Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 9**	
Title of Lab Assignment: Implementation and analysis of Apriori Algorithm using Market Basket Analysis.		
DOP: 21-10-2023	**DOS: 27-10-2023**	
CO Mapped: **CO6**	**PO Mapped:** **PO1, PO2, PO3, PO4, PO5,** **PO7, PO12, PSO1, PSO2**	**Signature:**

Practical No. 9

Aim: Implementation and analysis of Apriori Algorithm using Market Basket Analysis.

Theory:

Market Basket Analysis is a form of frequent itemset mining that examines consumer purchasing patterns by identifying relationships between the many goods in their "shopping baskets." By getting insight into which goods are commonly purchased together by customers, businesses may build marketing strategies based on the finding of these relationships. Market Basket Analysis is a method of determining the value of a market basket.

MBA is most often used to help in cross-selling and up-selling. If you know that customers who buy trousers also buy belts, for example, you may advertise the belts on the same page or offer them as part of a bundle to try to boost sales. You may also advertise one product while seeing an increase in the other. Customers' purchase patterns are depicted using "Association Rules" in Market Basket Analysis. A rule's interestingness is determined by two metrics: support and confidence.

Example:

Tea_powder => sugar [support = 4%, confidence = 70%]

a. A support of 2% for the above rule states that 2% of all the transactions under analysis show that tea powder and sugar are purchased together.
   support(B => C) = P(B U C)

b. A confidence of 70% means that 70% of the customers who purchased tea powder also bought the sugar.

c. Lift is a metric that helps us figure out if combining two products increases our chances of making a sale.

Packages / Functions Used:

a. arules: It is used for displaying, manipulating, and analyzing transaction data and patterns (frequent itemsets and association rules)

b. **inspect():** It summarizes all relevant options, plots and statistics that should be usually considered.

c. **apriori():** From a given collection of transaction data, apriori() creates the most relevant set of rules. It also demonstrates the rules' support, confidence, and lifting. The relative strength of the rules may be determined using these three criteria.

Problem Statement: Implementation and analysis of Apriori Algorithm using Market Basket Analysis.

Code (Script):

#reference:-https://www.analyticsvidhya.com/blog/2021/10/end-to-end-introduction-to-marketbasket-analysis-in-r/

```
# Install the libaries
# install.packages('arules')

# Load the libraries
library(arules)

# Load the data set
data(Groceries)

# Get the rules
grocery_rules = apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8))
grocery_rules # shows that it is a set of 410 rules

# Show top 10 rules out of total 410 rules,
# Show only 2 decimal digits after the decimal
options(digits = 2)
inspect(grocery_rules[1:10])

# Sorting rules by confidence
grocery_rules = sort(grocery_rules, by = "confidence", decreasing = TRUE)
```

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

# What type of customers will buy whole milk? (whole milk is rhs)
whole_milk_rules = apriori(data = Groceries,
                           parameter = list(supp = 0.001, conf = 0.08),
                           appearance = list(default = "lhs", rhs = "whole milk")
                          )

inspect(whole_milk_rules[1:10])
whole_milk_rules = sort(whole_milk_rules, by = "confidence", decreasing = TRUE)

inspect(whole_milk_rules[1:10])

# If a customer buys "whole milk" then what else will they buy? (whole milk is lhs)
whole_milk_rules = apriori(data = Groceries,
                           parameter = list(supp = 0.001, conf = 0.08, minlen = 2),
                           appearance = list(default = "rhs", lhs = "whole milk")
                          )

inspect(whole_milk_rules[1:10])

whole_milk_rules = sort(whole_milk_rules, by = "confidence", decreasing = TRUE)

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

Output:

Get the rules

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/
> # Load the libraries
> library(arules)
>
> # Load the data set
> data(Groceries)
>
> # Get the rules
> grocery_rules = apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5   0.001      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: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [157 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 ... [410 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> grocery_rules # shows that it is a set of 410 rules
set of 410 rules
>
```

Sort the rules in decreasing order of confidence

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/
> # Show top 10 rules out of total 410 rules,
> # show only 2 decimal digits after the decimal
> options(digits = 2)
> inspect(grocery_rules[1:10])
     lhs                                                rhs                support confidence coverage lift count
[1]  {rice, sugar}                                   => {whole milk}       0.0012  1          0.0012   3.9  12
[2]  {canned fish, hygiene articles}                 => {whole milk}       0.0011  1          0.0011   3.9  11
[3]  {root vegetables, butter, rice}                 => {whole milk}       0.0010  1          0.0010   3.9  10
[4]  {root vegetables, whipped/sour cream, flour}    => {whole milk}       0.0017  1          0.0017   3.9  17
[5]  {butter, soft cheese, domestic eggs}            => {whole milk}       0.0010  1          0.0010   3.9  10
[6]  {citrus fruit, root vegetables, soft cheese}    => {other vegetables} 0.0010  1          0.0010   5.2  10
[7]  {pip fruit, butter, hygiene articles}           => {whole milk}       0.0010  1          0.0010   3.9  10
[8]  {root vegetables, whipped/sour cream, hygiene articles} => {whole milk}       0.0010  1          0.0010   3.9  10
[9]  {pip fruit, root vegetables, hygiene articles}  => {whole milk}       0.0010  1          0.0010   3.9  10
[10] {cream cheese , domestic eggs, sugar}           => {whole milk}       0.0011  1          0.0011   3.9  11
>
> # Sorting rules by confidence
> grocery_rules = sort(grocery_rules, by = "confidence", decreasing = TRUE)
> inspect(grocery_rules[1:10])
     lhs                                                rhs                support confidence coverage lift count
[1]  {rice, sugar}                                   => {whole milk}       0.0012  1          0.0012   3.9  12
[2]  {canned fish, hygiene articles}                 => {whole milk}       0.0011  1          0.0011   3.9  11
[3]  {root vegetables, butter, rice}                 => {whole milk}       0.0010  1          0.0010   3.9  10
[4]  {root vegetables, whipped/sour cream, flour}    => {whole milk}       0.0017  1          0.0017   3.9  17
[5]  {butter, soft cheese, domestic eggs}            => {whole milk}       0.0010  1          0.0010   3.9  10
[6]  {citrus fruit, root vegetables, soft cheese}    => {other vegetables} 0.0010  1          0.0010   5.2  10
[7]  {pip fruit, butter, hygiene articles}           => {whole milk}       0.0010  1          0.0010   3.9  10
[8]  {root vegetables, whipped/sour cream, hygiene articles} => {whole milk}       0.0010  1          0.0010   3.9  10
[9]  {pip fruit, root vegetables, hygiene articles}  => {whole milk}       0.0010  1          0.0010   3.9  10
[10] {cream cheese , domestic eggs, sugar}           => {whole milk}       0.0011  1          0.0011   3.9  11
>
```

What type of customers will buy whole milk? (whole milk is rhs)

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/
> # What type of customers will buy whole milk? (whole milk is rhs)
> whole_milk_rules = apriori(data = Groceries,
+                            parameter = list(supp = 0.001, conf = 0.08),
+                            appearance = list(default = "lhs", rhs = "whole milk")
+                           )
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
       0.08    0.1    1 none FALSE            TRUE       5   0.001      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: 9

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [157 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 ... [3765 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> inspect(whole_milk_rules[1:10])
     lhs                   rhs          support confidence coverage lift count
[1]  {}                 => {whole milk} 0.2555  0.26       1.0000   1.0  2513
[2]  {honey}            => {whole milk} 0.0011  0.73       0.0015   2.9    11
[3]  {soap}             => {whole milk} 0.0011  0.42       0.0026   1.7    11
[4]  {cocoa drinks}     => {whole milk} 0.0013  0.59       0.0022   2.3    13
[5]  {pudding powder}   => {whole milk} 0.0013  0.57       0.0023   2.2    13
[6]  {cooking chocolate} => {whole milk} 0.0013  0.52       0.0025   2.0    13
[7]  {nuts/prunes}      => {whole milk} 0.0012  0.36       0.0034   1.4    12
[8]  {potato products}  => {whole milk} 0.0012  0.43       0.0028   1.7    12
[9]  {artif. sweetener} => {whole milk} 0.0011  0.34       0.0033   1.3    11
[10] {canned fruit}     => {whole milk} 0.0013  0.41       0.0033   1.6    13
> whole_milk_rules = sort(whole_milk_rules, by = "confidence", decreasing = TRUE)
> inspect(whole_milk_rules[1:10])
     lhs                                                rhs          support confidence coverage lift count
[1]  {rice, sugar}                                   => {whole milk} 0.0012  1          0.0012   3.9  12
[2]  {canned fish, hygiene articles}                 => {whole milk} 0.0011  1          0.0011   3.9  11
[3]  {root vegetables, butter, rice}                 => {whole milk} 0.0010  1          0.0010   3.9  10
[4]  {root vegetables, whipped/sour cream, flour}    => {whole milk} 0.0017  1          0.0017   3.9  17
[5]  {butter, soft cheese, domestic eggs}            => {whole milk} 0.0010  1          0.0010   3.9  10
[6]  {pip fruit, butter, hygiene articles}           => {whole milk} 0.0010  1          0.0010   3.9  10
[7]  {root vegetables, whipped/sour cream, hygiene articles} => {whole milk} 0.0010  1          0.0010   3.9  10
[8]  {pip fruit, root vegetables, hygiene articles}  => {whole milk} 0.0010  1          0.0010   3.9  10
[9]  {cream cheese , domestic eggs, sugar}           => {whole milk} 0.0011  1          0.0011   3.9  11
[10] {curd, domestic eggs, sugar}                    => {whole milk} 0.0010  1          0.0010   3.9  10
>
```

If a customer buys "whole milk" then what else will they buy? (whole milk is lhs)

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/
> # If a customer buys "whole milk" then what else will they buy? (whole milk is lhs)
> whole_milk_rules = apriori(data = Groceries,
+                            parameter = list(supp = 0.001, conf = 0.08, minlen = 2),
+                            appearance = list(default = "rhs", lhs = "whole milk")
+                           )
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
       0.08    0.1    1 none FALSE            TRUE       5   0.001      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: 9

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [23 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> inspect(whole_milk_rules[1:10])
     lhs             rhs                    support confidence coverage lift count
[1]  {whole milk} => {beef}                 0.021   0.083      0.26     1.6  209
[2]  {whole milk} => {curd}                 0.026   0.102      0.26     1.9  257
[3]  {whole milk} => {pork}                 0.022   0.087      0.26     1.5  218
[4]  {whole milk} => {frankfurter}          0.021   0.080      0.26     1.4  202
[5]  {whole milk} => {brown bread}          0.025   0.099      0.26     1.5  248
[6]  {whole milk} => {margarine}            0.024   0.095      0.26     1.6  238
[7]  {whole milk} => {butter}               0.028   0.108      0.26     1.9  271
[8]  {whole milk} => {newspapers}           0.027   0.107      0.26     1.3  269
[9]  {whole milk} => {domestic eggs}        0.030   0.117      0.26     1.9  295
[10] {whole milk} => {fruit/vegetable juice} 0.027  0.104      0.26     1.4  262
> whole_milk_rules = sort(whole_milk_rules, by = "confidence", decreasing = TRUE)
> inspect(whole_milk_rules[1:10])
     lhs             rhs                  support confidence coverage lift count
[1]  {whole milk} => {other vegetables}   0.075   0.29       0.26     1.5  736
[2]  {whole milk} => {rolls/buns}         0.057   0.22       0.26     1.2  557
[3]  {whole milk} => {yogurt}             0.056   0.22       0.26     1.6  551
[4]  {whole milk} => {root vegetables}    0.049   0.19       0.26     1.8  481
[5]  {whole milk} => {tropical fruit}     0.042   0.17       0.26     1.6  416
[6]  {whole milk} => {soda}               0.040   0.16       0.26     0.9  394
[7]  {whole milk} => {bottled water}      0.034   0.13       0.26     1.2  338
[8]  {whole milk} => {pastry}             0.033   0.13       0.26     1.5  327
[9]  {whole milk} => {whipped/sour cream} 0.032   0.13       0.26     1.8  317
[10] {whole milk} => {citrus fruit}       0.031   0.12       0.26     1.4  300
>
```

Conclusion: Demonstrated the implementation and analysis of Apriori Algorithm using Market Basket Analysis.