

PRACTICAL – 12

AIM: - Case Study Project or one of the below domains:

- **Implementation of an Association Rule Mining Algorithm**

❖ Overview:

- Association Rule Mining is a popular technique in data mining used to discover interesting relationships or patterns between variables in large datasets. This method is commonly applied in transactional datasets, like those in retail or e-commerce, to identify purchasing behaviors.

❖ Key Concepts:

1. Frequent Itemsets:

- Groups of items that frequently appear together in transactions.
- Example: In a supermarket, "bread and butter" appearing together in 60% of transactions.

2. Association Rules:

- Statements of the form: **If A, then B**, with a measure of confidence.
- Example: If a customer buys bread, they are 80% likely to buy butter.

3. Evaluation Metrics:

- **Support:** Frequency of an itemset appearing in transactions.

$$\text{Support} = \frac{\text{Transactions containing the itemset}}{\text{Total transactions}}$$

$$\text{Support} = \frac{\text{Transactions containing the itemset}}{\text{Total transactions}}$$
- **Confidence:** Probability of a consequent given an antecedent.

$$\text{Confidence} = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

$$\text{Confidence} = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$
- **Lift:** Strength of the rule compared to random chance.

$$\text{Lift} = \frac{\text{Confidence}}{\text{Support}(B)}$$

$$\text{Lift} = \frac{\text{Confidence}}{\text{Support}(B)}$$

❖ Algorithms:

1. Apriori Algorithm:

- Iteratively finds frequent itemsets and generates rules from them.
- Uses the **Apriori Property**: If an itemset is frequent, its subsets must also be frequent.
- Advantages: Simple and intuitive.
- Disadvantages: Can be slow for large datasets due to multiple scans.

Name: DHRUV SHERE

Enrollment No.:23012022021

Batch: 5IT-B-2

2. FP-Growth Algorithm:

- Uses a tree structure (Frequent Pattern Tree) to mine frequent itemsets without multiple scans of the dataset.
- Advantages: Faster than Apriori, especially for large datasets.
- Disadvantages: Requires more memory.

❖ Use Cases:**1. Market Basket Analysis:**

- Identify product bundles that sell well together.
- Example: Suggest "chips and soda" as a combo deal.

2. Recommendation Systems:

- Suggest additional products based on a customer's current choices.
- Example: Amazon's "Frequently Bought Together."

3. Inventory Optimization:

- Stock frequently co-purchased items together to reduce logistical effort.

❖ Example:

```
# Import necessary libraries
```

```
import pandas as pd
```

```
from mlxtend.frequent_patterns import apriori, association_rules
```

```
from mlxtend.preprocessing import TransactionEncoder
```

```
# Sample dataset - List of transactions
```

```
dataset = [  
    ['Milk', 'Bread', 'Butter'],  
    ['Milk', 'Bread'],  
    ['Milk', 'Butter'],  
    ['Bread', 'Butter'],  
    ['Bread', 'Butter', 'Milk']  
]
```

```
# Step 1: Convert the dataset into a DataFrame format using Transaction Encoder
```

```
te = TransactionEncoder()
```

```
te_ary = te.fit(dataset).transform(dataset)
```

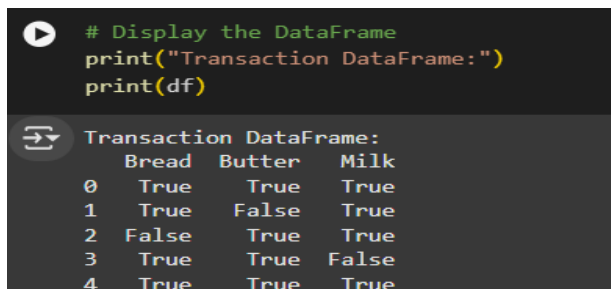
```
df = pd.DataFrame(te_ary, columns=te.columns_)
```

Name: DHRUV SHERE

Enrollment No.:23012022021

Batch: 5IT-B-2

```
# Display the DataFrame
print("Transaction DataFrame:")
print(df)
```

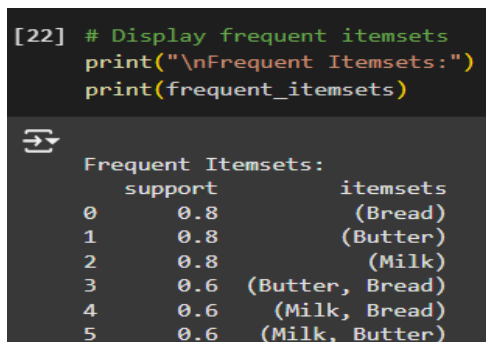


```
# Display the DataFrame
print("Transaction DataFrame:")
print(df)
```

	Bread	Butter	Milk
0	True	True	True
1	True	False	True
2	False	True	True
3	True	True	False
4	True	True	True

```
# Step 2: Apply the Apriori Algorithm to find frequent itemsets
# Set minimum support to 0.6 (60%)
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)
```

```
# Display frequent itemsets
print("\nFrequent Itemsets:")
print(frequent_itemsets)
```

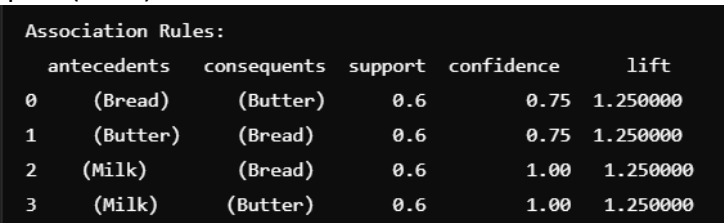


```
[22] # Display frequent itemsets
print("\nFrequent Itemsets:")
print(frequent_itemsets)
```

	support	itemsets
0	0.8	(Bread)
1	0.8	(Butter)
2	0.8	(Milk)
3	0.6	(Butter, Bread)
4	0.6	(Milk, Bread)
5	0.6	(Milk, Butter)

```
# Step 3: Generate association rules from the frequent itemsets
# Set minimum confidence to 0.7 (70%)
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
```

```
# Display the rules
print("\nAssociation Rules:")
print(rules)
```



	antecedents	consequents	support	confidence	lift
0	(Bread)	(Butter)	0.6	0.75	1.250000
1	(Butter)	(Bread)	0.6	0.75	1.250000
2	(Milk)	(Bread)	0.6	1.00	1.250000
3	(Milk)	(Butter)	0.6	1.00	1.250000

Name: DHRUV SHERE

Enrollment No.:23012022021

Batch: 5IT-B-2