

**COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD**

Data science

Class Task# 01

***Submitted by:***

Laiba binte tahir FA21-BSE-019

***Submitted to:***

Dr. Ghulam Mujtaba

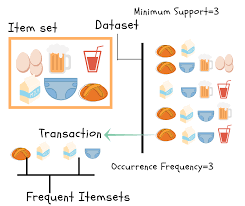
# Association Analysis Summary

Association analysis is a data mining technique used to discover interesting relationships, patterns, or associations among a set of items in transactional or categorical data. It is widely applied in market basket analysis, recommendation systems, and decision-making.

**Key Concepts:**

1. **Frequent Itemsets**: Groups of items that frequently appear together in transactions.
2. **Support**: The proportion of transactions in the dataset that contain a particular itemset. Support(A)=Transactions containing ATotal transactions\text{Support}(A) = \frac{\text{Transactions containing A}}{\text{Total transactions}}Support(A)=Total transactionsTransactions containing A​
3. **Confidence**: Measures the likelihood of item BBB being purchased when AAA is purchased. Confidence(A→B)=Support(A∪B)Support(A)\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}Confidence(A→B)=Support(A)Support(A∪B)​
4. **Lift**: Evaluates the strength of an association, indicating how much more likely two items are to be purchased together compared to if they were independent. Lift(A→B)=Confidence(A→B)Support(B)\text{Lift}(A \rightarrow B) = \frac{\text{Confidence}(A \rightarrow B)}{\text{Support}(B)}Lift(A→B)=Support(B)Confidence(A→B)​

**Algorithms Used:**

* **Apriori Algorithm**: Iteratively finds frequent itemsets by pruning combinations with insufficient support.
* **FP-Growth (Frequent Pattern Growth)**: Uses a compressed data structure called an FP-tree to extract frequent itemsets without candidate generation.
* 

# Priority Algorithm Summary

A priority algorithm is a decision-making approach that selects or prioritizes items based on predefined criteria or a ranking system. It is commonly applied in scheduling, task allocation, search algorithms, and optimization problems.

**Key Components:**

1. **Priority Queue**: A data structure where elements are dequeued in order of their priority (highest or lowest).
2. **Priority Function**: Defines the criteria for ranking elements. For example, it could be based on cost, deadline, or a score.

**Applications:**

* **Dijkstra’s Algorithm**: Uses a priority queue to find the shortest path in a graph.
* **Task Scheduling**: Prioritizes tasks based on deadlines or importance.
* **Search Algorithms**: Prioritizes nodes in A\* or Best-First Search algorithms.

**Steps in a Priority Algorithm:**

1. Define the priority criteria or function.
2. Initialize the priority queue with the elements to be prioritized.
3. Continuously extract the element with the highest (or lowest) priority.
4. Update priorities dynamically if needed (e.g., for changes in costs or scores).
5. Repeat until the goal is achieved or the queue is empty.

