Practical No. 9

Implementation and analysis of Apriori Algorithm using Market Basket Analysis.

Implementing and analyzing the Apriori algorithm for Market Basket Analysis using the R programming language involves several steps. Here's a general guide:

Step 1: Install Required Packages

Make sure to install the necessary packages. In this case, you'll need the `arules` package for association rule mining.

```
R
install.packages("arules")
library(arules)
```

Step 2: Load and Explore Data

Load your transaction data into R. The data should be in a transaction format where each row represents a transaction, and columns represent items.

```
R
# Example data
transactions <- read.transactions("your_data.csv", sep = ",")

# Check the structure of the transaction data
summary(transactions)
```

Step 3: Run Apriori Algorithm

Apply the Apriori algorithm to find frequent itemsets and association rules.

```
R
# Run Apriori algorithm
rules <- apriori(transactions, parameter = list(support = 0.1, confidence = 0.5))

# Display the generated rules
inspect(rules)
```

Step 4: Analyze and Interpret Results

Examine the generated association rules and their metrics.

```
R
# Display rules with additional information
summary(rules)

# Visualize rules (e.g., support vs. confidence)
plot(rules, method = "scatterplot")
```

Step 5: Adjust Parameters

Tweak the parameters of the Apriori algorithm to get desired results. Common parameters include `support` (minimum support threshold) and `confidence` (minimum confidence threshold).

R

```
# Example: Adjusting parameters
rules <- apriori(transactions, parameter = list(support = 0.05, confidence = 0.3))
```

Step 6: Interpretation and Business Insights

Interpret the generated rules and derive actionable insights for business decisions.

R

```
# Example: Filtering rules for specific conditions
rules_filtered <- subset(rules, lift > 1.2 & confidence > 0.6)
inspect(rules_filtered)
```

Step 7: Visualization

Visualize the association rules or other relevant information.

R

```
# Example: Visualizing rules
plot(rules, method = "graph")
```

Step 8: Evaluate and Iterate

Evaluate the performance of the generated rules and iterate if needed. Adjust parameters or preprocess the data to improve results.

R

```
# Example: Evaluating rules
quality(rules)
```

This is a basic outline, and you may need to customize it based on your specific dataset and analysis goals. Ensure that your dataset is appropriately formatted, and you may need to preprocess it to fit the requirements of the Apriori algorithm. Also, keep in mind that the interpretation of the results depends on the context of your specific market and business scenario.

```
>  w1 = read.table("G:/MCA-SEM-I-ADBMS/comm.csv")
>
>  trans = read.transactions("G:/MCA-SEM-I-ADBMS/comm.csv",format = "basket",sep=",")
>  w1
                          V1
1 mobile,headset,charger,pad
2           mobile,charger,,
3          charger,headset,,
4           mobile,headset,,
5        headset,pad,screen,
6   screen,pendrive,headset,
7     mobile,laptop,headset,
8        headset,mobile,pad,
```

```
>  trans
transactions in sparse format with
 8 transactions (rows) and
 7 items (columns)
>
>  itemFrequencyPlot(trans,topN=20,type="absolute")
>
>  rules <- apriori(data=trans,parameter = list(supp=0.001,conf=0.08),
+                   appearance = list(default="lhs",rhs="mobile"),control = list(verbose=F))
>
```

```
>  rules<-sort(rules,decreasing = TRUE,by="confidence")
>
>  inspect(rules[1:10])
     lhs                          rhs      support confidence coverage lift      count
[1]  {laptop}                  => {mobile} 0.125   1.0000000  0.125    1.6000000 1
[2]  {headset, laptop}         => {mobile} 0.125   1.0000000  0.125    1.6000000 1
[3]  {charger, pad}            => {mobile} 0.125   1.0000000  0.125    1.6000000 1
[4]  {charger, headset, pad}   => {mobile} 0.125   1.0000000  0.125    1.6000000 1
[5]  {charger}                 => {mobile} 0.250   0.6666667  0.375    1.0666667 2
[6]  {pad}                     => {mobile} 0.250   0.6666667  0.375    1.0666667 2
[7]  {headset, pad}            => {mobile} 0.250   0.6666667  0.375    1.0666667 2
[8]  {}                        => {mobile} 0.625   0.6250000  1.000    1.0000000 5
[9]  {headset}                 => {mobile} 0.500   0.5714286  0.875    0.9142857 4
[10] {charger, headset}        => {mobile} 0.125   0.5000000  0.250    0.8000000 1
>
>  plot(rules,method = "graph")
```