Association Rule Mining

Association rule mining is a technique used to identify patterns in large data sets. It involves finding relationships between variables in the data and using those relationships to make predictions or decisions. The goal of association rule mining is to uncover rules that describe the relationships between different items in the data set.

For example, consider a dataset of transactions at a grocery store. Association rule mining could be used to identify relationships between items that are frequently purchased together. For example, the rule "If a customer buys bread, they are also likely to buy milk" is an association rule that could be mined from this data set. We can use such rules to inform decisions about store layout, product placement, and marketing efforts.

Association rule mining typically involves using algorithms to analyze the data and identify the relationships. These algorithms can be based on statistical methods or machine learning techniques. The resulting rules are often expressed in the form of "if-then" statements, where the "if" part represents the antecedent (the condition being tested) and the "then" part represents the consequent (the outcome that occurs if the condition is met).

Association rule mining is an important technique in data analysis because it allows users to discover patterns or relationships within data that may not be immediately apparent. By identifying associations between variables, association rule mining can help users understand the relationships between different variables and how those variables may be related to one another.

This can be useful for various purposes, such as identifying market trends, detecting fraudulent activity, or understanding customer behavior. Association rule

mining can also be used as a stepping stone for other types of data analysis, such as predicting outcomes or identifying key drivers of certain phenomena. Overall, association rule mining is a valuable tool for extracting insights and understanding the underlying structure of data.

Association Rule Mining is a powerful technique used to uncover meaningful relationships between variables within large datasets. They are designed to discover **"if-then" patterns**, providing insights into how data items are related and frequently occur together. These rules are particularly useful in identifying correlations and dependencies, enabling data-driven decision-making.

For instance, in a retail dataset, an association rule might identify that **"if a customer buys bread, they are likely to buy butter"**. Such insights help businesses improve cross-selling strategies, inventory management, and customer satisfaction.

Key Components of Association Rules

1. **Antecedent:** The **"if"** part of the rule, representing the condition.
   - Example: A customer buys bread.
2. **Consequent:** The **"then"** part of the rule, representing the outcome.
   - Example: The customer also buys butter.

Association rules are derived through algorithms that evaluate the frequency and strength of these relationships. They use metrics like **support, confidence,** and **lift** to measure the relevance and reliability of discovered patterns. These rules have applications in various fields, such as retail, healthcare, and marketing, where analyzing customer behavior or trends is critical for success.

Rule Evaluation Metrics

Association rules are evaluated using key metrics that determine their relevance, strength, and reliability. These metrics include **support**, **confidence**, and **lift**, which quantify the frequency and strength of relationships between data items.

1. Support:

Support measures how frequently an itemset (both antecedent and consequent) appears in the dataset. It provides an indication of how common a particular association is.

For individual transactions

Support (X) = (Number of transactions containing X) / (Total transactions)

For two or more transactions

Support (XUY) = (Number of transactions containing X and Y) / (Total transactions)

2. Confidence:

Confidence measures the likelihood of the consequent occurring given that the antecedent has already occurred. It evaluates the reliability of the rule.

Confidence($X \rightarrow Y$) = Support($X \cup Y$) / Support(X)

3. Lift:

Lift measures the strength of an association compared to its random occurrence in the dataset. It identifies how much more likely the antecedent and consequent are to appear together than independently.

Lift($X \rightarrow Y$) = Confidence($X \rightarrow Y$) / Support(Y)

- Lift > 1 → Positive association
- Lift = 1 → No association
- Lift < 1 → Negative association

Example: Transaction Dataset

Transaction	Items Purchased
T1	Bread, Milk
T2	Bread, Diaper, Beer, Eggs
T3	Milk, Diaper, Beer, Coke
T4	Bread, Milk, Diaper, Beer
T5	Bread, Milk, Diaper, Coke

Total transactions = **5**

Example-1 (Bread → Diaper)

Step 1: Calculate Support

Support of Bread

Appears in T1, T2, T4, T5 → **4 times**

Support (Bread) = 4/5=0.8

Support of Diaper

Appears in T2, T3, T4, T5 → **4 times**

Support (Diaper) = 4/5=0.8

Support of (Bread ∪ Diaper)

Appears in T2, T4, T5 → **3 times**

Support (Bread, Diaper) = 3/5=0.6

Step 2: Association Rule

Bread → Diaper

Confidence

Confidence = 0.6/0.8 = 0.75 (75%)

Lift

Lift=0.75/0.8=0.9375

Result:

Customers who buy bread buy diapers **75% of the time**, but lift < 1 shows **weak association or no association**.

Example-2 (Milk → Bread)

Support (Milk)

Appears in T1, T3, T4, T5 → **4/5 = 0.8**

Support (Milk ∪ Bread)

Appears in T1, T4, T5 → **3/5 = 0.6**

Rule: Milk → Bread

Confidence = 0.6/0.8 = 0.75

Lift = 0.75/0.8 = 0.9375

Result:

75% of customers who buy milk also buy bread.

Example-3 (Diaper → Beer)

Support (Diaper ∪ Beer)

Appears in T2, T3, T4 → **3/5 = 0.6**

Support (Diaper)

= 0.8

Confidence

0.6/0.8=0.75

Support (Beer)

Appears in T2, T3, T4 → **3/5 = 0.6**

Lift

0.75/0.6 =1.25

Result :

Strong positive association ✔

Customers buying diapers are **25% more likely** to buy beer.

Other example

If:

- Support(A) = 40%
- Support(B) = 50%
- Support(A ∪ B) = 30%

Rule: A → B

Confidence

30/40=75%

Lift

0.75/0.5=1.5

Strong positive association ✔

How Does Association Rule Learning Work?

Association rule learning is a multi-step process designed to identify meaningful patterns and relationships in large datasets. It involves two main stages:

1. **Identifying Frequent Itemsets:** The process begins by identifying frequent itemsets—combinations of items that appear together in transactions with a frequency above a predefined threshold. Metrics like support are used to measure how often these itemsets occur in the dataset. For example, a frequent itemset might reveal that bread and butter are purchased together in 10% of transactions.
2. **Generating Association Rules:** Once frequent itemsets are identified, association rules are generated. These rules take the form of if-then statements that describe relationships between items (e.g., "If a customer

buys bread, they are likely to buy butter"). Metrics such as confidence and lift are applied to evaluate the strength and reliability of these rules.

Market Basket Analysis

One of the most well-known applications of association rule mining is in market basket analysis. This involves analyzing the items customers purchase together to understand their purchasing habits and preferences.

For example, a retailer might use association rule mining to discover that customers who purchase diapers are also likely to purchase baby formula. We can use this information to optimize product placements and promotions to increase sales.

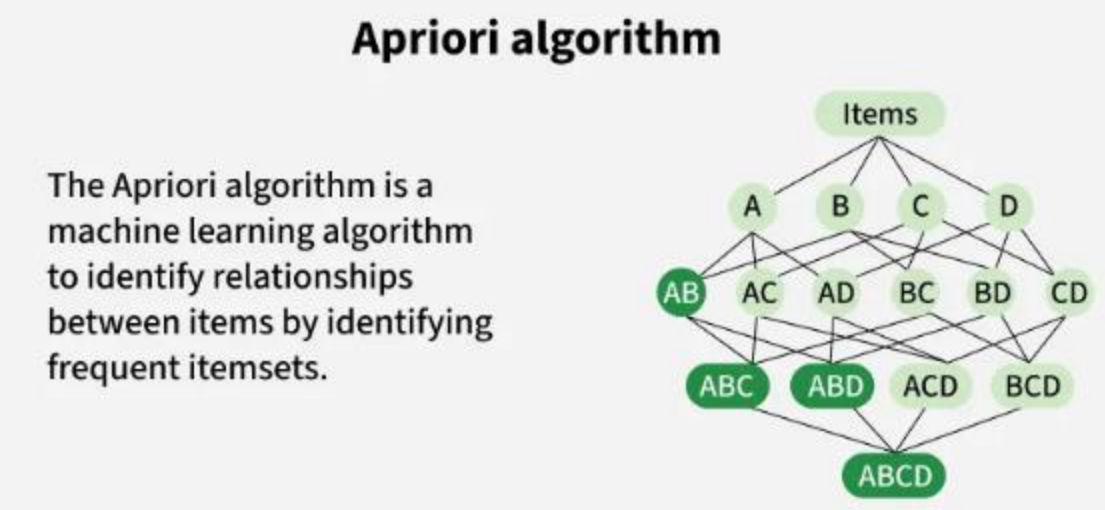
Types of Association Rule Learning Algorithms

Several algorithms are used for association rule learning, each with unique strengths and applications. The three most commonly used algorithms are:

- **Apriori Algorithm**
- **Eclat Algorithm**
- **FP-Growth Algorithm**

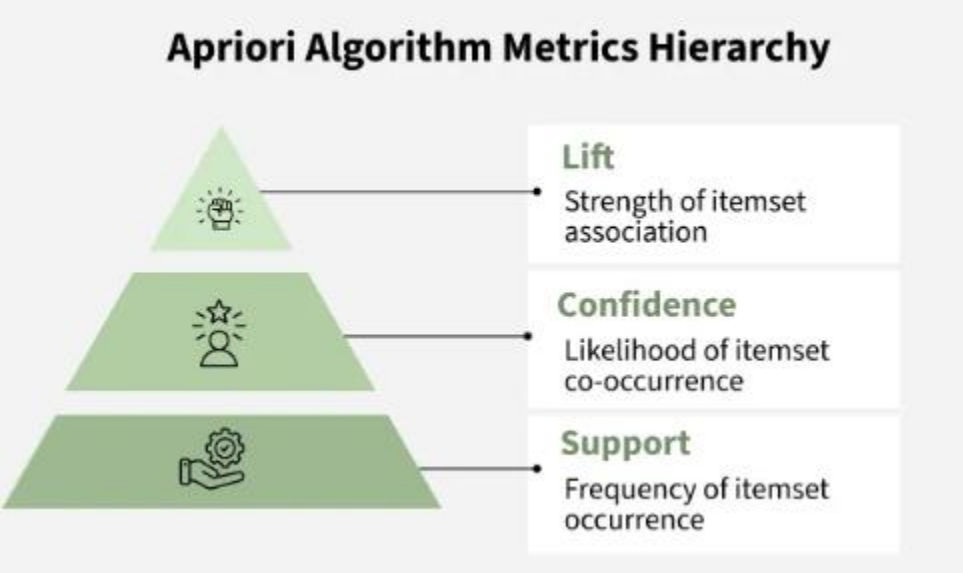
1. Apriori Algorithm

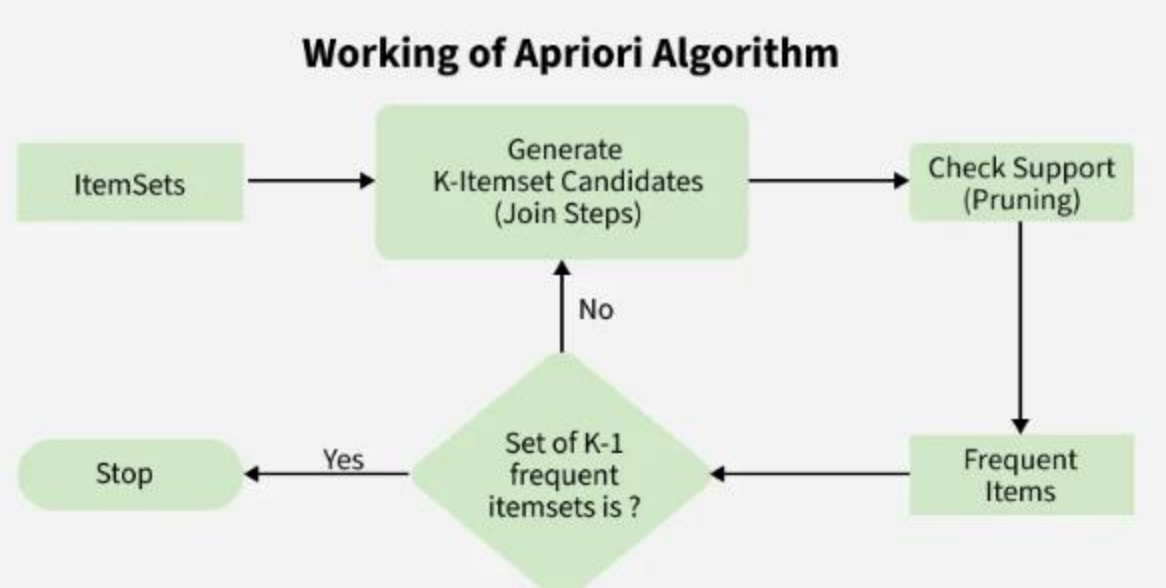
The **Apriori algorithm** employs a **breadth-first search** approach to identify frequent itemsets. It relies on the principle that all subsets of a frequent itemset must also be frequent, reducing the search space.



Principle (Apriori Property):

The Apriori algorithm is based on the principle that if an itemset is frequent, then all of its subsets must also be frequent. Conversely, if an itemset is infrequent, its supersets will also be infrequent.





Steps of the Apriori Algorithm

1. **Set Minimum Thresholds**:

   The user defines minimum support and confidence thresholds. Support is the minimum frequency an itemset must appear in the data to be considered significant, and confidence measures the strength of the association rule.

2. **Identify Frequent 1-Itemsets (L1)**:

   The algorithm scans the entire dataset to count the occurrences of each individual item (1-itemset). Items that meet or exceed the minimum support threshold are kept as "frequent 1-itemsets" (L1); others are pruned.

3. **Generate Candidate Itemsets (Ck)**: Using the frequent itemsets from the previous step, the algorithm generates candidate itemsets of the next size, which involves a **join step**. For example, frequent 2-itemsets are generated by combining frequent 1-itemsets.

4. **Prune Infrequent Candidate Itemsets**: The algorithm checks the support of all subsets of the new candidate itemsets. If any subset does not meet the minimum

support threshold (based on the Apriori property), the entire candidate itemset is removed in a **prune step**.

5. **Calculate Support and Find Frequent Itemsets (Lk)**: The database is scanned again to count the actual support for the remaining candidate itemsets. Those that meet or exceed the minimum support form the "frequent k-itemsets".
6. **Iterate**: Steps 3-5 are repeated, incrementing k (k=3, 4, and so on) until no more frequent itemsets can be generated.
7. **Generate Association Rules**: Once all frequent itemsets are found, strong association rules (e.g., "If {bread} is bought, then {butter} is also bought") are generated from them. The strength of these rules is typically evaluated using the confidence and lift metrics, and rules that do not meet the minimum confidence threshold are discarded.

Example (Short):

Transactions:

T1: {Milk, Bread}

T2: {Milk, Diaper, Beer}

T3: {Milk, Bread, Diaper}

- **Advantage:** Simple to implement and effective for small datasets with low dimensionality.
- **Limitation:** Performance degrades significantly with large or dense datasets due to repeated scanning of the database.

Example

Consider the below transaction in which B = Bread, J = Jelly, P = Peanut Butter, M = Milk and E = Eggs. Given that minimum threshold support = 40% and minimum threshold confidence = 80%.

TID	Items Bought
1	{B, J , P}
2	{B, P}
3	{B, M, P}
4	{E, B}
5	{E, M}

Step-1: Count the number of transactions in which each item occurs (Bread B occurs in 4 transactions and so on).

Items	Support	Support (in percentage) = (Support * 100) / No. of trans
B	4	(4 * 100) / 5 = 80%
J	1	(1 * 100) / 5 = 20%
P	3	(3 * 100) / 5 = 60%
M	2	(2 * 100) / 5 = 40%
E	2	(2 * 100) / 5 = 40%

Step-2: As minimum threshold support = 40%, So in this step we will remove all the items that are bought less than 40% of support or support less than 2.

Items	Support
B	4 (80%)
P	3 (60%)
M	2 (40%)
E	2 (40%)

The above table has single items that are bought frequently. Now let's find a pair of items that are bought frequently. We continue from the above table (Table in step 2)

Step-3: We start making pairs from the first item and below items like {B,P} ,{B,M} ,{B,E} and then we start with the second item and below items like {P,M} ,{P,E}. We do not make pair {P,B} because we already made {P,B} pair when we were making pairs of B. As buying a bread and Peanut Butter together is same as buying Peanut Butter and bread together. After making all the pairs we get,

Items	Support	Support (in percentage) = (Support * 100) / No. of trans
{B, P}	3	(3 * 100) / 5 = 60%
{B, M}	1	(1 * 100) / 5 = 20%
{B , E}	1	(1 * 100) / 5 = 20%
{P , M}	1	(1 * 100) / 5 = 20%
{P , E}	0	(0 * 100) / 5 = 0%
{M , E}	1	(1 * 100) / 5 = 20%

Step-4: As minimum threshold support = 40%, So in this step we will remove all the items that are bought less than 40% of support and we are left with

Items	Support
{B, P}	3

The above table has two items {B , P } that are bought together frequently.

Association Rule Generation :

Step-5: As we cannot generate large frequent item (itemset of 3) further because we are left with 1 frequent item set. We will start generating association rules from the frequent item set. As we have frequent item set of two, only two association rules will be generated which is shown below :

Items	Association Rule	Confidence (X => Y) = Occurrence (X U Y)/ Occurrence (X)
{B, P} (60%)	B => P	(3/4) * 100 = 75 %
	P => B	(3/3) * 100 = 100%

As P -> B has confidence 100% which is greater than minimum confidence threshold 80%, thus P -> B is a **Strong Association Rule.**

2. Eclat Algorithm

The **Eclat algorithm** uses a **depth-first search** strategy to discover frequent itemsets. Instead of scanning the database multiple times, it represents transactions as vertical itemsets and directly computes intersections.

- **Advantage:** Efficient for datasets with **sparse data** or where there are fewer frequent itemsets.

3. FP-Growth Algorithm

The **FP-Growth (Frequent Pattern Growth)** algorithm leverages a **prefix-tree structure** called the **FP-tree** to represent transactional data compactly. The **FP-**

Growth algorithm is a popular method in **Data Mining** used to find **frequent itemsets** without generating candidate sets like Apriori. It is **faster and more efficient** than Apriori because it uses a compact structure called an **FP-Tree (Frequent Pattern Tree)**.

Steps of FP-Growth Algorithm

Step 1: Scan Database & Find Frequent Items

- Count frequency of each item.
- Remove items below minimum support.
- Sort remaining items in descending order of frequency.

Step 2: Construct FP-Tree

- Create a root node labeled **NULL**.
- For each transaction:
    - Keep only frequent items.
    - Sort them by frequency order.
    - Insert into FP-Tree.

Step 3: Mine Frequent Patterns

- Start from least frequent item.
- Build **Conditional Pattern Base**.
- Construct **Conditional FP-Tree**.
- Extract frequent itemsets recursively.

Example of FP-Growth

Transaction Database

TID	Items
T1	A, B, D, E
T2	B, C, E
T3	A, B, C, E
T4	B, E
T5	A, B, C, E

Assume **Minimum Support = 30%**

Step 1: Count Item Frequencies

Item	Support
B	5
E	5
A	3
C	3
D	1 (remove)

Sorted order → **B > E > A > C**

Step 2: Build FP-Tree

Convert transactions (remove D & sort):

T1 → B E A

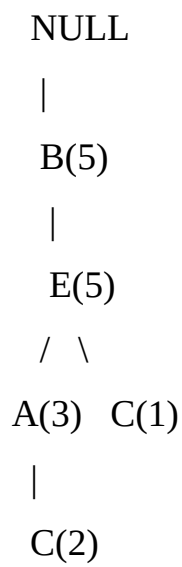
T2 → B E C

T3 → B E A C

T4 → B E

T5 → B E A C

FP-Tree Structure



Step 3: Mine Frequent Patterns

Start from bottom item.

For Item C

Conditional Pattern Base:

- B E C (1)
- B E A C (2)

Frequent patterns:

- C
- EC
- BC
- BEC
- AC
- EAC

For Item A

Conditional Pattern Base:

- B E A (3)

Frequent patterns:

- A
- EA
- BA
- BEA

For Item E

Conditional Pattern Base:

- B E (5)

Frequent patterns:

- E
- BE

For Item B

Frequent pattern:

- B

Final Frequent Itemsets

{B}, {E}, {A}, {C},

{BE}, {BA}, {EA}, {EC},

{BEA}, {BEC}, {EAC}, etc.

Advantages of FP-Growth

✔ No candidate generation

✔ Requires only two database scans

✔Significantly faster and more memory-efficient than Apriori, especially for large datasets.

✔ Works well for large datasets

FP-Growth vs. Apriori Algorithm

Feature	FP-Growth	Apriori
Candidate generation	Not required	Needed
Data scans	2	Several
Speed	Efficient for large datasets	Slower because of repeated scans
Memory use	Higher (requires a tree structure)	Lower
Complexity	More difficult to implement	Simpler to grasp and build

Applications of Association Rules

Association rules are widely applied across various industries to uncover patterns and relationships in data, enabling better decision-making and operational efficiency.

1. **Retail and Market Basket Analysis:** Retailers use association rules to identify frequently purchased product combinations, helping them optimize store layouts or create product bundles to increase sales.
    - **Example:** A supermarket discovers that customers who buy bread often purchase butter and jam, leading to strategic placement of these items together.
2. **Healthcare:** In healthcare, association rules help discover co-occurrence patterns in symptoms, aiding in diagnostic processes and treatment plans.
    - **Example:** Identifying that patients with high blood pressure often have a higher risk of developing diabetes can guide preventative care strategies.
3. **E-Commerce and Recommendation Systems:** E-commerce platforms leverage association rules to build recommendation systems that enhance user experiences and drive sales.
    - **Example:** Amazon's "Customers who bought this also bought" feature suggests complementary products, boosting cross-selling opportunities.
4. **Fraud Detection:** Association rules are used in financial services to identify unusual patterns in transaction data, which can help detect fraudulent activities.
    - **Example:** Flagging transactions that deviate significantly from established spending patterns for further investigation.