SAINT MARY'S UNIVERSITY SINCE 1802

One University. One World. Yours.

MSc in Computing and Data Analytics

MCDA 5580 – Data and Text Mining

Assignment – 3

Submitted to:

Trishla Shah

Prepared by:

Allen Mathew - A00432526

Gyaneshwar Rao - A00433014

Meghashyam - A00432392

Table of Contents

Executive Summary

The business has huge amount of transactions of the customers, which can be used to understand purchase patterns of the customers. The association between the products purchase would help to boost the revenue of the business. By considering the transactions and the items purchased, the analysis of the data revealed insights, which has projected that, the purchase of few products always influences the purchase of their associated products. This insight would help the business to have the relevant items in stock, and the placement of the product in the store to boost the sales by allowing the customer to go through other products. 23 rules were generated with a lift range from 1 to 10 and a confidence of 50% purchase with a support of 1%.

Objective

The complete dataset "OnlineRetail" will be taken and cleaned to eliminate the improper records from going through the analysis. The filtering/cleaning of the data will be explained in the section "Data preparation". From the cleaned data set, only the columns with Invoice number and all the stock code descriptions from the dataset will be retrieved. The extracted data will be transformed using "ddply" function in R which helps us to flatten the data and get all the distinct records in a row per invoice number which will help us for further analysis. "Apriori" algorithm will be used to generate the association rules for the cleaned and transformed dataset using the function "Apriori" in R. Association rules will be generated by configuring the optimal values for the constraints to parameters like Support, Confidence and Lift until a small number of association rules are generated which will be meaningful and interesting to users. Maximally frequent itemset is derived from the rules and projected as the most associated itemset with a maximum frequency of occurrence in the baskets.

Data Summary

For our analysis, we will be using the “OnlineRetail” dataset. It consists of 541,909 records which are from December 2011 to December 2012, of the various products bought by customers from multiple countries. The following are the attributes of the data set:

Attributes	Description
Invoice Number (InvoiceNo)	It is generally a 6-digit number that uniquely identifies the transaction made by a customer.
Product Item ID (StockCode)	The code is alphanumeric, consisting of 1 - 6 characters. It is used to uniquely identifies the product item.
Description	It is text that is used to describe the stock code. It mainly provides details for the product item.
Quantity	It indicates the number of products brought or returned by the customer.
Unit Price	It is a positive float, that indicates the cost of a single product. But for some records the Unit Price is negative, it was done to “adjust for bad debt” as mentioned in the Description.
CustomerID	It is generally a 6-digit number that uniquely identifies a customer.
Country	It is text that describes the location where the product was bought.
InvoiceDateTime	It consists of the date and time when the product was purchased. In general, the records are between December 2011 to December 2012.

Table 1:Attributes of the “OnlineRetail” Dataset

Observations:

The following table briefly describes the observation made on each of the attributes in the "OnlineRetail" data set:

Attributes	Description
Invoice Number	• All records follow the same pattern, i.e. The code is numeric, consisting of 5 characters. • There are 9,292 records containing Invoice Number 0. • Majority of the records with Invoice Number 0 have a negative value for Quantity.
StockCode	• The code follows 2 distinct patterns: 1. The code is numeric, consisting of 5 characters. 2. The code is alphanumeric, consisting of 5 numeric characters and a single letter. • Apart from the above-mentioned patterns there are 15 unique stock codes that don't follow the above patterns, but they are used to indicate Discount, Bank Charges, Amazon Fee, Samples, Postage, etc.
Description	• For most of the records it displays the title of the product. • It provides describes for the 15 unique stock codes as mentioned in the previous observation. • There are 1,454 records that have no description.
Quantity	• There are 10,624 records that have a negative value. • The Maximum Quantity of a product brought and not returned by a customer is 12,540
Unit Price	• There is 1 record that has a negative value. • The records are between 0.00 to 9.99.
CustomerID	• All records follow the same pattern, i.e. The code is numeric, consisting of 6 characters. • There are 135,080 records containing Customer ID 0. • There are 1,719 records with Customer ID 0 and have a negative value for Quantity. • There are 386 records with Customer ID 0 and Invoice No 0.
Country	• There are 38 distinct counties in the dataset. • Majority of the purchases is done in the United Kingdom. • The least number of purchases is done in Lebanon, RSA and Brazil.
InvoiceDateTime	• All records follow the same pattern of when the product was purchased. • The records are between December 2011 to December 2012.

Table 2:Observations made on the Attributes of the "OnlineRetail" Dataset

Limitations:

Invoice Number 0, CustomerID 0 and a few StockCode items are not clearly defined in the data set. These attributes are interlinked with other attributes in the dataset like Quantity, UnitPrice, etc.

Also, there is no common identifier to remove the product that was returned since each transaction ID is unique for a visit/ one trip to the store

Outcome:

From the "OnlineRetail" dataset we will create a dataset or tables in MySQL called **temp.** The table will be utilized the following attributes:

1. **Invoice Number**
2. **Description**

The **temp** table will consist of all the records/transactions where consider from the "OnlineRetail" dataset except for the one where the product was returned. To do so the following cases needs to be considered for cleaning up the data(assumptions):

- Invoice Number 0 mainly contains data on the items that were returned by the customer,
- Customer 0 has many transactions compared to the rest which is unusual
- StockCode "POST" appears frequently in many transactions
- UnitPrice and Quantity needs to be greater than 0

To prevent the above cases from causing problem in the analysis we will be removing all records where:

- CustomerID is equal to 0
- InvoiceNo is equal to 0
- StockCode is equal to "POST"
- `Quantity` And `UnitPrice` are less than 0

The SQL command used to create the table **temp** along with the necessary conditions to filter the dataset is give in the Appendix.

Association Mining

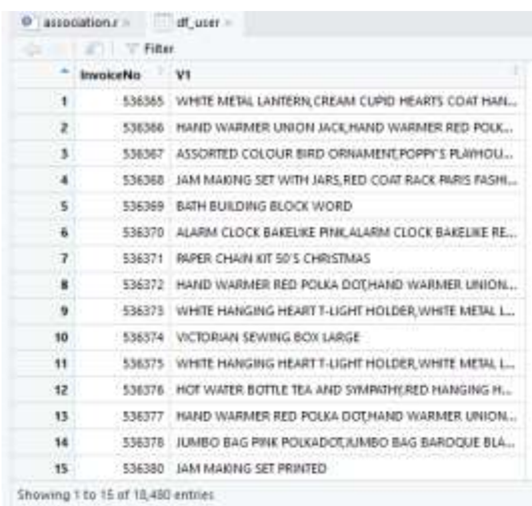
To discover the association between items, large retailers often use Market Basket Analysis. It helps to identify the relationship between the combination of products that occur frequently in a transaction. In simpler terms, Association Mining is the process of defining rules and finding out the likelihood of a purchase/event to occur based on the occurrence of another purchase/event. (Li, 2019)

The "OnlineRetail" Dataset will be used to generate Association Rules to identify the relationship between the combination of products bought in every transaction.

1. Transforming the Dataset

Currently, the dataset is in a dataframe format where each row consists of a transaction number ("InvoiceNo") and an individual product bought in that transaction ("Description"). The dataset needs to be transformed in such a way that each row contains all the products that were brought in a unique transaction.

The data frame is passed to the ddply() function, it then creates subsets based on the InvoiceNo variable. Then a function is applied to return the new data frame that combine all the Descriptions of various products to the particular InvoiceNo variable, each product Description is separated by a comma(.i.e. ","). (Rdocumentation.org, 2019)


association.r df_user
Filter

	InvoiceNo	V1
1	536365	WHITE METAL LANTERN,CREAM CUPID HEARTS COAT HAN...
2	536366	HAND WARMER UNION JACK,HAND WARMER RED POLK...
3	536367	ASSORTED COLOUR BIRD ORNAMENT,POPPY'S PLAYHOU...
4	536368	JAM MAKING SET WITH JARS,RED COAT RACK PARIS FASHI...
5	536369	BATH BUILDING BLOCK WORD
6	536370	ALARM CLOCK BAKELIKE PINK,ALARM CLOCK BAKELIKE RE...
7	536371	PAPER CHAIN KIT 50'S CHRISTMAS
8	536372	HAND WARMER RED POLKA DOT,HAND WARMER UNION...
9	536373	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL L...
10	536374	VICTORIAN SEWING BOX LARGE
11	536375	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL L...
12	536376	HOT WATER BOTTLE TEA AND SYMPATHY,RED HANGING H...
13	536377	HAND WARMER RED POLKA DOT,HAND WARMER UNION...
14	536378	JUMBO BAG PINK POLKADOT,JUMBO BAG BAROQUE BLA...
15	536380	JAM MAKING SET PRINTED

Showing 1 to 15 of 18,480 entries

Figure 1: Successfully transformed the data frame

The “InvoiceNo” column is then removed since only the product “Description” for every transaction is required. For further analysis, the new data frame is written into a *CSV* file (*Milestones2.csv*).

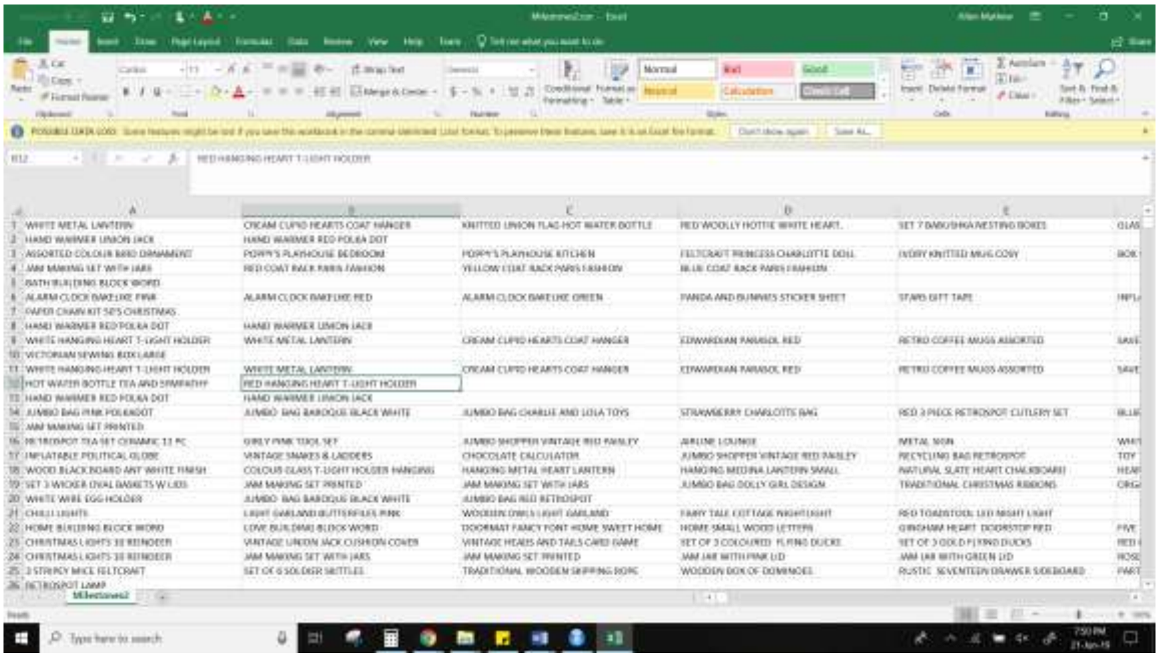


Figure 2: Successfully created the transaction dataset (Milestones2.csv)

The above image shows the transaction dataset that consists of all the products that where brought in every transaction.

2. Analysis

Further analysis is done on the transaction's dataset. To do so, the above *CSV* file is read and stored as a variable in R. The summary function is then used, to have a better understanding of the transaction's dataset.

```
> summary(tr)
transactions as itemMatrix in sparse format with
 18480 rows (elements/itemsets/transactions) and
 7790 columns (items) and a density of 0.002278257

most frequent items:
WHITE HANGING HEART T-LIGHT HOLDER           REGENCY CAKESTAND 3 TIER
                              1760                               1532
           JUMBO BAG RED RETROSPOT                      PARTY BUNTING
                              1418                               1267
     ASSORTED COLOUR BIRD ORNAMENT                            (Other)
                              1240                             320759

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
1557  847  761  771  744  704  642  644  656  584  599  534  495  512  551  520  453  441
  19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36
 482  412  385  312  305  262  240  250  229  217  223  211  160  164  135  139  139  102
  37   38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54
 115   86  113   91   92   87   89   66   60   69   61   63   54   50   63   42   42   46
  55   56   57   58   59   60   61   62   63   64   65   66   67   68   69   70   71   72
  44   37   29   37   32   27   27   18   24   25   20   26   24   22   16   20   18   14
  73   74   75   76   77   78   79   80   81   82   83   84   85   86   87   88   89   90
  15   16   11   15   12    7    9   14   15   12    9    9   10   11   14    8    7    4
  91   92   93   94   95   96   97   98   99  100  101  102  103  104  105  106  107  108
   7   10    6    4    4    4    5    5    2    4    2    4    4    3    2    2    6    3
 109  110  111  112  113  114  116  117  118  120  121  122  123  125  126  127  131  132
   4    3    2    1    3    1    3    3    3    1    2    2    1    3    2    2    1    1
 133  134  140  141  142  143  145  146  147  149  154  157  168  169  171  177  178  180
   2    1    1    2    2    1    1    2    1    1    3    2    2    1    1    1    1    1
 202  204  228  236  249  250  285  320  400  419
   1    1    1    1    1    1    1    1    1    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    5.00   13.00   17.75   23.00  419.00

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

Figure 3: Summary of the transaction's dataset

Observation:

<table>
<tr><td>Transactions:</td><td>Total Number of Transactions: 18480</td></tr>
<tr><td>Items:</td><td>Total Number Items: 7790</td></tr>
<tr><td>Density:</td><td>The calculated Density: 0.00227
The percentage of non-empty cells in the sparse matrix.
Formula :
It is the total number of items that are purchased / The total number of possible items in that matrix
The items that were purchased can be calculated using density:
➔18480x7790x0.00227</td></tr>
<tr><td>Frequent Items:</td><td>The following is a list of most frequent items:
1. WHITE HANGING HEART T-LIGHT HOLDER : 1760
2. REGENCY CAKESTAND 3 TIER : 1532
3. JUMBO BAG RED RETROSPOT : 1418
4. PARTY BUNTING : 1267
5. ASSORTED COLOUR BIRD ORNAMENT : 1240
6. (OTHER) : 320759

The itemFrequencyPlot function is used to create a bar plot of the top 10 items that were frequently bought.

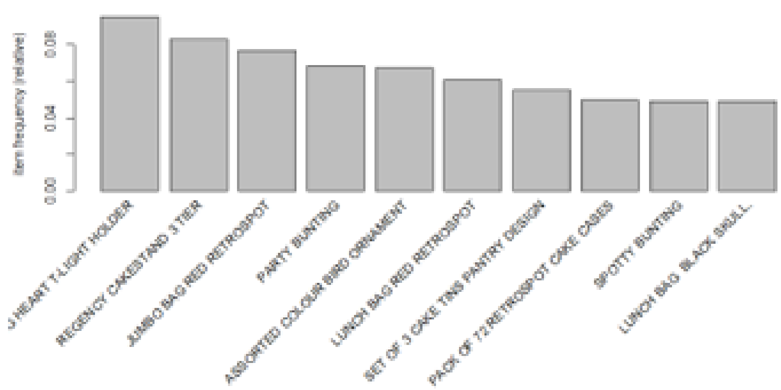


Figure 2: Bar chart of top 10 frequent items.</td></tr>
<tr><td>Transaction Size:</td><td>The Minimum and Maximum products bought in a transaction is 1 & 419 respectively.</td></tr>
<tr><td>Data Distribution:</td><td>The distribution of the data is right skewed. This indicates that most of the customers buy a small number of items in each transaction.

Min. 1st Qu. Median Mean 3rd Qu. Max.
1.00 5.00 13.00 17.75 23.00 419.00

includes extended item information - examples:
labels
1 1 HANGER
2 10 COLOUR SPACEBOY PEN
3 12 COLOURED PARTY BALLOONS
></td></tr>
</table>

Table 3: Observation made from the Summary of the transaction's dataset

To carry forward with the analysis Apriori algorithm (Arules library in R) is used. It is an effective tool to generate association rules and mine frequent itemset. The algorithm applies level-wise inspection for commonly occurring itemset.

A. Lift / Confidence / Support:

Lift:

Lift is a measure of confidence that an antecedent provides us for having the consequent to happen. In mathematical terminology, Lift is the amount of rise in probability for having an item (consequent) on the cart with the knowledge of another item (antecedent) being present/purchased already divided by the probability of having consequent on the cart without any knowledge about presence of antecedent. (Garg, 2019)

Formula:

X – Antecedent | Y – Consequent

P(Y|X) => What is the probability of Y to happen given that you already knew that X happened?

P(Y|X) = P (X and Y)/P(X)

Lift (X -> Y) => What is the value of Lift that {X} actually gives to {Y} to be present on the cart.

Mathematically, Lift (X -> Y) is derived as P(Y|X) divided by P(Y)

Outcome:

For the analysis, we have taken the Lift value to be greater than 1 and less than 10. A value of lift greater than 1 shows that having an antecedent on the cart increases the chances of occurrence of consequent on the cart despite the confidence value. A value of lift greater than 1 account for the high association between the antecedent and consequent.

Confidence:

Confidence works on the rule of conditional probability where we would calculate the probability of an event X given an event Y already occurred. (Garg, 2019)

Formula:

P(X|Y) -> What is the probability of X given Y.

The value from the above condition gives us insight but sometimes it could mislead us as it doesn't check if the Y is popular too.

If both the products X, Y are very popular, both P(X|Y) and P(Y|X) will have higher confidence.

Outcome:

For the analysis, we have taken the confidence (i.e. conf) value as 0.5 or 50% because it is the minimum amount of confidence or strength that we wanted to have for the conditional probability between any two products. Moreover, any value which is above or below than 0.5were either generating too many association rules or limiting them drastically. Hence, we chose 0.5 /50% as a tradeoff and an optimal value for the further analysis.

Support:

Support is sort of a cut-off that we would like to keep to only select the portion of products/events that are popular and are bought/occurred often. This way the analysis is only done on the products/events that occur above a certain threshold and thus leading us to work on a small group of products/events that will have a significant/meaningful effect on business. Selecting a support is a key step to keep a restriction on the different products/events that we would work with for the further analysis. (Garg, 2019)

Formula:

Total number of occurrences of a product from all the records / Total number of records

Outcome:

A trial and error process are conducted to find the optimal support value:

- When we took support (i.e. supp) as 0.03 or 3%.
    - We did not get any rules.
    - Hence to generate rules we will need to take a supp value lesser than 0.03.
- When we took support (i.e. supp) as 0.02 or 2%,
    - We got 17 rules, i.e. we got a small set of rules.
    - We can get rules for specific products, like
        - If customers buy PINK REGENCY TEACUP AND SAUCER they will buy GREEN REGENCY TEACUP AND SAUCER
        - If customers buy ROSES REGENCY TEACUP AND SAUCER they will buy PINK REGENCY TEACUP AND SAUCER
    - The rules have a high lift (>1) which indicates that the purchase of the item(s) on the left-hand side (Antecedent) has a higher likeliness that the item(s) on the right-hand side (Consequent) will also occur on the same invoice.
- When we took support (i.e. supp) as 0.01 or 1%.
    - We got 163 rules, i.e. we got a set of rules with an appropriate size.
    - We can get rules for generic products, like
        - If customers buy SUGAR they will buy COFFEE
        - If customers buy BACK DOOR they will buy KEY FOB

Finally, the value for support was chosen as 0.01. With the Confidence of 0.05 and Lift value between 1 and 10 there were no association rules being generated until the value for support is lowered to 0.01. This was done by changing the values for the Support and keeping the values for Lift and Confidence as static.

B. Rules Generated:

As mentioned in the previous section, the ***support of 0.01***, ***confidence of 0.5*** was used to generate the rules. Then a sub-set is made from the generated rules where ***lift is in between 1 and 10.*** The sub-set rules are then sorted based on the <u>descending decreasing order of the lift.</u> The summary function is then used to have a better understanding of the sub-set rules that was generated.

```
> summary(rules.sub)
set of 23 rules

rule length distribution (lhs + rhs):sizes
 2  3  4
17  5  1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   2.000   2.304   2.500   4.000

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.01001   Min.   :0.5030   Min.   :6.338   Min.   :185.0
 1st Qu.:0.01250   1st Qu.:0.5389   1st Qu.:6.880   1st Qu.:231.0
 Median :0.01494   Median :0.5620   Median :7.279   Median :276.0
 Mean   :0.01583   Mean   :0.5744   Mean   :7.584   Mean   :292.6
 3rd Qu.:0.01981   3rd Qu.:0.6018   3rd Qu.:8.142   3rd Qu.:366.0
 Max.   :0.02592   Max.   :0.7256   Max.   :9.934   Max.   :479.0

mining info:
 data ntransactions support confidence
   tr         18480    0.01        0.5
> |
```

Figure 4: Summary of the sub-set rules

Observation:

Rules:	The Total number of rules generated: 23.
Rules Length Distribution	The most rules that were generated had a length of 2 items/products.
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.	

Table 4: Observation made on the summary of the sub-set rules

The inspect function is then used to view the top 15 individual association rules and have a better understanding of the sub-set rules that was generated.

```
> inspect(rules.sub[1:15])
     lhs                                   rhs                                  support    confidence lift     count
[1]  {LUNCH BAG SPACEBOY DESIGN,
      LUNCH BAG SUKI DESIGN}           => {LUNCH BAG RED RETROSPOT}             0.01001082 0.6026059 9.934127  185
[2]  {LUNCH BAG CARS BLUE,
      LUNCH BAG SPACEBOY DESIGN}       => {LUNCH BAG RED RETROSPOT}             0.01082251 0.6006006 9.901070  200
[3]  {LUNCH BAG PINK POLKADOT}         => {LUNCH BAG RED RETROSPOT}             0.02478355 0.5585366 9.207633  458
[4]  {LUNCH BAG WOODLAND}              => {LUNCH BAG RED RETROSPOT}             0.01980519 0.5176803 8.534106  366
[5]  {LUNCH BAG DOLLY GIRL DESIGN}     => {LUNCH BAG RED RETROSPOT}             0.01352814 0.5030181 8.292395  250
[6]  {JUMBO BAG STRAWBERRY}            => {JUMBO BAG RED RETROSPOT}             0.01980519 0.6354167 8.281030  366
[7]  {JUMBO BAG PINK POLKADOT}         => {JUMBO BAG RED RETROSPOT}             0.02591991 0.6141026 8.003255  479
[8]  {CANDLEHOLDER PINK HANGING HEART} => {WHITE HANGING HEART T-LIGHT HOLDER} 0.01244589 0.7255521 7.618297  230
[9]  {JUMBO BAG SCANDINAVIAN BLUE PAISLEY} => {JUMBO BAG RED RETROSPOT}         0.01255411 0.5843829 7.615935  232
[10] {JUMBO STORAGE BAG SUKI}          => {JUMBO BAG RED RETROSPOT}             0.02034632 0.5620329 7.324660  376
[11] {GREEN REGENCY TEACUP AND SAUCER,
      PINK REGENCY TEACUP AND SAUCER,
      ROSES REGENCY TEACUP AND SAUCER} => {REGENCY CAKESTAND 3 TIER}            0.01141775 0.6063218 7.313856  211
[12] {JUMBO  BAG BAROQUE BLACK WHITE}  => {JUMBO BAG RED RETROSPOT}             0.01704545 0.5585106 7.278756  315
[13] {PINK REGENCY TEACUP AND SAUCER,
      ROSES REGENCY TEACUP AND SAUCER} => {REGENCY CAKESTAND 3 TIER}            0.01271645 0.6010230 7.249938  235
[14] {JUMBO BAG SPACEBOY DESIGN}       => {JUMBO BAG RED RETROSPOT}             0.01255411 0.5550239 7.233316  232
[15] {JUMBO BAG PINK VINTAGE PAISLEY}  => {JUMBO BAG RED RETROSPOT}             0.01536797 0.5409524 7.049929  284
> |
```

Figure 5: Inspecting the top 15 association rules of the sub-set rules

From the above image we can interpret the above rules as follows:

- 72% customers who bought "CANDLEHOLDER PINK HANGING HEART" also bought "WHITE HANGING HEART T-LIGHT HOLDER".
- 60% customers who bought "GREEN REGENCY TEACUP" AND "SAUCER & PINK REGENCY TEACUP AND SAUCER" & "ROSES REGENCY TEACUP AND SAUCER" also bought "REGENCY CAKESTAND 3 TIER".

The rest of the rules can be interpreted in the same way.

The following image is a plot of the top 15 rules:

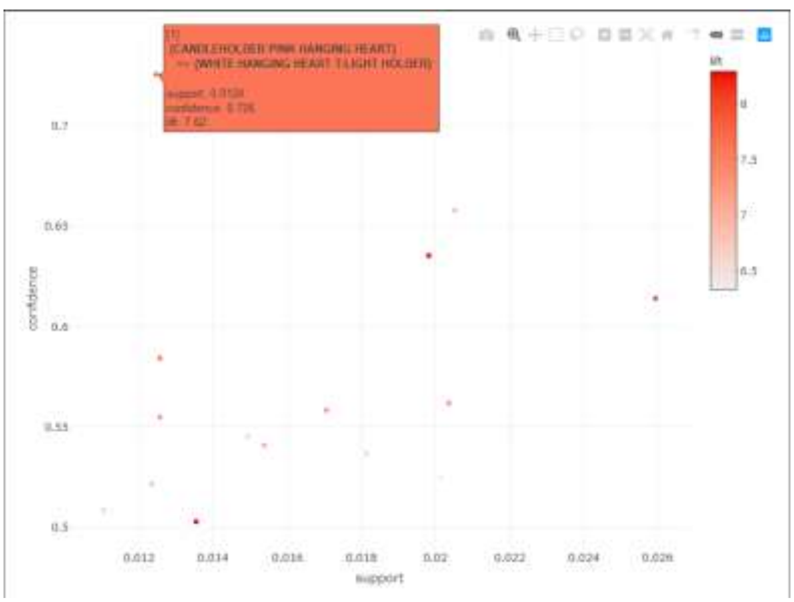


Figure 6: Graph Plot of the top 15 Rules

3. Maximal Frequent Itemset:

Maximal frequent itemset is defined as the superset which is a frequent itemset and which doesn't have another superset which falls under frequent item set. (Kumaresan, 2019)

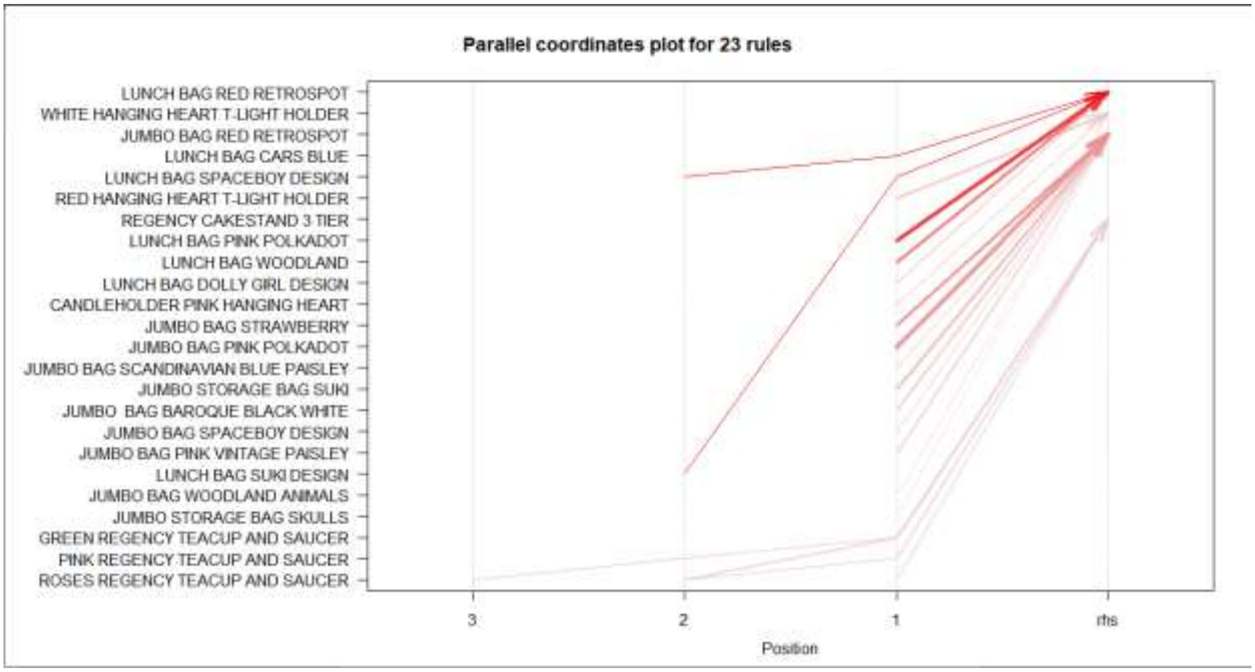


Figure 7: Association of items that are purchased together.

The above figure depicts the below rules and the superset which is influencing the purchase patterns of the items. (Analytics Vidhya, 2019)

The line which is plotted from the highest position on x axis gives us the combination of all the items which are maximal supersets for the respective dataset that we have considered.

Rule No	LHS	RHS	Support	Confidence	Lift	Count
11	{GREEN REGENCY TEACUP AND SAUCER, PINK REGENCY TEACUP AND SAUCER, ROSES REGENCY TEACUP AND SAUCER}	{REGENCY CAKESTAND 3 TIER}	0.01141775	0.6063218	7.313856	211
13	{PINK REGENCY TEACUP AND S AUCER, ROSES REGENCY TEACUP AND SAUCER}	{REGENCY CAKESTAND 3 TIER}	0.01271645	0.6010230	7.249938	235

16	{GREEN REGENCY TEACUP AND SAUCER, PINK REGENCY TEACUP AND SAUCER}	{REGENCY CAKESTAND 3 TIER}	0.01304113	0.5821256	7.021985	241
1	{LUNCH BAG SPACEBOY DESIGN , LUNCH BAG SUKI DESIGN}	{LUNCH BAG RED RETROSPOT}	0.01001082	0.6026059	9.934127	185
2	{LUNCH BAG CARS BLUE, LUNCH BAG SPACEBOY DESIGN}	{LUNCH BAG RED RETROSPOT}	0.01082251	0.6006006	9.901070	200

Table 5: list of 5 Maximally Frequent Item sets

From the above diagram and the table, we conclude that Rule number 1,2,11,13,16 with LHS + RHS can be considered as 5 maximally frequent item sets.

We observe the RHS in the below table are the most frequent items as projected in *Figure 3.*

```
lhs                                 rhs                                 support confidence    lift count
[1]  {LUNCH BAG SPACEBOY DESIGN,
      LUNCH BAG SUKI DESIGN}         => {LUNCH BAG RED RETROSPOT}           0.01001082  0.6026059 9.934127   185
[2]  {LUNCH BAG CARS BLUE,
      LUNCH BAG SPACEBOY DESIGN}     => {LUNCH BAG RED RETROSPOT}           0.01082251  0.6006006 9.901070   200
[3]  {LUNCH BAG PINK POLKADOT}       => {LUNCH BAG RED RETROSPOT}           0.02478355  0.5585366 9.207633   458
[4]  {LUNCH BAG WOODLAND}            => {LUNCH BAG RED RETROSPOT}           0.01980519  0.5176803 8.534106   366
[5]  {LUNCH BAG DOLLY GIRL DESIGN}   => {LUNCH BAG RED RETROSPOT}           0.01352814  0.5030181 8.292395   250
[6]  {JUMBO BAG STRAWBERRY}          => {JUMBO BAG RED RETROSPOT}           0.01980519  0.6354167 8.281030   366
[7]  {JUMBO BAG PINK POLKADOT}       => {JUMBO BAG RED RETROSPOT}           0.02591991  0.6141026 8.003255   479
[8]  {CANDLEHOLDER PINK HANGING HEART}      => {WHITE HANGING HEART T-LIGHT HOLDER} 0.01244589  0.7255521 7.618297   230
[9]  {JUMBO BAG SCANDINAVIAN BLUE PAISLEY} => {JUMBO BAG RED RETROSPOT}     0.01255411  0.5843829 7.615935   232
[10] {JUMBO STORAGE BAG SUKI}        => {JUMBO BAG RED RETROSPOT}           0.02034632  0.5620329 7.324660   376
[11] {GREEN REGENCY TEACUP AND SAUCER,
      PINK REGENCY TEACUP AND SAUCER,
      ROSES REGENCY TEACUP AND SAUCER}      => {REGENCY CAKESTAND 3 TIER}        0.01141775  0.6063218 7.313856   211
[12] {JUMBO  BAG BAROQUE BLACK WHITE}       => {JUMBO BAG RED RETROSPOT}           0.01704545  0.5585106 7.278756   315
[13] {PINK REGENCY TEACUP AND SAUCER,
      ROSES REGENCY TEACUP AND SAUCER}      => {REGENCY CAKESTAND 3 TIER}          0.01271645  0.6010230 7.249938   235
[14] {JUMBO BAG SPACEBOY DESIGN}     => {JUMBO BAG RED RETROSPOT}           0.01255411  0.5550239 7.233316   232
[15] {JUMBO BAG PINK VINTAGE PAISLEY}       => {JUMBO BAG RED RETROSPOT}           0.01536797  0.5409524 7.049929   284
[16] {GREEN REGENCY TEACUP AND SAUCER,
      PINK REGENCY TEACUP AND SAUCER}       => {REGENCY CAKESTAND 3 TIER}          0.01304113  0.5821256 7.021985   241
[17] {RED HANGING HEART T-LIGHT HOLDER}     => {WHITE HANGING HEART T-LIGHT HOLDER} 0.02050866  0.6579861 6.908854   379
[18] {GREEN REGENCY TEACUP AND SAUCER,
      ROSES REGENCY TEACUP AND SAUCER}      => {REGENCY CAKESTAND 3 TIER}          0.01493506  0.5679012 6.850401   276
[19] {JUMBO BAG WOODLAND ANIMALS}    => {JUMBO BAG RED RETROSPOT}           0.01233766  0.5217391 6.799534   228
[20] {JUMBO STORAGE BAG SKULLS}      => {JUMBO BAG RED RETROSPOT}           0.01103896  0.5087282 6.629970   204
[21] {PINK REGENCY TEACUP AND SAUCER}       => {REGENCY CAKESTAND 3 TIER}          0.01493506  0.5454545 6.579634   276
[22] {GREEN REGENCY TEACUP AND SAUCER}      => {REGENCY CAKESTAND 3 TIER}          0.01812771  0.5368590 6.475949   335
[23] {ROSES REGENCY TEACUP AND SAUCER}      => {REGENCY CAKESTAND 3 TIER}          0.02012987  0.5254237 6.338009   372
```

Conclusion:

The Apriori algorithm helps us to understand and evaluate the association of the products and understand the pattern of frequent purchase. Using the support, confidence, lift, count parameters, we can make business decisions on the products which has to stay in the store and how it is going to influence the purchase of other products and increase the revenues to the business. Further we can analyze the supersets with different support values and confidence and understand different purchase patterns. Market basket analysis is made easy and performed efficiently with association mining algorithms which is useful to the retail businesses and the applications of this association is huge in various fields.

References:

Analytics Vidhya. (2019). *Mining frequent items bought together using Apriori Algorithm (code in R)*. [online] Available at: https://www.analyticsvidhya.com/blog/2017/08/mining-frequent-items-using-apriori-algorithm/ [Accessed 22 Jun. 2019].

Garg, A. (2019). *Complete guide to Association Rules (1/2)*. [online] Towards Data Science. Available at: https://towardsdatascience.com/association-rules-2-aa9a77241654 [Accessed 22 Jun. 2019].

Kumaresan, D. (2019). *maximal frquent itemset*. [online] YouTube. Available at: https://www.youtube.com/watch?v=3A4I7sgD9uk [Accessed 22 Jun. 2019].

Li, S. (2019). *A Gentle Introduction on Market Basket Analysis — Association Rules*. [online] Towards Data Science. Available at: https://towardsdatascience.com/a-gentle-introduction-on-market-basket-analysis-association-rules-fa4b986a40ce [Accessed 22 Jun. 2019].

Rdocumentation.org. (2019). *ddply function | R Documentation*. [online] Available at: https://www.rdocumentation.org/packages/plyr/versions/1.8.4/topics/ddply [Accessed 22 Jun. 2019].

Appendix:

SQL Code:

```sql
CREATE TABLE temp (SELECT `InvoiceNo`, `Description` FROM
dataset04.OnlineRetail WHERE `UnitPrice` > 0 AND `Quantity` > 0 AND
`CustomerID` <> 0 AND `InvoiceNo` <> 0 AND `StockCode` <> "POST")
```

R Code:

```r
setwd("D:/Workspace/r-workspace/MCDA 5580/Assignment3")
getwd()

# install.packages("arules")
# install.packages("plyr", dependencies = TRUE)
# install.packages("arulesViz")

library(arules)
library(plyr)

df_user= read.csv("temp.csv")
df_user <- df_user[df_user$InvoiceNo != "0", ]
View(df_user)
df_user = ddply(df_user,c("InvoiceNo"),function(dfl)paste(dfl$Description,
collapse = ","))
df_user$InvoiceNo = NULL
write.table(df_user,"Milestones2.csv", quote=FALSE, row.names = FALSE,
col.names = FALSE)
tr = read.transactions("Milestones2.csv",format="basket",sep=",")
summary(tr)
```

```
itemFrequencyPlot(tr, topN=10)

#----------------------------------------------------------------
#supp = 0.03
rules = apriori(tr,parameter = list(supp=0.03,conf=0.5))
inspect(rules)
#supp = 0.03 (Gives No Rules)
#
#----------------------------------------------------------------
#supp = 0.02
rules = apriori(tr,parameter = list(supp=0.02,conf=0.5))
inspect(rules)
#supp = 0.02 (Gives 17 Rules)
#
#----------------------------------------------------------------
#supp = 0.01
rules = apriori(tr,parameter = list(supp=0.01,conf=0.5))
inspect(rules)
#supp = 0.01 (Gives 163 Rules)

rules.sub = subset(rules, subset = lift > 1 & lift < 10)
inspect(rules.sub)
rules.sub = sort(rules.sub,by='lift')
inspect(rules.sub)

itemsets=unique(generatingItemsets(rules.sub))
itemsets
inspect(itemsets)

#----------------------------------------------------------------
#getting the maximally frequent itemsets
help(apriori)
maxrules = apriori(tr,list(supp=0.02,conf=0.5, target="maximally frequent
itemsets"))
inspect(sort(maxrules))

#----------------------------------------------------------------
#plotting the graph.
#install.packages("arulesViz")
library(arulesViz)
plot(rules.sub[1:5],method = "graph",control = list(type = "items"))
plot(rules.sub[1:23],method = "matrix",control = list(type =
"items",reorder))
arulesViz::plotly_arules(rules.sub)
arulesViz::plotly_arules(rules.sub[1:15])
plot(sort(rules.sub,by='lift')[1:23],method = "paracoord",control =
list(reorder = TRUE))
```