Association Analysis

Joshua Kibuye

2022-04-04

Carrefour Marketing Analysis (Associative Analysis)

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. # 1. Defining the Question ## a) Specifying the Data Analytic Question. Create association rules that will allow you to identify relationships between variables in the dataset.

b) Defining the Metric for Success

Creating association rules to identify relationships between variables. ## c) Understanding the context 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.

d) Recording the Experimental Design

- Data cleaning
- Performing extensive exploratory data analysis where applicable.
- Detecting anomalies in our data.

```
#Loading packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 4.1.3

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

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

2. Data Understanding

```
#Loading the dataset
df <- read.transactions('http://bit.ly/SupermarketDatasetII', sep = ",", rm.duplicates=TRUE)
```

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

```
#Looking at information of our dataset
df
```

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

```
#Checking the classes of the dataset
class(df)
```

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

```r
#Loading items
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}
```

```r
#Getting a summary of our data
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
```

```
# Exploring the frequency of some items
itemFrequency(df[, 7:10],type = "absolute")
```

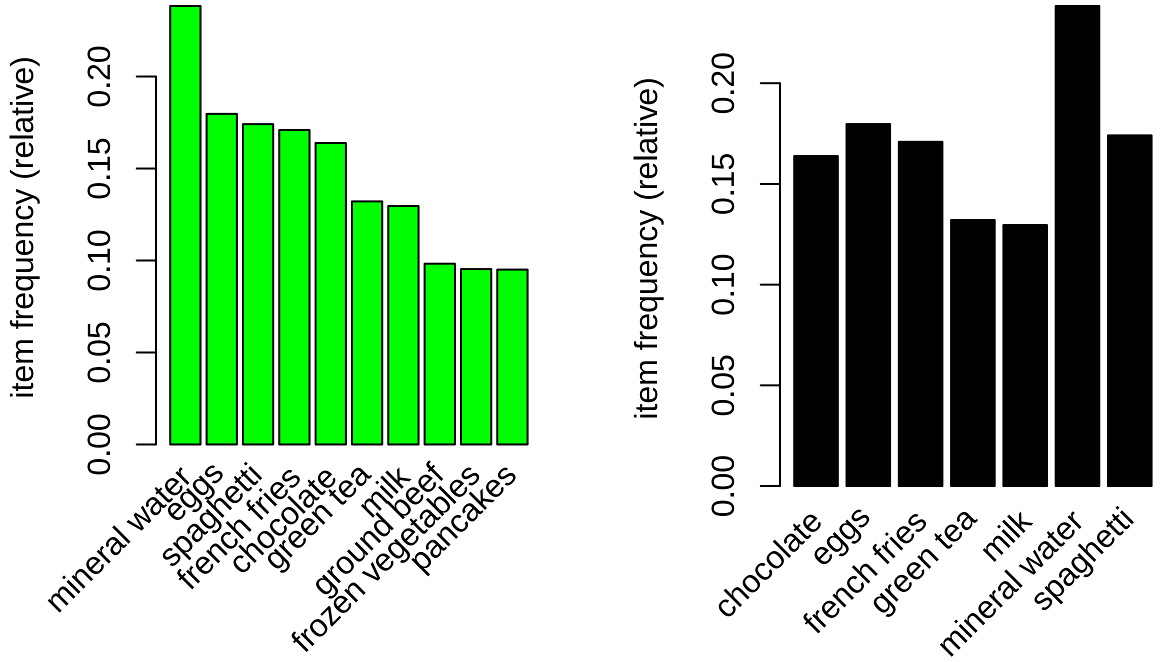
```
## barbecue sauce      black tea    blueberries     body spray
##             81            107             69             86
```

```
#Relative frequency of some items
round(itemFrequency(df[, 7:10],type = "relative")*100,2)
```

```
## barbecue sauce      black tea    blueberries     body spray
##           1.08           1.43           0.92           1.15
```

```
#Visualizing the most common items. We will go for 10 and for relative importance of at least 10%
par(mfrow = c(1, 2))

# plot the frequency of items
itemFrequencyPlot(df, topN = 10,col="green")
itemFrequencyPlot(df, support = 0.1,col="black")
```



# Building Association Rules Model

```
# We use Min Support as 0.001 and confidence as 0.8
rules <- apriori (df, 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 [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [74 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#Looking at the number of rules
rules
```

```
## set of 74 rules
```

We get a set of 74 rules using 0.001 support and 0.8 as confidence.

```
#Inspecting the rules
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
```

Person buying frozen smoothie and spinach is likely to buy mineral water and so on.

```
rules<-sort(rules, by="confidence", decreasing=TRUE)
inspect(rules[1:10])
```

```
##      lhs                        rhs                 support confidence    coverage      lift count
## [1]  {french fries,                                                                              
##        mushroom cream sauce,                                                                     
##        pasta}                 => {escalope}      0.001066524  1.0000000 0.001066524 12.606723     8
## [2]  {ground beef,                                                                               
##        light cream,                                                                              
##        olive oil}             => {mineral water} 0.001199840  1.0000000 0.001199840  4.195190     9
## [3]  {cake,                                                                                      
##        meatballs,                                                                                
##        mineral water}         => {milk}          0.001066524  1.0000000 0.001066524  7.717078     8
## [4]  {cake,                                                                                      
##        olive oil,                                                                                
##        shrimp}                => {mineral water} 0.001199840  1.0000000 0.001199840  4.195190     9
## [5]  {mushroom cream sauce,                                                                      
##        pasta}                 => {escalope}      0.002532996  0.9500000 0.002666311 11.976387    19
## [6]  {red wine,                                                                                  
##        soup}                  => {mineral water} 0.001866418  0.9333333 0.001999733  3.915511    14
## [7]  {eggs,                                                                                      
##        mineral water,                                                                            
##        pasta}                 => {shrimp}        0.001333156  0.9090909 0.001466471 12.722185    10
## [8]  {herb & pepper,                                                                             
##        mineral water,                                                                            
##        rice}                  => {ground beef}   0.001333156  0.9090909 0.001466471  9.252498    10
## [9]  {ground beef,
##       pancakes,
##       whole wheat rice}      => {mineral water} 0.001333156  0.9090909 0.001466471  3.813809    10
## [10] {frozen vegetables,
##       milk,
##       spaghetti,
##       turkey}                => {mineral water} 0.001199840  0.9000000 0.001333156  3.775671     9
```

```
# Making a promotion for mineral water by creating a subset of rules concerning the product
# This would tell us the items that the customers bought before purchasing mineral water

mineral_water <- subset(rules, subset = rhs %pin% "mineral water")

# Ordering by condidence
mineral_water<-sort(mineral_water, by="confidence", decreasing=TRUE)
inspect(mineral_water[1:5])
```

```
##     lhs                     rhs                 support confidence    coverage     lift count
## [1] {ground beef,
##      light cream,
##      olive oil}          => {mineral water} 0.001199840  1.0000000 0.001199840 4.195190     9
## [2] {cake,
##      olive oil,
##      shrimp}             => {mineral water} 0.001199840  1.0000000 0.001199840 4.195190     9
## [3] {red wine,
##      soup}               => {mineral water} 0.001866418  0.9333333 0.001999733 3.915511    14
## [4] {ground beef,
##      pancakes,
##      whole wheat rice}   => {mineral water} 0.001333156  0.9090909 0.001466471 3.813809    10
## [5] {frozen vegetables,
##      milk,
##      spaghetti,
##      turkey}             => {mineral water} 0.001199840  0.9000000 0.001333156 3.775671     9
```

The above are items that people buy most with mineral water.

```
#Creating a promotion based on milk
milk <- subset(rules, subset = rhs %pin% "milk" )

# Then order by confidence
milk <-sort(milk, by="confidence", decreasing=TRUE)
inspect(milk[])
```

```
##     lhs                                    rhs    support     confidence
## [1] {cake, meatballs, mineral water}    => {milk} 0.001066524 1.0000000
## [2] {escalope, hot dogs, mineral water} => {milk} 0.001066524 0.8888889
## [3] {meatballs, whole wheat pasta}      => {milk} 0.001333156 0.8333333
## [4] {black tea, frozen smoothie}        => {milk} 0.001199840 0.8181818
## [5] {burgers, ground beef, olive oil}   => {milk} 0.001066524 0.8000000
##     coverage    lift     count
## [1] 0.001066524 7.717078  8
## [2] 0.001199840 6.859625  8
## [3] 0.001599787 6.430898 10
## [4] 0.001466471 6.313973  9
## [5] 0.001333156 6.173663  8
```

People are like to buy cake, meatballs, and mineral water with milk than any other items. # Conclusion Mineral water is the most purchased items and should be used more for promotion. Many customers buy egg and milk. Spaghetti was a top 3 bought item with eggs and mineral water.

Reccomendations

Most bought items can be put in package deals. Associated items should be close to each other.