🧭 Unit 3: Mining Frequent Patterns, Associations, and Correlations

Subject: Data Mining (2101CS521)
In Easy English – For Theory Exam

🟩 1. What Kinds of Patterns Can Be Mined

Data mining tasks are divided into two types:

🔹 1. Descriptive Tasks

- Describe the **general properties** or patterns of data.
- Used to find patterns like **frequent items**, **associations**, or **correlations**.
- Example: Finding that people often buy “milk and bread” together.

🔹 2. Predictive Tasks

- Used to **predict future data or behavior**.
- Example: Predicting product sales during a festival.

🧩 Frequent Patterns

Frequent patterns are the patterns that **appear again and again** in data.

Types of frequent patterns:

Type	Meaning	Example
Frequent Itemset	Items appearing together frequently	{Milk, Bread}
Frequent Subsequence	Order of events that occur often	Buy Laptop → Camera → Memory Card
Frequent Substructure	Common structure like graphs or trees	Common social network connections

---

🟩 2. Market Basket Analysis

Market Basket Analysis is used to find **relationships among items** that are bought together.

👉 It helps to answer:

“If a customer buys certain items, what else are they likely to buy?”

🛒 Example:

If someone buys a **car**, they are likely to also buy **insurance**.
So, rule: **{Car, Accessories} → {Insurance}**

📊 Purpose:

- Understand customer buying behavior
- Decide product placement in stores
- Create combo offers or discounts

🟩 3. Association Rule Mining

Association rule mining finds **relationships between items** in transactions.

It uses rules of the form **X → Y**,
meaning: If X is bought, Y is also likely to be bought.

🧩 Important Terms

Term	Meaning	Example
Itemset	Group of items	{Milk, Bread}
Support	Fraction of transactions containing X and Y	3 out of 10 → 30%
Confidence	How often Y appears when X appears	3 out of 5 → 60%

⚙️ Steps:

1. Find all **frequent itemsets** (meet minimum support).
2. Generate **rules** from these frequent itemsets.
3. Evaluate rules using **support** and **confidence**.

🟩 4. Maximal and Closed Frequent Itemsets

🔹 Closed Frequent Itemset:

A frequent itemset is **closed** if no larger itemset has **the same support**.
👉 Keeps detailed frequency information.

Example:
{A} = 4, {A, E} = 4 → {A} is *not closed*, {A, E} is *closed*.

🔹 Maximal Frequent Itemset:

A frequent itemset is **maximal** if **none of its supersets are frequent**.
👉 Only shows the largest frequent sets.

Example:
{A, D, E} is maximal if adding any item makes it non-frequent.

🧠 Difference:

Closed Itemset	Maximal Itemset
No superset with same support	No superset is frequent
More detailed info	Only largest sets
Used for accurate data	Used for faster mining

🟩 5. Apriori Algorithm

Apriori is a **basic algorithm** to find frequent itemsets.

⚙️ Main Idea:

If an itemset is frequent, then **all its subsets must also be frequent.**

🔹 Steps:

1. Find **frequent 1-itemsets (L1)**.
2. Generate **candidate k-itemsets (Ck)** from **L(k–1)**.
3. Remove candidates whose subsets are not frequent (**pruning**).
4. Count supports and keep itemsets ≥ min_support (**frequent itemsets**).
5. Repeat until no more frequent sets are found.

🧠 Apriori Property:

All non-empty subsets of a frequent itemset must be frequent.

🔹 Advantages:

- Simple and easy to implement.

❌ Disadvantages:

- Generates many candidates.
- Requires multiple database scans (slow for large data).

🟩 6. Methods to Improve Apriori Efficiency

To make Apriori faster, several techniques are used:

Method	Idea	Benefit
Hash-Based Technique	Use a hash table to reduce candidates	Fewer candidate itemsets

Method	Idea	Benefit
Transaction Reduction	Remove transactions that don’t contain frequent itemsets	Less scanning
Partitioning	Divide data into parts and find local frequent sets	Only two database scans
Sampling	Use small random sample of database	Saves time and memory
Dynamic Itemset Counting	Update counts as new data arrives	Good for real-time data

---

🟩 7. FP-Growth Algorithm (Frequent Pattern Growth)

FP-Growth is a **fast algorithm** that finds frequent patterns **without generating candidate sets**.

⚙️ Steps:

1. **Scan database once** → find frequent items.
2. **Sort frequent items** in decreasing order.
3. **Build FP-Tree** (Frequent Pattern Tree).
4. For each item, find its **Conditional Pattern Base** (paths ending with that item).
5. Create **Conditional FP-Trees** and mine them recursively.

🧠 Advantages:

- No candidate generation.
- Requires only 2 database scans.
- Much faster than Apriori.

❌ Disadvantages:

- FP-Tree may not fit in memory for very large data.

🟩 8. Pattern Evaluation Methods

After mining, we must evaluate which rules are **useful and interesting**.

🧩 Two Main Measures:

1. **Support:** How frequently the rule occurs.

$$Support = \frac{Transactions(X \cup Y)}{TotalTransactions}$$

2. **Confidence:** How often Y appears when X appears.

$$Confidence = \frac{Transactions(X \cup Y)}{Transactions(X)}$$

⚙️ Two Types of Evaluation:

Type	Description
Subjective	Based on user’s interest or experience
Objective	Based on numerical measures (support, confidence, lift)

❌ Limitation:

High confidence may not always mean a strong relation —
example: {Brush} → {Toothpaste} (toothpaste is common anyway).

🟩 9. Correlation Analysis

Correlation checks whether two items are **actually related** or **just occur together by chance**.

⚙️ Measure Used: Lift

$$Lift(A \rightarrow B) = \frac{P(A \cup B)}{P(A) \times P(B)}$$

🧮 Interpretation of Lift:

Lift Value	Meaning	Type
> 1	A and B are **positively correlated**	Go together
= 1	A and B are **independent**	No relation
< 1	A and B are **negatively correlated**	Rarely together

🧠 Example (from PPT):

- P(A) = 0.6, P(B) = 0.75, P(A ∪ B) = 0.4

$$Lift = \frac{0.4}{0.6 \times 0.75} = 0.88$$

Lift < 1 → A and B are **negatively correlated**.

🧾 Overall Unit Summary

Topic	Main Idea
1. Patterns	Descriptive and Predictive types
2. Market Basket Analysis	Finds relations between items bought together
3. Association Rules	Rules like X → Y (Support, Confidence)
4. Maximal & Closed Sets	Special frequent itemsets
5. Apriori Algorithm	Finds frequent patterns using candidate generation
6. Efficiency Methods	Techniques to make Apriori faster
7. FP-Growth	Fast algorithm without candidates
8. Pattern Evaluation	Uses support and confidence to check interesting rules
9. Correlation Analysis	Uses lift to find true relationships

✅ **In Short (For 2–3 Marks Questions):**

- **Support** → How frequent the rule is.
- **Confidence** → How strong the rule is.
- **Lift > 1** → Positive relation.
- **Apriori Property** → All subsets of frequent itemsets must also be frequent.
- **FP-Growth** → Finds frequent patterns without generating candidates.
- **Closed Itemset** → No superset has same support.
- **Maximal Itemset** → No superset is frequent.