Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 4 Date of Implementation : 31/10/2023

****

Q1) Use the Apriori algorithm on the grocery dataset with minimum support to 0.001 and minimum confidence of 0.8 indicate the top 5 association rules that are generated and highlight the strong ones, sort them by confidence.

Program :

```
install.packages("arules")
library(arules)
install.packages("arulesViz")
library(arulesViz)
install.packages("RColorBrewer")
library(RColorBrewer)
#import dataset
data("Groceries")
#applying apriori function with min support=0.001 and min confidence=0.8
rules<-apriori(Groceries,parameter=list(supp=0.001,conf=0.8))
#sorting
rules<-sort(rules,decreasing=TRUE,by="confidence")
#applying inspect element to display 5 strong association rules
inspect(rules[1:5])
#applying itemFrequencyPlot() function
arules::itemFrequencyPlot(Groceries, topN=20,
                          col=brewer.pal(8,'Pastel2'),
                          main='Relative Item Frequency Plot',
                          type="relative",
                          ylab="Item Frequency(Relative)")
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 4 Date of Implementation : 31/10/2023

****

Output :

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
        0.8    0.1    1 none FALSE            TRUE       5   0.001      1     10
 target  ext
  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.04s].
writing ... [410 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> #Sorting
> rules <- sort(rules, decreasing=TRUE,by="confidence")
> # using inspect() function
> inspect(rules[1:5])
    lhs                     rhs              support confidence    coverage     lift cou
nt
[1] {rice,
     sugar}              => {whole milk} 0.001220132          1 0.001220132 3.913649
12
[2] {canned fish,
     hygiene articles}   => {whole milk} 0.001118454          1 0.001118454 3.913649
11
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

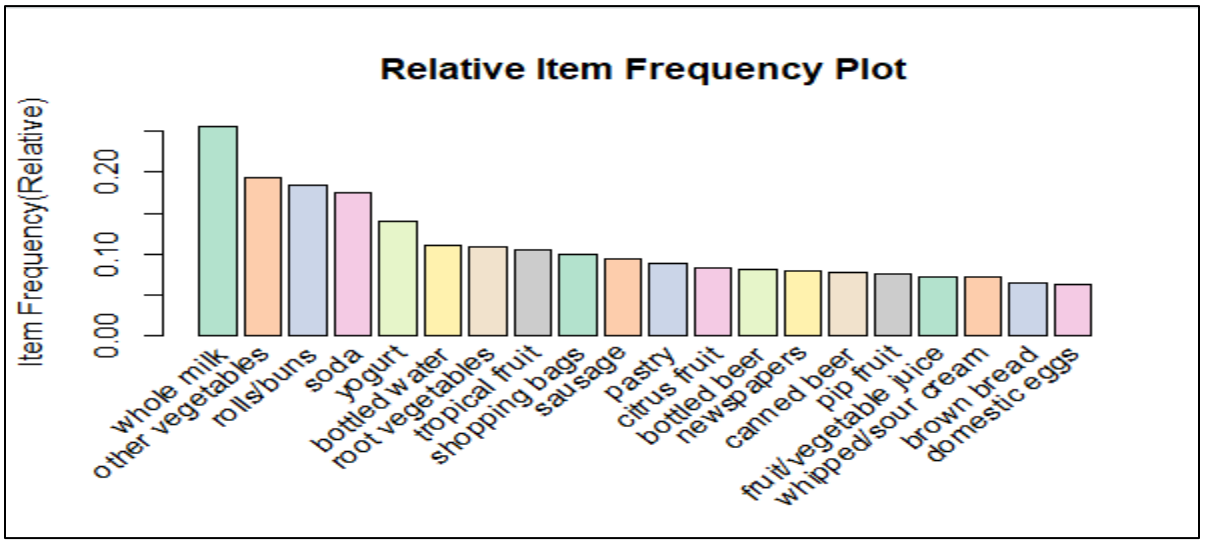
****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 4 Date of Implementation : 31/10/2023

****

```
[3] {root vegetables,
     butter,
     rice}                => {whole milk} 0.001016777          1 0.001016777 3.913649
10
[4] {root vegetables,
     whipped/sour cream,
     flour}               => {whole milk} 0.001728521          1 0.001728521 3.913649
17
[5] {butter,
     soft cheese,
     domestic eggs}       => {whole milk} 0.001016777          1 0.001016777 3.913649
10
> #Ploting graph of frequency
> arules::itemFrequencyPlot(Groceries, topN = 20,
+                           col = brewer.pal(8, 'Pastel2'),
+                           main = 'Relative Item Frequency Plot',
+                           type = "relative",
+                           ylab = "Item Frequency (Relative)")
>
```



Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 4 Date of Implementation : 31/10/2023

****

Q2) Use the Eclat algorithm on given Market Basket Dataset and predict the items which are bought frequently.

Program :

```
install.packages("tidyverse")
library(arules)
library(arulesViz)
library(tidyverse)
dataset=read.csv("E:\\MCA\\SY (1)\\Data Science\\Practical\\Market_Basket_Optimisation.csv",header=FALSE)
dataset=read.transactions("E:\\MCA\\SY (1)\\Data
Science\\Practical\\Market_Basket_Optimisation.csv",sep=",",rm.duplicates=TRUE)
associa_rules=eclat(data=dataset,parameter = list(support=0.004,minlen=2))
inspect(sort(associa_rules,by="support")[1:20])
```

Output :

```
> dataset=read.transactions("E:\\MCA\\SY (1)\\Data Science\\Practical\\Market_Basket_Opti
misation.csv",sep=",",rm.duplicates=TRUE)
distribution of transactions with duplicates:
1
5
> associa_rules=eclat(data=dataset,parameter = list(support=0.004,minlen=2))
Eclat

parameter specification:
 tidLists support minlen maxlen            target  ext
    FALSE   0.004      2     10 frequent itemsets TRUE

algorithmic control:
 sparse sort verbose
      7   -2    TRUE

Absolute minimum support count: 30

create itemset ...
set transactions ...[119 item(s), 7501 transaction(s)] done [0.01s].
sorting and recoding items ... [114 item(s)] done [0.00s].
creating sparse bit matrix ... [114 row(s), 7501 column(s)] done [0.01s].
writing  ... [845 set(s)] done [0.03s].
Creating S4 object  ... done [0.00s].
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 4 Date of Implementation : 31/10/2023

****

```
> inspect(sort(associa_rules,by="support")[1:20])
     items                                 support    count
[1]  {mineral water, spaghetti}            0.05972537 448
[2]  {chocolate, mineral water}            0.05265965 395
[3]  {eggs, mineral water}                 0.05092654 382
[4]  {milk, mineral water}                 0.04799360 360
[5]  {ground beef, mineral water}          0.04092788 307
[6]  {ground beef, spaghetti}              0.03919477 294
[7]  {chocolate, spaghetti}                0.03919477 294
[8]  {eggs, spaghetti}                     0.03652846 274
[9]  {eggs, french fries}                  0.03639515 273
[10] {frozen vegetables, mineral water}    0.03572857 268
[11] {milk, spaghetti}                     0.03546194 266
[12] {chocolate, french fries}             0.03439541 258
[13] {mineral water, pancakes}             0.03372884 253
[14] {french fries, mineral water}         0.03372884 253
[15] {chocolate, eggs}                     0.03319557 249
[16] {chocolate, milk}                     0.03212905 241
[17] {green tea, mineral water}            0.03106252 233
[18] {eggs, milk}                          0.03079589 231
[19] {burgers, eggs}                       0.02879616 216
[20] {french fries, green tea}             0.02852953 214
>
```