PRACTICAL NO. 06

Student Name : Samuel **UID : 24MAI10018**

Branch : ME - CSE (AI & ML) **Section : 24MAI – 1**

Semester : 2nd **Date of Perf. :**

Subject : Machine Learning Lab **Subject Code : 24CSH - 667**

AIM ::

Implementation of Apriori Algorithm using Python.

SOFTWARE REQUIREMENT ::

- Windows 11
- Python IDE (Anaconda Distributor)
- Jupyter Notebook
- Python Libraries (NumPy, Pandas, Matplotlib).

THEORY ::

UNSUPERVISED LEARNING ::

Unsupervised learning is a type of machine learning where the model is trained on data that is neither labelled nor categorized. The system tries to learn the underlying patterns and structure from the input data. Let's break it down further:

KEY CHARACTERISTICS OF UNSUPERVISED LEARNING:

1. No Labelled Data:

   - The training data consists of input features without any associated output labels.
   - The goal is to discover hidden patterns or intrinsic structures within the data.

2. Exploratory:

- Used for exploratory data analysis to identify patterns, groupings, or anomalies.
- Commonly applied in clustering, dimensionality reduction, and association problems.

COMMON ALGORITHMS IN UNSUPERVISED LEARNING:

1. Clustering:

- K-Means Clustering: Partitions the data into K distinct clusters based on similarity.
- Hierarchical Clustering: Builds a hierarchy of clusters by either merging or splitting them.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Forms clusters based on the density of points.

2. Dimensionality Reduction:

- Principal Component Analysis (PCA): Reduces the dimensionality of the data while preserving as much variance as possible.
- t-SNE (t-Distributed Stochastic Neighbour Embedding): Reduces high-dimensional data to two or three dimensions for visualization.
- Autoencoders: Neural networks used to learn compressed representations of data.

3. Association Rule Learning:

- Apriori Algorithm: Identifies frequent item sets and generates association rules.
- Eclat Algorithm: Uses a depth-first search approach to discover frequent item sets.

APRIORI ALGORITHM ::

The Apriori algorithm is a frequent itemset mining algorithm used in association rule learning. It is widely applied in market basket analysis, where it identifies relationships between items in transactional datasets.

STEPS OF APRIORI ALGORITHM

- Set a minimum support threshold to filter frequent itemsets.
- Generate candidate itemsets by combining frequent itemsets.
- Prune infrequent itemsets based on support count.
- Extract association rules from frequent itemsets using confidence and lift.

KEY CONCEPTS OF APRIORI ALGORITHM

- Frequent Itemsets – Groups of items that frequently appear together in transactions.
- Support – Measures how often an itemset appears in the dataset.
- Confidence – Probability of occurrence of item B given item A is purchased.
- Lift – Measures how much more likely item B is bought when item A is present.

APPLICATION OF APRIORI ALGORITHM

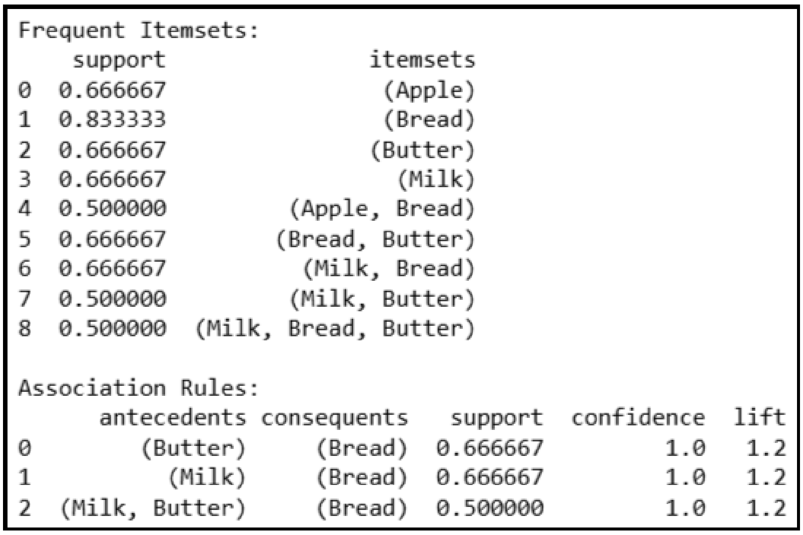
- Market Basket Analysis (Retail & E-commerce)
- Recommendation Systems
- Fraud Detection
- Healthcare (Disease Correlation)

SOURCE CODE ::

```
#import the necessary libraries,
from mlxtend.frequent_patterns import apriori,
association_rules from mlxtend.preprocessing import
TransactionEncoder import pandas as pd


# Sample transaction dataset
transactions = [
    ['Milk', 'Bread', 'Butter'],
    ['Bread', 'Butter', 'Apple'],
    ['Apple', 'Chocolate'],
    ['Milk', 'Bread', 'Apple'],
    ['Milk', 'Bread', 'Butter', 'Chocolate'],
    ['Milk', 'Bread', 'Butter', 'Apple']
]


# Convert transaction data into a format suitable for Apriori
te = TransactionEncoder()
te_array = te.fit(transactions).transform(transactions)
df = pd.DataFrame(te_array, columns=te.columns_)


# Apply Apriori algorithm
min_support = 0.5  # 50% support threshold
frequent_itemsets = apriori(df, min_support=min_support, use_colnames=True)


# Generate association rules
min_confidence = 1  # 100% confidence threshold
rules = association_rules(frequent_itemsets, metric="confidence",
min_threshold=min_confidence)
```

```
# Display results
print("Frequent Itemsets:")
print(frequent_itemsets)

print("\nAssociation Rules:")
print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```

OUTPUT :



```
Frequent Itemsets:
    support              itemsets
0  0.666667               (Apple)
1  0.833333               (Bread)
2  0.666667              (Butter)
3  0.666667                (Milk)
4  0.500000        (Apple, Bread)
5  0.666667       (Bread, Butter)
6  0.666667         (Milk, Bread)
7  0.500000        (Milk, Butter)
8  0.500000  (Milk, Bread, Butter)

Association Rules:
      antecedents consequents   support  confidence  lift
0        (Butter)     (Bread)  0.666667         1.0   1.2
1          (Milk)     (Bread)  0.666667         1.0   1.2
2  (Milk, Butter)     (Bread)  0.500000         1.0   1.2
```

LEARNING OUTCOMES ::

1. Understand association rule mining and its applications.
2. Implement the Apriori algorithm using Python.
3. Analyze frequent itemsets and association rules.
4. Interpret support, confidence, and lift metrics.
5. Apply Apriori for market basket analysis and decision-making.