interactively) and view all quality measures (support, confidence and lift).

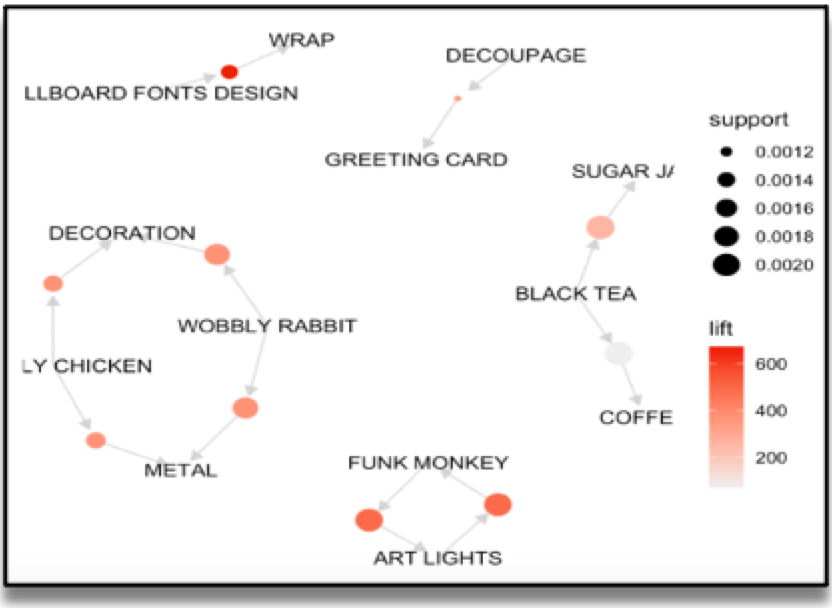
```
plot(Rules, engine = "plotly")
plot(top10Rules, engine = "plotly")
```

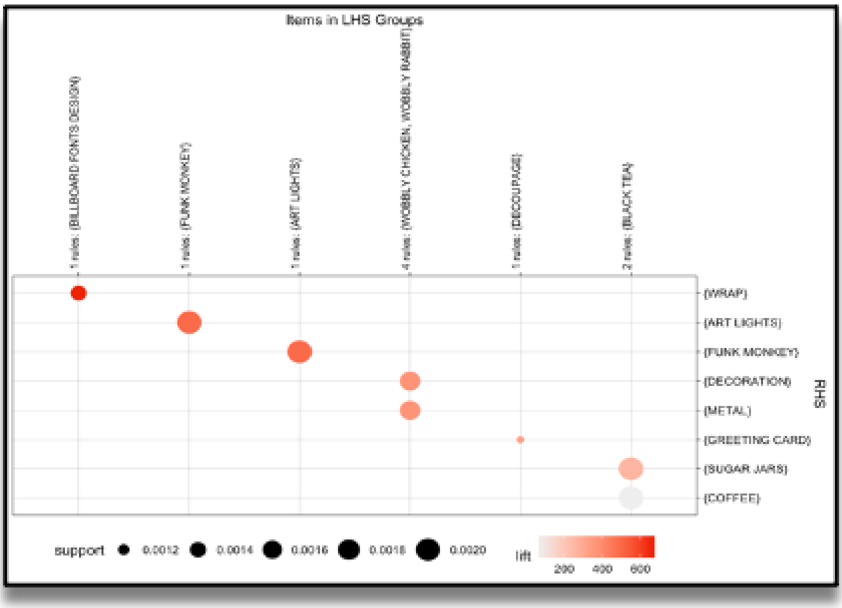




Graph - Based Visualization and Group Method:

Graph plots are a great way to visualize rules but tend to become congested as the number of rules increases. So, it is better to visualize a smaller number of rules with graph-based visualizations. We can see as well group method for top 10 items.





Conclusion

Based on the results of these calculations can be used as a recommendation for retail owners to arrange the arrangement of product catalogs and take strategic steps to improve product marketing.. By utilizing the association rules which are discovered as a result of the analyses, the retailer can apply effective marketing and sales promotion strategies, he will be able increase customer engagement and improve customer experience and identify customer behavior.