Market Basket Analysis in R, Market Basket Analysis is very popular. In this tutorial, the main idea is to identify the purchase pattern of the products, “what goes with what”.

Based on this information Data Scientist can make decisions for increasing business profit.

Many examples are available, suppose if you are login into amazon prime, they will suggest some of the interesting movies to you based on your previous watch views.

Ultimately, they analyze our viewing pattern and present it to you in a very beautiful way.

pdftools & pdftk in R

What is market basket analysis?

Basically, it is the study of “what goes with what”.

Examples are customers who bought X item also bought Y item or in another case what symptoms go with what diagnosis.

In most cases, companies are not interested why Y bought with X, they just want to identify the patterns.

Its also called association rules or affinity analysis.

Majorly used for recommender systems like Netflix, Amazon, Big basket, etc..

Market Basket Analysis in R

How to clean datasets in R

Getting Data

```
mydata<-read.csv("D:/RStudio/MarketBasketAnalysis/
MarketBasketData.csv",header=T, colClasses = "factor")
str(mydata)
data.frame':  1000 obs. of  14 variables:
 $ Bag            : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2
...
 $ Blush          : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 1 1 2
...
 $ Nail.Polish    : Factor w/ 2 levels "No","Yes": 2 2 1 2 1 1 2 2 1 2
...
 $ Brushes        : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 2 1 2
...
 $ Concealer      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 1
...
 $ Eyebrow.Pencils: Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 1 1
...
 $ Bronzer        : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1
...
 $ Lip.liner      : Factor w/ 2 levels "No","Yes": 2 2 2 1 2 1 2 1 1 1
...
 $ Mascara        : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 2 1 2
```

```
...
 $ Eye.shadow     : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 1 2 2 1 2
...
 $ Foundation     : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 2 1 2 1
...
 $ Lip.Gloss      : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 1 2 2 1 1
...
 $ Lipstick       : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 1 1 1 2 1
...
 $ Eyeliner       : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 1 1 1 1
...
```

Total 1000 observations and 14 variables and all the columns are loaded as factor variables.

Association rules

```
library(arules)
rules <- apriori(mydata)
Apriori
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1
1     10  rules TRUE
Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
Absolute minimum support count: 100
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[28 item(s), 1000 transaction(s)] done [0.00s].
sorting and recoding items ... [26 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.03s].
writing ... [68880 rule(s)] done [0.04s].
creating S4 object  ... done [0.03s].
```

Its provides default setting results with 80% confidence and maximum of number of items is 10.

68880 rules is very huge we need to cut down the rules for easy analysis and interpretation.

Naïve Bayes Classification in R

Rules with specified parameter values

```
rules <- apriori(mydata,parameter = list(minlen=2, maxlen=10,supp=.7,
conf=.8))
rules
Apriori
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.7
2     10  rules TRUE
```

```
Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
Absolute minimum support count: 700
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[28 item(s), 1000 transaction(s)] done [0.00s].
sorting and recoding items ... [6 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [15 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Now total number of rule is 15. We considered minimum purchase of 2 items and maximum is 10.

```
inspect(rules)
```

```
lhs                                    rhs                   support
confidence coverage lift     count
 [1]  {Nail.Polish=No}                    => {Brushes=No}         0.720
1.0000000  0.720    1.175088 720
 [2]  {Brushes=No}                        => {Nail.Polish=No}     0.720
0.8460635  0.851    1.175088 720
 [3]  {Lip.liner=No}                      => {Bag=No}             0.732
0.9556136  0.766    1.010162 732
 [4]  {Lip.liner=No}                      => {Eyebrow.Pencils=No} 0.734
0.9582245  0.766    1.000234 734
 [5]  {Brushes=No}                        => {Bag=No}             0.817
0.9600470  0.851    1.014849 817
 [6]  {Bag=No}                            => {Brushes=No}         0.817
0.8636364  0.946    1.014849 817
 [7]  {Brushes=No}                        => {Eyebrow.Pencils=No} 0.820
0.9635723  0.851    1.005817 820
 [8]  {Eyebrow.Pencils=No}                => {Brushes=No}         0.820
0.8559499  0.958    1.005817 820
 [9]  {Bag=No}                            => {Eyebrow.Pencils=No} 0.909
0.9608879  0.946    1.003015 909
 [10] {Eyebrow.Pencils=No}                => {Bag=No}             0.909
0.9488518  0.958    1.003015 909
 [11] {Bag=No,Lip.liner=No}               => {Eyebrow.Pencils=No} 0.703
0.9603825  0.732    1.002487 703
 [12] {Eyebrow.Pencils=No,Lip.liner=No} => {Bag=No}             0.703
0.9577657  0.734    1.012437 703
 [13] {Bag=No,Brushes=No}                 => {Eyebrow.Pencils=No} 0.789
0.9657283  0.817    1.008067 789
 [14] {Brushes=No,Eyebrow.Pencils=No}   => {Bag=No}             0.789
0.9621951  0.820    1.017120 789
 [15] {Bag=No,Eyebrow.Pencils=No}       => {Brushes=No}         0.789
0.8679868  0.909    1.019961 789
```

Finding interesting rules-1

Now we need to identify the interesting rules. The item is not purchased then the particular rule

we are interested, so we want to identify the rules with items are purchased.

Deep Neural Network in R

```
rules <- apriori(mydata,parameter = list(minlen=2, maxlen=3,supp=.01,
conf=.7),appearance=list(rhs=c("Foundation=Yes"),lhs=c("Bag=Yes",
"Blush=Yes"),default="lhs"))
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
minlen maxlen target  ext
        0.7    0.1    1 none FALSE            TRUE       5    0.01
2      3  rules TRUE
Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
Absolute minimum support count: 10
set item appearances ...[3 item(s)] done [0.00s].
set transactions ...[28 item(s), 1000 transaction(s)] done [0.00s].
sorting and recoding items ... [28 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [19 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
inspect(rules)
     lhs                                    rhs              support
confidence coverage lift     count
[1]  {Lip.Gloss=Yes}                     => {Foundation=Yes} 0.356
0.7265306  0.490    1.355468 356
[2]  {Bag=Yes,Lip.Gloss=Yes}             => {Foundation=Yes} 0.021
0.7000000  0.030    1.305970  21
[3]  {Lip.Gloss=Yes,Lipstick=Yes}        => {Foundation=Yes} 0.116
0.7341772  0.158    1.369734 116
[4]  {Mascara=Yes,Lip.Gloss=Yes}         => {Foundation=Yes} 0.130
0.7182320  0.181    1.339985 130
[5]  {Mascara=Yes,Eye.shadow=No}         => {Foundation=Yes} 0.026
0.7222222  0.036    1.347430  26
[6]  {Eye.shadow=Yes,Lip.Gloss=Yes}      => {Foundation=Yes} 0.146
  0.7263682  0.201    1.355164 146
[7]  {Mascara=No,Eye.shadow=Yes}         => {Foundation=Yes} 0.045
0.7500000  0.060    1.399254  45
[8]  {Lip.Gloss=Yes,Eyeliner=No}         => {Foundation=Yes} 0.200
0.7604563  0.263    1.418762 200
[9]  {Concealer=No,Lip.Gloss=Yes}        => {Foundation=Yes} 0.215
0.7904412  0.272    1.474704 215
[10] {Eye.shadow=No,Lip.Gloss=Yes}       => {Foundation=Yes} 0.210
0.7266436  0.289    1.355678 210
[11] {Blush=No,Lip.Gloss=Yes}            => {Foundation=Yes} 0.237
0.7596154  0.312    1.417193 237
[12] {Mascara=No,Lip.Gloss=Yes}          => {Foundation=Yes} 0.226
0.7313916  0.309    1.364537 226
[13] {Lip.Gloss=Yes,Lipstick=No}         => {Foundation=Yes} 0.240
0.7228916  0.332    1.348678 240
```

```
[14] {Nail.Polish=No,Lip.Gloss=Yes}     => {Foundation=Yes} 0.267
0.7500000  0.356    1.399254 267
15] {Bronzer=No,Lip.Gloss=Yes}         => {Foundation=Yes} 0.295
0.8452722  0.349
1.577000 295
[16] {Lip.liner=No,Lip.Gloss=Yes}       => {Foundation=Yes} 0.310
0.8288770  0.374    1.546412 310
[17] {Brushes=No,Lip.Gloss=Yes}         => {Foundation=Yes} 0.313
0.7417062  0.422    1.383780 313
[18] {Bag=No,Lip.Gloss=Yes}             => {Foundation=Yes} 0.335
0.7282609  0.460    1.358696 335
[19] {Eyebrow.Pencils=No,Lip.Gloss=Yes} => {Foundation=Yes} 0.345
0.7278481  0.474    1.357926 345
In this support indicate the size of the transactions. Support is
higher that much good it is.
```

Finding interesting rules-2

```
rules <- apriori(mydata,parameter = list(minlen=2, maxlen=5,supp=.1,
conf=.5),appearance=list(rhs=c("Foundation=Yes"),lhs=c("Bag=Yes",
"Blush=Yes", "Nail.Polish=Yes", "Brushes=Yes", "Concealer=Yes",
"Eyebrow.Pencils=Yes", "Bronzer=Yes", "Lip.liner=Yes", "Mascara=Yes",
"Eye.shadow=Yes","Lip.Gloss=Yes", "Lipstick=Yes",
"Eyeliner=Yes"),default="none"))
quality(rules)<-round(quality(rules),digits=3)
rules.sorted <- sort(rules, by="lift")
```

Remove Redundancy

```
redundant <- is.redundant(rules, measure="confidence")
which(redundant)
[1] 11 12 13 14 15 16 17 18 19 20 21 22
rules.pruned <- rules[!redundant]
rules.pruned <- sort(rules.pruned, by="lift")
inspect(rules.pruned)
     lhs                              rhs              support
confidence coverage lift  count
[1]  {Lip.Gloss=Yes,Lipstick=Yes} => {Foundation=Yes} 0.116
0.734      0.158    1.370 116
[2]  {Lip.Gloss=Yes}              => {Foundation=Yes} 0.356
0.727      0.490    1.355 356
[3]  {Eye.shadow=Yes}             => {Foundation=Yes} 0.211
0.554      0.381    1.033 211
[4]  {Blush=Yes,Mascara=Yes}      => {Foundation=Yes} 0.101
0.549      0.184    1.024 101
[5]  {Mascara=Yes}                => {Foundation=Yes} 0.192
  0.538      0.357    1.003 192
[6]  {Blush=Yes}                  => {Foundation=Yes} 0.192
0.529      0.363    0.987 192
[7]  {Concealer=Yes}              => {Foundation=Yes} 0.231
0.523      0.442    0.975 231
[8]  {Eyeliner=Yes}               => {Foundation=Yes} 0.238
```

```
0.521      0.457    0.972 238
[9]  {Lipstick=Yes}               => {Foundation=Yes} 0.167    0.519
     0.322    0.968 167
[10] {Nail.Polish=Yes}            => {Foundation=Yes} 0.143
0.511      0.280    0.953 143
```

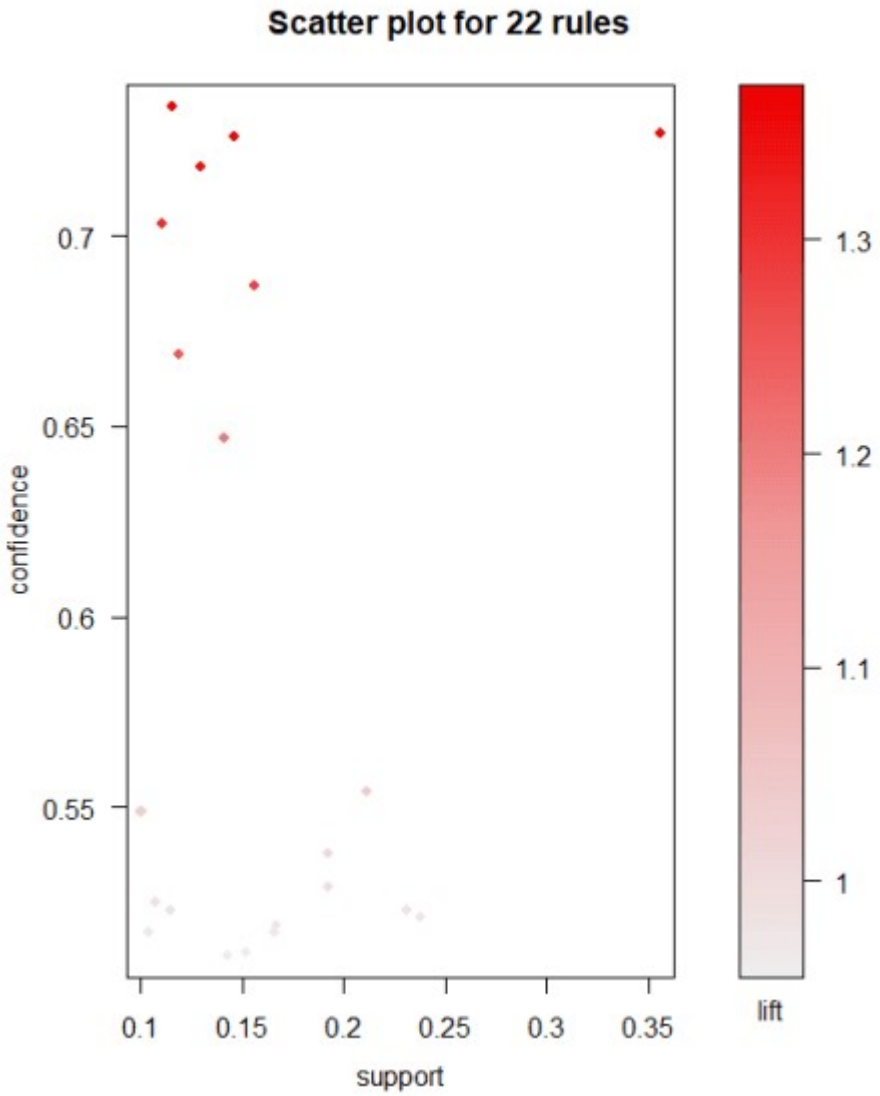
Now you can see count and confidence is much higher.

Graphs and Charts

Lets visualize the association rules.

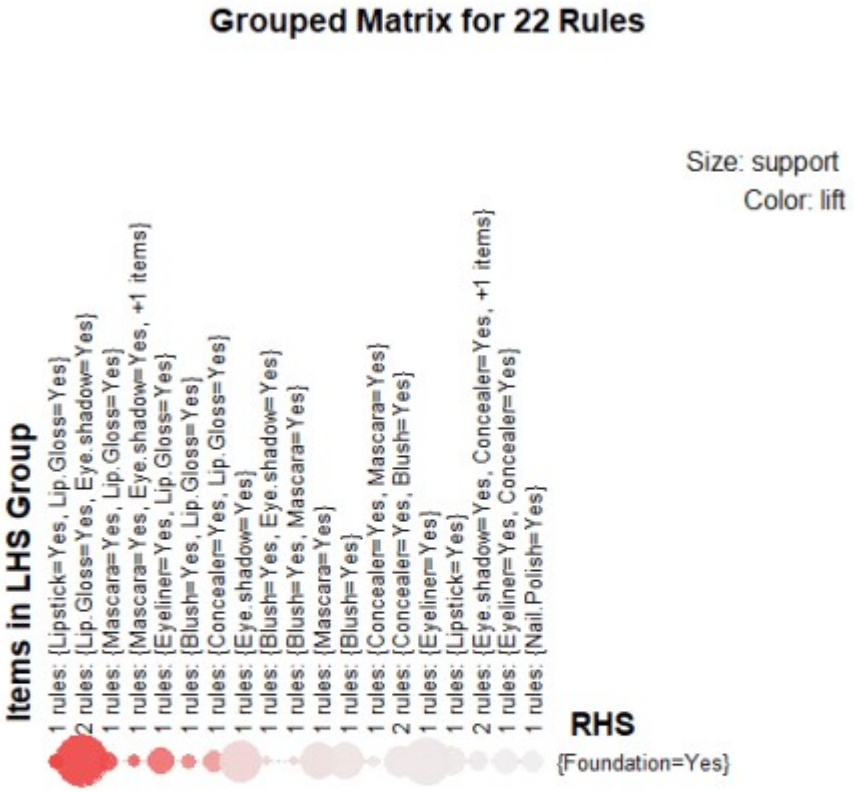
LSTM Networks in R

```
library(arulesViz)
plot(rules)
```



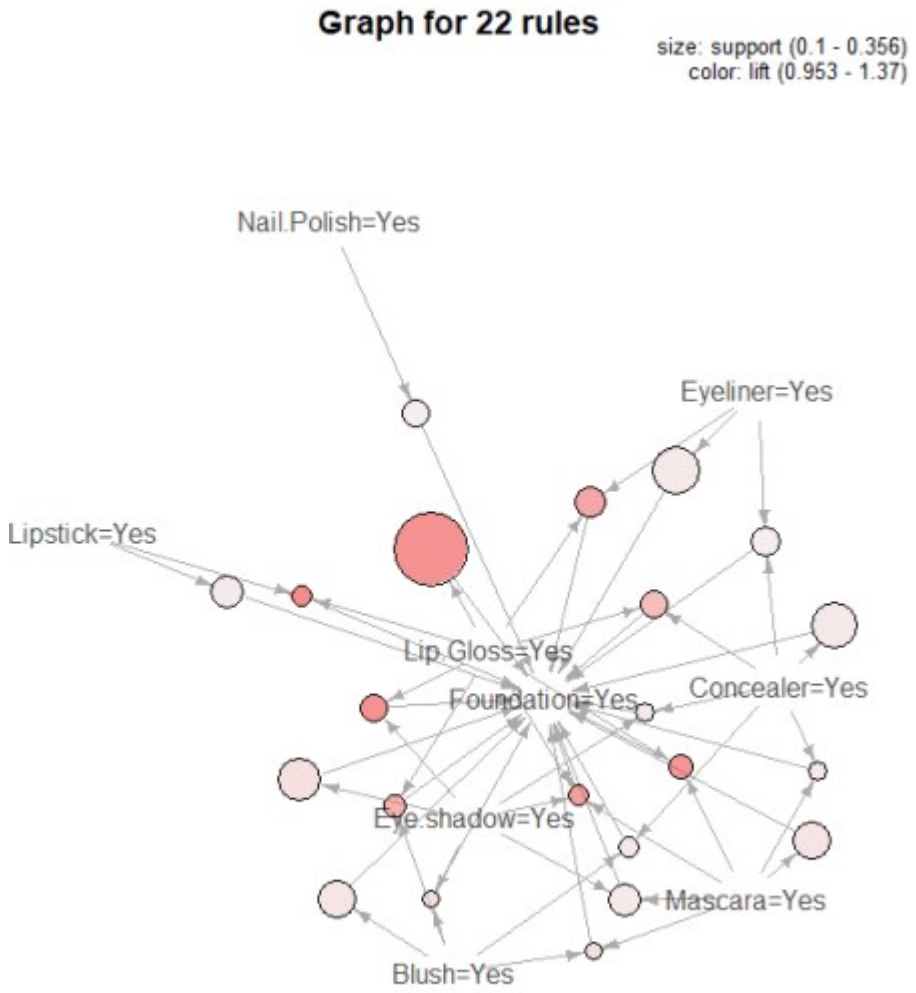
Scatterplot with all rules with confidence and support.

```
plot(rules,method="grouped")
```



Ballon plot listed with all the rules.

```
plot(rules,method="graph")
```



Network diagram with support and confidence. Whenever purchasing lip gloss consumers are purchasing foundation also.