Detailed Answers to Selected Data Mining Questions

1 What are the different types of data warehouses? Explain their uses.

A data warehouse is a centralized repository that allows for storage, retrieval, and analysis of large volumes of data. It is used for decision-making, reporting, and data analysis.

Types of Data Warehouses

1. **Enterprise Data Warehouse (EDW)**
   A centralized warehouse that provides decision support services across the enterprise.

   Use Cases: Used by large organizations to analyze data across various business units.

   Example: An EDW for a retail chain might integrate sales, customer, inventory, and supplier data.

   Advantages:

   - Holistic view of the organization
   - Supports cross-functional analysis

2. **Operational Data Store (ODS)**
   A store that holds data in near real-time for short-term operational decisions.

   Use Cases: Used for daily operations like viewing up-to-date customer information.

   Example: Banks use ODS to fetch current balances and transaction records.

   Advantages:

   - Real-time data updates
   - Lower latency for operational queries

3. **Data Mart**
   A smaller, subject-oriented version of a data warehouse focused on a specific department or function.

   Use Cases: Department-level analysis like sales trends in a particular region.

   Example: A sales data mart could include monthly sales targets, actual sales, and sales rep performance.

   Advantages:

- Faster access
- Easier to manage and query

2 Why do you need a data warehouse?

Data warehouses are essential for the following reasons:

- **Integrated Data Analysis:** Combines data from multiple sources such as ERP and CRM systems.
- **Data Consistency:** Cleansed and standardized data ensures accuracy and uniformity.
- **Historical Intelligence:** Stores large volumes of historical data for trend and pattern analysis.
- **Improved Decision-Making:** Supports OLAP operations and Business Intelligence tools.
- **Performance Optimization:** Offloads complex analytical queries from transactional systems.

3 What is warehousing?

Warehousing refers to the process of collecting, cleaning, transforming, and storing data in a central repository known as a data warehouse.

Key Aspects of Warehousing

- Extracting data from heterogeneous sources.
- Transforming and cleaning data to ensure consistency and quality.
- Loading the cleaned data into the warehouse (ETL process).
- Making the data available for querying and analysis.

Purpose: To support business intelligence, reporting, and decision support systems.

4 Define Knowledge Discovery in Databases (KDD). What are the steps involved in the KDD workflow?

Knowledge Discovery in Databases (KDD) is the process of discovering useful patterns and knowledge from large datasets. Data mining is a key step within the broader KDD process.

Steps in the KDD Process

1. **Data Selection:** Identify and retrieve relevant data from various sources.

   Example: Selecting transaction data from a sales database.

2. **Data Preprocessing (Cleaning):** Remove noise, handle missing values, and deal with inconsistencies.

   Techniques include imputation, deduplication, and outlier handling.

3. **Data Transformation:** Normalize or transform data into suitable formats for mining.

   Example: Converting categorical data into numerical codes.

4. **Data Mining:** Apply algorithms such as classification, clustering, or association to extract patterns.

   Example: Apriori algorithm to find frequent itemsets.

5. **Pattern Evaluation:** Evaluate the discovered patterns using measures such as support, confidence, lift, and accuracy.

6. **Knowledge Representation:** Present findings using visualizations, reports, or dashboards.

   Example: Decision trees, heatmaps, graphs.

5 What is association analysis? How can it be used in data mining or recommendation systems?

Association analysis is a rule-based method for discovering interesting relationships or patterns among items in large datasets.

Key Concepts

- **Itemset:** A collection of one or more items.
- **Support:** The frequency of occurrence of an itemset.
- **Confidence:** The likelihood of the consequent given the antecedent.
- **Lift:** The strength of a rule over random chance.

Example Rule

{Bread, Butter} $\rightarrow$ {Milk} with 80% confidence and 40% support.

This means 40% of all transactions include Bread, Butter, and Milk; and 80% of those who buy Bread and Butter also buy Milk.

Algorithms Used

- **Apriori:** Iteratively finds frequent itemsets by pruning infrequent ones.
- **FP-Growth:** Builds a prefix tree (FP-tree) to find frequent patterns without candidate generation.

Applications in Recommendation Systems

- **E-commerce:** Recommend products frequently bought together (e.g., Amazon).
- **Streaming Services:** Suggest content based on user viewing patterns (e.g., Netflix).
- **Retail:** Optimize store layout by placing commonly purchased items close together.

Answers to Questions 6–10

Your Name

6. Define Data Scaling and Normalization with Examples

Data scaling and normalization are preprocessing techniques used to make data consistent and suitable for algorithms that rely on distance or assume a specific distribution.

1. Normalization (Min-Max Scaling)

Formula:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Range: $[0, 1]$
Use Case: Used in algorithms like k-NN and neural networks.
Example: Let raw values be 45, 55, 65.
Then,

$$X_{\text{norm}} = \frac{55 - 45}{65 - 45} = \frac{10}{20} = 0.5$$

2. Standardization (Z-score Scaling)

Formula:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

Range: Data is centered at 0 with standard deviation 1.
Use Case: Suitable for SVM, logistic regression, etc.
Example: Let mean (μ) = 50 and std. deviation (σ) = 10. Then,

$$X_{\text{scaled}} = \frac{60 - 50}{10} = 1$$

7. Write and Explain the k-NN Algorithm (with Steps and Example)

k-Nearest Neighbors (k-NN) is a simple, non-parametric, instance-based learning algorithm.

Steps:

1. Choose the number of neighbors k.
2. Compute distance between test sample and all training samples.
3. Sort and select k nearest neighbors.
4. Assign the most frequent class among the k neighbors.

Example:

To classify a new email:

- Compute distance between new email and all labeled emails.
- If $k = 3$, choose the 3 closest ones.
- Assign the class (spam or not spam) based on majority vote.

Pros:

- Easy to understand and implement.
- No training phase required.

Cons:

- Slow for large datasets.
- Sensitive to irrelevant or unscaled features.

8. Explain How Classification Can Be Used for Data Mining with Examples

Classification is a supervised learning method used in data mining to categorize data into predefined classes.

Process:

1. **Training:** A model is built using labeled data.
2. **Prediction:** Model is used to classify new data.

Popular Classification Algorithms:

- Decision Trees
- k-NN
- Naive Bayes
- SVM
- Random Forest

Examples:

- Email classification (spam or not spam)
- Credit approval (approve or reject)
- Medical diagnosis (type of disease)
- Sentiment analysis (positive, negative, neutral)

9. List and Explain the Uses of Different Types of Data Visualization Techniques

Technique	Purpose	Use Case
Bar Chart	Compare quantities across categories	Sales by region
Line Chart	Show trends over time	Website traffic analysis
Pie Chart	Show proportions	Market share distribution
Histogram	Show frequency distribution	Student score distribution
Box Plot	Detect outliers and show data spread	Salary comparison
Scatter Plot	Show correlation between variables	Age vs income
Heatmap	Use color to visualize data matrices	Correlation matrix
TreeMap	Hierarchical proportional representation	Disk space usage
Dendrogram	Show hierarchical clustering	Gene expression clusters

10. Write and Explain the ID3 Algorithm. Discuss Pros and Cons of Decision Trees

ID3 (Iterative Dichotomiser 3) is a decision tree algorithm developed by Ross Quinlan that builds a tree by choosing attributes with the highest **Information Gain**.

Steps of ID3:

1. Compute the entropy of the dataset.
2. For each attribute, calculate the information gain.
3. Select the attribute with the highest gain as a decision node.
4. Recursively repeat for each branch.

Entropy Formula:

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

where p_i is the probability of class i.

Advantages:

- Simple to understand and interpret.
- Can handle both categorical and numerical data.
- No need for scaling or normalization.

Disadvantages:

- Overfitting is common if tree is too deep.
- Biased toward attributes with many levels.
- Cannot handle continuous attributes directly (needs preprocessing).

Answers to Questions 11–15

Your Name

11. What is a Decision Tree? How to Build a Classification Model Using Decision Trees?

A **Decision Tree** is a supervised learning model used for classification and regression. It splits data into branches based on feature values. The structure includes:

- Internal nodes representing attribute tests
- Branches representing outcomes
- Leaves representing class labels

Steps to Build a Decision Tree:

1. Select the best attribute using **Information Gain** or **Gini Index**.
2. Split the dataset based on the chosen attribute.
3. Recursively repeat the process on each subset.
4. Stop when:
   - All samples in a node belong to one class, or
   - No more attributes remain.

Example:

Predicting car purchase:

- `Age < 30?` → Yes: `Income > $50K?`, No: `Owns House?`

Common Algorithms:

- ID3 (Information Gain)
- C4.5 (Gain Ratio, handles continuous attributes)
- CART (Gini Index)

12. What is Data Preprocessing? List Useful Preprocessing Techniques for Numerical Data.

Data Preprocessing is the step of transforming raw data into a usable format by cleaning and structuring it.

Techniques for Numerical Data:

1. **Handling Missing Values:**
    - Mean/Median/Mode Imputation
    - Interpolation
    - Row Deletion (if negligible)
2. **Normalization/Scaling:**
    - Min-Max Scaling
    - Z-score Standardization
3. **Outlier Detection and Removal:**
    - Z-score Method
    - IQR Method
    - LOF (Local Outlier Factor)
4. **Discretization:**
    - Binning numeric data into categories
5. **Transformation:**
    - Log, square root, or power transformation
6. **Noise Removal:**
    - Smoothing techniques like moving average

13. List the Differences Between OLAP and OLTP

Feature	**OLTP (Online Transaction Processing)**	**OLAP (Online Analytical Processing)**
Purpose	Transaction processing	Analytical querying
Data Source	Operational data	Historical data from Data Warehouse

Queries	Simple, read-write	Complex, read-only
Normalization	Highly normalized	Denormalized (star/snowflake schema)
Speed	Fast for short transactions	Fast for large queries
Data Volume	Small, real-time updates	Large, periodic updates
Example	ATM withdrawal, sales entry	Sales trend report, customer segmentation

14. What are the Limitations of OLTP and How Does OLAP Solve Them?

Limitations of OLTP:

- Not optimized for complex analytical queries
- High normalization leads to slow joins
- No historical or trend data
- Query performance degrades for large reports

How OLAP Solves Them:

- Designed for fast reporting and analysis
- Supports multidimensional analysis (slicing, dicing, pivoting)
- Stores historical and aggregated data
- Uses denormalized schema for better performance

15. Explain Clustering with DBSCAN as an Example

Clustering is an unsupervised learning technique that groups similar data into clusters.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

- Groups data points based on density.
- Handles arbitrary-shaped clusters and outliers.

Key Parameters:

- ε: Radius of neighborhood
- MinPts: Minimum number of points to form a dense region

Steps:

1. For each point, count neighbors within ε
2. If neighbors $\geq$ MinPts: mark as **core point**
3. Points within ε of core: **density-reachable**
4. Points not reachable from any cluster: **noise**

Example:

In telecom fraud detection:

- Legitimate users form tight clusters (similar call patterns)
- Fraudsters are isolated (marked as noise)

Advantages:

- Detects arbitrarily shaped clusters
- Identifies noise/outliers
- No need to specify number of clusters

Disadvantages:

- Sensitive to choice of ε and MinPts
- Struggles with clusters of varying densities

Answers to Questions 16–20

16. Explain clustering and how it can detect outliers or unusual patterns in data (e.g., telecom).

Clustering groups similar data points based on patterns or characteristics.

Outlier Detection via Clustering

Outliers are data points that **do not belong to any cluster** or are far from their assigned cluster center.

Example in Telecom

- **Normal users**: Call during business hours, regular intervals → form dense clusters.
- **Fraudulent users**: Unusual patterns (e.g., calling multiple countries at odd hours) → remain isolated or in sparse clusters.

Techniques Used

- **K-Means**: Points far from cluster centroids can be outliers.
- **DBSCAN**: Points marked as "noise" are considered outliers.
- **Hierarchical Clustering**: Outliers may appear as single-member branches.

Applications

- Telecom fraud detection
- Network intrusion detection
- Banking anomaly detection

17. Explain anomaly detection and types of outliers.

Anomaly Detection is identifying data points that deviate significantly from normal behavior.

Types of Outliers

1. **Point Outliers**: A single data instance is far from the rest.
   Example: A person aged 120 in a population dataset.
2. **Contextual Outliers**: An instance is normal in one context but not in another.
   Example: 35°C in winter is abnormal, but not in summer.
3. **Collective Outliers**: A group of instances is anomalous together.
   Example: A sudden spike in internet usage at midnight across several accounts.

Detection Techniques

- Statistical methods
- Distance-based (e.g., LOF)
- Clustering (e.g., DBSCAN)
- Machine learning (e.g., Isolation Forest, Autoencoders)

18. What is the FP-Growth algorithm? How does it improve over Apriori?

FP-Growth (Frequent Pattern Growth) is an efficient method for mining frequent itemsets without candidate generation.

Steps in FP-Growth

1. **Build FP-Tree**:
   - Scan database once to find frequent items.
   - Sort items in descending frequency.
   - Construct a compact tree structure (FP-tree).
2. **Pattern Extraction**:
   - Recursively mine the FP-tree using a divide-and-conquer approach.

Advantages over Apriori

Feature	**FP-Growth**	**Apriori**
Candidate Generation	No	Yes
Scans of DB	2	Multiple
Efficiency	Higher	Lower for large datasets
Memory Usage	More compact (tree structure)	High (many candidate sets)

Use Case Example

In a retail system, quickly find frequent purchase combinations without scanning the whole dataset repeatedly.

19. Describe evaluation techniques for clustering models.

Evaluating clustering is difficult since we often don't have true labels. Still, several techniques exist:

1. Internal Evaluation

- **Silhouette Score**:
  $$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$
  where:
  - $a(i)$ = average intra-cluster distance
  - $b(i)$ = average nearest-cluster distance

  Ranges from -1 to 1 (higher is better).
- **Dunn Index**: Ratio of minimal inter-cluster distance to maximal intra-cluster distance.
- **Davies-Bouldin Index**: Lower values indicate better separation.

2. External Evaluation (if labels exist)

- Adjusted Rand Index (ARI)
- Normalized Mutual Information (NMI)
- Fowlkes–Mallows Index

20. How can web-based association-based recommendation systems be built?

Association-based Recommendation Systems suggest items based on frequently co-occurring patterns in user behavior.

Steps

1. **Collect data**: e.g., user-item interactions (clicks, purchases).
2. **Mine frequent itemsets**:
   - Use **Apriori** or **FP-Growth**.
   - Example: “Users who viewed X also viewed Y.”
3. **Generate rules**:
   - Example: $\{\text{Shoes}\} \rightarrow \{\text{Socks}\}$ (if confidence is high).
4. **Deploy system**:
   - Integrate with a website using frameworks (e.g., Flask, Django).
   - Use real-time or batch-based updates.

Example

In an online bookstore:

- If users frequently buy {Data Mining Book, Python Book} together,
- Recommend “Python Book” to someone viewing “Data Mining Book”.

Enhancements

Combine with **collaborative filtering** or **content-based filtering** for hybrid systems.

Answers to Questions 21–25

21. Explain Support Vector Machines (SVM), kernel functions, and how they work.

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression. It finds the optimal hyperplane that separates data points of different classes with the maximum margin.

Key Concepts:

- **Support Vectors:** Data points closest to the hyperplane; critical in defining the margin.
- **Margin:** Distance between the hyperplane and the nearest data point from any class.

Linear SVM: For linearly separable data, SVM finds the straight line (or hyperplane) that separates classes.

Non-linear SVM: When data isn't linearly separable, we use the **Kernel Trick** to map data to a higher-dimensional space.

Common Kernel Functions:

- Linear Kernel: $K(x, y) = x^T y$
- Polynomial Kernel: $K(x, y) = (x^T y + c)^d$
- RBF (Gaussian): $K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

Applications: Text classification, handwriting recognition, bioinformatics (e.g., cancer detection).

22. Describe ensemble techniques and their advantages. Why are ensemble methods better?

Ensemble techniques combine multiple models to produce a stronger, more accurate model.

Types of Ensembles:

- **Bagging (Bootstrap Aggregating):** Train multiple models on random subsets of the data.
  Example: Random Forest.
- **Boosting:** Train models sequentially, where each model focuses on errors of the previous one.
  Example: AdaBoost, Gradient Boosting, XGBoost.
- **Stacking:** Combine predictions of different models using a meta-learner.

Advantages:

- Improved accuracy (reduces bias and variance)
- Robustness (less sensitive to noise and overfitting)
- Better generalization on unseen data

Why better? They average out weaknesses and combine strengths of individual models.

23. Explain outliers, outlier detection, and Local Outlier Factor (LOF) with examples.

Outliers: Data points that deviate significantly from other observations.

Outlier Detection Methods:

- Statistical: Z-score, IQR
- Distance-based: k-NN
- Density-based: **Local Outlier Factor (LOF)**

Local Outlier Factor (LOF): LOF measures the local density of a point compared to its neighbors.

Steps:

1. Compute k-distance (distance to the k-th nearest neighbor)
2. Calculate reachability distance
3. Compute local reachability density (LRD)
4. Compute LOF as the ratio of the average LRD of the k-nearest neighbors to the LRD of the point

Interpretation:

- LOF ≈ 1: Not an outlier
- LOF > 1: Likely an outlier

Example: A remote city in GPS data (far from population clusters) may have high LOF.

24. How do you handle missing data in datasets?

Common Strategies:

- **Deletion:**
  - Listwise Deletion: Remove rows with any missing value.
  - Pairwise Deletion: Use available data for each analysis.
- **Imputation:**
  - Mean/Median/Mode Imputation
  - Regression Imputation
  - k-NN Imputation
  - Multiple Imputation (e.g., MICE)
- **Model-Based Methods:** Random Forest, EM Algorithm
- **Flagging:** Add a binary indicator variable to denote missingness.

Choice of method depends on the missingness type: MCAR, MAR, or MNAR.

25. List different data mining techniques with use cases.

- **Classification:** Assign categories.
  Use case: Spam detection in emails.

- **Clustering:** Group similar instances.
  Use case: Customer segmentation.

- **Association Rule Mining:** Find item co-occurrence patterns.
  Use case: Market basket analysis.

- **Regression:** Predict continuous values.
  Use case: Sales forecasting.

- **Anomaly Detection:** Find rare/unusual patterns.
  Use case: Credit card fraud detection.

- **Sequential Pattern Mining:** Discover frequent sequences.
  Use case: User purchase prediction.

- **Recommendation Systems:** Suggest items to users.
  Use case: Netflix or Amazon product recommendations.

26. Describe the Confusion Matrix, accuracy, precision, TPR, TNR, FPR, and FNR.

Confusion Matrix:

A 2×2 table used to evaluate the performance of a classification model:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Metrics:

- **Accuracy:** Overall correctness.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** How many predicted positives are correct?

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **True Positive Rate (TPR) / Recall / Sensitivity:**

$$\text{TPR} = \frac{TP}{TP + FN}$$

- **True Negative Rate (TNR) / Specificity:**

$$\text{TNR} = \frac{TN}{TN + FP}$$

- **False Positive Rate (FPR):**

$$\text{FPR} = \frac{FP}{FP + TN}$$

- **False Negative Rate (FNR):**

$$\text{FNR} = \frac{FN}{FN + TP}$$

Use Cases:

- Medical diagnosis (minimize FNR).
- Spam filters (minimize FPR).

27. How is cluster quality measured?

Key Cluster Quality Metrics:

- **Silhouette Score:**
  - Measures cohesion (intra-cluster) and separation (inter-cluster).
  - Range: -1 to 1 (higher is better).
- **Davies-Bouldin Index:**
  - Average similarity between clusters.
  - Lower values indicate better clustering.
- **Dunn Index:**
  - Ratio of the minimum inter-cluster distance to the maximum intra-cluster distance.
  - Higher is better.
- **Within-Cluster Sum of Squares (WCSS):**
  - Total distance of points from their cluster centroid.
  - Lower WCSS = more compact clusters.
- **Elbow Method (for K-means):**
  - Plot WCSS vs number of clusters.
  - "Elbow" point suggests optimal cluster count.

Application: Used to validate clustering results and choose the best algorithm or number of clusters.

28. What is an outlier? Why are they important? Types of outliers.

Outlier:

An observation that significantly deviates from other data points.

Why Important:

- Indicate data entry errors or fraud.
- Affect statistical summaries and model accuracy.
- May represent rare but important phenomena (e.g., disease, intrusion).

Types of Outliers:

- **Point Outliers:**
  - Single values far from the norm.
  - E.g., A person with 0 income in a high-income dataset.
- **Contextual Outliers:**
  - Values abnormal in specific contexts.
  - E.g., 30°C in winter.
- **Collective Outliers:**
  - Group of points that are anomalous together.
  - E.g., Sudden burst of traffic at odd hours.

29. Describe distance-based outlier detection.

Concept:

Points are considered outliers if they are far from most other points.

Steps:

- Define distance metric (e.g., Euclidean).
- For each point, compute distance to its k-nearest neighbors.
- If average distance is above a threshold → mark as outlier.

Advantages:

- Simple and intuitive.
- Works well for numerical data.

Disadvantages:

- Sensitive to choice of distance metric and threshold.
- Struggles with high-dimensional data.

30. K-means clustering with Manhattan distance and SSE calculation.

K-means Algorithm:

- Initialize k centroids.
- Assign each point to nearest centroid.
- Update centroids (mean of assigned points).
- Repeat until convergence.

Manhattan Distance:

$$D(x, y) = \sum |x_i - y_i|$$

Used instead of Euclidean when absolute differences are preferred.

Sum of Squared Errors (SSE):

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

Where:

- C_i: Cluster i
- μ_i: Centroid of cluster i

Measures compactness of clusters (lower is better).

Example:

Given 2D points and 2 clusters:

- Use Manhattan distance to assign points.
- Recalculate centroids.
- Compute SSE using squared differences from centroids.

31. Fuzzy C-means clustering with example

Fuzzy C-means (FCM):

An extension of K-means where each data point belongs to all clusters with varying degrees of membership.

Key Concepts:

- **Membership values:** Range from 0 to 1 for each data point per cluster.
- **Objective function:**

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \cdot ||x_i - c_j||^2$$

  where:

  - u_{ij} is the degree of membership of x_i in cluster j,
  - c_j is the centroid of cluster j,
  - m is the fuzziness coefficient (typically $m = 2$).

Algorithm Steps:

1. Initialize membership matrix randomly.
2. Calculate cluster centers using weighted averages.
3. Update membership values.
4. Repeat until convergence (small change in membership matrix).

Example:

For data: (1, 2), (2, 1), (5, 6), (6, 5), try 2 clusters.

- Initial random memberships assigned.
- Cluster centers calculated.
- Membership updated iteratively until stable.

Advantages:

- Better for overlapping clusters.
- More flexible than K-means.

Disadvantages:

- Slower convergence.
- Requires setting fuzziness parameter.

32. Hierarchical clustering and dendrogram construction

Hierarchical Clustering:

Builds a hierarchy of clusters using either:

- **Agglomerative (Bottom-up)** – Start with individual points and merge.
- **Divisive (Top-down)** – Start with all points and split.

Steps (Agglomerative):

1. Start with each data point as a separate cluster.
2. Merge closest clusters based on linkage criteria:
   - **Single linkage:** min distance.
   - **Complete linkage:** max distance.
   - **Average linkage:** mean distance.
3. Repeat until one cluster remains.

Dendrogram:

- A tree-like diagram showing the merging steps.
- Y-axis = distance (or dissimilarity).
- Cut the tree at a desired level to form k clusters.

Advantages:

- No need to specify number of clusters beforehand.
- Useful for visual analysis.

Disadvantages:

- Computationally expensive ($O(n^2)$).
- Sensitive to noise and outliers.

33. Summary:

Clustering Algorithm	**Key Feature**	**Distance Used**	**Cluster Assignment**
K-means	Hard clusters	Euclidean or Manhattan	Each point in one cluster
Fuzzy C-means	Soft clusters	Euclidean (typically)	Membership values per cluster
Hierarchical	Dendrogram-based	Various (linkage-based)	Based on tree cuts