Times asked: 7 times

6 times

5 times

4 times

3 times

2 times

1 time

Data Warehousing & Mining Question bank

 # Included 1 example of numerical from previous paper

1. Data warehousing fundamentals

1. **Design Star and Snowflake schema for given records: #**

   A social media platform wants to analyse user engagement data to improve content recommendations and user experience. The INTERACTIONS fact table contains information about user interactions, including interaction details, user information, content details, and time periods.

   The dimension tables provide additional context about users, content, categories, and time periods.
2. **Differentiate between Star schema and Snowflake schema.**
3. **Compare OLTP and OLAP.**
4. **Perform OLAP operations: #**

   The college wants to record the marks for the courses completed by students using the dimensions:

   I) Course, II) Student, III) Time & a measure Aggregate marks.

   Create a cube and describe following OLAP operations

   I) Slice, II) Dice, III) Roll up, IV) Drill down, V) Pivot
5. **Differentiate between ER Modelling vs Dimensional modelling.**
6. **Illustrate major steps in ETL process.**
7. **Write a short note on: Techniques of data loading.**

2. Introduction to Data mining, exploration and pre-processing

1. **Describe any 5 issues in data mining.**
2. **Explain KDD process with neat diagram**
3. **Explain different data visualization techniques.**
4. **Explain data preprocessing. Explain different steps involved in data preprocessing.**
5. **State any five applications of data mining.**
6. **Describe various methods for handling the problem of missing values in attributes.**
7. **Explain in brief what is data discretization and concept hierarchy generation.**

3. Classification

1. Describe in detail about how to evaluate accuracy of the classifier.

2. Apply Naïve Bayes classification: #

A data sample is given below. Find whether Patient X has flu or not using Naïve Bayes classifier.

If X= (chills=Y, runny nose=N, headache=Mild, fever=Y, flu=?)

chills	Runny nose	headache	fever	Flu
Y	N	Mild	Y	N
Y	Y	No	N	Y
Y	N	Strong	Y	Y
N	Y	Mild	Y	Y
N	N	No	N	N
N	Y	Strong	Y	Y
N	Y	Strong	N	N
Y	Y	Mild	Y	Y

3. Build decision tree for given problem: #

A company wants to predict whether a customer will subscribe to a premium membership based on their demographic and browsing behaviour data. The dataset contains information about customers, including age, gender, income, browsing time, and subscription status.

Age	Gender	Income	Browsing Time	Subscription
20-30	Male	High	10am-12pm	Yes
20-30	Female	Medium	2pm-4pm	Yes
30-40	Male	Low	8am-10am	No
30-40	Female	High	4pm-6pm	Yes
>40	Male	Medium	6pm-8pm	Yes
>40	Female	Medium	8am-10am	No
>40	Male	High	12pm-2pm	Yes
20-30	Female	Low	10am-12pm	No
20-30	Male	Medium	2pm-4pm	Yes
30-40	Female	High	8am-10am	Yes

Use ID3 to build the decision tree and predict the following example:

4. Explain Decision tree-based classification approach with example.

5. Explain how Naive Bayes classification makes predictions and discuss the "naive" assumption in Naive Bayes. Provide an example to illustrate the application of Naive Bayes in a real-world scenario.

4. Clustering

1. **Cluster given data using k-means algorithm: #**

   Suppose the data for clustering is {6,14,18,22,1,40,50,11,25} consider k=2, cluster the given data using k means algorithm.

2. **Explain K-means clustering algorithm and draw flowchart. Discuss its advantages and limitations.**
3. **Differentiate between Agglomerative and Divisive clustering method.**
4. **Describe K-medoids algorithm.**
5. **Apply single linkage clustering and construct dendrogram: #**

   Find the clusters for the following dataset using a single link technique. Use Euclidean distance and draw the dendrogram.

Sample No	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

6. **Apply complete linkage clustering and construct dendrogram: #**

   Consider the data given below. Create adjacency matrix. Apply complete link algorithm to cluster the given data set and draw the dendrogram.

	A	B	C	D	E
A	0	2	6	10	9
B	2	0	3	9	8
C	6	3	0	7	5
D	10	9	7	0	4
E	9	8	5	4	0

5. Mining frequent patterns and associations

1. **Explain Multilevel association rules mining and Multidimensional association rules mining with examples.**
2. **Write a short note on: FP tree.**
3. **Explain market basket analysis with an example.**

4. Apply Apriori algorithm to find frequent pattern set and strong association rules: #

For the table given perform Apriori algorithm and show frequent item set and strong association rules. Assume minimum support of 30% and maximum confidence of 70%.

TID	Items
1	1,4,6,8
2	2,5,3
3	7,1,3,8
4	9,10
5	1,5

5. Create FP Tree to find frequent pattern sets: #

A database has five transactions:

T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, K, I, E}

Let minimum support = 3, Find all frequent item sets using FP-growth algorithm.

6. Web mining

1. **Explain page rank algorithm with example**
2. **Explain web structure mining? List the approaches used to structure the web pages to improve on the effectiveness of search engines and crawlers.**
3. **Write a note on web usage mining. Also state any two of its applications.**
4. **Write a short note on: Web content mining.**
5. **Explain CLARANS extension in web mining.**

	1	2	3	4	5	6
2024 May	25	25	25	15	25	15
2023 Dec	30	30	15	15	20	20
2023 May	30	30	10	25	20	15
2022 Dec	25	15	15	25	25	25
Last 4 Avg	25	25	15	20	25	20
*2022 May	15	15	20	15	10	15
Total	125	115	85	95	100	90

Data Warehousing & Mining Answer bank

1. Data warehousing fundamentals

1. Design Star and Snowflake schema for given records:

Consider a data warehouse for hotel occupancy, where there are four dimensions namely

(a) Hotel

(b) Room

(c) Time

(d) Customer

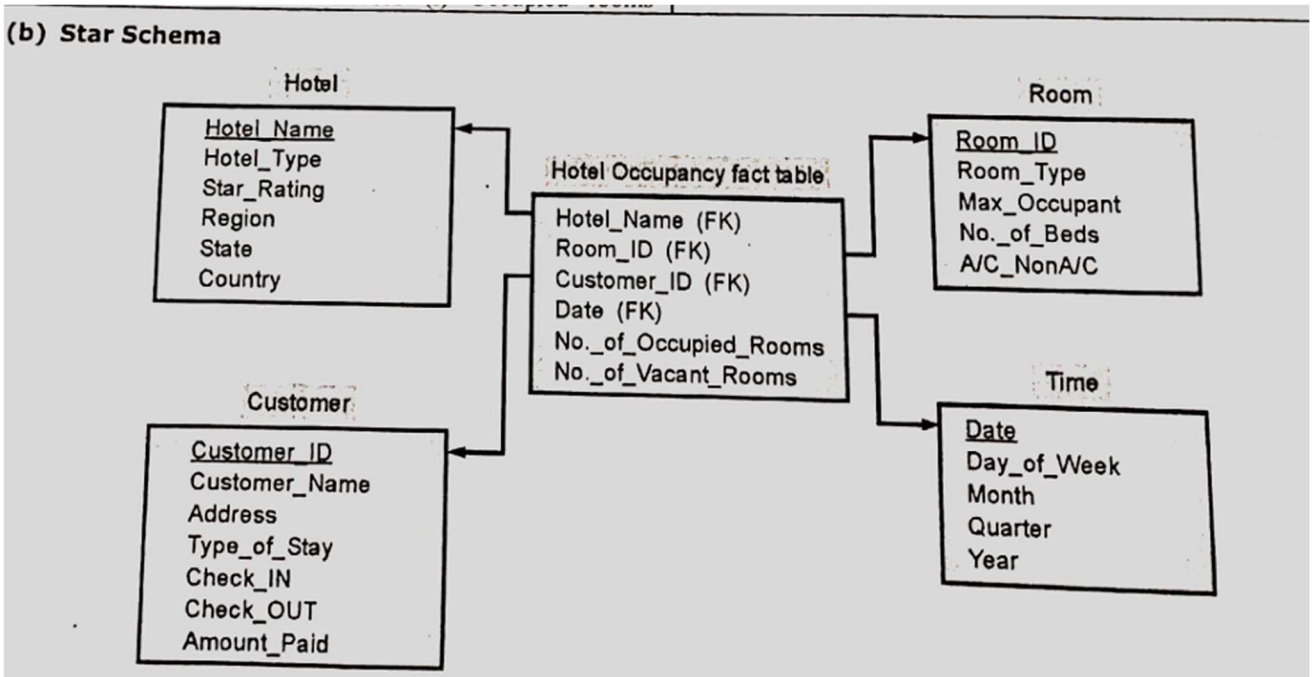
and two measures

(i) Occupied rooms

(ii) Vacant rooms.

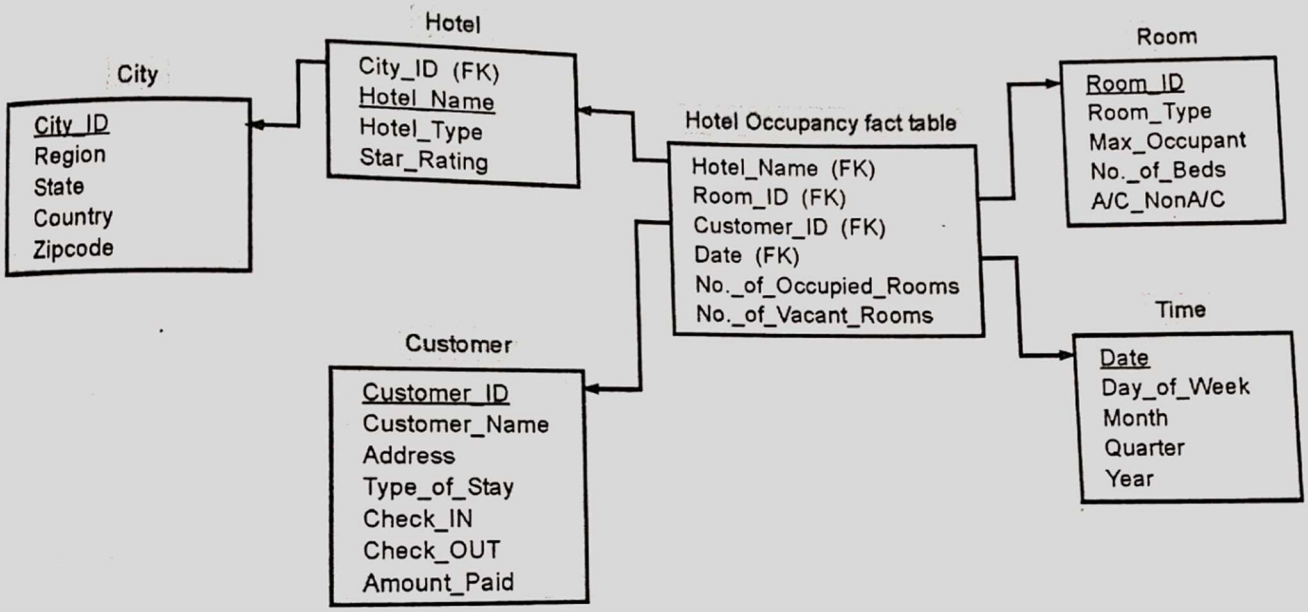
Draw information package diagram, star schema and Snowflake Schema.

(b) Star Schema



(1A9)**Fig. P. 1.7.1(a): Star Schema for Hotel Occupancy**

(c) Snowflake Schema



(1A10)**Fig. P. 1.7.1(b) : Snowflake Schema for Hotel Occupancy**

(a) Information Package Diagram

Hierarchies/Categories	Dimensions					
	Hotel	**Room**	**Time**	**Customer**	...	...
	Hotel Name	Room ID	Date	Customer ID		
	Hotel Type	Room Type	Day of Week	Customer Name		
	Star Rating	Max. Occupant	Month	Address		
	Region	No. of Beds	Quarter	Type of Stay		
	State	A/C Non A/C	Year	Check IN		
	Country			Check OUT		
				Amount Paid		
	Facts : Occupied Rooms, Vacant Rooms					

2. Differentiate between Star schema and Snowflake schema.

Feature	Star Schema	Snowflake Schema
Structure	Central fact table linked to denormalized dimension tables.	Central fact table linked to normalized dimension tables.
Complexity	Simple, easier to understand.	More complex due to normalization.
Normalization	Dimension tables are typically denormalized.	Dimension tables are normalized.
Query Performance	Faster due to fewer joins.	Slower as more joins are required.
Storage Requirement	Higher, due to redundancy in data.	Lower, as data redundancy is minimized.
Use Case	Suitable for small to medium data models.	Suitable for complex or large data models.

3. Compare OLTP and OLAP.

Feature	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Characteristic	Operational processing	Informational processing
Orientation	Transaction	Analysis
Primary Users	Clerks, DBAs, database professionals	Knowledge workers (e.g., managers, executives, analysts)
Purpose	Day-to-day operations	Long-term decision-making and information support
Database Design	ER-based, application-oriented	Star/snowflake schema, subject-oriented
Data	Current, always up-to-date	Historical, maintains accuracy over time
Data Summarization	Primitive and highly detailed	Summarized and consolidated
View	Detailed, flat relational structure	Summarized, multidimensional
Unit of Work	Short, simple transactions	Complex queries
Access Pattern	Read/write	Primarily read
Focus	Data input	Data output
Operations	Indexing or hashing on primary keys	Extensive scans
Records Accessed	Tens	Millions
Number of Users	Thousands	Hundreds
Database Size	100 MB to GB	100 GB to TB
Performance Priority	High performance, high availability	High flexibility, end-user autonomy
Key Metric	Transaction throughput	Query throughput, response time

4. **Perform OLAP operations:**

Q) There are four tables, out of 3-dimension tables and 1 fact table.

Dimension tables:

1. **Doctor (DID , name, phone, location, pin, specialization)**
2. **Patient (PID, name, phone, state, city, location, pin)**
3. **Time (TID, day, month, quarter, year)**

Fact Table:

Fact, table (DID.PID, TID, count, charge)

Perform OLAP operations on the above tables.

Creating tables

1. Doctor Table

DID	Name	Phone	Location	Pin	Specialization
1	Dr. Smith	1234567890	New York	10001	Cardiology
2	Dr. Johnson	0987654321	Los Angeles	90001	Neurology
3	Dr. Williams	5551234567	Chicago	60601	Orthopedics

2. Patient Table

PID	Name	Phone	State	City	Location	Pin
1	John Doe	1112223333	California	Los Angeles	LA Downtown	90001
2	Jane Smith	2223334444	New York	New York City	Manhattan	10001
3	Emily Johnson	3334445555	Illinois	Chicago	Loop	60601

3. Time Table

TID	Day	Month	Quarter	Year
1	15	1	1	2023
2	10	3	1	2023
3	20	6	2	2023

4. Fact Table

DID	PID	TID	Count	Charge
1	1	1	2	300.00
2	2	2	1	150.00
3	3	3	3	450.00

Performing OLAP Operations

1. Roll-Up Operation

- **Scenario**: Roll-up from the day level to the month level.
- **Result:**

DID	Month	TotalCharge
1	1	300.00
2	3	150.00
3	6	450.00

2. Drill-Down Operation

- **Scenario:** Drill-down from quarter to month.
- **Result:**

DID	Quarter	Month	TotalCharge
1	1	1	300.00
2	1	3	150.00
3	2	6	450.00

3. Slice Operation

- **Scenario:** Slice to view data for Year = 2023 only.
- **Result:**

DID	PID	TotalVisits
1	1	2
2	2	1
3	3	3

4. Dice Operation

- **Scenario:** Dice to view data for Quarter = 1 and Doctor Location = New York.
- **Result:**

Doctor Name	Patient Name	TotalCharge
Dr. Smith	Jane Smith	300.00

\

5. Pivot Operation

- **Scenario:** Pivot to swap Doctor and Patient dimensions.
- **Result:**

Patient Name	Doctor Name	TotalCharge
John Doe	Dr. Smith	300.00
Jane Smith	Dr. Johnson	150.00
Emily Johnson	Dr. Williams	450.00

5. Differentiate between ER Modelling vs Dimensional modelling.

Aspect	ER Modeling	Dimensional Modeling
Purpose	Focuses on capturing detailed transactional data.	Optimized for analytical and reporting purposes.
Structure	Complex with many entities, attributes, and relationships.	Simpler with fact and dimension tables.
Normalization	Highly normalized to eliminate redundancy.	Denormalized to improve query performance.
Schema Type	Typically results in **Relational Schema**.	Results in **Star Schema** or **Snowflake Schema**.
Usage	Used in OLTP systems (transactional systems).	Used in OLAP systems (data warehouses).
Focus	Entity relationships and operational details.	Facts (measures) and dimensions (context).
Query Performance	Slower for analytical queries due to joins.	Faster for analytical queries due to fewer joins.
Flexibility	Flexible for changes in data structure.	Less flexible due to predefined hierarchies.
Data Redundancy	Minimal (due to normalization).	Higher (due to denormalization).
Examples	Banking systems, order processing systems.	Sales analysis, customer behavior analysis.

6. Illustrate major steps in ETL process.

ETL (Extract, Transform, Load) is a process used in data integration and data warehousing to collect, process, and move data from multiple sources into a unified data store. Here are the three main steps with detailed explanations:

1. Extract

- **Purpose**: Retrieve data from various sources such as databases, files, APIs, or web services.
- **Key Activities**:
    - Identifying the data sources (e.g., relational databases, spreadsheets, cloud storage).
    - Extracting the relevant data while maintaining its integrity.
    - Handling different formats (structured, semi-structured, or unstructured data).

- **Example**: Extracting customer transaction data from a sales database and product details from an inventory system.

2. Transform

- **Purpose**: Convert and clean the extracted data into a format suitable for analysis or reporting.
- **Key Activities**:
    - **Data Cleaning**: Removing errors, handling missing values, and resolving inconsistencies.
    - **Data Integration**: Combining data from multiple sources and aligning it under a unified schema.
    - **Data Transformation**:
        - Normalizing or standardizing data.
        - Aggregating or summarizing data.
        - Encoding categorical data into numeric values.
        - Applying business rules for derived attributes.
- **Example**: Converting transaction dates into a standard format, normalizing product prices to a common currency, and removing duplicate records.

3. Load

- **Purpose**: Store the transformed data into a target system, such as a data warehouse, data lake, or analytical database.
- **Key Activities**:
    - Choosing between **full load** (loading all data) or **incremental load** (loading only changes).
    - Validating the data to ensure it has been loaded correctly.
    - Optimizing the load process for speed and efficiency.
- **Example**: Loading the cleaned and integrated sales and inventory data into a data warehouse for business intelligence reports.

7. **Write a short note on: Techniques of data loading.**

Data loading refers to the process of transferring data into a target system, such as a data warehouse. It is a critical phase in the ETL process. There are two primary techniques for data loading:

1. **Full Load**
    - Involves completely erasing existing data in the target system and reloading it with new data.
    - Typically used for initial loads or small datasets where performance is not a concern.
    - **Pros**: Simple to implement; ensures data consistency.
    - **Cons**: Time-consuming; high system downtime; unsuitable for large datasets.
2. **Incremental Load**
    - Only the new or updated records are added or modified in the target system.
    - Used for ongoing, periodic updates to maintain efficiency.
    - **Pros**: Faster and more efficient; minimal downtime.
    - **Cons**: Complex implementation; requires change data capture (CDC) mechanisms.

2. Introduction to Data mining, exploration and pre-processing

1. Describe any 5 issues in data mining.

Mining Methodology and User Interaction Issues

- Mining different types of knowledge in databases.
- Interactive mining of knowledge at various levels of abstraction.
- Incorporating background knowledge into the mining process.
- Development of data mining query languages and support for ad hoc mining.
- Presentation and visualization of data mining results.
- Handling noisy or incomplete data effectively.
- Pattern evaluation for meaningful insights.

Performance Issues

- Ensuring the efficiency and scalability of data mining algorithms.
- Supporting parallel, distributed, and incremental data mining processes.

Issues Related to the Diversity of Database Types

- Managing relational and complex data types effectively.
- Extracting information from heterogeneous databases and global information systems.

2. Explain KDD process with neat diagram

Knowledge discovery in the database(KDD) is the process of searching for hidden knowledge in the massive amounts of data that we are technically capable of generating and storing.

The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. Data Cleaning

- Removal of noise, inconsistent data, and outliers.
- Strategies to handle missing data fields.

2. Data Integration

- Data from various sources such as databases, data warehouse, and transactional data are integrated.
- Multiple data sources may be combined into a single data format.

3. Data Selection

- Data relevant to the analysis task is retrieved from the database.
- Collecting only necessary information to the model.
- Finding useful features to represent data depending on the goal of the task.

4. Data Transformation

- Data is transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.

- By using transformation methods invariant representations for the data is found.

5. Data Mining

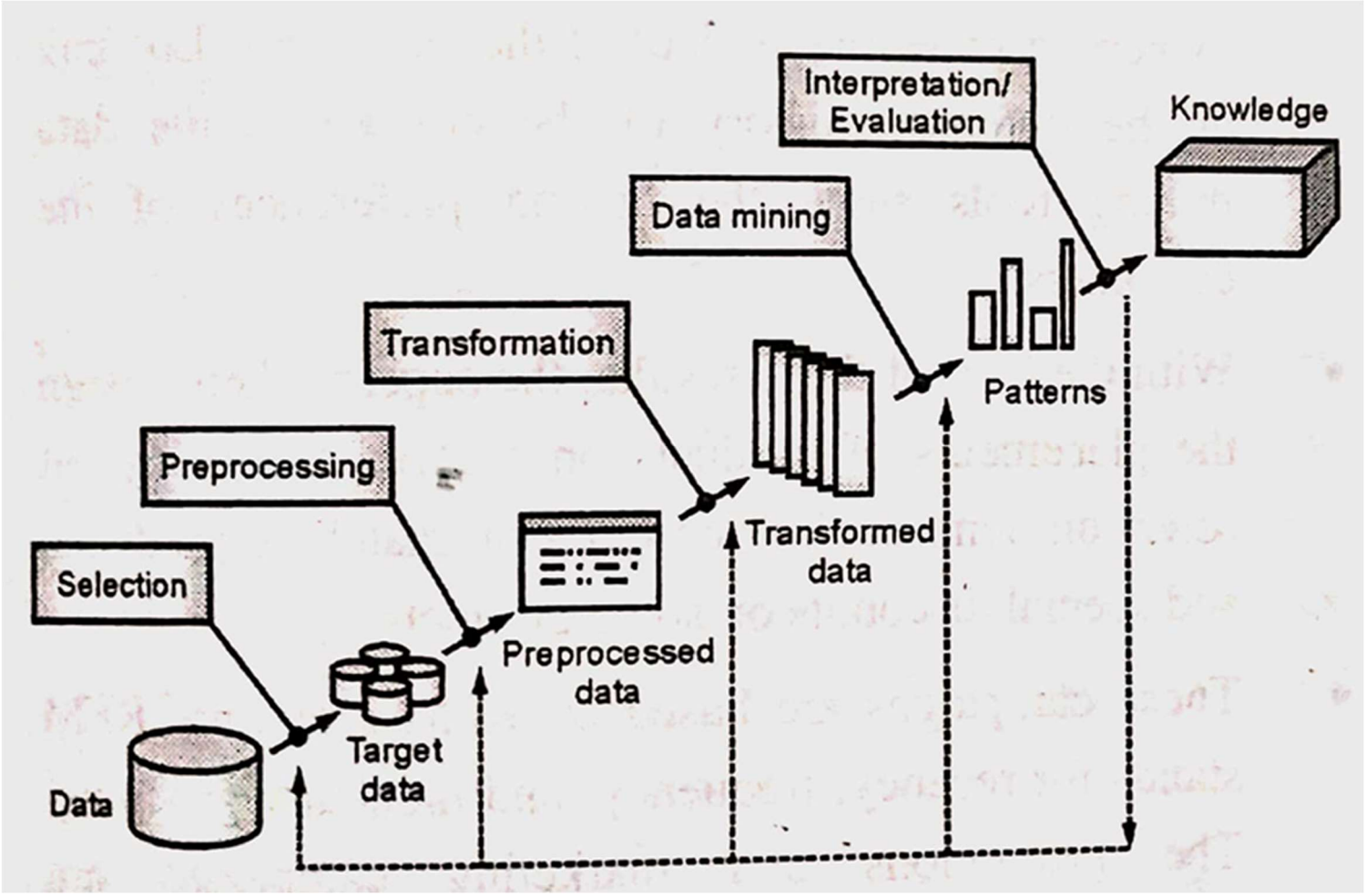
- An essential process where intelligent methods are applied to extract data patterns.
- Deciding which model and parameter may be appropriate.

6. Pattern Evaluation

- To identify the truly interesting patterns representing knowledge based on interesting measures.

7. Knowledge Presentation

- Visualization and knowledge representation techniques are used to present mined knowledge to users.
- Visualizations can be in form of graphs, charts or table.



3. Explain different data visualization techniques.

1. **Pixel-oriented visualization techniques**
   - For a data set of m dimensions, create m windows on the screen, one for each dimension.
   - The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows.
   - The colours of the pixels reflect the corresponding values.

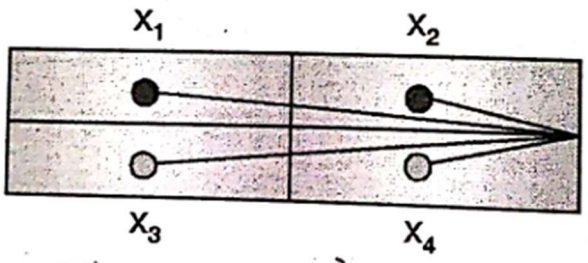
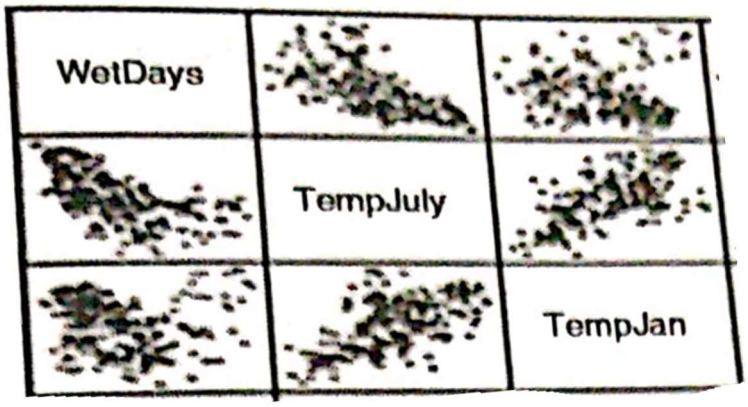


Fig. 3.10.1 : Pixel visualisation with four variables

2. Geometric Projection Visualization Techniques

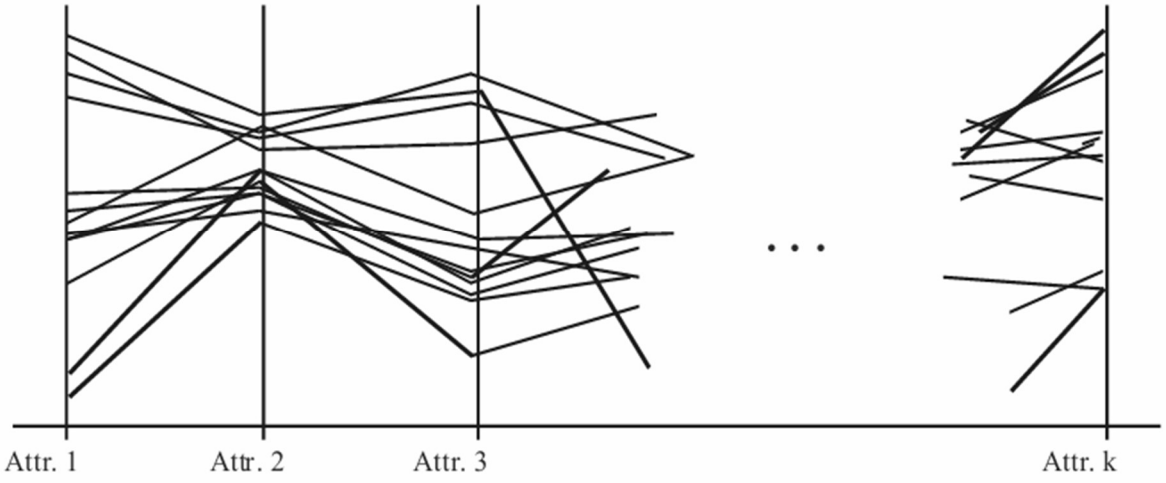
Landscapes: Represent high-dimensional data as 3D surfaces where heights indicate values. Used to detect clusters, peaks, or valleys in complex data.

Scatterplot Matrices: It is composed of scatterplots of all possible pairs of variables in a dataset.
It is used to identify relationships and correlations among variables.



Hyperslice: Visualize high-dimensional data as 2D slices for better comprehension. Used to examine multiple variable relationships individually.

Parallel Coordinates: Represent high-dimensional data using parallel axes, connecting data points with lines.
Every data item corresponds to a polygonal line which intersects each of the axes at the point which corresponds to the value for the attribute.

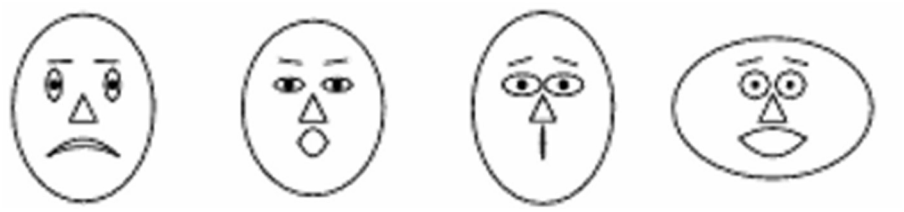


3. Icon-Based Visualization Techniques

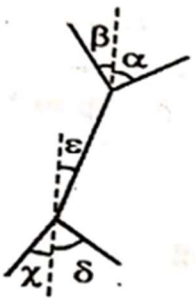
Visualization of the data values as features of icons, typical visualization methods

■ Chernoff Faces ■Stick Figures

a. **Chernoff faces:** A way to display variables on a two-dimensional surface, e.g., let x be eyebrow slant, y be eye size, z be the nose length, etc.
The figure shows faces produced using 10 characteristics--head eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening): Each assigned one of 10 possible values.



b. **Stick figures:** Represent data attributes as line segments forming a "stick figure."
It is used to highlight patterns and outliers intuitively.

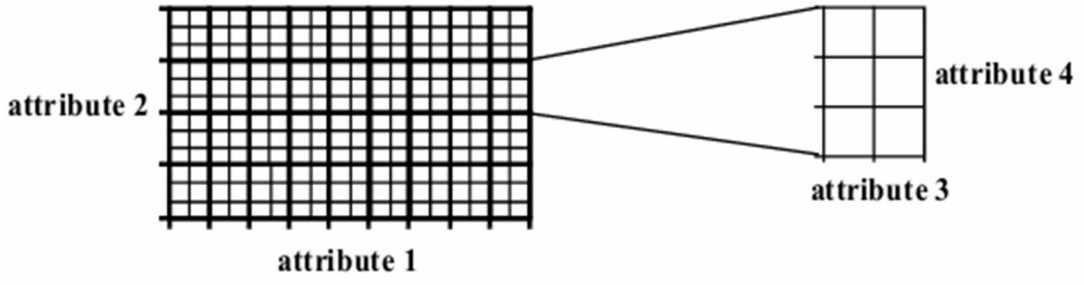


(a) A five stick figure icon with orientation

4. Hierarchical Visualization technique:

Dimensional Stacking: Partitioning of the n-dimensional attribute space in 2-D subspaces, which are 'stacked' into each other.

Partitioning of the attribute value ranges into classes. The important attributes should be used on the outer levels.



Mosaic Plot: Rectangles are used to represent the count of categorical data and at every stage rectangles are split parallel.
It is used to show relationships between categorical variables.

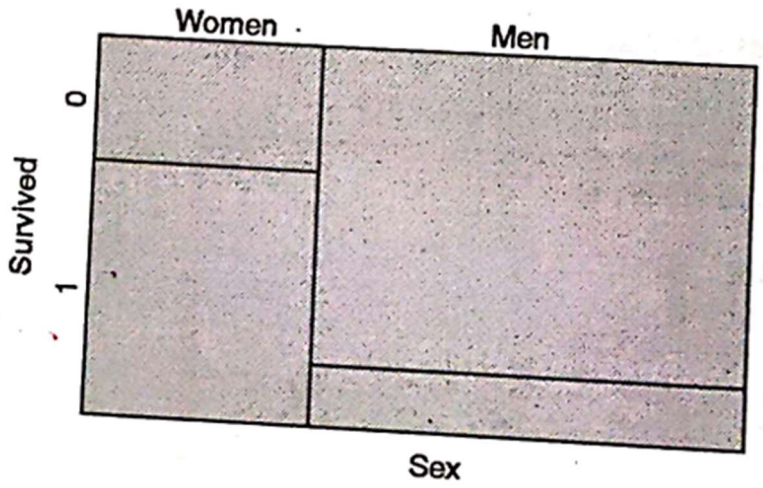


Fig. 3.10.7 : Mosaic Plot for Titanic

Worlds Within Worlds: Used to generate interactive hierarchy for display.
Innermost world must have a function and two most important parameters. Remaining parameters are fixed with constant value.

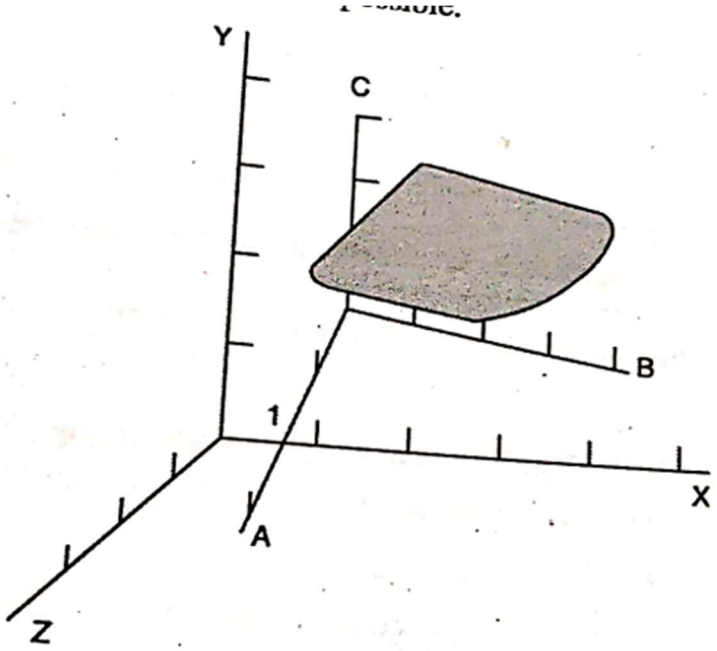
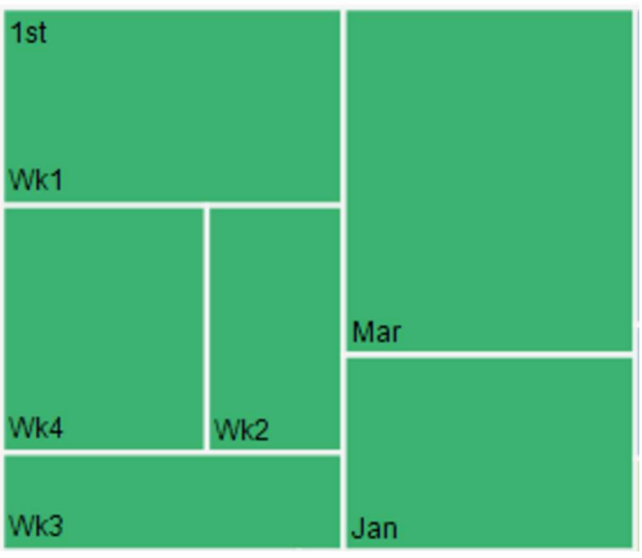


Fig. 3.10.8 : Worlds within worlds visualization

Tree Maps: Show hierarchical data as nested rectangles, with sizes indicating magnitude.



4. Explain data preprocessing. Explain different steps involved in data preprocessing.

Data preprocessing is the process of transforming raw data into a clean, usable format before applying data mining or machine learning techniques. This step is crucial for improving the accuracy, efficiency, and interpretability of analytical models.

Steps in Data Preprocessing

1. Data Cleaning

- **Objective**: Handle missing, noisy, and inconsistent data.
- **Techniques**:
    - Fill missing values with mean, median, mode, or predictions.
    - Remove or smooth noisy data using binning, regression, or clustering.
    - Resolve inconsistencies through manual corrections or domain-specific rules.

2. Data Integration

- **Objective**: Combine data from multiple sources into a cohesive dataset.
- **Techniques**:
    - Use schema integration or entity matching to merge datasets.
    - Resolve redundancies and conflicts between sources.

3. Data Transformation

- **Objective**: Convert data into a suitable format or scale for analysis.
- **Techniques**:
    - Normalization: Rescale data to a specific range (e.g., [0, 1]).
    - Aggregation: Summarize data to reduce granularity.
    - Encoding: Convert categorical data into numerical form (e.g., one-hot encoding).

4. Data Reduction

- **Objective**: Reduce the size of the dataset while retaining critical information.
- **Techniques**:
    - Sampling: Select a representative subset of data.
    - Feature selection: Identify the most relevant features.

5. Data Discretization

- **Objective**: Convert continuous data into discrete bins for analysis.
- **Techniques**:
    - Binning: Divide data into intervals.
    - Decision tree-based discretization.

5. State any five applications of data mining.

Market Basket Analysis:

- Used in retail to analyse customer purchasing patterns by identifying items frequently bought together.
- Example: Recommending products like "bread" when a customer adds "butter" to their cart.

Fraud Detection:

- Identifies unusual patterns or anomalies in transactions that may indicate fraudulent activity.
- Example: Detecting credit card fraud based on sudden, high-value purchases in foreign locations.

Healthcare and Medicine:

- Assists in diagnosing diseases by analysing patient records and medical data.
- Example: Predicting potential illnesses based on patient history and lifestyle factors.

Customer Relationship Management (CRM):

- Helps companies understand customer behaviour, segment customers, and personalize marketing strategies.
- Example: Offering discounts to high-value customers likely to switch to a competitor.

Educational Data Mining:

- Analyses student performance data to enhance teaching strategies and improve learning outcomes.
- Example: Predicting which students are at risk of dropping out based on attendance and grades.

6. Describe various methods for handling the problem of missing values in attributes.

1. Ignore the Tuple: Skip the record if the class label is missing. Useful when there are many missing attributes, but less effective otherwise.

2. Fill in Missing Values Manually: Manually input missing values. Effective for small datasets with limited missing data.

3. Use a Global Constant: Replace missing values with a constant (e.g., "Unknown" or -∞).

4. Use Central Tendency Measures: Replace missing values with the mean or median of the attribute (e.g., average customer income is $25,000).

5. Use Class-Specific Central Tendency: Replace missing values using the mean or median of samples from the same class as the given record.

6. Use the Most Probable Value: Predict missing values using methods like regression, Bayesian classification, or decision-tree induction.

7. **Explain in brief what is data discretization and concept hierarchy generation.**

Data Discretization

- **Definition**:
  Data discretization is the process of converting continuous numerical data into discrete, categorical intervals or bins. It simplifies data analysis by reducing the number of distinct values and making patterns more apparent.
- **Purpose**:
  - Simplify complex data for analysis.
  - Improve the efficiency of machine learning algorithms, especially those that perform better with categorical data.
  - Facilitate interpretation by grouping similar values.
- **Example**:
  - A continuous attribute like "age" can be discretized into intervals such as:
    - 0–18: "Child"
    - 19–35: "Young Adult"
    - 36–60: "Adult"
    - 61+: "Senior"

Concept Hierarchy Generation

- **Definition**:
  Concept hierarchy generation involves organizing data into multiple levels of abstraction, where higher levels represent broader categories, and lower levels provide finer details.
- **Purpose**:
  - Enable analysis at varying levels of granularity.
  - Provide meaningful aggregation for summarization and visualization.
  - Facilitate roll-up and drill-down operations in OLAP (Online Analytical Processing).
- **Example**:
  - For a location attribute:
    - **City Level**: New York, Los Angeles
    - **State Level**: New York, California
    - **Country Level**: United States

3. Classification

1. Describe in detail about how to evaluate accuracy of the classifier.

1. Holdout Method

This is the most basic method:

- **How it works:** You split your data into two parts: one for training the model and the other for testing how well it works. A common split is 70% for training and 30% for testing.
- **Pros:** It's quick and easy.
- **Cons:** The result can change depending on how you split the data.

2. Random Subsampling

It's similar to the Holdout method but repeated multiple times.

- **How it works:** You randomly split the data many times and train/test the model each time. Then, you average the results.
- **Pros:** More reliable than Holdout because you do multiple splits.
- **Cons:** Still can be biased if some important data points are always left out.

3. Cross-Validation (k-Fold Cross-Validation)

A more reliable method that splits the data into equal parts, used many times.

- **How it works:** You divide the data into several parts (say 5 parts). You train the model on 4 parts and test it on the 5th part. Repeat this process 5 times, using each part as the test set once. Then, average the results.
- **Pros:** Gives a more accurate idea of how the model will perform on new data.
- **Cons**: Takes longer because you're training and testing multiple times.

4. Bootstrap

This method creates many samples from your data by picking some data points randomly, with replacement.

- **How it works:** You make many copies of your data by sampling, then train the model on each copy and test on the leftover data. Average the results.
- **Pros:** Works well for small datasets and gives a more stable result.
- **Cons:** Takes a lot of computing power.

Summary:

- **Holdout:** Quick but less reliable.
- **Random Subsampling:** Better than Holdout but not perfect.
- **Cross-Validation:** More reliable and widely used.
- **Bootstrap:** Stable results, good for small datasets, but slow to run.

2. Apply Naïve Bayes classification:

The following table consists of training data from a database Apply Naive Bayes Algorithm and classify the tuple = X(age "<=30". Income = "low", student = "yes" and credit - rating = "Excellent")

Age	Income	Student	Credit_Rating	Class : buys_computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31..40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31..40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31..40	Medium	No	Excellent	Yes
31..40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

Soln :

Class label Attribute is Buys - computer.

C_1 : Buys - computer = Yes = 9 samples.

C_2 : Buys - computer = No = 5 samples

$\therefore P(C_1) = \frac{9}{14}$ and $P(C_2) = \frac{5}{14}$

Let event X_1 be age = "<=30"

event X_2 be income = "low"

event X_3 be student = "yes"

event X_4 be credit_rating = "Excellent"

Formula :

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

1) To compute $P(X|C_1)$:

$$P(X|C_1) = P(X_1|C_1) \cdot P(X_2|C_1) \cdot P(X_3|C_1) \cdot P(X_4 \cdot C_1)$$

$$\therefore P(X_1|C_1) = P\left(\frac{\text{Age "<=30"}}{\text{Buys-computer = Yes}}\right) = \frac{2}{9}$$

$$P(X_2|C_1) = P\left(\frac{\text{Income = "low"}}{\text{buys-computer = Yes}}\right) = \frac{3}{9}$$

$$P(X_3|C_1) = P\left(\frac{\text{Student = "Yes"}}{\text{buys-computer = Yes}}\right) = \frac{6}{9}$$

$$P(X_4|C_1) = P\left(\frac{\text{credit-rating = "Excellent"}}{\text{buys-computer = Yes}}\right) = \frac{3}{9}$$

$$\therefore P(X|C_1) = \frac{2}{9} \times \frac{3}{9} \times \frac{6}{9} \times \frac{3}{9} = 0.016$$

$$P(X|C_1) \cdot P(C_1) = 0.016 \times \frac{9}{14} = 0.0105 \approx 0.011 \quad \text{— (1)}$$

2) To compute $P(X|C_2)$

$$P(X|C_2) = P(X_1|C_2) \cdot P(X_2|C_2) \cdot P(X_3|C_2) \cdot P(X_4|C_2)$$

$$P(X_1|C_2) = P\left(\frac{\text{Age "<=30"}}{\text{buys-computer = ~~Yes~~ No}}\right) = \frac{3}{5}$$

$$P(X_2|C_2) = P\left(\frac{\text{Income = "low"}}{\text{buys-computer = No}}\right) = \frac{1}{5}$$

$$P(X_3 | C_2) = P\left(\frac{\text{Student} = \text{"Yes"}}{\text{buys\_computer} = \text{No}}\right) = \frac{1}{5}$$

$$P(X_4 | C_2) = P\left(\frac{\text{credit\_rating} = \text{"Excellent"}}{\text{buys\_computer} = \text{No}}\right) = \frac{3}{5}$$

$$P(X | C_2) = \frac{3}{5} * \frac{1}{5} \times \frac{1}{5} \times \frac{3}{5} = 0.014$$

$$\therefore P(X|C_2) \cdot P(C_2) = 0.014 \times \frac{5}{14} = 0.005 \quad -(2)$$

Naive Bayes classification will assign sample X to class c_i if and only if

$$P(c_i|X) > P(c_j|X)$$

$$\therefore \text{i.e } \frac{P(X|c_i) \cdot P(c_i)}{P(X)} > \frac{P(X|c_j) \cdot P(c_j)}{P(X)}$$

$P(X)$ is constant for both classes c_i & c_j

$$\therefore P(X|c_i) \cdot P(c_i) > P(X|c_j) \cdot P(c_j) \quad -(3)$$

$\therefore$ from ① and ② and ③

$$P(X|C_1) \cdot P(C_1) > P(X|C_2) \cdot P(C_2)$$
$$0.011 > 0.005$$

$\therefore$ We conclude that the unknown sample $X = (\text{Age} = \text{"<=30"}, \text{Income} = \text{"low"}, \text{student} = \text{"Yes"}, \text{credit\_rating} = \text{Excellent})$ belongs to the class $C_1 = \text{Buys\_computer} = \text{Yes}$.

3. Explain Decision tree-based classification approach with example. OR Build decision tree for given problem:

Ex. 4.2.4 : Using the following training data set. Create classification model using decision-tree and hence classify following tuple.

Table P. 4.2.4

Tid	Income	Age	Own House
1.	Very High	Young	Yes
2.	High	Medium	Yes
3.	Low	Young	Rented
4.	High	Medium	Yes
5.	Very high	Medium	Yes
6.	Medium	Young	Yes
7.	High	Old	Yes
8.	Medium	Medium	Rented
9.	Low	Medium	Rented
10.	Low	Old	Rented
11.	High	Young	Yes
12.	medium	Old	Rented

Soln. :

Class P : Own house = "yes"

Class N: Own house = "rented"

Total number of records 12

Count the number of records with "yes" class and "rented" class.

So number of records with "yes" class = 7 and "no" class = 5

$$\text{So Information gain} = I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$I(p, n) = I(7, 5) = -(7/12)\log_2(7/12) - (5/12)\log_2(5/12) = 0.979$$

Step 1 : Compute the entropy for Income : (Very high, high, medium, low)

For Income = Very high,

p_i = with "yes" class = 2 and n_i = with "no" class = 0

Therefore, $I(p_i, n_i) = I(2,0) = 0$

Similarly for different Income ranges $I(p_i, n_i)$ is calculated as given below :

Table P. 4.2.4(a)

Income	p_I	n_I	$I(p_I, n_I)$
Very high	2	0	0
High	4	0	0
Medium	1	2	0.918
Low	0	3	0

Calculate entropy using the values from the Table P. 4.2.4(a) and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(\text{Income}) = 2/12 * I(2,0) + 4/12 * I(4,0) + 3/12 * I(0,3) = 0.229$$

Note : S is the total training set.

Hence

$$\text{Gain}(S, \text{Income}) = I(p, n) - E(\text{Income})$$

$$= 0.979 - 0.229 = 0.75$$

Step 2 : Compute the entropy for Age : (Young , medium, old)

Similarly for different age ranges $I(p_i, n_i)$ is calculated as given below :

Table P. 4.2.4(b)

Age	p_I	n_I	$I(p_i, n_i)$
Young	3	1	0.811
Medium	3	2	0.971
Old	1	2	0.918

Calculate entropy using the values from the Table P. 4.2.4(b) and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_I + n_I}{p + n} I(p_I, n_I)$$

$$E(Age) = 4/12 * I(3,1) + 5/12*I(3,2) + 3/12*I(1,2)$$
$$= 0.904$$

Note : S is the total training set.

Hence

$$Gain(S, age) = I(p, n) - E(age)$$
$$= 0.979 - 0.904$$
$$= 0.075$$

Income attribute has the highest gain, therefore it is used as the decision attribute in the root node.

Since income has four possible values, the root node has four branches (very high, high, medium, low).

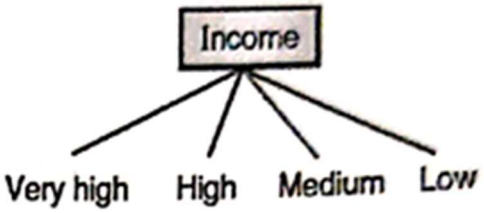


Fig. P. 4.2.4

Step 3 :

Since we have used income at the root, now we have to decide on the age attribute.

Consider income = "very high" and count the number of tuples from the original given training set

$$S_{very\ high} = 2$$

Since both the tuples have class label ="yes" , so directly give "yes" as a class label below "very high".

Similarly check the tuples for income= "high" and income = "low", are having the class label "yes" and "rented" respectively.

Now check for income = "medium", where number of tuples having "yes" class label is 1 and tuples having "rented" class label are 2.

So put the age label below income = "medium".

So the final decision tree is :

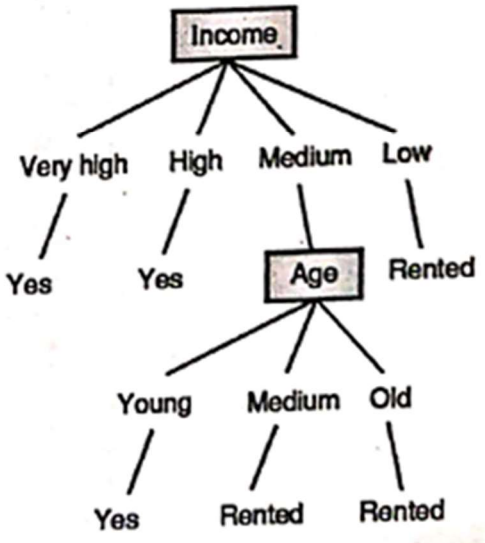


Fig. P. 4.2.4(a)

4. **Explain how Naive Bayes classification makes predictions and discuss the "naive" assumption in Naive Bayes. Provide an example to illustrate the application of Naive Bayes in a real-world scenario.**

How Naive Bayes Classification Makes Predictions

Naive Bayes is a probabilistic classifier based on **Bayes' Theorem**, which calculates the probability of a class given certain features. It assumes that all features are **conditionally independent** given the class, hence the term "naive."

Steps for Prediction:

1. **Calculate Prior Probabilities**: The probability of each class in the dataset, denoted as P(Class)
2. **Calculate Likelihoods**: For each feature value compute the likelihood i.e., the probability of the feature value given the class.
3. **Apply Bayes' Theorem**: Compute the posterior probability for each class:

$$P(\text{Class}|\text{Features}) \propto P(\text{Class}) \times \prod_{i=1}^{n} P(x_i|\text{Class})$$

4. **Predict the Class**: Choose the class with the highest posterior probability.

The "Naive" Assumption

The naive assumption in Naive Bayes is that all features are **independent** of each other given the class label. This means the probability of multiple features occurring together is simply the product of their individual probabilities. Mathematically:

$$P(x_1, x_2, \ldots, x_n|\text{Class}) = \prod_{i=1}^{n} P(x_i|\text{Class})$$

Example: Spam Email Detection

A Naive Bayes classifier is used to determine if an email is "Spam" or "Not Spam" based on words in the email.

Steps:

1. **Training Data**: Emails labelled as "Spam" or "Not Spam." Features are the words (e.g., "Buy," "Offer," "Meeting").
2. **Model Training**: Calculate probabilities for each word appearing in spam and not spam emails.
3. **Prediction**: For a new email, like *"Buy now for a great offer"*, the model calculates:
    - P(Spam|Email)
    - P(Not Spam|Email)

    The class with the higher probability is the prediction (e.g., "Spam").

Real-World Impact:

This approach enables email systems to filter spam effectively, improving user experience.

4. Clustering

1. Cluster given data using k-means algorithm:

1.1] For 2 clusters

Q) Suppose the data for clustering is {2,4,10,12,3,20,30,11,25} consider k=2.

1. Given : {2,4,10,12,3,20,30,11,25}, Randomly assign alternative values to each cluster.
2. Number of cluster = 2, therefore

K_1 = {2,10,3,30,25}, Mean = 14

K_2 = {4,12,20,11}, Mean = 11.75

3. Re-assign

K_1 = {20,30,25}, Mean = 25

K_2 = {2,4,10,12,3,11}, Mean = 7

4. Re-assign

K_1 = {20,30,25}, Mean = 25

K_2 = {2,4,10,12,3,11}, Mean = 7

So the final answer is K_1 = {2,3,4,10,11,12}, K_2 = {20,30,25}

1.2] For 3 clusters:

Ex. 4.8.2 : Use K-means algorithm to create 3 - clusters for given set of values : {2, 3, 6, 8, 9, 12, 15, 18, 22}

Soln. :

1. 2, 3, 6, 8, 9, 12, 15, 18, 22 – break into 3 clusters (Randomly assign data to three clusters) and calculate the mean value.

K_1 = 2, 8, 15 – mean = 8.3 ; K_2 = 3, 9, 18 – mean = 10

K_3 = 6, 12, 22 – mean = 13.3

2. Re-assign

K_1 = 2, 3, 6, 8, 9 – mean = 5.6 ; K_2 = mean = 0

K_3 = 12, 15, 18, 22 – mean = 16.75

3. Re-assign

K_1 = 3, 6, 8, 9 – mean = 6.5 ; K_2 = 2 – mean = 2

K_3 = 12, 15, 18, 22 – mean = 16.75

4. Re-assign

$K_1 = 6, 8, 9$ – mean = 7.6 ; $K_2 = 2, 3$ – mean = 2.5

$K_3 = 12, 15, 18, 22$ – mean = 16.75

5. Re-assign

$K_1 = 6, 8, 9$ – mean = 7.6 ; $K_2 = 2, 3$ - mean = 2.5

$K_3 = 12, 15, 18, 22$ – mean = 16.75

6. Last two groups are same. So finally we got 3 clusters

Cluster 1 = {6,8,9} , Cluster 2 = { 2,3 } , Cluster 3 = { 12,15,18,22 }

1.3] For points:

Ex. 4.8.6 : Suppose that the data mining task is to cluster the following points (with (x,y) representing locations) into 3 clusters.

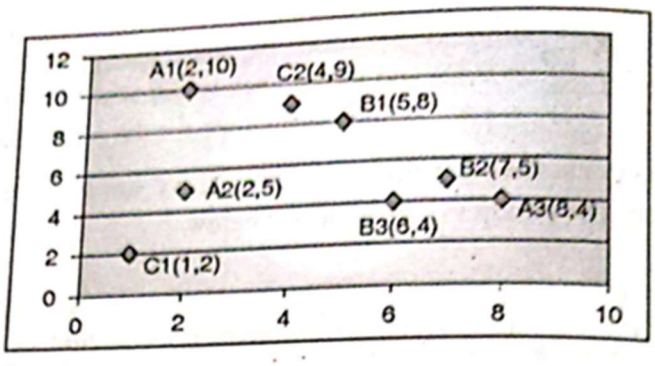
A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)

The distance function is Euclidean distance. Suppose initially we assign A1, B1 and C1 as the centre of each cluster respectively, use the K-means algorithm to show only

(a) The three cluster centers after the first round execution.

(b) The final three clusters.

Soln. :



1. **Initial value of centroids :** In this we use A1 , B1and C1 as the first centroids. Let X1 and X2, X3 denote the coordinate of the centroids, then X1 = A1(2,10), X2 = B1(5,8) and X3 = C1(1,2)

2. **Objects-centroids distance :** We calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

D^0 =

A1	A2	A3	B1	B2	B3	C1	C2	
0	5	8.48	3.61	7.07	7.21	8.06	2.24	X1=A1(2,10)
3.61	4.24	5	0	3.61	4.12	7.21	1.41	X2=B1(5,8)
8.06	3.16	7.28	7.21	6.71	5.39	0	7.62	X3=C1(1,2)

3. **Objects clustering :** We assign each object based on the minimum distance. Thus, A1 is assigned to group 1, point A3, B1, B2, B3, C2 are assigned to group 2 and A2 and C1 are assigned to group 3. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$G^0 =$

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	0	0	0	0	0	X1=A1(2,10)
0	0	1	1	1	1	0	1	X2=B1(5,8)
0	1	0	0	0	0	1	0	X3=C1(1,2)

4. **Iteration-1, determine centroids**

$$X1 = (2,10)$$

$$X2 = \left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right) = (6,6)$$

$$X3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1.5,3.5)$$

Iteration-1, objects-centroids distances : The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

$D^1 =$

A1	A2	A3	B1	B2	B3	C1	C2	
0	.5	8.48	3.61	7.07	7.21	8.06	2.24	X1(2,10)
5.66	4.12	2.83	2.24	1.41	2	6.40	3.61	X2(6,6)
6.52	1.58	6.52	5.70	5.70	4.52	1.58	6.04	X3(1.5,3.5)

Iteration-1, objects clustering : Similar to step 3, we assign each object based on the minimum distance. The Group matrix is shown below.

$G^1 =$

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	0	0	0	0	1	X1(2,10)
0	0	1	1	1	1	0	0	X2(6,6)
0	1	0	0	0	0	1	0	X3(1.5,3.5)

Iteration-2, determine centroids

$$X1 = ((2+4)/2,\ (10+9)/2) = (3, 9.5)$$

$$X2 = ((8+5+7+6)/4, (4+8+5+4)/4) = (6.5, 5.25)$$

$$X3 = ((2+1)/2, (5+2)/2) = (1.5, 3.5)$$

Iteration-2, objects-centroids distances

$D^2 =$

A1	A2	A3	B1	B2	B3	C1	C2	
1.12	2.35	7.43	2.5	6.02	6.26	7.76	1.12	X1(3,9.5)
6.54	4.51	1.95	3.13	0.56	1.35	6.38	7.68	X2(6.5 ,5.25)
6.52	1.58	6.52	5.70	5.70	4.52	1.58	6.04	X3(1.5,3.5)

Iteration-2, objects clustering : We assign each object based on the minimum distance. The Group matrix is shown below.

G^2 =

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	1	0	0	0	1	X1(3,9.5)
0	0	1	0	1	1	0	0	X2(6.5 ,5.25)
0	1	0	0	0	0	1	0	X3(1.5,3.5)

10. Iteration-3, determine centroids

$$X1 = ((2+5+4)/3,\ (10+9+8)/3) = (3.67, 9)$$

$$X2 = ((8+7+6)/3, (4+5+4)/3) = (7, 4.33)$$

$$X3 = ((2+1)/2, (5+2)/2) = (1.5, 3.5)$$

11. Iteration-3, objects-centroids distances

D^3 =

A1	A2	A3	B1	B2	B3	C1	C2	
1.95	4.33	6.61	1.66	5.20	5.52	7.49	0.33	X1(3.67,9)
6.01	5.04	1.05	4.17	0.67	1.05	6.44	5.55	X2(7 ,4.33)
6.52	1.58	6.52	5.70	5.70	4.52	1.58	6.04	X3(1.5,3.5)

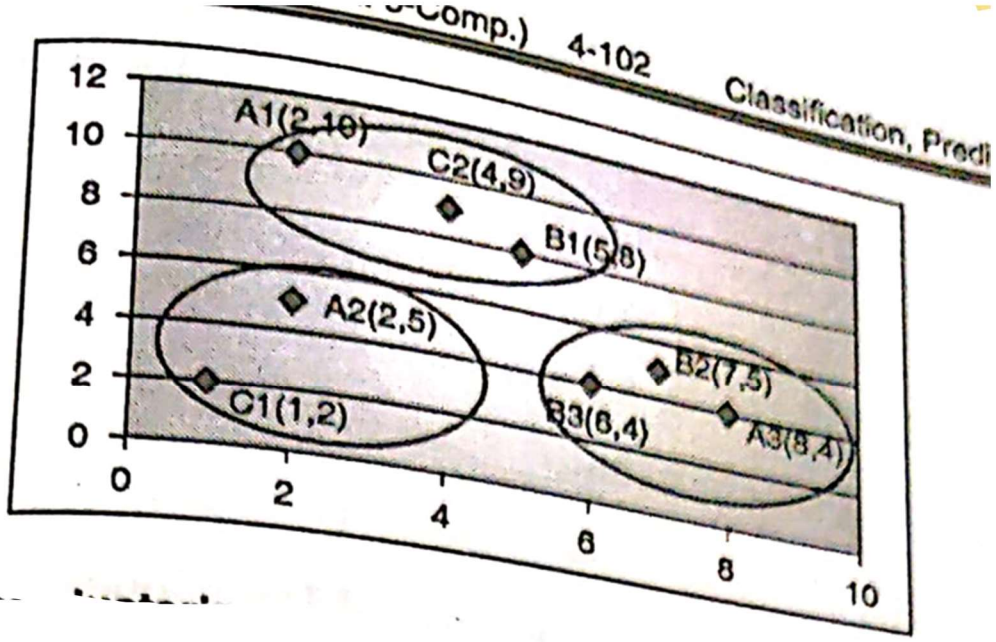
12. Iteration-3, objects clustering : We assign each object based on the minimum distance. The Group matrix is shown below.

G^3 =

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	1	0	0	0	1	X1(3.67,9)
0	0	1	0	1	1	0	0	X2(7 ,4.33)
0	1	0	0	0	0	1	0	X3(1.5,3.5)

By comparing $G^3 = G^2$ we see that the objects do not move to new group therefore we can say that K means has reached its stability.

So final clusters are Group 1= { A1, B1, C2 } and Group 2 = { A3, B2 , B3} and Group 3 = {A2, C1}.



2. Explain K-means clustering algorithm and draw flowchart. Discuss its advantages and limitations.

K-Means is an iterative clustering algorithm used to partition a dataset into k distinct, non-overlapping clusters based on their similarity. It minimizes the variance within each cluster and requires the number of clusters k to be specified in advance.

Flowchart

1. **Start**
   ↓
2. Input the number of clusters k
   ↓
3. Initialize k centroids randomly
   ↓
4. Assign each data point to the nearest centroid
   ↓
5. Update centroids based on cluster members
   ↓
6. Check for convergence (centroids stop changing or max iterations reached)
   - If **No**, repeat steps 4–5
   - If **Yes**, proceed
     ↓
7. Output clusters and centroids
   ↓
8. **End**

Advantages of K-Means

- **Simplicity**: Easy to implement and computationally efficient.
- **Scalability**: Works well for large datasets.
- **Interpretability**: Produces distinct, non-overlapping clusters.
- **Speed**: Quick for small to medium-sized data.

Limitations of K-Means

- **Fixed k**: Requires k (number of clusters) to be pre-specified, which is not always intuitive.
- **Sensitivity to Initialization**: Different initial centroids can lead to different results (may converge to a local minimum).
- **Poor with Non-Spherical Clusters**: Assumes clusters are spherical and equal in size, struggling with complex shapes.

3. **Differentiate between Agglomerative and Divisive clustering method.**

Aspect	Agglomerative Clustering	Divisive Clustering
Approach	Bottom-up: Starts with each data point as a separate cluster.	Top-down: Starts with all data points in one cluster.
Process	Merges clusters iteratively until only one cluster remains or desired number of clusters is reached.	Splits clusters iteratively until each data point is its own cluster or desired number of clusters is reached.
Granularity at Start	Begins with the finest granularity (single data points).	Begins with the coarsest granularity (entire dataset).
Complexity	Generally faster as fewer decisions are needed at each step.	More computationally intensive as splitting decisions are more complex.
Use Cases	More commonly used due to its simplicity and lower computational cost.	Used less frequently but can be effective for specific datasets with clear top-level separation.
Output Hierarchy	Builds a hierarchical tree by merging clusters.	Builds a hierarchical tree by splitting clusters.
Examples	Single-linkage, complete-linkage, average-linkage clustering.	Rarely specific examples, but uses similar linkage criteria as agglomerative clustering for splitting.

4. **Describe K-medoids algorithm.**

K-Medoids is a clustering algorithm similar to K-Means but is more robust to outliers and noise. Instead of using the mean as the centre of clusters, K-Medoids selects actual data points (called medoids) as cluster centres, minimizing the sum of dissimilarities between points and their medoid.

Steps of K-Medoids Algorithm

1. **Initialization**: Select k random data points as initial medoids.
2. **Assign Clusters**: Assign each data point to the nearest medoid based on a dissimilarity metric (e.g., Manhattan distance).
3. **Update Medoids**:
    - For each cluster, choose a data point as the new medoid if it minimizes the total distance (dissimilarity) within the cluster.
4. **Iterate**: Repeat steps 2 and 3 until the medoids do not change or the improvement in clustering stops.
5. **Output**: Final medoids and clusters.

5. Apply single linkage clustering and construct dendrogram:

Ex. 4.9.5 : Following table gives fat and proteins content of items. Apply single linkage clustering and construct dendrogram.

Food Item	Protein	Fat
1	1.1	60
2	8.2	20
3	4.2	35
4	1.5	21
5	7.6	15
6	2.0	55
7	3.9	39

Soln. :

Step 1 : Calculate the distance matrix using Euclidian distance formula :

Cluster number	C1	C2	C3	C4	C5	C6	C7
C1	0	40.62	25.19	39.00	45.46	5.08	21.18
C2		0	15.52	6.77	5.03	35.54	19.48
C3			0	14.25	20.28	20.12	4.01
C4				0	8.55	34.00	18.19
C5					0	40.39	24.28
C6						0	16.11

From the above table, the minimum distance between any two points is 4.01 and this distance is between C3 and C7. So, these two points can be merged into a single point (cluster) and is called the C37.

Step 2 : Calculate the new distance matrix with C37 using single linkage clustering.

Therefore,

$$dis(C37,4) = \min(dis(3,4), dis(7,4)) = \min(14.25, 18.19) = 14.25$$

Similarly, calculate the other distances to get the distance matrix.

Cluster number	C1	C2	C37	C4	C5	C6
C1	0	40.62	21.18	39.00	45.46	5.08
C2		0	15.52	6.77	5.03	35.54
C37			0	14.25	20.28	16.11
C4				0	8.55	34.00
C5					0	40.39
C6						0

In the above matrix distance between points 2 and 5 is minimum i.e. 5.03. So combine the cluster as C25.

Step 3 : Calculate the new distance matrix with C25 using single linkage clustering.

Cluster number	C1	C25	C37	C4	C6
C1	0	40.62	21.18	39.00	5.08
C25		0	15.52	6.77	35.54
C37			0	14.25	16.11
C4				0	34.00
C6					0

In the above matrix distance between C1 and C6 in minimum which is 5.08. So newly formed new cluster is C16.

Step 4 : Calculate the new distance matrix with cluster C16.

Cluster number	C16	C25	C37	C4
C16	0	35.54	16.11	34.00
C25		0	15.52	6.77
C37			0	14.25
C4				0

The minimum distance is 6.77. So combine clusters C25 and C4

Step 5 : Calculate new distance matrix.

Cluster number	C16	C254	C37
C16	0	34.00	16.11
C254		0	14.25
C37			0

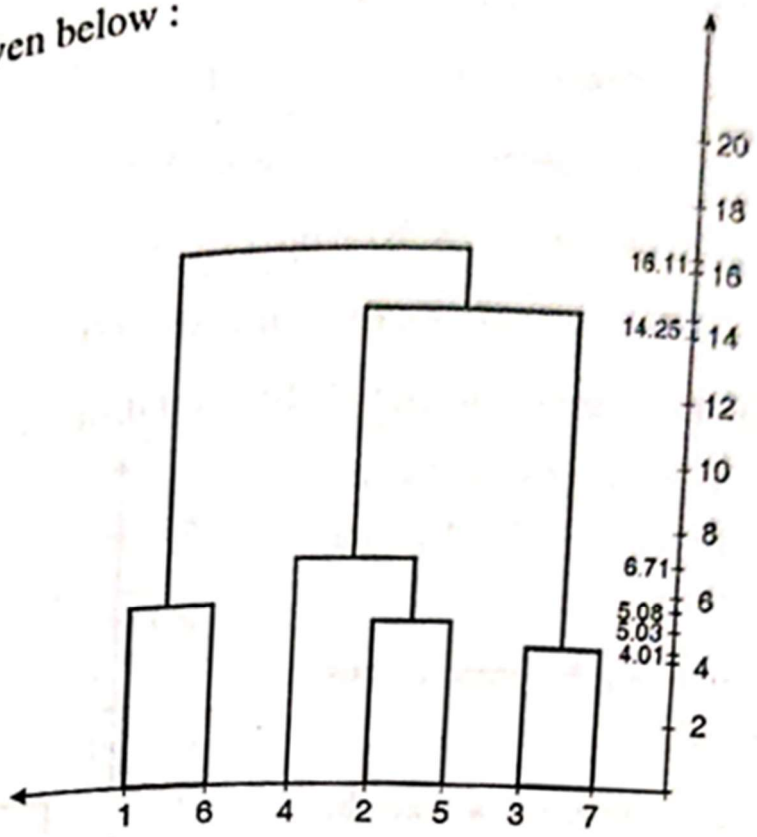
Combine the clusters C254 and C37 which has minimum distance 14.25.

Step 6 : New distance matrix is

Cluster number	C16	C25437
C16	0	16.11
C25437		0

So finally combine the clusters C25437 and C16.

Dendrogram is given below :

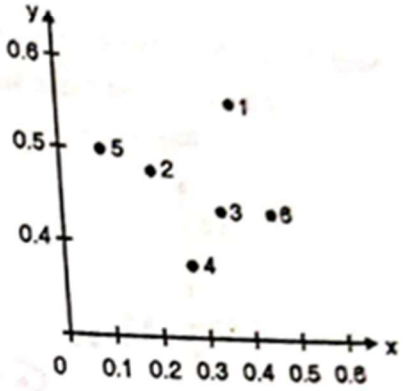


6. Apply complete linkage clustering and construct dendrogram:

		X	Y
	p1	0.40	0.53
	p2	0.22	0.38
D =	p3	0.35	0.32
	p4	0.26	0.19
	p5	0.08	0.41
	p6	0.45	0.30

Soln. :

Step 1 : Plot the objects in n-dimensional space (where n is the number of attributes). In our case we have 2 attributes x and y, so we plot the objects p1, p2, ... p6 in 2-dimensional space :



Step 2 : Calculate the distance from each object (point) to all other points, using Euclidean distance measure, and place the numbers in a distance matrix.

The formula for Euclidean distance between two points i and j is :

$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j1}|^2 + \ldots + |x_{ip} - x_{jp}|^2}$$

where x_{i1} is the value of attribute 1 for i and x_{j1} is the value of attribute 1 for j and so on, as many attributes we have ... shown up to p i.e. x_{ip} in the formula.

In our case, we only have 2 attributes. So, the Euclidean distance between our points p1 and p2, which have attributes x and y would be calculated as follows:

$$d(p1, p2) = \sqrt{|xp_1 - xp_1|^2 + |yp_1 - yp_2|^2}$$

$$= \sqrt{|0.40 - 0.22|^2 + |0.53 - 0.38|^2}$$

$$= \sqrt{|0.18|^2 + |0.15|^2}$$

$$= \sqrt{0.0324 + 0.0225}$$

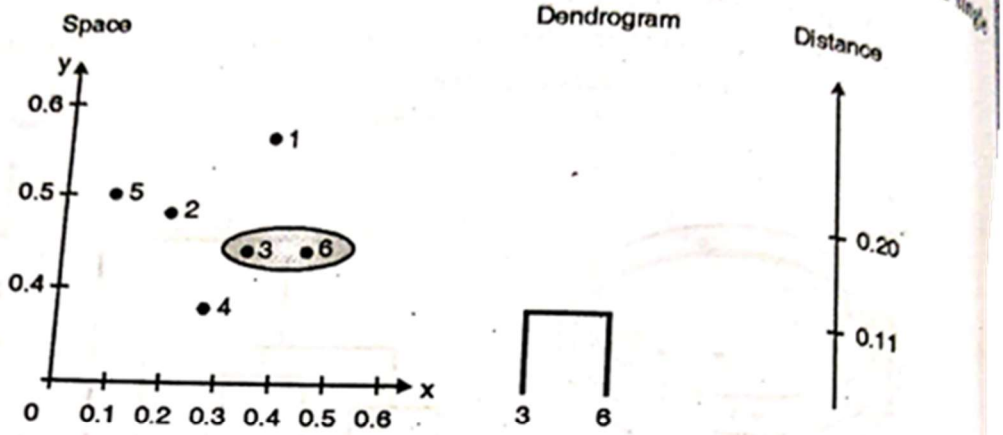
$$= \sqrt{0.0549} = 0.2343$$

Analogically, we calculate the distance to the remaining points and we will receive the following values :

Distance matrix :

p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0
	p1	p2	p3	p4	p5	p6

Step 3 : Identify the two clusters with the shortest distance in the matrix, and merge them together. Re-compute the distance matrix, as those two clusters are now in a single cluster.



By looking at the distance matrix above, we see that p3 and p6 have the smallest distance from all i.e. 0.11 So, we merge those two in a single cluster and re-compute the distance matrix.

$dist((p3, p6), p1) = MAX(dist(p3, p1), dist(p6, p1))$

$= MAX(0.22, 0.23)$ //from original matrix

$= 0.23$

$dist((p3, p6), p2) = MAX(dist(p3, p2), dist(p6, p2))$

$= MAX(0.15, 0.25)$ //from original matrix

$= 0.25$

$dist((p3, p6), p4) = MAX(dist(p3, p4), dist(p6, p4))$

$= MAX(0.15, 0.22)$ //from original matrix

$= 0.22$

$dist((p3, p6), p5) = MAX(dist(p3, p5), dist(p6, p5))$

$= MAX(0.28, 0.39)$ //from original matrix

$= 0.39$

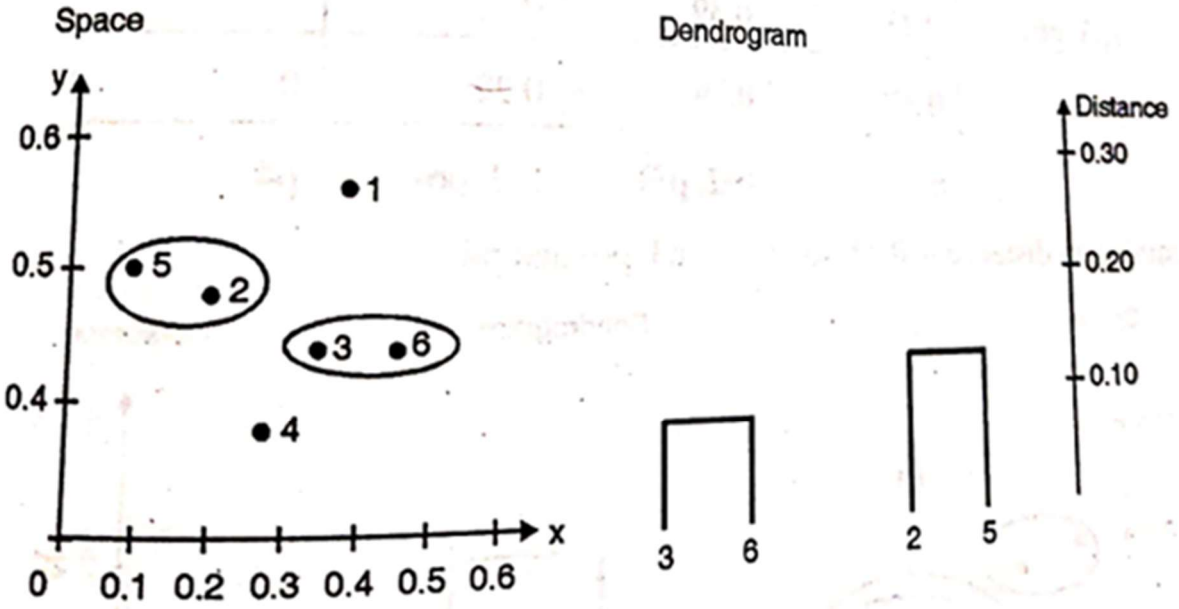
New distance matrix

	p1	p2	(p3, p6)	p4	p5
p1	0				
p2	0.24	0			
(p3, p6)	**0.23**	**0.25**	0		
p4	0.37	0.20	**0.22**	0	
p5	0.34	0.14	**0.39**	0.29	0

Step 4 : Consider the following distance matrix

	p1	p2	(p3, p6)	p4	p5
p1	0				
p2	0.24	0			
(p3, p6)	**0.23**	**0.25**	0		
p4	0.37	0.20	**0.22**	0	
p5	0.34	0.14	**0.39**	0.29	0

So, looking at the above distance matrix, we see that p2 and p5 have the smallest distance from all – 0.14. So, we merge those two in a single cluster, and re-compute the distance matrix using the following calculations.



$dist((p2, p5), p1) = MAX(dist(p2, p1), dist(p5, p1))$

$= MAX(0.24, 0.34)$ //from original matrix

$= 0.34$

dist((p2, p5), (p3,p6)) = MAX (*dist*(p2, p3) , *dist*(p2, p6), *dist*(p5, p3), *dist*(p5, p6))

= MAX (0.15, 0.25, 0.28, 0.39) //from original matrix

= 0.39

dist((p2, p5), p4) = MAX (*dist*(p2, p4) , *dist*(p5, p4))

= MAX (0.20, 0.29) //from original matrix

= 0.29

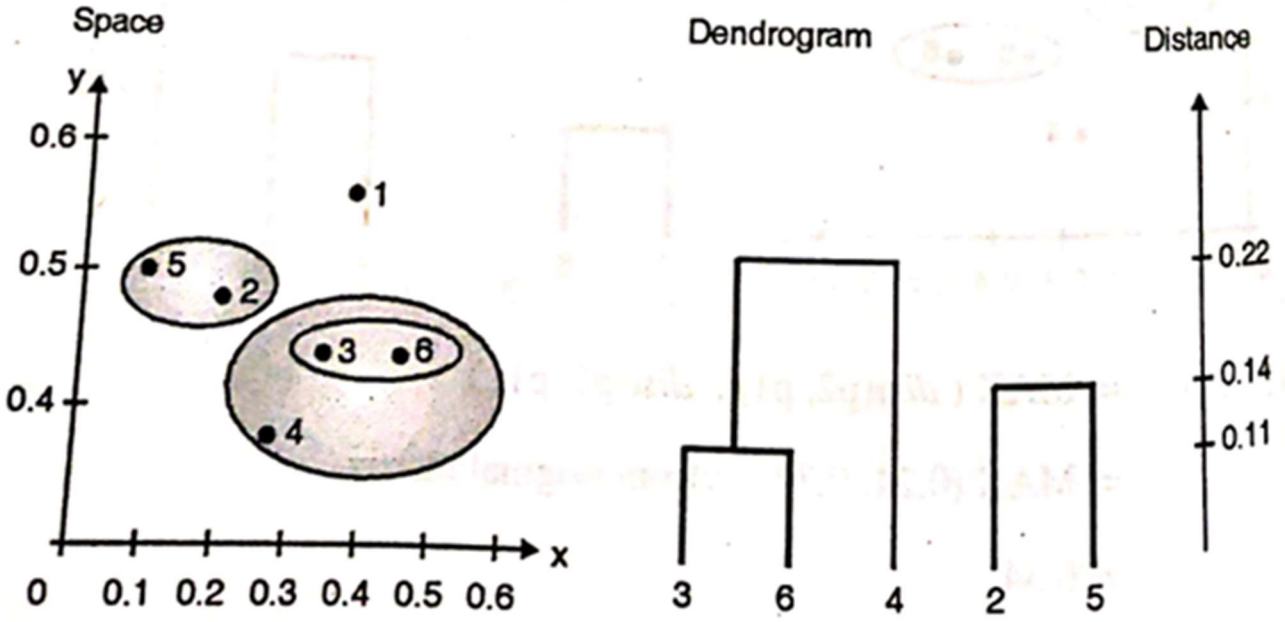
Therefore new distance matrix is :

p1	0			
(p2, p5)	**0.34**	0		
(p3, p6)	0.23	**0.39**	0	
p4	0.37	**0.29**	0.22	0
	p1	**(p2, p5)**	(p3, p6)	p4

Step 5 : Consider the following matrix

p1	0			
(p2, p5)	**0.34**	0		
(p3, p6)	0.23	**0.39**	0	
p4	0.37	**0.29**	0.22	0
	p1	**(p2, p5)**	(p3, p6)	p4

The minimum distance is 0.22, so merge (p3, p6) and p4



$dist(\text{(p3, p6, p4)}, p1) = \text{MAX}(dist(p3, p1), dist(p6, p1), dist(p4, p1))$

$= \text{MAX}(0.22, 0.23, 0.37)$ //from original matrix

$= 0.37$

$dist(\text{(p3, p6, p4)}, (p2, p5)) = \text{MAX}(dist(p3, p2), dist(p3, p5), dist(p6, p2), dist(p6, p5), dist(p4, p2), dist(p4, p5))$

$= \text{MAX}(0.15, 0.28, 0.25, 0.39, 0.20, 0.29)$

$= 0.39$

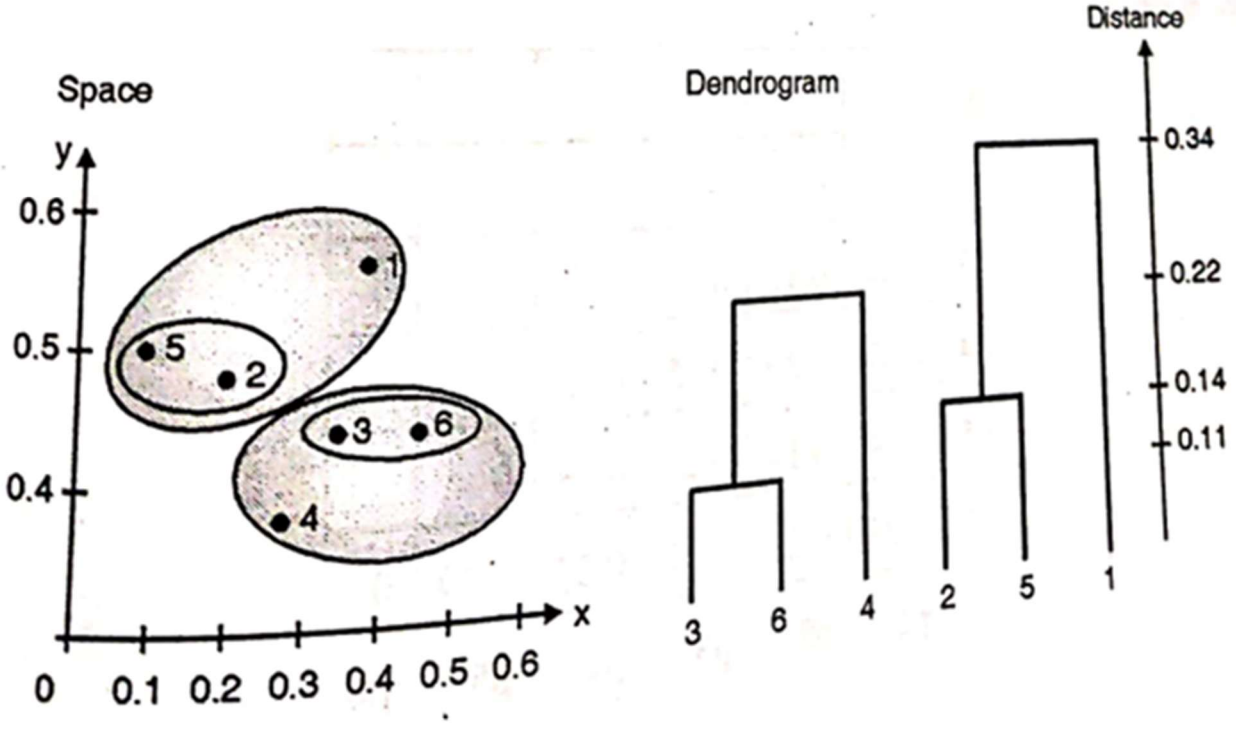
Therefore new distance matrix

p1	0		
(p2, p5)	**0.34**	0	
(p3, p6, p4)	0.37	**0.39**	0
	p1	**(p2, p5)**	(p3, p6, P4)

Step 6 : Now consider the following distance matrix

p1	0		
(p2, p5)	**0.34**	0	
(p3, p6, p4)	0.37	**0.39**	0
	p1	**(p2, p5)**	(p3, p6, P4)

Since the minimum distance is 0.34, merge (p2, p5) with p1.

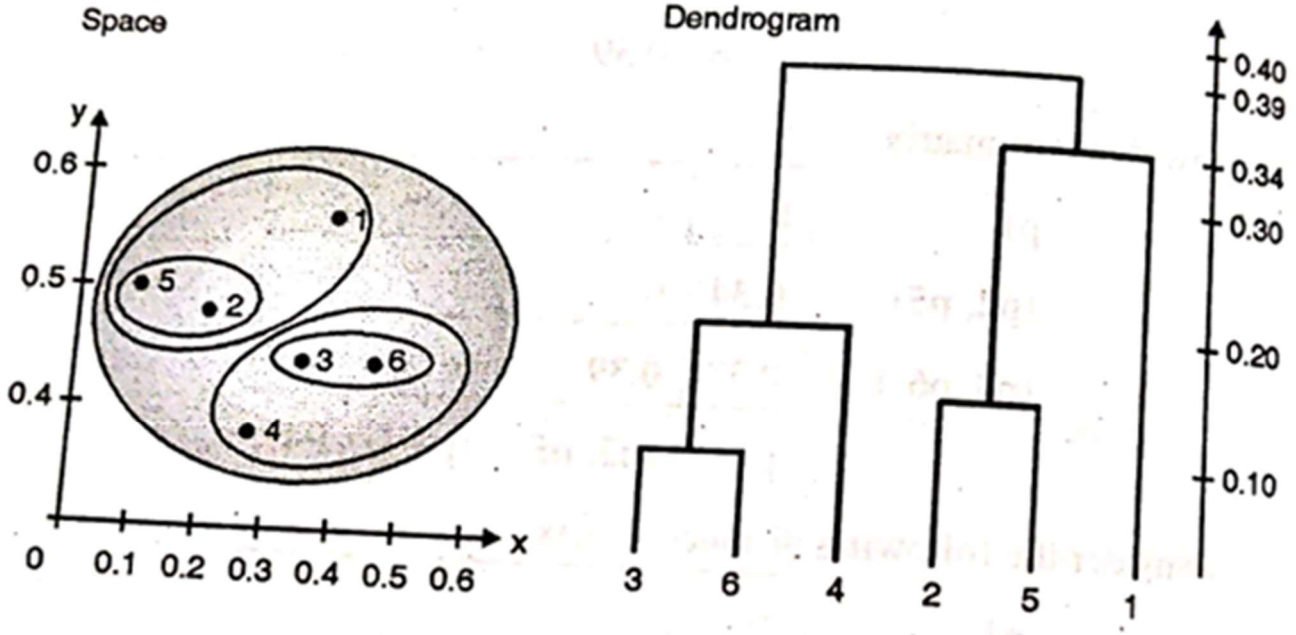


dist ((p2, p5, p1), (p3, p6, p4) = 0.39

Therefore new distance matrix

(p2, p5,p1)	0	
(p3, p6, p4)	0.39	0
	(p2, p5,p1)	(p3, p6, P4)

Finally, merge the cluster (p2, p5,p1) and (p3,p6,p4)



5. Mining frequent patterns and associations

1. Explain Multilevel association rules mining and Multidimensional association rules mining with examples.

Multilevel Association Rules refer to rules that are mined across different levels of a data hierarchy. This allows for associations to be discovered at different levels of abstraction, which can provide both generalized and more specific insights into the relationships between items.

Types of Multilevel Associations:

1. **Uniform Support**:
   - The same minimum support is used across all levels of the hierarchy.

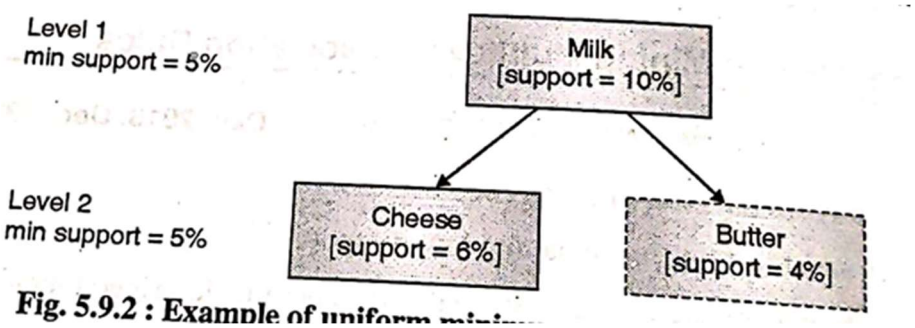


Fig. 5.9.2 : Example of uniform mi...

2. **Reduced Support**:
   - A lower minimum support is applied as you go down the hierarchy

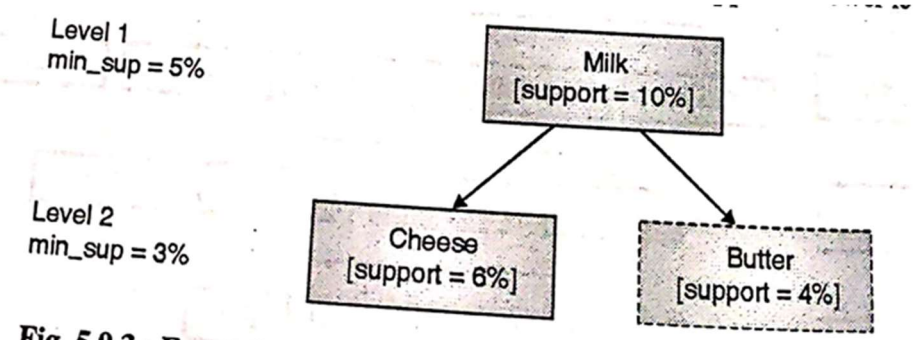


Fig. 5.9.3 : Example of ...

Examples of Multilevel Association Rules:

1. **High-Level Rule (General)**:
   - **"If a customer buys Electronics, they are likely to buy Clothing."**
   - This is a general rule that applies to broad categories.
2. **Low-Level Rule (Specific)**:
   - **"If a customer buys Mobile Phones, they are likely to buy Shirts."**
   - This is a more specific rule that links particular subcategories (Mobile Phones and Shirts).

Multidimensional Association Rules are an extension of traditional association rules that involve multiple dimensions or attributes in the data. While traditional association rules focus on relationships between items in a single dimension (e.g., products bought together in transactions), multidimensional association rules capture patterns across multiple attributes or dimensions, such as time, location, customer demographics, or product categories.

Types:

1. **Intra-dimensional**: Involves multiple attributes of the same dimension (e.g., items bought from different categories).
2. **Inter-dimensional**: Involves attributes from different dimensions (e.g., customer demographics and transaction details).

Examples:

- A traditional association rule might be:
  - "If a customer buys bread, they are likely to buy butter."
- A **multidimensional association rule** might be:
  - "If a customer aged 25-35 buys bread in the evening from store X, they are likely to buy butter."

Weather Condition + Sales:

- **Rule**: If the **weather** is **rainy** and the **day** is a **weekday**, customers are likely to buy **umbrellas**.
- **Dimensions**:
  - Weather Condition (Rainy)
  - Day Type (Weekday)
  - Product (Umbrella)

2. <u>Write a short note on: FP tree.</u>

Definition of FP-tree

An FP-tree (Frequent Pattern Tree) is a tree structure consisting of:

One root labelled as "null".

A set of item-prefix sub-trees, where each node contains: item-name, count, and node-link.

A frequent-item header table with two fields: item-name and head of node-link.

It stores complete information for frequent pattern mining.

The size of the FP-tree is limited by the database size but is usually much smaller due to frequent item sharing.

Frequent items are placed closer to the root for better compression and sharing.

The FP-tree contains all necessary information for mining frequent patterns.

Advantages of FP-tree

The size of the FP-tree is ≤ the candidate sets generated in association rule mining.

Efficiency is achieved through:

1. Compression of a large database into a compact tree structure.
2. Frequent pattern growth using a divide-and-conquer strategy.
3. Preserving the original information of the database in the FP-tree.

FP-tree Usage

The database is compressed into an FP-tree for mining.

Each database subset is mined separately using the tree structure.

3. Explain market basket analysis with an example.

Market Basket Analysis (MBA) is a technique used in data mining to understand the purchase behaviour of customers by discovering associations between different products that are frequently bought together. This analysis is widely used in retail to help businesses optimize product placement, cross-sell products, and improve sales strategies.

Key Concepts in Market Basket Analysis:

1. **Itemset**: A set of items that appear together in a transaction.
2. **Association Rule**: A rule that suggests that if a customer buys one item, they are likely to buy another. The rule has the form **{A} → {B}**, which means if a customer buys item A, they are likely to also buy item B.
3. **Support**: The proportion of transactions in the dataset that contain a particular itemset. For example, if 10 out of 100 transactions contain {A, B}, the support for {A, B} is 10%.
4. **Confidence**: The likelihood that item B is purchased when item A is purchased. For example, if 8 out of 10 customers who bought item A also bought item B, the confidence of {A} → {B} is 80%.
5. **Lift**: The ratio of observed support to expected support, showing the strength of an association. Lift > 1 indicates a strong association.

Example of Market Basket Analysis:

Consider a retail store with the following transaction dataset:

- **T1**: {Milk, Bread, Butter}
- **T2**: {Milk, Bread}

- **T3**: {Milk, Butter}
- **T4**: {Bread, Butter}
- **T5**: {Milk, Bread, Butter}

Step 1: Identify Itemsets

We identify item sets of interest, like **{Milk, Bread}**, **{Milk, Butter}**, etc.

Step 2: Calculate Support

- **Support({Milk, Bread})** = 3/5 = 0.60 (60% of the transactions contain both Milk and Bread).
- **Support({Milk, Butter})** = 3/5 = 0.60.

Step 3: Calculate Confidence

- **Confidence({Milk} → {Bread})** = 3/4 = 0.75 (75% of customers who bought Milk also bought Bread).
- **Confidence({Bread} → {Milk})** = 3/4 = 0.75.

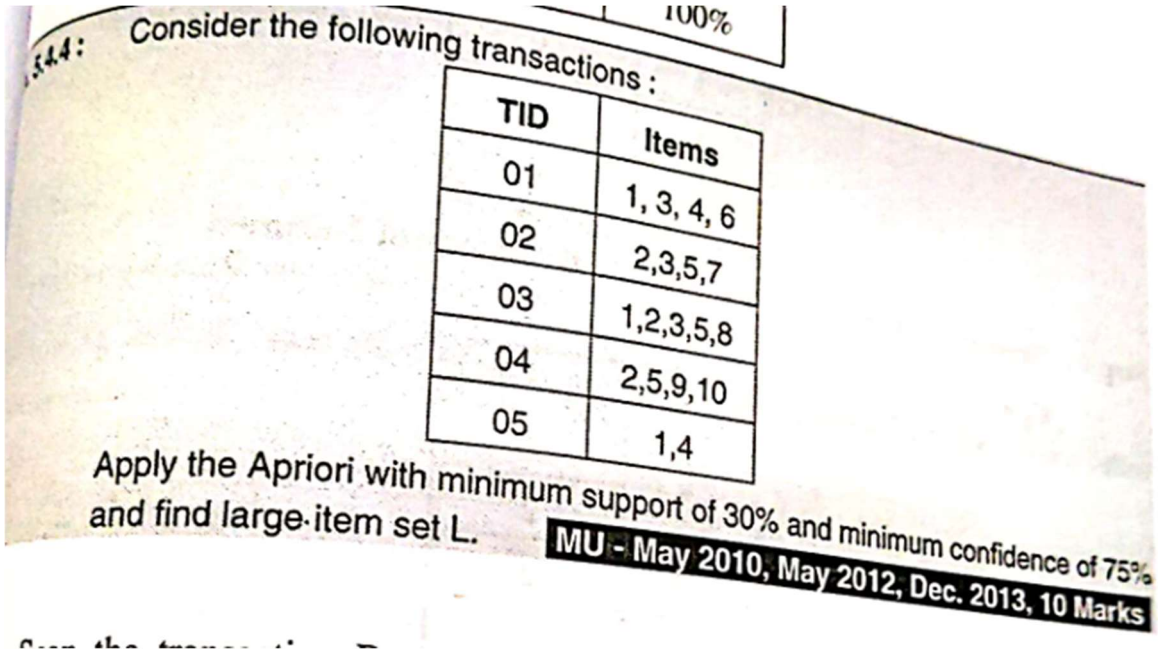
Step 4: Generate Association Rules

- **{Milk} → {Bread}** (75% of customers who bought Milk bought Bread).
- **{Bread} → {Milk}** (75% of customers who bought Bread bought Milk).

Step 5: Evaluate the Lift

- **Lift({Milk} → {Bread})** = Support({Milk, Bread}) / (Support({Milk}) × Support({Bread})) = 0.60 / (0.80 × 0.80) = 0.60 / 0.64 = 0.9375.

4. Apply Apriori algorithm to find frequent pattern set and strong association rules: #



100%

5.44: Consider the following transactions :

TID	Items
01	1, 3, 4, 6
02	2,3,5,7
03	1,2,3,5,8
04	2,5,9,10
05	1,4

Apply the Apriori with minimum support of 30% and minimum confidence of 75% and find large item set L. MU - May 2010, May 2012, Dec. 2013, 10 Marks

Step 1: Scan the transaction database D and find the count for item-1 set which is the candidate. The candidate list is {1,2,3,4,5,6,7,8,9,10}, find the support.

C_1 =

Itemset	Sup-count
1	3
2	3
3	3
4	2
5	3
6	1
7	1
8	1
9	1
10	1

Step 2: Find out wheter each candidate item is present in atleast 30% of transactions. (As support count given is 30%)

L_1 =

Itemset	Sup-count
1	3
2	3
3	3
4	2
5	3

Step 3: Generate C_2 from L_1 and find the support of 2-itemsets.

C_2 =

Itemset	Sup-count
1,2	1
1,3	2
1,4	2
1,5	1
2,3	2
2,4	0
2,5	3
3,4	1
3,5	2

Step 4: Compare candidate C_2 generated in step 3 with the support count, and prune the item sets which do not satisfy the minimum support count.

L_2 =

Itemset	Sup-count
1,3	2
1,4	2
2,3	2
2,5	3

Step 5: Generate candidate C_3 from L_2 and find the support.

C_3 =

Itemset	Sup-count
1,2,3	1
2,3,5	2
1,3,4	1

Step 6: Compare candidate C_3 support count with minimum support count.

L_3=

Itemset	Sup-count
2,3,5	2

Therefore, the database contains the frequent itemset{2,3,5}.

Following are the association rules that can be generated from L_3 as shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
2^3=>5	2	2/2=1	100%
3^5=>2	2	2/2=1	100%
2^5=>3	2	2/3=0.66	66%
2=>3^5	2	2/3=0.66	66%
3=>2^5	2	2/3=0.66	66%
5=>2^3	2	2/3=0.66	66%

Given minimum confidence threshold is 75% so only the first and second rules above are output, since these are the only ones generated that are strong.

Final rules are: Rule 1: 2^3=>5

and Rule 2: 3^5=>2

5. Create FP Tree to find frequent pattern sets:

FP-Growth Algorithm - Example

Minimum Support = 2

FP-Tree

TID	Items
1	A B C E F O
2	A C G
3	E I
4	A C D E G
5	A C E G L
6	E J
7	A B C E F P
8	A C D
9	A C E G M
10	A C E G N

Step:1
Freq. 1-Itemsets.
Min_Sup ≥ 2

Order
A : 8
C : 8
E : 8
G : 5
B : 2
D : 2
F : 2

✓

Remaining all O,I,J,L,P,M & N is with min_sup = 1 ✗

Step:2
Transactions with items sorted based on frequencies, and ignoring the infrequent items.

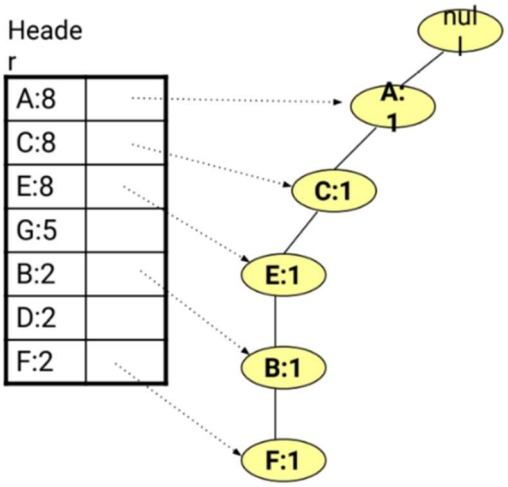
A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G

Building the FP-Tree

- ✔ Scan data to determine the support count of each item.
- ✔ Infrequent items are discarded, while the frequent items are sorted in decreasing support counts.
- ✔ Make a second pass over the data to construct the FP-tree.
- ✔ As the transactions are read, before being processed, their items are sorted according to the above order.

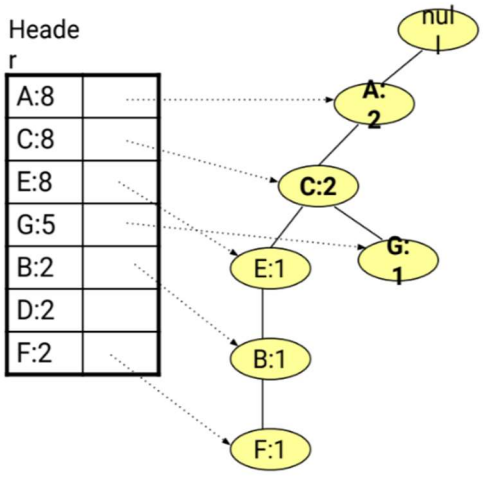
FP-Tree after reading 1st transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



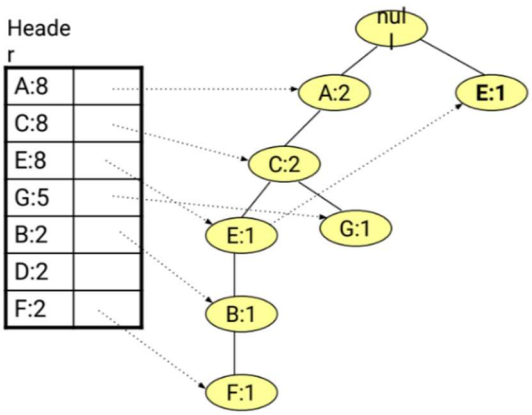
FP-Tree after reading 2nd transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



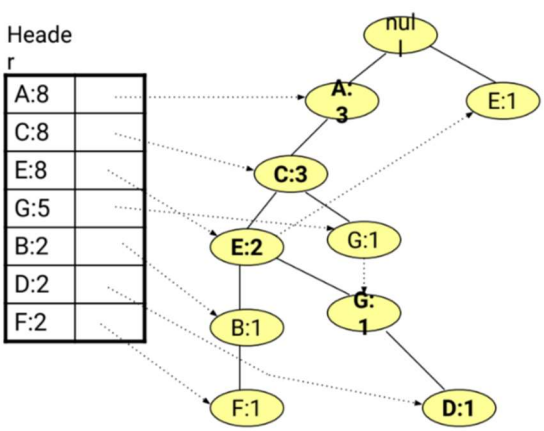
FP-Tree after reading 3rd transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



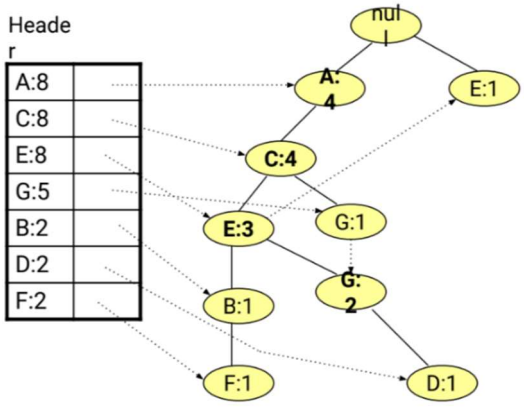
FP-Tree after reading 4th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



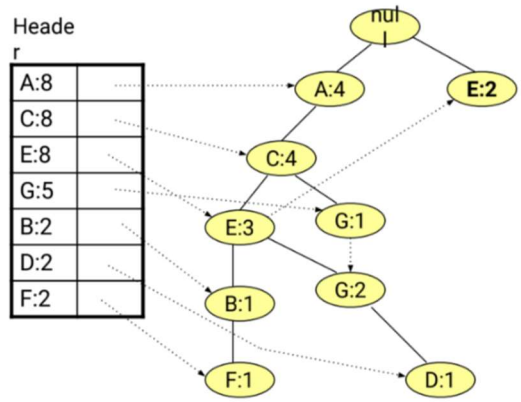
FP-Tree after reading 5th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



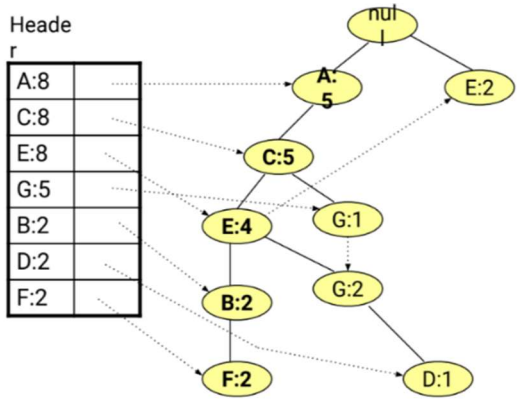
FP-Tree after reading 6th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



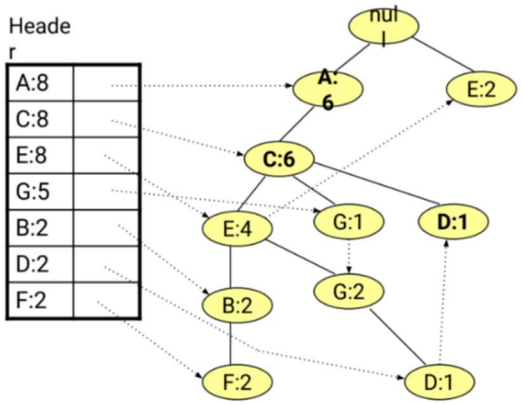
FP-Tree after reading 7th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



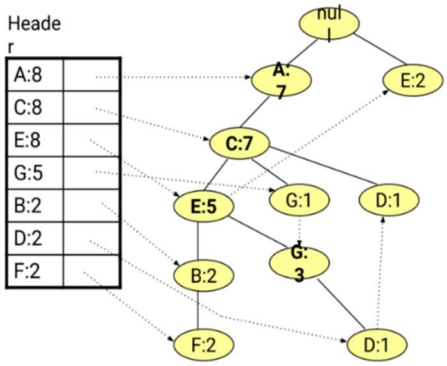
FP-Tree after reading 8th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



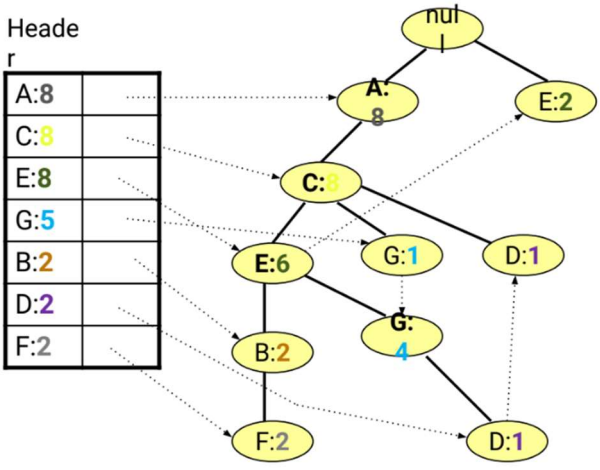
FP-Tree after reading 9th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



FP-Tree after reading 10th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



6. Web mining

1. Explain page rank algorithm with example.

The PageRank technique was designed to both increase the effectiveness of search engines and improve their efficiency.

PageRank is used to measure the importance of a page and to prioritize pages returned from a traditional search engine using keyword searching.

The effectiveness of this measure has been demonstrated by the success of Google.

The PageRank value for a page is calculated based on the number of pages that point to it. This is actually a measure based on the number of backlinks to a page.

A backlink is a link pointing to a page rather than pointing out from a page.

The measure is not simply a count of the number of backlinks because a weighting is used to provide more importance to backlinks coming from important pages.

Given a page p, we use B_p to be the set of pages that point to p, and F_p to be the set of links out of p. The PageRank of a page p is defined as

$$PR(p) = c \sum_{q \in B_p} \frac{PR(q)}{N_q}$$

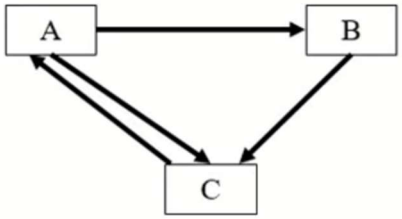
Example: Find the page rank of the following graph with damping factor =0.85

Iteration 1:

- Initially Page Rank (PR) for all the web pages = 1

PR(A) = (1-d) + d(PR(Ti)/C(Ti) +…+PR(Tn)/C(Tn))

PR(A) = (1-d) + d [PR(C) / C(C)]
=(1-0.85) + 0.85 [1/1]
= 0.15 + 0.85[1]
= 0.15 + 0.85
= 1

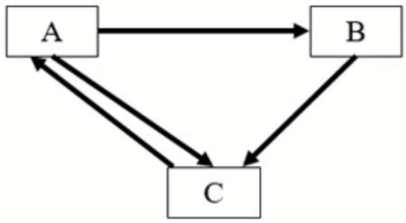


PR(B) = (1-d) + d [PR(A) / C(A)]
= (1-0.85) + 0.85 [(1) / 2]
= 0.15 + 0.85 [0.5]
= 0.15 + 0.425
= 0.575

$$PR(C) = (1-d) + d\,[\,PR(A)/C(A) + PR(B)/C(B)\,]$$
$$=(1-0.85) + 0.85\,[\,(1/2) + (0.575/1)]$$
$$= 0.15 + 0.85[\,0.5 + 0.575]$$
$$= 0.15 + 0.85\,[\,1.075]$$
$$= 0.15 + 0.91375$$
$$= 1.06375$$

Iteration 2:

$$PR(A) = (1-d) + d\,[\,PR(C)/C(C)\,]$$
$$=(1-0.85) + 0.85\,[\,1.06375/1]$$
$$= 0.15 + 0.85[\,1.06375]$$
$$= 0.15 + 0.9041875$$
$$= 1.0541875$$



$$PR(B) = (1-d) + d\,[\,PR(A)/C(A)]$$
$$= (1-0.85) + 0.85\,[\,(1.0541875)/2]$$
$$= 0.15 + 0.85\,[0.52709375]$$
$$= 0.15 + 0.4480296875$$
$$= 0.5980296875$$

$$PR(C) = (1-d) + d\,[\,PR(A)/C(A) + PR(B)/C(B)\,]$$
$$=(1-0.85) + 0.85\,[\,(1.0541875/2) + (0.5980296875/1)]$$
$$= 0.15 + 0.85[0.52709375 + 0.5980296875]$$
$$= 0.15 + 0.85\,[\,1.125123438]$$
$$= 0.15 + 0.9563549219$$
$$= 1.06354922$$

Iteration	A	B	C
0	1	1	1
1	1	0.575	1.06375
2	1.0541875	0.5980296875	1.06354922

2. **Explain web structure mining? List the approaches used to structure the web pages to improve on the effectiveness of search engines and crawlers.**

Web Structure Mining is a branch of web mining that focuses on analysing the hyperlink structure of the web to discover patterns, relationships, and hierarchies among webpages. It uses graph theory to model and interpret the interconnected web pages, treating each webpage as a node and each hyperlink as an edge.

Goals

- Improve search engine rankings (e.g., PageRank, HITS).
- Discover clusters or communities of related pages.
- Optimize web crawling and indexing processes.

Web structure mining techniques like PageRank and Clever (HITS) focus on analysing the link structure of webpages to determine their importance and relevance. Here's how they work:

1. PageRank Algorithm

Developed by Google founders Larry Page and Sergey Brin, PageRank measures the importance of a webpage based on the number and quality of incoming links.

PageRank enhances search engine effectiveness and efficiency by measuring page importance to prioritize search results. It calculates a page's value based on the number and quality of backlinks, with higher weight given to links from important pages. This approach, exemplified by Google's success, emphasizes backlinks as a key factor in determining relevance.

2. HITS (Hyperlink-Induced Topic Search) or Clever Algorithm

The **HITS algorithm** is a web mining technique developed by Jon Kleinberg to analyse web pages based on their link structure. It identifies two types of pages:

1. **Hubs**: Pages that link to many authoritative pages.
2. **Authorities**: Pages that are heavily linked by other pages.

How It Works:

1. A root set of pages is retrieved based on a user query.
2. These pages are expanded to include those they link to or are linked by.
3. Each page gets two scores: a hub score and an authority score.
4. Scores are updated iteratively:
    - Hub scores increase if they link to strong authorities.
    - Authority scores increase if linked by strong hubs.

3. **Write a note on web usage mining. Also state any two of its applications.**

Web Usage Mining is the process of analysing user interactions and behaviour on websites by extracting useful patterns from web log data. It involves studying data generated by user activity, such as clickstreams, navigation paths, and session information, to gain insights into user preferences and improve web applications.

Steps in Web Usage Mining

1. **Data Collection**:
   Gather raw data from web server logs, browser cookies, and application logs. This data records user actions like page views, timestamps, and clickstreams.
2. **Data Preprocessing**:
   Clean and prepare data by removing irrelevant entries (e.g., failed requests), identifying users (using IP or cookies), segmenting sessions, and completing missing paths in navigation sequences.
3. **Pattern Discovery**:
   Extract meaningful insights using techniques like clustering (grouping similar users), association rule mining (finding relationships between items), and sequential pattern mining (analysing navigation paths).
4. **Pattern Analysis**:
   Interpret and evaluate patterns to improve user experience, enhance navigation, and tailor marketing strategies.

Applications of Web Usage Mining

a) **Personalization**:
   Customizes content based on user preferences, such as recommending products or articles.

b) **Website Optimization**:
   Improves site navigation by analysing user paths and identifying bottlenecks.

4. **Write a short note on: Web content mining.**

Web Content Mining involves extracting useful information from the content of web pages, such as text, images, videos, and structured data like tables. It focuses on analysing the information within webpages to understand their relevance, structure, and meaning.

Techniques used in web content mining include:

- **Text Mining**: Extracting insights from textual content using techniques like natural language processing (NLP) and sentiment analysis.
- **Multimedia Mining**: Analysing images, videos, and audio for patterns or classifications.

- **Structured Data Mining**: Extracting information from structured formats like tables or metadata.

Applications:

- Enhancing search engine performance by improving the indexing of web content.
- Summarizing webpage content for recommendation systems.
- Detecting trends and topics in online articles and blogs.

5. **Explain CLARANS extension in web mining.**

CLARANS (Clustering Large Applications based on Randomized Search) is an extension of the k-medoids clustering algorithm designed to handle large datasets, like those encountered in web mining, by combining clustering with randomized search for efficiency. It addresses the challenges of scalability and computational complexity when dealing with large and high-dimensional web data.

Key Features:

1. **Randomized Search**: Reduces computational cost by randomly selecting medoid candidates rather than performing exhaustive searches.
2. **Medoid-based Clustering**: Uses medoids (central points) instead of centroids, making it more robust to outliers.
3. **Efficiency**: Faster and more scalable for large web data like user sessions or browsing patterns.

Applications:

- **User Behaviour Analysis**: Clusters users based on browsing patterns.
- **Content Recommendation**: Groups similar content to improve recommendations.
- **Website Optimization**: Identifies frequent page clusters to enhance site structure.