Association Rules

Dian Ramadhani

08/01/2020

Association Rules

Association rules merupakan suatu metode data mining yang bertujuan untuk mencari sekumpulan items yang sering muncul bersamaan.

Install Packages

```
# Menginstall package(s)
install.packages("readr") # membaca file
install.packages("here") # menampilkan direktori
install.packages("arules") # algoritma asosiasi
install.packages("arulesViz") # visualisasi asosiasi
```

Import Library

```
# Mengaktifkan package(s)
library(here)
library(arules)
library(arulesViz)
```

Menampilkan Direktori

```
# Mengetahui direktori proyek
here()
```

Import Data

Data yang digunakan kali ini yaitu dataset Groceries yang telah tersedia di package arules

```
# Mengimport data bawaan dari package arules
data(Groceries)

# Melihat Contoh 6 Data Transaksi
inspect(head(Groceries))
```

```
##     items
## [1] {citrus fruit,
##      semi-finished bread,
##      margarine,
##      ready soups}
## [2] {tropical fruit,
##      yogurt,
##      coffee}
## [3] {whole milk}
## [4] {pip fruit,
##      yogurt,
##      cream cheese ,
```

```
##      meat spreads}
## [5] {other vegetables,
##      whole milk,
##      condensed milk,
##      long life bakery product}
## [6] {whole milk,
##      butter,
##      yogurt,
##      rice,
##      abrasive cleaner}
```

Data ini merupakan daftar barang - barang yang dibelanjakan oleh konsumen dalam satu transaksi. Melalui data ini, dapat diketahui barang - barang apa saja yang sering dibeli secara bersamaan oleh konsumen.

Frequent Itemset

Mencari itemset yang dianggap sering muncul.

```
# Menampilkan attribute dan struktur data
freq.itemset <- apriori(Groceries,
                        parameter = list(minlen = 1,
                                         maxlen = 1,
                                         target = "frequent itemsets",
                                         support = 0.002))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5   0.002      1
##  maxlen            target   ext
##       1 frequent itemsets FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 19
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [147 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1
```

```
## Warning in apriori(Groceries, parameter = list(minlen = 1, maxlen = 1,
## target = "frequent itemsets", : Mining stopped (maxlen reached). Only
## patterns up to a length of 1 returned!
```

```
##  done [0.00s].
## writing ... [147 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Menampilkan summary frequent itemset
summary(freq.itemset)
```

```
## set of 147 itemsets
```

```
##
## most frequent items:
## frankfurter     sausage  liver loaf         ham        meat     (Other)
##           1           1           1           1           1         142
##
## element (itemset/transaction) length distribution:sizes
##   1
## 147
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
##
## summary of quality measures:
##     support             count
##  Min.   :0.002034   Min.   :  20.0
##  1st Qu.:0.005897   1st Qu.:  58.0
##  Median :0.014235   Median : 140.0
##  Mean   :0.029851   Mean   : 293.6
##  3rd Qu.:0.035486   3rd Qu.: 349.0
##  Max.   :0.255516   Max.   :2513.0
##
## includes transaction ID lists: FALSE
##
## mining info:
##       data ntransactions support confidence
##  Groceries          9835   0.002          1
```

```
# Menampilkan 10 data dengan nilai support tertinggi
inspect(head(sort(freq.itemset, by = "support"), 10))
```

```
##      items              support    count
## [1]  {whole milk}       0.25551601 2513
## [2]  {other vegetables} 0.19349263 1903
## [3]  {rolls/buns}       0.18393493 1809
## [4]  {soda}             0.17437722 1715
## [5]  {yogurt}           0.13950178 1372
## [6]  {bottled water}    0.11052364 1087
## [7]  {root vegetables}  0.10899847 1072
## [8]  {tropical fruit}   0.10493137 1032
## [9]  {shopping bags}    0.09852567  969
## [10] {sausage}          0.09395018  924
```

Membuat Rules

Data yang dianggap sering muncul kemudian digunakan untuk membuat rules

```
# Membuat rules
rules <- apriori(Groceries, parameter = list(support = 0.002, confidence = 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 FALSE
```

```
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 19
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [147 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [376 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Menampilkan summary rules
summary(rules)
```

```
## set of 376 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4   5
##   2 136 211  27
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   4.000   3.699   4.000   5.000
##
## summary of quality measures:
##     support           confidence          lift           count
##  Min.   :0.002034   Min.   :0.6000   Min.   :2.348   Min.   :20.00
##  1st Qu.:0.002237   1st Qu.:0.6208   1st Qu.:2.491   1st Qu.:22.00
##  Median :0.002491   Median :0.6485   Median :2.735   Median :24.50
##  Mean   :0.002914   Mean   :0.6657   Mean   :2.912   Mean   :28.66
##  3rd Qu.:0.003254   3rd Qu.:0.6983   3rd Qu.:3.192   3rd Qu.:32.00
##  Max.   :0.009354   Max.   :0.8857   Max.   :5.838   Max.   :92.00
##
## mining info:
##       data ntransactions support confidence
##  Groceries          9835   0.002        0.6
```

```
# Menampilkan 5 data dengan nilai lift tertinggi
inspect(head(sort(rules, by = "lift"), 5))
```

```
##     lhs                   rhs                   support confidence     lift count
## [1] {beef,
##      citrus fruit,
##      other vegetables}  => {root vegetables} 0.002135231  0.6363636 5.838280    21
## [2] {citrus fruit,
##      tropical fruit,
##      other vegetables,
##      whole milk}        => {root vegetables} 0.003152008  0.6326531 5.804238    31
## [3] {citrus fruit,
##      other vegetables,
##      frozen vegetables} => {root vegetables} 0.002033554  0.6250000 5.734025    20
## [4] {beef,
##      tropical fruit,
```

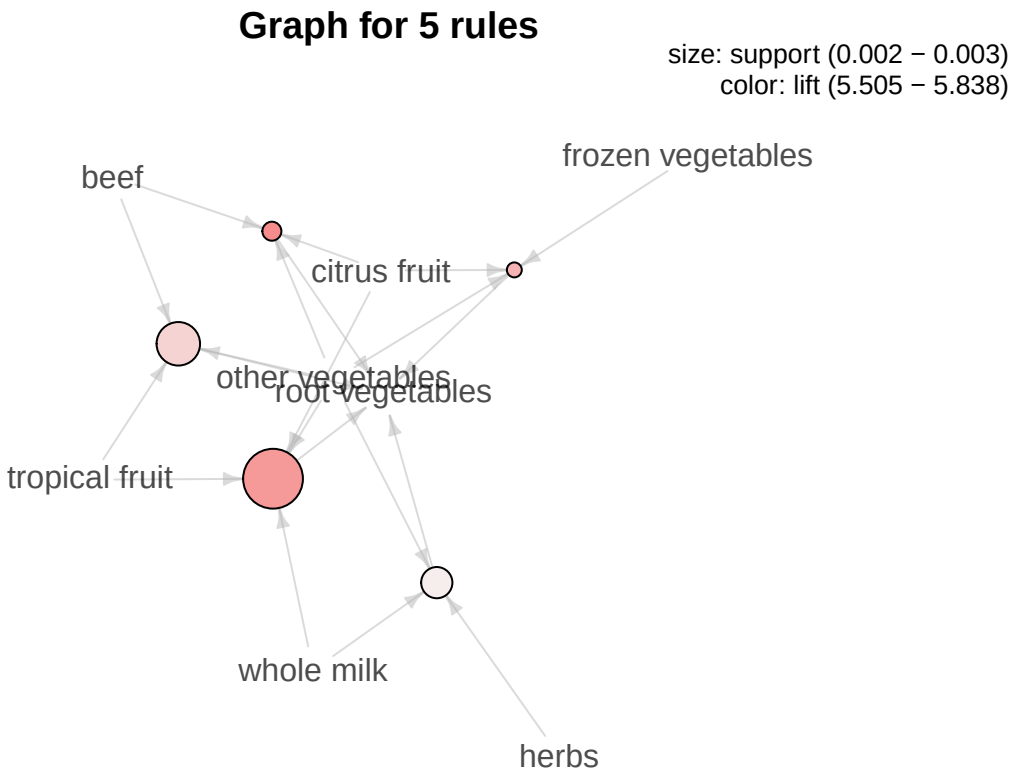
```
##      other vegetables}  => {root vegetables} 0.002745297  0.6136364 5.629770    27
## [5] {herbs,
##      other vegetables,
##      whole milk}        => {root vegetables} 0.002440264  0.6000000 5.504664    24
```

Visualisasi Rules

```r
# Visualisasi rules
toprules <- head(sort(rules, by = "lift"), 5)
plot(toprules, method = "graph", control = list(type = "items"))
```

```
## Warning: Unknown control parameters: type
```

```
## Available control parameters (with default values):
## main  =  Graph for 5 rules
## nodeColors    =  c("#66CC6680", "#9999CC80")
## nodeCol   =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE121
## edgeCol   =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353
## alpha     =  0.5
## cex   =  1
## itemLabels    =  TRUE
## labelCol  =  #000000B3
## measureLabels     =  FALSE
## precision     =  3
## layout    =  NULL
## layoutParams  =  list()
## arrowSize     =  0.5
## engine    =  igraph
## plot  =  TRUE
## plot_options  =  list()
## max   =  100
## verbose   =  FALSE
```



Simpan Rules

```
# Simpan rules dalam bentuk Excel
df <- as(rules, "data.frame")
write_csv(df, here("data", "processed", "association_rules.csv"))
```