ASSOCIATION RULE LEARNING

In this type of learning, the training data is unlabeled, i.e. the system tries to learn the information without a teacher. ***Association Rule Learning*** is an example of ***Unsupervised Learning***, it is used in Market Basket Analysis, Intrusion Detection, Web Usage Mining etc. This algorithm's goal is to dig into large amounts of data and discover interesting relations between attributes.

For example, you find out that people who purchase milk and bread, also tend to purchase butter.

This algorithm counts the frequency of complimentary occurrences, or associations, across a very large dataset with over thousands of attributes.

to measure the associations between thousands of data items, there are several metrics.

- **Support** — This says how popular an itemset is, i.e. it is used to find the frequency of a certain itemset appearing in the dataset.

$$Support(A) = Frequency(A)$$

- **Confidence** — This says how likely item B is purchased when item A is purchased, expressed as (A -> B).

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

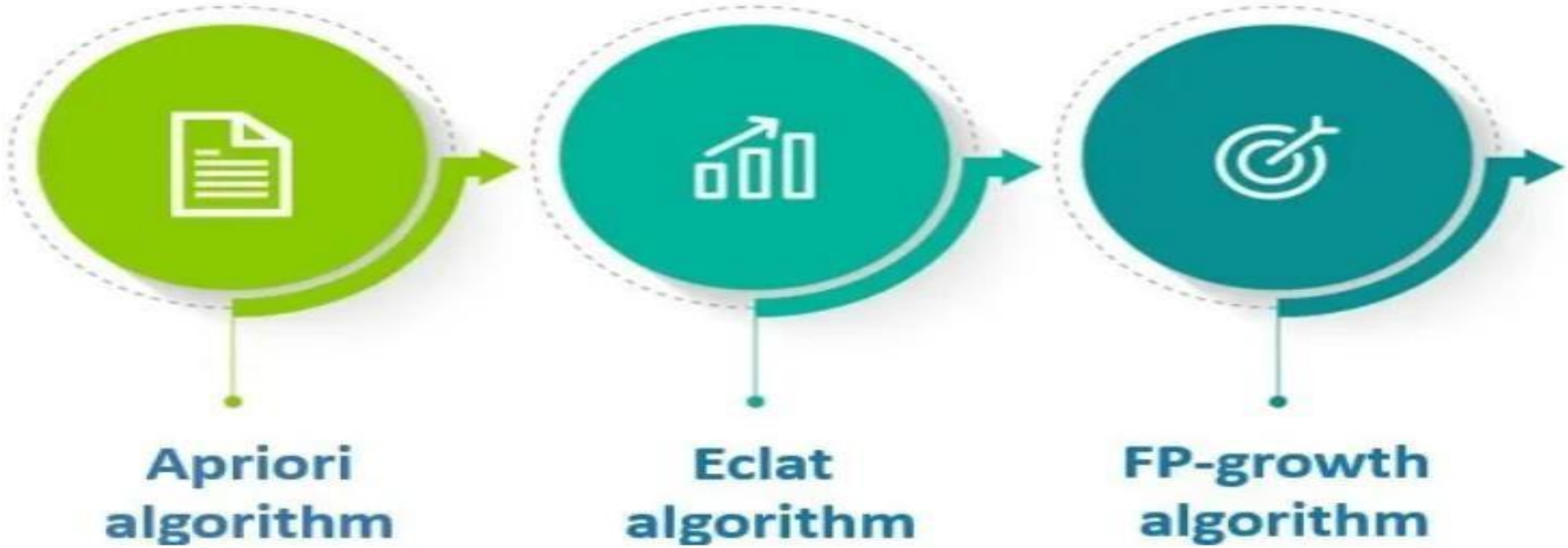
- **Lift** — This says how likely an item A is purchased while controlling how popular item B is.

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

Lift has three possible values —

- **Lift = 1** — The probability of occurrence of A and B is independent of each other.
- **Lift > 1** — It determines the degree to which A and B are dependent on each other.
- **Lift < 1** — It tells us that A is a substitute for B, which means A has a negative effect on item B.

Different types of Association Rule Learning



Association Rule Learning can be divided into three algorithms —

- **Apriori**— This algorithm uses frequent datasets to generate association rules. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets. This algorithm uses a *Breadth-First Search* algorithm and *Hash-Tree* to calculate the itemset efficiently.

- **Eclat** — Eclat algorithm stands for *Equivalence Class Transformation*. While the Apriori algorithm works in a horizontal sense imitating the *Breadth-First Search* of a graph, the ECLAT algorithm works in a vertical manner just like the *Depth-First Search* of a graph.. It performs faster execution than Apriori Algorithm.

- **F-P Growth**— The F-P Growth algorithm stands for ***Frequent Pattern***, and it is the improved version of the Apriori Algorithm. The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance. It uses a ***Divide-and-Conquer*** strategy and the core of this method is the usage of a special data structure named ***Frequent-Pattern Tree (FP-tree)***, which retains the item set association information. The purpose of this frequent tree is to extract the most frequent patterns.

Association Rules Exercise

- Here are a dozen sales transactions.
- The objective is to use this transaction data to find affinities between products, that is, which products sell together often.
- The support level will be set at 33 percent; the confidence level will be set at 50 percent.

$$Rule: \quad X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

1-item Sets	Frequency	2-item Sets	Frequency
Milk	9	Milk, Bread	7
Bread	10	Milk, Butter	7
Butter	10	Milk, Cookies	3
Egg	3	Bread, Butter	9
Ketchup	3	Butter, Cookies	3
Cookies	5	Bread, Cookies	4

Frequent 1-item Sets	Frequency
Milk	9
Bread	10
Butter	10
Cookies	5

Frequent 2-item Sets	Frequency
Milk, Bread	7
Milk, Butter	7
Bread, Butter	9
Bread, Cookies	4

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

Milk, Bread, Butter, Cookies

3-item Sets	Frequency
Milk, Bread, Butter	6
Milk, Bread, Cookies	1
Bread, Butter, Cookies	3
Milk, Butter, Cookies	2

Frequent 3-item Sets	Frequency
Milk, Bread, Butter	6

Association Rule Mining - Subset Creation

- Frequent 3-Item Set = I => {Milk, Bread, Butter}
- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
- How to form Association Rule…?
  - For every non-empty subset S of I, the association rule is,
    - **S →(I-S)**
    - **If support(I) / support(S) >= min_confidence**

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 1: {Milk} → {Bread, Butter} {S=50%, C=66.67%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk) = $\frac{6/12}{9/12}$ = 6/9 = 66.67% > 60%
  - Valid

- Rule 2: {Bread} → {Milk, Butter} {S=50%, C=60%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Bread) = 6/10 = 60% >= 60%
  - Valid

- Rule 3: {Butter} → {Milk, Bread} {S=50%, C=60%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Butter) = 6/10 = 60%>=60
  - Valid

- Rule 4: {Milk, Bread} → {Butter} {S=50%, C=85.7%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk, Bread) = 6/7 = 85.7% > 60%
  - Valid

Association Rule Mining - Subset Creation

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 5: {Milk, Butter} → {Bread} {S=50%, C=85.7%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk, Butter) = 6/7 = 85.7% >= 60%
  - Valid

- Rule 6: {Bread, Butter} → {Milk} {S=50%, C=66.67%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Bread, Butter) = 6/9 = 66.67%>=60
  - Valid