

Exp 5

Prof Meenakshi Garg

8/2/2020

Implementation and analysis of Apriori Algorithm using Market Basket Analysis.

```
setwd("E:/R Orientation")
setwd("E:/R Orientation")
# we are creating a data frame by importing csv file
mba_data<-read.csv("data_apriori.csv")
# you can check top 6 observation using head() function
trans <- split(mba_data$Products, mba_data$Customer_Id,"transactions")
head(trans)

## $`1`
## [1] "bread" "butter" "eggs" "milk"
##
## $`2`
## [1] "beer" "bread" "cheese" "chips" "mayo" "soda"
##
## $`3`
## [1] "bread" "butter" "eggs" "milk" "oranges"
##
## $`4`
## [1] "bread" "butter" "eggs" "milk" "soda"
##
## $`5`
## [1] "buns" "chips" "beer" "mustard" "pickels" "soda"
##
## $`6`
## [1] "bread" "butter" "chocolate" "eggs" "milk"
# loading arules library
library(arules)

## Loading required package: Matrix
##
## Attaching package: 'arules'
## The following objects are masked from 'package:base':
##
## abbreviate, write
rules = apriori(trans, parameter=list(support=0.5, confidence=0.9,maxlen=3,minlen=2))

## Warning in asMethod(object): removing duplicated items in transactions ## Apriori

##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen ## 0.9 0.1
```

```

1 none FALSE TRUE 5 0.5 2 ## maxlen target ext
## 3 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 ...[15 item(s), 15 transaction(s)] done [0.00s]. ## sorting and
recoding items ... [4 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3

## Warning in apriori(trans, parameter = list(support = 0.5, confidence = 0.9, : ## Mining
stopped (maxlen reached). Only patterns up to a length of 3 returned!

## done [0.00s].
## writing ... [11 rule(s)] done [0.00s].
## creating S4 object ... done [0.00s].
# to get the rules
inspect(rules)

## lhs rhs support confidence coverage lift count ## [1] {eggs} => {milk} 0.6000000 1
0.6000000 1.666667 9 ## [2] {milk} => {eggs} 0.6000000 1 0.6000000 1.666667 9 ## [3]
{butter} => {bread} 0.6000000 1 0.6000000 1.250000 9 ## [4] {butter,eggs} => {milk}
0.5333333 1 0.5333333 1.666667 8 ## [5] {butter,milk} => {eggs} 0.5333333 1 0.5333333
1.666667 8 ## [6] {bread,eggs} => {milk} 0.5333333 1 0.5333333 1.666667 8 ## [7]
{bread,milk} => {eggs} 0.5333333 1 0.5333333 1.666667 8 ## [8] {butter,eggs} => {bread}
0.5333333 1 0.5333333 1.250000 8 ## [9] {bread,eggs} => {butter} 0.5333333 1 0.5333333
1.666667 8 ## [10] {butter,milk} => {bread} 0.5333333 1 0.5333333 1.250000 8 ## [11]
{bread,milk} => {butter} 0.5333333 1 0.5333333 1.666667 8

```