Association Rules

2022-06-13

1. Defining the question

a) Specifying the question

form the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).

b) Defining the metrics of success

explore a recent marketing dataset by performing various unsupervised learning techniques and later providing recommendations based on your insights. Check if there are any anomalies in the dataset

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.

Part 3: 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.

d) Data relevance

The data is relevant since it was provided by the company itself and can be used to answer the question

Import necessary Libraries

```
library(arules)
```

```
## Loading required package: Matrix

##
## Attaching package: 'arules'

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

Loading the dataset

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

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

```
head(ts)
```

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

```
# Verification of the objects class
class(ts)
```

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

```
# Preview the head of dataset
inspect(ts[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
# Preview the last items in the dataset
last <- as.data.frame(itemLabels(ts))
colnames(last) <- "Item"
head(last, 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
```

```r
# Summary to see the most purchased item
summary(ts)
```

```
## 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
```

The most purchased item was mineral water

```r
# Getting the absolute frequency
itemFrequency(ts, type="absolute")
```

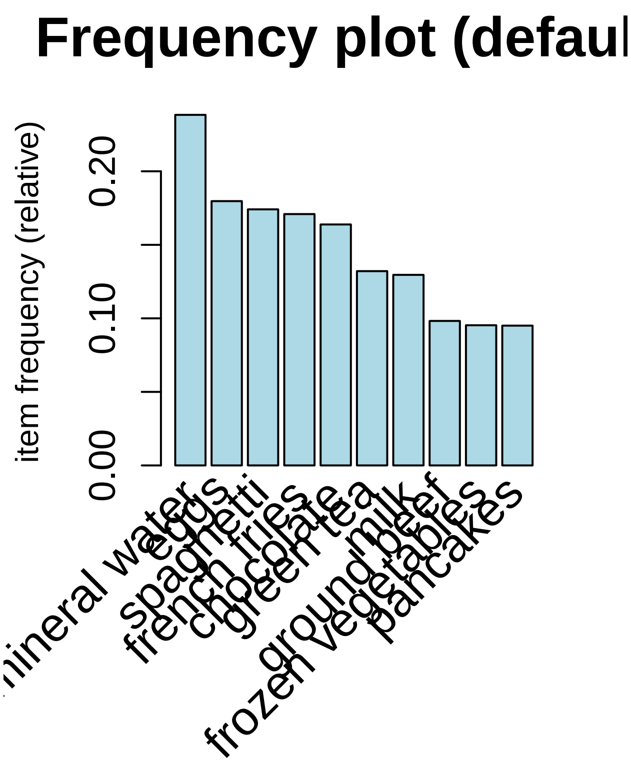
```
##              almonds    antioxydant juice            asparagus
##                  153                   67                   36
##              avocado          babies food                bacon
```

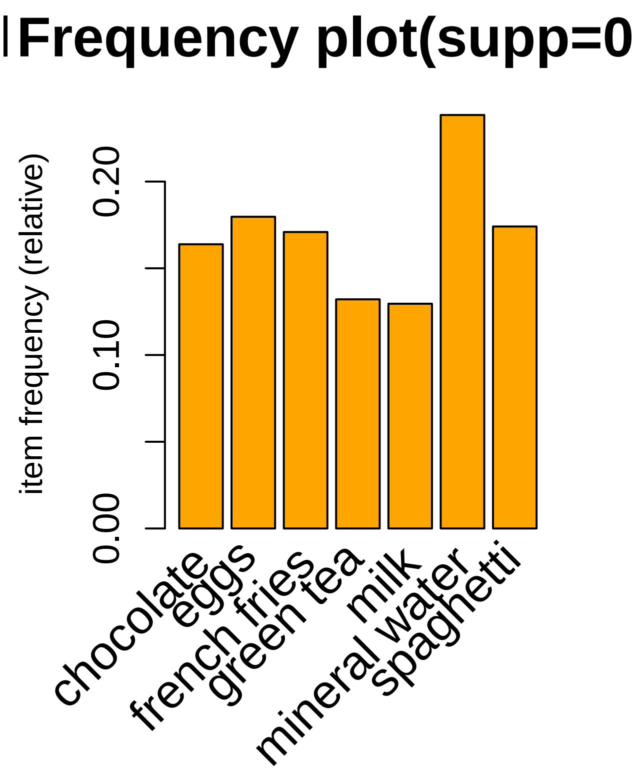
```
##                  250                   34                   65
##       barbecue sauce            black tea           blueberries
##                   81                  107                   69
##           body spray              bramble             brownies
##                   86                   14                  253
##            bug spray         burger sauce              burgers
##                   65                   44                  654
##               butter                 cake           candy bars
##                  226                  608                   73
##              carrots          cauliflower              cereals
##                  115                   36                  193
##            champagne              chicken                chili
##                  351                  450                   46
##            chocolate      chocolate bread              chutney
##                 1229                   32                   31
##                cider  clothes accessories              cookies
##                   79                   63                  603
##          cooking oil                 corn       cottage cheese
##                  383                   36                  239
##                cream         dessert wine             eggplant
##                    7                   33                   99
##                 eggs           energy bar         energy drink
##                 1348                  203                  200
##             escalope extra dark chocolate            flax seed
##                  595                   90                   68
##         french fries          french wine          fresh bread
##                 1282                  169                  323
##           fresh tuna       fromage blanc      frozen smoothie
##                  167                  102                  475
##    frozen vegetables       gluten free bar        grated cheese
##                  715                   52                  393
##          green beans         green grapes            green tea
##                   65                   68                  991
##          ground beef                 gums                  ham
##                  737                  101                  199
##     hand protein bar        herb & pepper                honey
##                   39                  371                  356
##             hot dogs              ketchup          light cream
##                  243                   33                  117
##           light mayo        low fat yogurt            magazines
##                  204                  574                   82
##        mashed potato           mayonnaise            meatballs
##                   31                   46                  157
##               melons                 milk        mineral water
##                   90                  972                 1788
##                 mint       mint green tea              muffins
##                  131                   42                  181
## mushroom cream sauce              napkins          nonfat milk
##                  143                    5                   78
##              oatmeal                  oil            olive oil
##                   33                  173                  494
##             pancakes      parmesan cheese                pasta
##                  713                  149                  118
##               pepper             pet food              pickles
```

```
##                199                49                45
##        protein bar          red wine              rice
##                139               211               141
##              salad            salmon              salt
##                 37               319                69
##           sandwich           shallot           shampoo
##                 34                58                37
##             shrimp              soda              soup
##                536                47               379
##          spaghetti   sparkling water           spinach
##               1306                47                53
##       strawberries     strong cheese               tea
##                160                58                29
##       tomato juice      tomato sauce          tomatoes
##                228               106               513
##         toothpaste            turkey    vegetables mix
##                 61               469               193
##        water spray        white wine  whole weat flour
##                  3               124                70
##  whole wheat pasta whole wheat rice              yams
##                221               439                86
##        yogurt cake          zucchini
##                205                71
```

```
# Plotting the most frequent items

options(repr.plot.width = 40, repr.plot.height = 10)
par(mfrow = c(1, 2))
itemFrequencyPlot(ts, topN = 10,col="lightblue", main = "Frequency plot (default)", cex = 1.5, cex.main=
itemFrequencyPlot(ts, support = 0.1,col="orange", main = "Frequency plot(supp=0.1)", cex = 1.5, cex.mai
```





Building a model based on association rules

```r
# supp = 0.001, conf = 0.8

rule <- apriori (ts, 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 [0.01s].
```

```
## writing ... [74 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
rule
```

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

We get a set of 74 rules

```r
# Min support 0.002 and confidence 0.8
rule_1 <- apriori (ts, 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].
```

```r
rule_1
```

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

We get a set of 2 rules

```r
# Min supp = 0.001, confidence = 0.6

rule_2 <- apriori (ts, 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 [0.01s].
## writing ... [545 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rule_2
```

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

We get 545 rules

We shall use the first one that gave us 74 rules

```
# Observing rules built in our model i.e. first 10 model rules

inspect(rule[1:10])
```

```
##      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
## [6]  {milk, pasta}                    => {shrimp}        0.001599787 0.8571429
## [7]  {cooking oil, fromage blanc}     => {mineral water} 0.001199840 0.8181818
## [8]  {black tea, salmon}              => {mineral water} 0.001066524 0.8000000
## [9]  {black tea, frozen smoothie}     => {milk}          0.001199840 0.8181818
## [10] {red wine, tomato sauce}         => {chocolate}     0.001066524 0.8000000
##      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
## [6]  0.001866418 11.995203 12
## [7]  0.001466471  3.432428  9
## [8]  0.001333156  3.356152  8
## [9]  0.001466471  6.313973  9
## [10] 0.001333156  4.882669  8
```

If a customer buys frozen smoothie, spinach they have a 88% chance to buy mineral water. This is seen by 8 transactions in the data set

```
# Inspecting the first 5 rules with the highest lift

inspect(head(rule, n = 5, by = "lift"))
```

```
##     lhs                          rhs                      support confidence    coverage      lift coun
## [1] {eggs,
##      mineral water,
##      pasta}                   => {shrimp}             0.001333156   0.9090909 0.001466471 12.722185     1
## [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
```

We can conclude that, if a customer buys eggs, mineral water and pasta, they are 90% most likely to buy shrimp

```r
# Inspecting the first 5 rules with the highest confidence

inspect(head(rule, n = 5, by = "confidence"))
```

```
##     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
```

We can conclude that, if a customer buys french fries, mushroom cream sauce, and pasta, they are 100% most likely to buy escalope

```r
# Looking at the least popular transactions

itm <- itemFrequency(ts, type = "relative")
head(sort(itm), n = 10)
```

```
##     water spray         napkins           cream         bramble             tea
##    0.0003999467    0.0006665778    0.0009332089    0.0018664178    0.0038661512
##         chutney  mashed potato chocolate bread    dessert wine         ketchup
##    0.0041327823    0.0041327823    0.0042660979    0.0043994134    0.0043994134
```

```r
# If we are interested in making a promotion relating to the sale of eggs
# Let us look at what people buy after buying eggs

eggs = subset(rule, subset = lhs %pin% "eggs")

# Then order by confidence
eggs = sort(eggs, by="confidence", decreasing=TRUE)
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
```

We should market shrimp to those who buy eggs since there is a 90.1% chance they will buy it

```r
# Let us look at what people buy after buying ground beef

beef = subset(rule, subset = lhs %pin% "ground beef")
beef
```

```
## set of 12 rules
```

```r
# Then order by confidence
beef = sort(beef, by="confidence", decreasing=TRUE)
inspect(beef[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] {ground beef,
##      pancakes,
##      whole wheat rice} => {mineral water}     0.001333156  0.9090909 0.001466471 3.813809    10
## [3] {brownies,
##      eggs,
##      ground beef}       => {mineral water}     0.001066524  0.8888889 0.001199840 3.729058     8
## [4] {ground beef,
##      salmon,
##      shrimp}            => {spaghetti}         0.001066524  0.8888889 0.001199840 5.105326     8
## [5] {chocolate,
##      ground beef,
##      milk,
##      mineral water,
##      spaghetti}         => {frozen vegetables} 0.001066524  0.8888889 0.001199840 9.325253     8
```

We should market mineral water to those who buy ground beef because there is a 100% chance they will make a purchase

Conclusion

Mineral water was the most purchased item If a customer makes a purchase of french fries, mushroom cream sauce and pasta they are 100% most likely to buy escalope We should market shrimp to those who buy eggs since there is a 90.1% chance they will buy it

Recommendations

We recommend that the supermarket restocks on eggs, mineral water, spaghetti and french fries since they were the most purchased items