Association Rules

2022-03-30

Research Question

You are a Data analyst at Carrefour Kenya and are currently undertaking a project that will inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax). Your project has been divided into four parts where you'll explore a recent marketing dataset by performing various unsupervised learning techniques and later providing recommendations based on your insights.

Association Rules

This section will require that you create association rules that will allow you to identify relationships between variables in the dataset. You are provided with a separate dataset that comprises groups of items that will be associated with others. Just like in the other sections, you will also be required to provide insights for your analysis.

Defining the question

i)Specifying the Data Analytic Question

Create association rules that will allow you to identify relationships between variables in the dataset. You are provided with a separate dataset that comprises groups of items that will be associated with others.

ii)Defining the Metric for Success

To be able to establish relationships between items in our dataset using the apropriate algorithm.

Understanding the context

Association analysis is an unsupervised method that is used to discover patterns that occur within a given dataset by identifying relationships between observations and variables from a dataset.

We use the apriori algorithm to build the association rules.

The 3 important measure parameters of association rules include;

1. Support- How popular an itemset is, as measured by the proportion of transactions in which an itemset appears
2. Confidence - How often one item A appears whenever another item B appears in a transaction. This is usually a conditional probability.

3. Lift- Used to measure the performance of the rule when compared against the entire data set.

Dataset link http://bit.ly/SupermarketDatasetII

```
#Import necessary libraries
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

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

```
# We will use read.transactions fuction which will load data from comma-separated files and convert the
path <-"http://bit.ly/SupermarketDatasetII"
df<-read.transactions(path, sep = ",", rm.duplicates=TRUE)
```

```
## distribution of transactions with duplicates:
## 1
## 5
```

```
df
```

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

```
#Verifying the object's class
# This should show us transactions as the type of data that we will need
class(df)
```

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

```
#Previewing our first 5 transactions
inspect(df[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}
```

```
# alternatively we can preview the items that make up our dataset
items<-as.data.frame(itemLabels(df))
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
```

```
#Generating a summary of the sales dataset to get a sense of the most purchased items.
summary(df)
```

```
## 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
```

From the summary above, we can see that the most purchased item in our dataset is mineral water followed by eggs.

```
#Exploring the absolute and relative frequency of some items from some transactions
itemFrequency(df[, 8:10],type = "absolute")
```

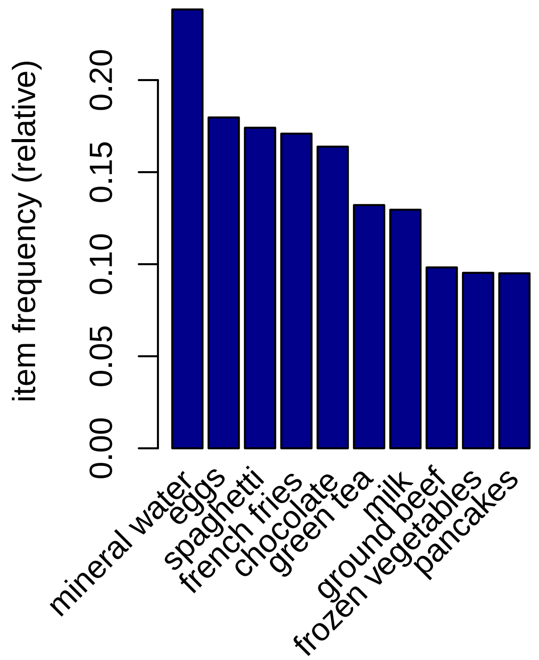
```
##   black tea blueberries  body spray
##         107          69          86
```

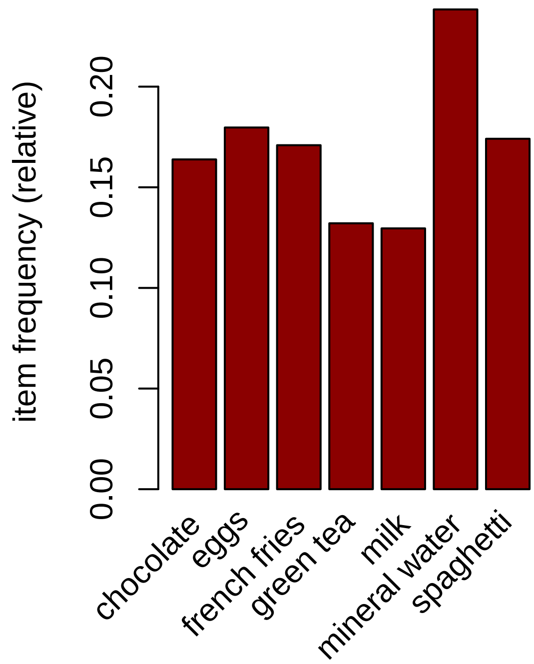
```
round(itemFrequency(df[, 8:10],type = "relative")*100,2)
```

```
##   black tea blueberries  body spray
##        1.43        0.92        1.15
```

Visualizing the top 10 most common items in the sales dataset and the items whose relative importance is at least 5%

```
par(mfrow = c(1, 2))
# plot the frequency of items
itemFrequencyPlot(df, topN = 10,col="darkblue")
itemFrequencyPlot(df, support = 0.1,col="darkred")
```





From the first graph we can see that mineral water, eggs,spaghetti were the most bought items. From the second graph we can see that chocolate, eggs,french fries had the highest support

Building a model based on association rules

```r
# Building apriori model with Min Support as 0.001 and confidence as 0.7.
rules1 <- apriori (df, parameter = list(supp = 0.001, conf = 0.7))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    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 [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.01s].
## writing ... [200 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules1
```

```
## set of 200 rules
```

Using the minimum support as 0.001 and confidence as 0.8, we were able to attain a set of 200 rules.

```
# Building a apriori model with Min Support as 0.002 and confidence as 0.8.
rules2 <- apriori (df,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].
```

```
# Building apriori model with Min Support as 0.002 and confidence as 0.6.
rules3 <- apriori (df, parameter = list(supp = 0.001, conf = 0.6))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.001      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 [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [545 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules2
```

```
## set of 2 rules
```

```
rules3
```

```
## set of 545 rules
```

e increased the minimum support of 0.001 to 0.002 , confidence of 0.7 to 0.8 and model rules went from 200 to only 2. This would lead us to understand that using a high level of support can make the model lose interesting rules. In the second example, we decreased the minimum confidence level to 0.6 and the number of model rules went from 200 to 545. This would mean that using a low confidence level increases the number of rules to quite an extent and many will not be useful.

```
#checking the summary of the rules 1
summary(rules1)
```

```
## set of 200 rules
##
## rule length distribution (lhs + rhs):sizes
##   3   4   5   6
##  44 122  33   1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   4.000   4.000   3.955   4.000   6.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001067   Min.   :0.7000   Min.   :0.001067   Min.   : 2.937
##  1st Qu.:0.001067   1st Qu.:0.7273   1st Qu.:0.001466   1st Qu.: 3.088
##  Median :0.001200   Median :0.7500   Median :0.001466   Median : 3.616
##  Mean   :0.001330   Mean   :0.7767   Mean   :0.001728   Mean   : 4.160
##  3rd Qu.:0.001466   3rd Qu.:0.8139   3rd Qu.:0.001866   3rd Qu.: 4.418
##  Max.   :0.003066   Max.   :1.0000   Max.   :0.004133   Max.   :12.722
##      count
##  Min.   : 8.00
##  1st Qu.: 8.00
##  Median : 9.00
##  Mean   : 9.98
##  3rd Qu.:11.00
##  Max.   :23.00
##
```

```
## mining info:
##  data ntransactions support confidence
##    df          7501   0.001        0.7
##                                                                      call
##  apriori(data = df, parameter = list(supp = 0.001, conf = 0.7))
```

```r
# Observing rules built in our model i.e. first 5 model rules
inspect(rules1[1:5])
```

```
##     lhs                              rhs             support     confidence
## [1] {frozen smoothie, spinach} => {mineral water} 0.001066524 0.8888889
## [2] {spaghetti, spinach}       => {mineral water} 0.001333156 0.7142857
## [3] {olive oil, strong cheese} => {spaghetti}     0.001066524 0.7272727
## [4] {milk, strong cheese}      => {mineral water} 0.001599787 0.7058824
## [5] {green beans, ground beef} => {spaghetti}     0.001066524 0.7272727
##     coverage    lift     count
## [1] 0.001199840 3.729058  8
## [2] 0.001866418 2.996564 10
## [3] 0.001466471 4.177085  8
## [4] 0.002266364 2.961311 12
## [5] 0.001466471 4.177085  8
```

From the preview above, we can conclude ;

-If a customer buys frozen smoothie,spinach they are 88.9% likely to also buy mineral water as this was observed in 8 transactions within our dataset.

-If a customer buys olive oil,strong cheese, they are 72.7% likely to also buy spaghetti as this was observed in 8 transactions within our dataset.

```r
# Ordering the rules the level of confidence then looking at the first five rules.
rules1<-sort(rules1, by="confidence", decreasing=TRUE)
inspect(rules1[1:5])
```

```
##     lhs                           rhs                 support confidence    coverage      lift count
## [1] {french fries,
##      mushroom cream sauce,
##      pasta}                    => {escalope}      0.001066524       1.00 0.001066524 12.606723     8
## [2] {ground beef,
##      light cream,
##      olive oil}                => {mineral water} 0.001199840       1.00 0.001199840  4.195190     9
## [3] {cake,
##      meatballs,
##      mineral water}            => {milk}          0.001066524       1.00 0.001066524  7.717078     8
## [4] {cake,
##      olive oil,
##      shrimp}                   => {mineral water} 0.001199840       1.00 0.001199840  4.195190     9
## [5] {mushroom cream sauce,
##      pasta}                    => {escalope}      0.002532996       0.95 0.002666311 11.976387    19
```

From the above preview, we can conclude the following :

-If a customer buys french fries, mushroom cream sauce and pasta they are 100% likely to also buy escalope as this was observed in 8 transactions within our dataset.

-If a customer buys ground beef,light cream,olive oil, they are 100% likely to also buy mineral water as this was observed in 9 transactions within our dataset.

```
# If we're interested in making a promotion relating to the sale of escalope,
# we could create a subset of rules concerning these products
# ---
# This would tell us the items that the customers bought before purchasing escalope
escalope <- subset(rules1, subset = rhs %pin% "escalope")

# Then order by confidence
escalope<-sort(escalope, by="confidence", decreasing=TRUE)
inspect(escalope[1:2])
```

```
##     lhs                          rhs            support confidence    coverage     lift count
## [1] {french fries,
##      mushroom cream sauce,
##      pasta}                   => {escalope} 0.001066524       1.00 0.001066524 12.60672     8
## [2] {mushroom cream sauce,
##      pasta}                   => {escalope} 0.002532996       0.95 0.002666311 11.97639    19
```

From the above preview, we can conclude the following :

-We should market escalope to people who buy french fries,mushroom cream sauce,pasta because there is 100% chance that they will buy escalope

-We should market escalope to people who buy mushroom cream sauce,pasta because there is 95% chance that they will buy escalope

```
# What if we wanted to determine items that customers might buy
# who have previously bought eggs?
# Subset the rules
eggs <- subset(rules1, subset = lhs %pin% "eggs")
# Order by confidence
eggs<-sort(eggs, by="confidence", decreasing=TRUE)
# inspect top 5
inspect(eggs[1:5])
```

```
##     lhs                        rhs                 support confidence    coverage      lift count
## [1] {eggs,
##      mineral water,
##      pasta}                 => {shrimp}        0.001333156  0.9090909 0.001466471 12.722185    10
## [2] {brownies,
##      eggs,
##      ground beef}           => {mineral water} 0.001066524  0.8888889 0.001199840  3.729058     8
## [3] {chocolate,
##      eggs,
##      frozen vegetables,
##      ground beef}           => {mineral water} 0.001466471  0.8461538 0.001733102  3.549776    11
## [4] {chocolate,
##      eggs,
##      olive oil,
##      spaghetti}             => {mineral water} 0.001199840  0.8181818 0.001466471  3.432428     9
## [5] {cooking oil,
##      eggs,
##      olive oil}             => {mineral water} 0.001066524  0.8000000 0.001333156  3.356152     8
```

From the above preview, we can conclude the following :

-We should market shrimp to people who buy eggs because there is 90.9% chance that they will buy shrimp

CONCLUSIONS

-Mineral Water is the most purchased item in our dataset followed by eggs,spaghetti, french fries and chocolate.

-If a customer buys french fries, mushroom cream sauce and pasta they are 100% likely to also buy escalope as this was observed in 8 transactions within our dataset.

-If a customer buys ground beef,light cream,olive oil, they are 100% likely to also buy mineral water as this was observed in 9 transactions within our dataset.

RECOMMENDATIONS

-We recommend that the supermarket should stock up on items such as eggs,spaghetti, french fries, chocolate,mineral water, fat and yoghurt as they were the most purchased and thus will guarantee the highest number of sales. -We recommend that the supermarket should market escalope to people who buy french fries,mushroom cream sauce,pasta because there is 100% chance that they will buy escalope

-We recommend that the supermarket should consider placing french fries, mushroom cream sauce and pasta in similar or neighboring aisles as we are 100% confident that a customer would purchase all these items and thus this would reduce the time they take looking for the items separately.