Carrefour-Marketing-Project Applying Association Rules

Margaret Gathoni

2022-07-29

Introduction

Carrefour has 13 outlets mostly located in the suburbs of Kenya's capital city, Nairobi. Their mission is to provide our customers with quality services, products and food accessible to all across all distribution channels.

Problem Statement

The project aim to inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).

Metrics of success

a. To uncover how the items are associated with each other
b. Provide insights for our analysis.

Association Analysis

Loading important libraries

```
library(arules)

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##     abbreviate, write

library(arulesViz)
```

Loading the data set

```
path <- "http://bit.ly/SupermarketDatasetII"

data <-read.transactions(path, sep = ",")

## Warning in asMethod(object): removing duplicated items in transactions

data
```

```
## transactions in sparse format with
##  7501 transactions (rows) and
##  119 items (columns)
```

Previewing objects in our data set

```
dim(data)

## [1] 7501  119
```

Verifying the object's class

```
class(data)

## [1] "transactions"
## attr(,"package")
## [1] "arules"
```

Previewing our first 5 items

```
inspect(data[1:5])

##     items
## [1] {almonds,
##      antioxydant juice,
##      avocado,
##      cottage cheese,
##      energy drink,
##      frozen smoothie,
##      green grapes,
##      green tea,
##      honey,
##      low fat yogurt,
##      mineral water,
##      olive oil,
##      salad,
##      salmon,
##      shrimp,
##      spinach,
##      tomato juice,
##      vegetables mix,
##      whole weat flour,
##      yams}
## [2] {burgers,
##      eggs,
##      meatballs}
## [3] {chutney}
## [4] {avocado,
##      turkey}
## [5] {energy bar,
##      green tea,
##      milk,
```

```
##      mineral water,
##      whole wheat rice}
```

The results above shows combination of items in first five items

We can also view the items as follows

```
items<-as.data.frame(itemLabels(data))
colnames(items) <- "Item"
head(items, 10)
```

```
##                  Item
## 1             almonds
## 2  antioxydant juice
## 3           asparagus
## 4             avocado
## 5         babies food
## 6               bacon
## 7      barbecue sauce
## 8           black tea
## 9         blueberries
## 10         body spray
```

Previewing the summary of data

```
summary(data)
```

```
## transactions as itemMatrix in sparse format with
##  7501 rows (elements/itemsets/transactions) and
##  119 columns (items) and a density of 0.03288973
##
## most frequent items:
## mineral water          eggs     spaghetti  french fries     chocolate
##          1788          1348          1306          1282          1229
##       (Other)
##         22405
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
16
## 1754 1358 1044  816  667  493  391  324  259  139  102   67   40   22   17
4
##   18   19   20
##    1    2    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.914   5.000  20.000
##
## includes extended item information - examples:
##              labels
```

```
## 1           almonds
## 2 antioxydant juice
## 3         asparagus
```

This shows the most purchased items as almonds, antioxidant juice, and asparagus. It also shows the distribution of the item sets (no. of items purchased in each transaction),

Exploring the frequency of some items

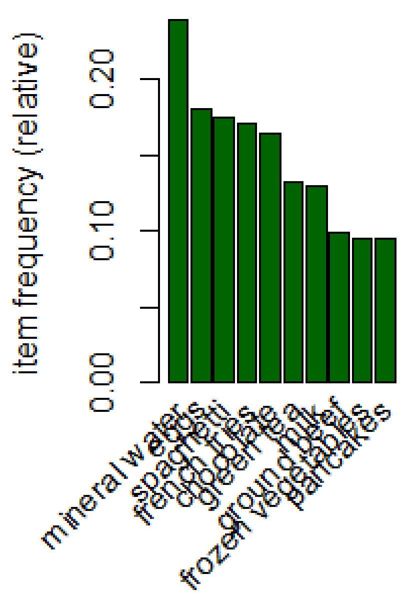
```
itemFrequency(data[, 5:12],type = "absolute")
```

```
##    babies food           bacon barbecue sauce       black tea     blueberries
##             34              65             81             107              69
##     body spray         bramble       brownies
##             86              14            253
```

The results above shows transactions ranging from 5 to 12

Plotting a frequency chart

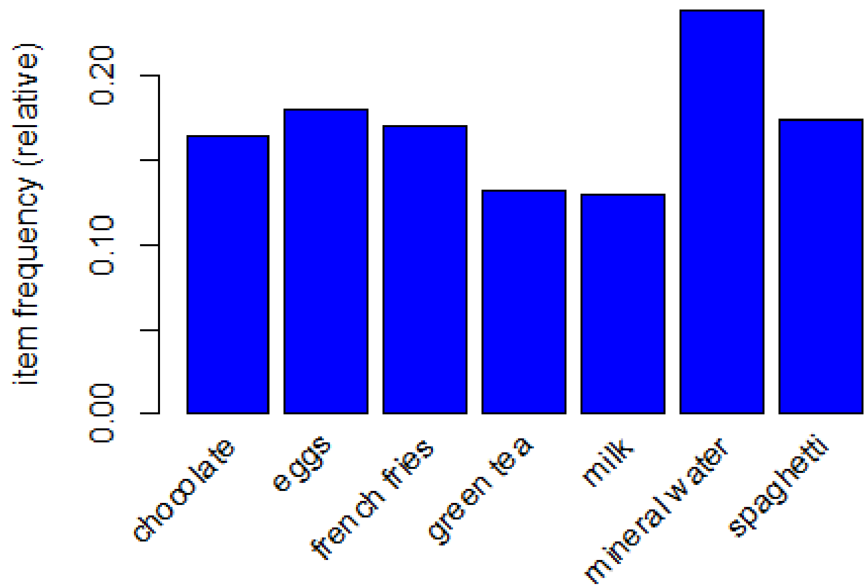
```
par(mfrow = c(1, 2))
itemFrequencyPlot(data, topN = 10,col="darkgreen")
```



The above graph shows top 10 most common items in the data set.

Displaying items whose relative importance is at least 10%

```
itemFrequencyPlot(data, support = 0.1,col="blue")
```



Let's now Build a model based on association rules a. Using Minimum support of 0.001 and CI of 0.8

```
rules <- apriori (data, parameter = list(supp = 0.001, conf = 0.8))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 7
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.00s].
## sorting and recoding items ... [116 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 done [0.02s].
## writing ... [74 rule(s)] done [0.01s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules)

## set of 74 rules
##
## rule length distribution (lhs + rhs):sizes
##  3  4  5  6
## 15 42 16  1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   4.000   4.000   4.041   4.000   6.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001067   Min.   :0.8000   Min.   :0.001067   Min.   : 3.356
##  1st Qu.:0.001067   1st Qu.:0.8000   1st Qu.:0.001333   1st Qu.: 3.432
##  Median :0.001133   Median :0.8333   Median :0.001333   Median : 3.795
##  Mean   :0.001256   Mean   :0.8504   Mean   :0.001479   Mean   : 4.823
##  3rd Qu.:0.001333   3rd Qu.:0.8889   3rd Qu.:0.001600   3rd Qu.: 4.877
##  Max.   :0.002533   Max.   :1.0000   Max.   :0.002666   Max.   :12.722
##      count
##  Min.   : 8.000
##  1st Qu.: 8.000
##  Median : 8.500
##  Mean   : 9.419
##  3rd Qu.:10.000
##  Max.   :19.000
##
## mining info:
##  data ntransactions support confidence
##  data          7501   0.001        0.8
##                                                                 call
##  apriori(data = data, parameter = list(supp = 0.001, conf = 0.8))
```

Observing rules built in our mode

```
inspect(rules[1:5])

##     lhs                               rhs             support
confidence
## [1] {frozen smoothie, spinach}    => {mineral water} 0.001066524 0.8888889
## [2] {bacon, pancakes}             => {spaghetti}     0.001733102 0.8125000
## [3] {nonfat milk, turkey}         => {mineral water} 0.001199840 0.8181818
## [4] {ground beef, nonfat milk}    => {mineral water} 0.001599787 0.8571429
## [5] {mushroom cream sauce, pasta} => {escalope}      0.002532996 0.9500000
##     coverage    lift      count
## [1] 0.001199840  3.729058  8
## [2] 0.002133049  4.666587 13
## [3] 0.001466471  3.432428  9
## [4] 0.001866418  3.595877 12
## [5] 0.002666311 11.976387 19
```

The table above shows that if someone buys mushroom cream sauce, pasta they are 88% likely to buy escalope too.

We can plot this as follows

```
inspect(sort(rules, by = 'lift')[1:10])

##      lhs                           rhs                     support  confidence
coverage      lift count
## [1]  {eggs,
##       mineral water,
##       pasta}                    => {shrimp}            0.001333156  0.9090909
0.001466471 12.722185    10
## [2]  {french fries,
##       mushroom cream sauce,
##       pasta}                    => {escalope}          0.001066524  1.0000000
0.001066524 12.606723     8
## [3]  {milk,
##       pasta}                    => {shrimp}            0.001599787  0.8571429
0.001866418 11.995203    12
## [4]  {mushroom cream sauce,
##       pasta}                    => {escalope}          0.002532996  0.9500000
0.002666311 11.976387    19
## [5]  {chocolate,
##       ground beef,
##       milk,
##       mineral water,
##       spaghetti}                => {frozen vegetables} 0.001066524  0.8888889
0.001199840  9.325253     8
## [6]  {herb & pepper,
##       mineral water,
##       rice}                     => {ground beef}       0.001333156  0.9090909
0.001466471  9.252498    10
## [7]  {grated cheese,
##       mineral water,
##       rice}                     => {ground beef}       0.001066524  0.8888889
0.001199840  9.046887     8
## [8]  {cake,
##       meatballs,
##       mineral water}            => {milk}              0.001066524  1.0000000
0.001066524  7.717078     8
## [9]  {escalope,
##       hot dogs,
##       mineral water}            => {milk}              0.001066524  0.8888889
0.001199840  6.859625     8
## [10] {meatballs,
##       whole wheat pasta}        => {milk}              0.001333156  0.8333333
0.001599787  6.430898    10

plot(rules, method = "graph",
     measure = "confidence", shading = "lift")
```

```
## Warning: ggrepel: 6 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

The graph show association rules by confidence and lift. From the previous table we saw that if someone buys mushroom cream sauce, pasta they are 88% likely to buy escalope too. On the graph we can clearly see this association via lift.

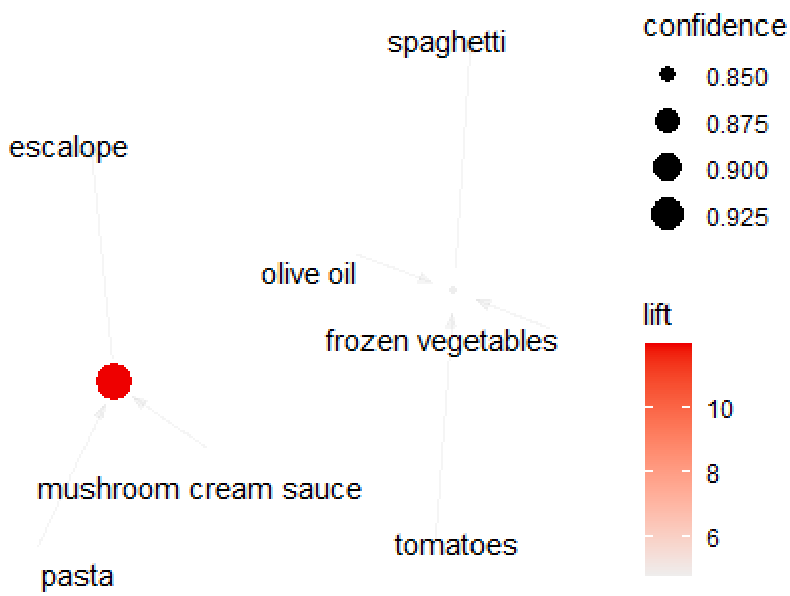
We can try using different support and confidence interval

b. Using Minimum support of 0.002 and CI of 0.8

```
rules2 <- apriori (data, parameter = list(supp = 0.002, conf = 0.8))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.002      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
```

```
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.00s].
## sorting and recoding items ... [115 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [2 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
plot(rules2, method = "graph",
     measure = "confidence", shading = "lift")
```



The graph shows less feature due to use of higher support value

c. Using Minimum support of 0.002 and CI of 0.6

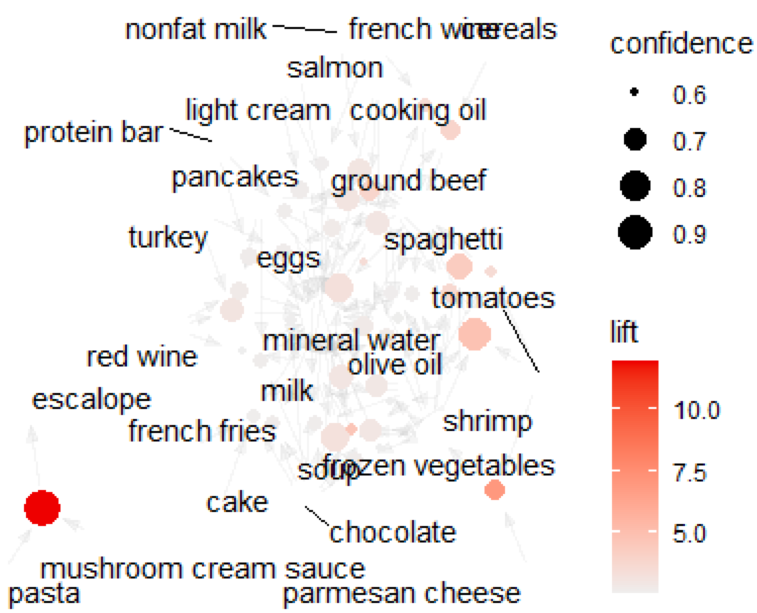
```
rules3 <- apriori (data, parameter = list(supp = 0.002, conf = 0.6))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.002      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
```

```
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.01s].
## sorting and recoding items ... [115 item(s)] done [0.00s].
## creating transaction tree ... done [0.02s].
## checking subsets of size 1 2 3 4 5 done [0.02s].
## writing ... [43 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(rules3[1:5])
```

```
##     lhs                              rhs             support
confidence
## [1] {nonfat milk, spaghetti}      => {mineral water} 0.002399680 0.7200000
## [2] {mushroom cream sauce, pasta} => {escalope}      0.002532996 0.9500000
## [3] {ground beef, light cream}    => {mineral water} 0.002133049 0.6400000
## [4] {light cream, spaghetti}      => {mineral water} 0.003199573 0.6000000
## [5] {ground beef, protein bar}    => {mineral water} 0.002266364 0.6296296
##     coverage    lift      count
## [1] 0.003332889  3.020537 18
## [2] 0.002666311 11.976387 19
## [3] 0.003332889  2.684922 16
## [4] 0.005332622  2.517114 24
## [5] 0.003599520  2.641416 17
```

```
plot(rules3, method = "graph",
     measure = "confidence", shading = "lift")
```



The difference of using higher alpha or support and lower confidence is observed above but with the same results.

Conclusion

For the marketing department my advise would be to stack some of the items customers likely to buy after buying a specific item, for example

- Stacking: mushroom cream sauce, pasta together with escalope
- Stacking: ground beef, nonfat milk together with mineral water
- Stacking: bacon, pancakes together with spaghetti

This is will also help in customer marketing reach, it would be a waste to approach someone who buys bacon and pancakes to buy escalope cause its not an item they buy, however, approaching them for spaghetti is a wise marketing decision.