National College of Ireland

Customer Visit Segmentation based on Clustering and Association Rules

Configuration Manual

MSc Research Project
Data Analytics

Vishakha Kale
Student ID: x18181643

School of Computing
National College of Ireland

Supervisor: Dr. Paul Stynes, Dr. Pramod Pathak

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vishakha Balkrishna Kale
Student ID:	X18181643
Programme:	M.Sc. in Data Analytics **Year:** 2020
Module:	Research Project
Lecturer:	Dr. Paul Stynes, Dr. Pramod Pathak
Submission Due Date:	17th August 2020
Project Title:	Customer Visit Segmentation based on Clustering and Association Rules.
Word Count:	1060 **Page Count:** 11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Vishakha Kale

Date: 17th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Customer Visit Segmentation based on Clustering and Association Rules

Vishakha Kale
Student ID: x18181643

1 Introduction

This configuration manual explains every hardware requirement and steps to follow for implementing the research experiment of customer visit segmentation using clustering and association rule.

2 Hardware Setup

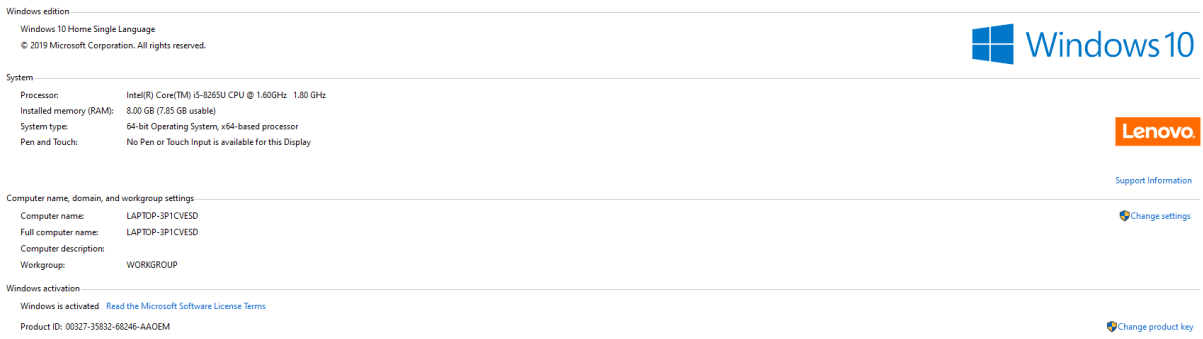


Figure 1 : Computer Hardware

The computer on which the project is implemented has 8 GB RAM and Intel Core i5 processor with 1.60GHZ CPU. All the experiments and environments implemented smoothly on this computer without any glitch.

3 Environment Setup

Environments used for this project are as follows and essential to setup to execute this project.

1. SQL Server.
2. SQL Server Management Studio.
3. Visual Studio with SSAS extension.
4. RStudio

3.1 SQL Server

SQL Server is used for Data pre-processing and Data storage in this research. To install SQL Server, download installation pack from the link below (Developer version is used in this research):

https://www.microsoft.com/en-us/sql-server/sql-server-downloads

SQL Server has installed by creating a new instance as shown in figure 2:

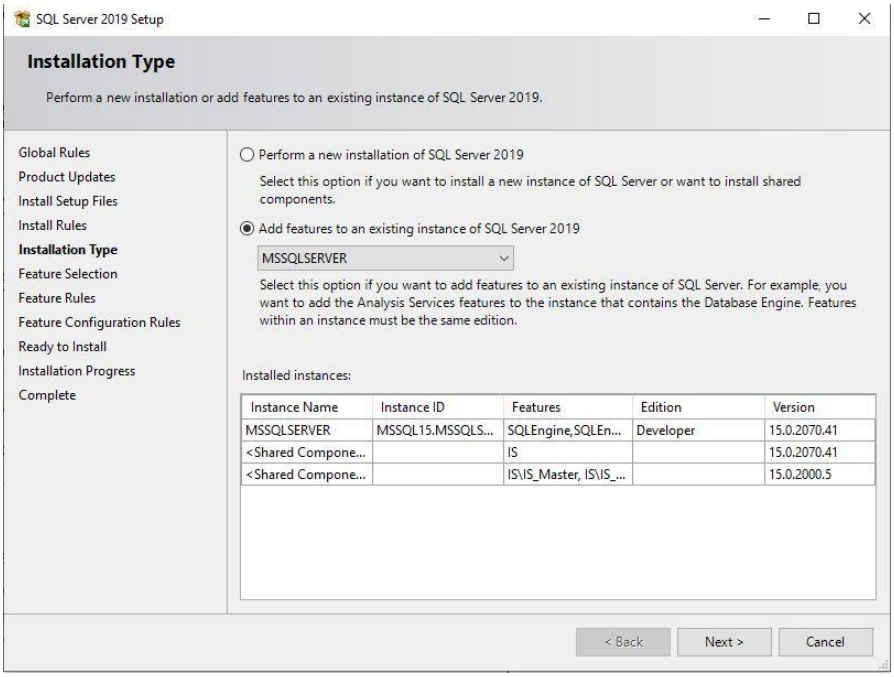


Figure 2 : SQL Server Instance

The SQL instance is created with features as given in figure 3

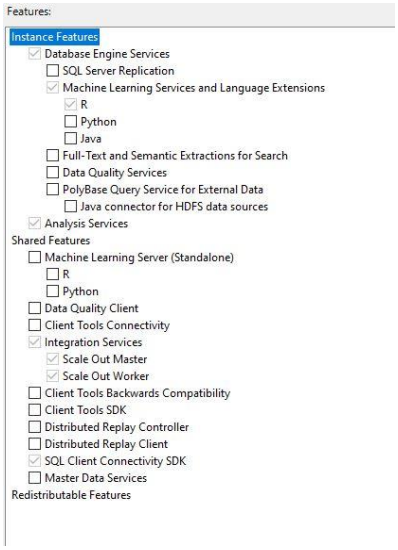


Figure 3 : SQL Server Instance Features

3.2 SQL Server Management Studio

SQL Server Management studio is essential to manage the data stored in SQL Server. SSMS installation wizard can be downloaded from following link:

https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

The installation can be performed by selecting appropriate destination in the installation wizard as given in figure 4

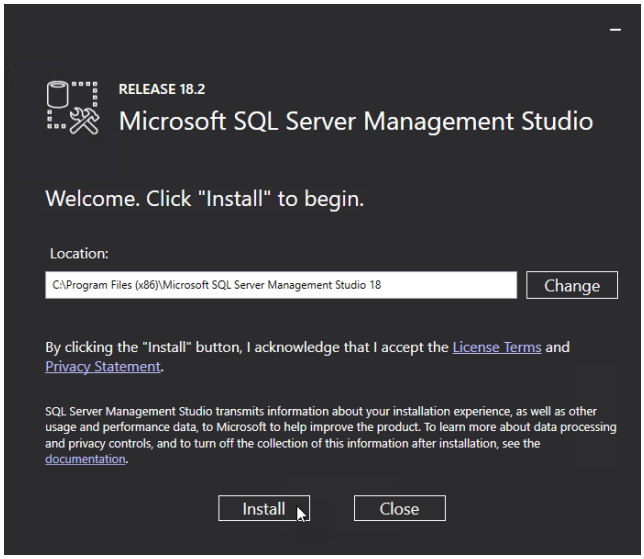


Figure 4 : SQL Server Management Studio Installation Wizard

3.3 Visual Studio 2019 with SSAS Extension

Visual studio 2019 is used as a platform to perform Microsoft Data mining algorithm using SQL Server Analysis Services (SSAS). Visual studio installation wizard can be downloaded from following link:

Features of Data storage and processing, Data science and analytical applications as given in figure 5 are to be selected while installing Visual studio from workloads section.

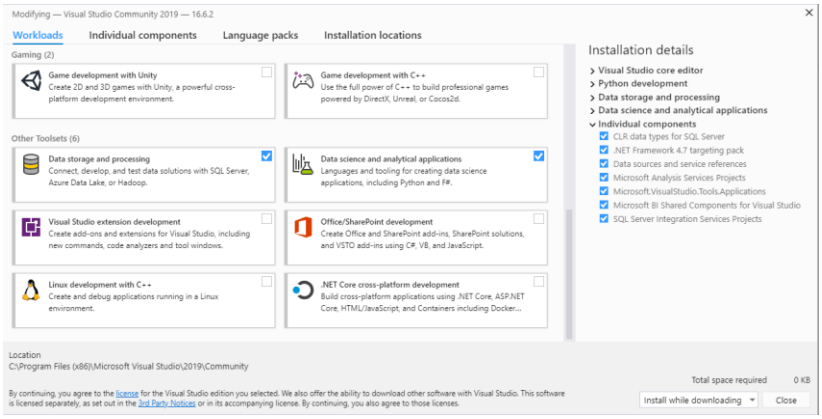


Figure 5 : Visual Studio Features

3.4 RStudio

To Install R:

1. Open an internet browser and go to www.r-project.org.
2. Click the "download R" link in the middle of the page under "Getting Started."
3. Select a CRAN location (a mirror site) and click the corresponding link.
4. Click on the "Download R for Windows" link at the top of the page.
5. Click on the "install R for the first time" link at the top of the page.
6. Click "Download R for Windows" and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.
7. Now that R is installed, you need to download and install RStudio.

To Install RStudio

1. Go to www.rstudio.com and click on the "Download RStudio" button.
2. Click on "Download RStudio Desktop."
3. Click on the version recommended for your system, or the latest Windows version, and save the executable file. Run the .exe file and follow the installation instructions.

To Install the SDSFoundations Package

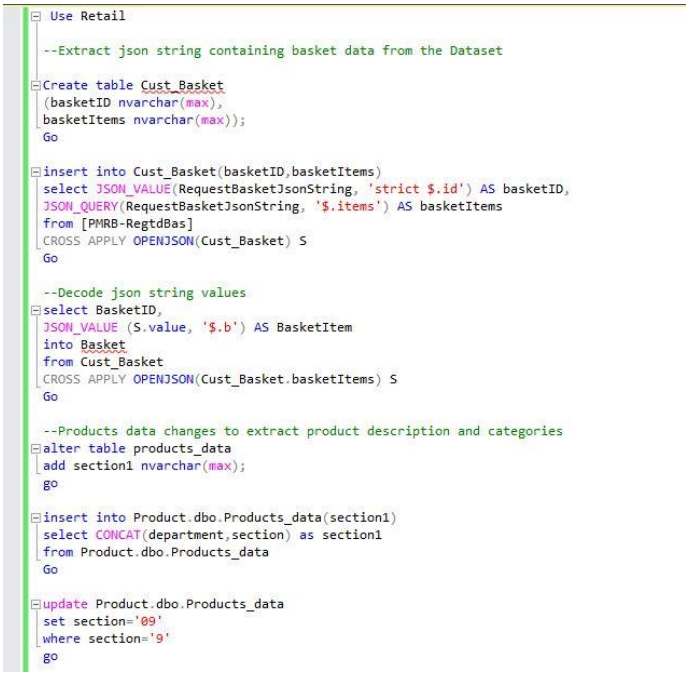
1. Download SDSFoundations to your desktop (make sure it has the ".zip" extension).
2. Open RStudio.
3. Click on the Packages tab in the bottom right window.
4. Click "Install."
5. Select install from "Package Archive File."
6. Select the SDSFoundations package file from your desktop.
7. Click install. You are done! You can now delete the SDSpackage file from your desktop.

Figure 6 : Steps for installing R and RStudio

4 Data Pre-processing

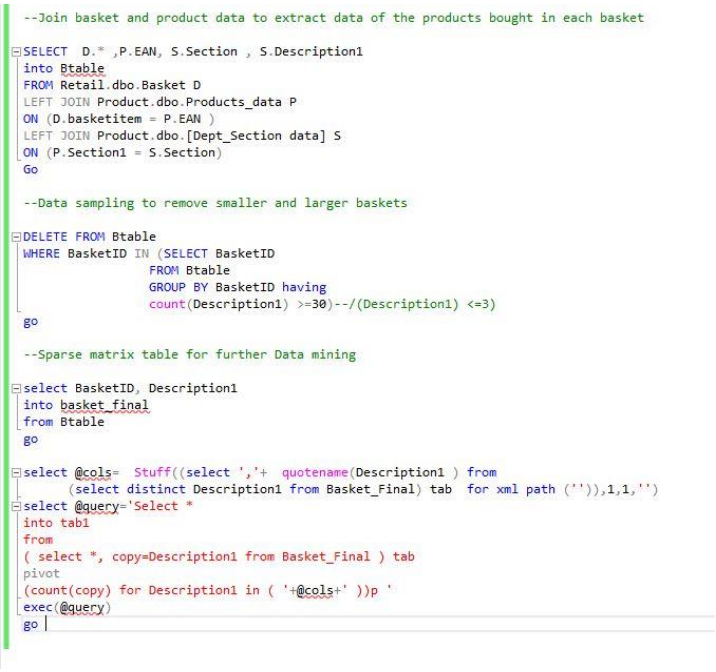
The Data used for this research is provided by NCI_IPP Team named Glantus Data. As per the signed consent with the company, research is not allowed to share the data with anyone.

The following Data pre processing is performed on the provided data given in figure 6 and figure 7 to improve data mining results.



```
Use Retail

--Extract json string containing basket data from the Dataset

Create table Cust_Basket
(basketID nvarchar(max),
basketItems nvarchar(max));
Go

insert into Cust_Basket(basketID,basketItems)
select JSON_VALUE(RequestBasketJsonString, 'strict $.id') AS basketID,
JSON_QUERY(RequestBasketJsonString, '$.items') AS basketItems
from [PMRB-RegtdBas]
CROSS APPLY OPENJSON(Cust_Basket) S
Go

--Decode json string values
select BasketID,
JSON_VALUE (S.value, '$.b') AS BasketItem
into Basket
from Cust_Basket
CROSS APPLY OPENJSON(Cust_Basket.basketItems) S
Go

--Products data changes to extract product description and categories
alter table products_data
add section1 nvarchar(max);
go

insert into Product.dbo.Products_data(section1)
select CONCAT(department,section) as section1
from Product.dbo.Products_data
Go

update Product.dbo.Products_data
set section='09'
where section='9'
go
```

Figure 7 : Data Preprocessing part 1



```
--Join basket and product data to extract data of the products bought in each basket

SELECT  D.* ,P.EAN, S.Section , S.Description1
into Btable
FROM Retail.dbo.Basket D
LEFT JOIN Product.dbo.Products_data P
ON (D.basketitem = P.EAN )
LEFT JOIN Product.dbo.[Dept_Section data] S
ON (P.Section1 = S.Section)
Go

--Data sampling to remove smaller and larger baskets

DELETE FROM Btable
WHERE BasketID IN (SELECT BasketID
                   FROM Btable
                   GROUP BY BasketID having
                   count(Description1) >=30)--/(Description1) <=3)
go

--Sparse matrix table for further Data mining

select BasketID, Description1
into basket_final
from Btable
go

select @cols=  Stuff((select ','+  quotename(Description1 ) from
       (select distinct Description1 from Basket_Final) tab  for xml path ('')),1,1,'')
select @query='Select *
into tab1
from
( select *, copy=Description1 from Basket_Final ) tab
pivot
(count(copy) for Description1 in ( '+@cols+' ))p '
exec(@query)
go
```

Figure 8 : Data Preprocessing part 2

After implementing the stated processing, the data was received in the form of sparse matrix for further processing as in figure 8.

	BasketID	BEER/LAGER & CIDER	FRESH CHICKEN	PIES	BABY NEEDS - FOOD	SUGAR	CREAM CAKES	ROSE TABLE WINES	HOME BAKING
1	340112801273	0	0	0	0	0	0	0	0
2	340313443085	0	0	0	0	0	0	0	0
3	340313443096	0	0	0	0	0	0	0	0
4	340313443178	0	0	0	0	0	0	0	0
5	340313443572	0	0	0	0	0	0	0	0
6	340313443595	0	0	0	0	0	0	0	0
7	340313443907	0	0	0	0	0	0	0	0
8	34032462001	0	0	0	0	0	0	0	0
9	340612895338	0	0	0	0	0	0	0	0
10	340612895578	0	0	0	0	0	0	0	0
11	340612895727	0	0	0	0	0	0	0	0
12	340612895760	0	0	1	0	0	0	0	0
13	340612896098	0	0	0	0	0	0	0	0
14	34062517201	0	0	0	0	0	0	0	0
15	34062517206	0	0	0	0	0	0	0	0
16	340812467224	0	0	0	0	0	0	0	1
17	340912998949	0	0	0	0	0	0	0	0
18	340912999274	0	0	0	0	0	0	0	0
19	340912999378	0	0	0	0	0	0	0	0

Figure 9 : Sparse Matrix format for Basket Data

5 Data Mining

Data Mining for this research is performed in 3 steps:

1) Elbow method in RStudio
2) K-means Clustering in Visual studio
3) Apriori Algorithm in Visual Studio
4) Eclat algorithm in Visual studio

5.1 Elbow method in RStudio

Elbow method is implemented for the evaluation of K-means clustering to get the exact value of K based on which accurate clusters are to be mined.
The code as per figure 9 is implemented on RStudio for the same.

```
#Connect SQL Server with R

library(DBI)
con <- DBI::dbConnect(odbc::odbc(),
                      Driver   = "SQL Server",
                      Server   = "LAPTOP-3P1CVESD",
                      Database = "Retail")

#Fetch the required table from SQL
input_query <- "SELECT * from dbo.NoBasket"

#Save the data in R Dataset
library(tidyverse)
tab1 <- as.tibble(dbGetQuery(con, input_query))

#Perform Elbow method
wss <- (nrow(tab1) - 1) * sum(apply(tab1, 2, var))
for (i in 2:20)
  wss[i] <- sum(kmeans(tab1, centers = i)$withinss)
plot(1:20, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")
```

Figure 10 : Elbow method code

It would plot the elbow method graph as shown in figure 10.

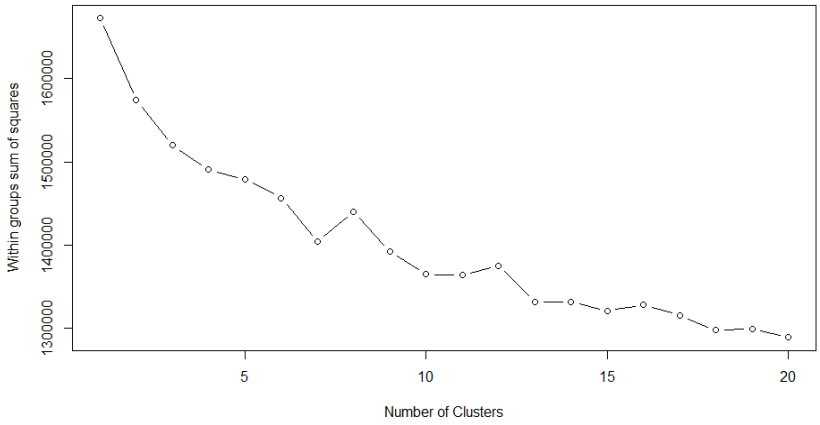


Figure 11 : Elbow method graph

5.2 K-means Clustering

K-means Clustering is implemented using Microsoft clustering in visual studio.
As shown in figure 9, Data source and Data source views are created as required and New mining structure is created from mining structure tab.

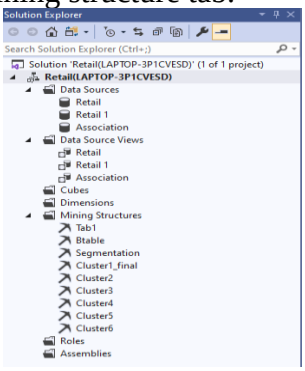


Figure 12 : Visual Studio Solution Explorer

The following parameters as shown in figure 10 are set for CLUSTERING_METHOD as K-means clustering, CLUSTER_COUNT as 6 and SAMPLE_SIZE as 0 to include entire data.

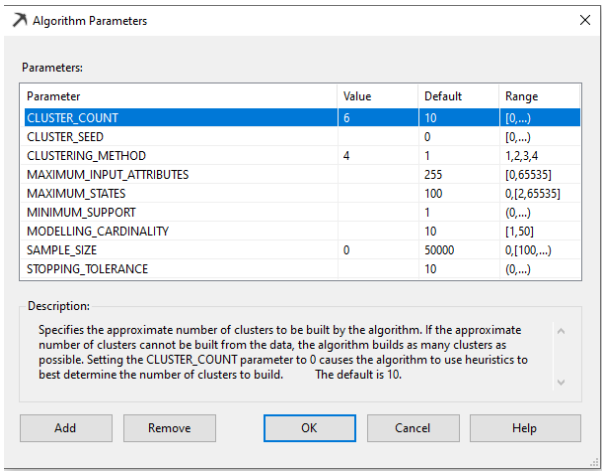


Figure 13 : Clustering Algorithm Parameters

After processing the model, clusters can be viewed in model viewer as shown in figure 11

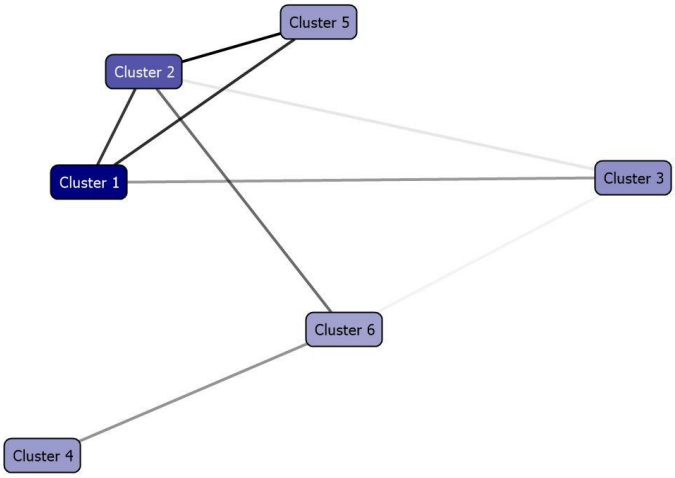


Figure 14 : Cluster Segments

5.3 Apriori Algorithm

For Apriori Algorithm, each cluster data is extracted using DMX query in SSMS as shown in figure 12

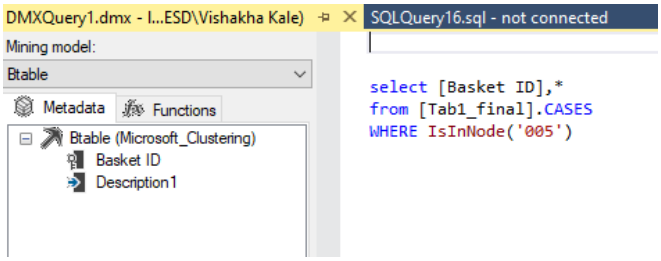


Figure 15 : DMX Query

Similarly, new mining structure is created in solution explorer of visual studio and model as processed for each cluster to get the association rules as shown in figure 13.

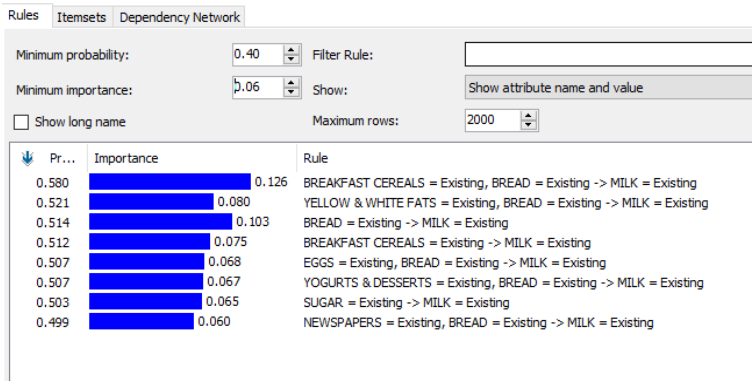


Figure 16 : Apriori Algorithm Rules

5.4 Eclat Algorithm

Data format required by RStudio is different than Visual studio, the data is formatted as per given in figure 14 to prepare the data in basket format.

```
#Import Cluster Data

RCluster1=read.csv('RCluster1.csv')


#Data Formatting

df_sorted <- RCluster1[order(RCluster1$Basket_ID),]
df_sorted$Basket_ID <- as.numeric(df_sorted$Basket_ID)


library(plyr)

df_itemList <- ddply(df_sorted,c("Basket_ID"),
                     function(df1)paste(df1$Description1,
                                        collapse = ","))
df_itemList$Basket_ID <- NULL

colnames(df_itemList) <- c("itemList")

write.csv(df_itemList,"ItemList.csv", row.names = TRUE)
```

Figure 17 : Data Formatting

Eclat algorithm is implemented in RStudio as figure 15 and figure 16 to give out the result as shown in figure 16.

```
library(arules)

Basket = read.csv('ItemList.csv', header = TRUE)

Basket = read.transactions('ItemList.csv', sep = ',', rm.duplicates = TRUE)

summary(Basket)

rules = eclat(data = Basket, parameter = list(support = 0.05 , minlen = 2))

inspect(sort(rules, by = 'support') [1:8])
```

Figure 18 : Eclat Algorithm

```
> summary(Basket)
transactions as itemMatrix in sparse format with
 125532 rows (elements/itemsets/transactions) and
 139 columns (items) and a density of 0.03578244

most frequent items:
                MILK               BREAD CRISPS/SNACKS & NUTS  BAKERY- INSTORE/CDF     CHEESE (PRE PACK)
               54782               35988               26964               23243               17856
             (Other)
              465533

element (itemset/transaction) length distribution:
sizes
    1     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19
    1     7 64056 31146 15350  7506  3668  1905   908   458   205   136    77    44    21    13    10     8
   20    21    22    23    24    25
    2     4     2     3     1     1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   4.000   4.000   4.974   5.000  25.000

includes extended item information - examples:
                    labels
1              ACCESSORIES
2      ACCESSORIES/FASHION
3 AFFORDABLE/LOW ALC WINES
```

Figure 19 : Summary of Basket

```
> inspect(sort(rules, by = 'support') [1:8])
    items                          support    transIdenticalToItemsets count
[1] {BREAD,MILK}                   0.14730905 18492                    18492
[2] {CRISPS/SNACKS & NUTS,MILK}    0.07869707  9879                     9879
[3] {BAKERY- INSTORE/CDF,MILK}     0.07386961  9273                     9273
[4] {BISCUITS,MILK}                0.05863047  7360                     7360
[5] {MILK,MORNING GOODS}           0.05530064  6942                     6942
[6] {CHEESE (PRE PACK),MILK}       0.05514132  6922                     6922
[7] {MILK,YOGURTS & DESSERTS}      0.05268776  6614                     6614
[8] {BREAD,CRISPS/SNACKS & NUTS}   0.05220980  6554                     6554
>
```

Figure 20 : Eclat Algorithm Rules

References

UTAustinX: UT.7.01x Foundations of Data Analysis. (n.d.). Retrieved August 17, 2020,from https://courses.edx.org/courses/UTAustinX/UT.7.01x/3T2014/56c5437b88fa43cf828bff5371c6a924/