

**ML UNIT – 5 (Unsupervised Learning) – END-SEM PYQ Answers**➤ **MAY / JUN 2023****Q5) a) With reference to Clustering explain the issue of “Optimization of Clusters”. [6]**

Optimization of clusters in clustering refers to the process of improving the quality and correctness of the cluster formation so that the results are more accurate, meaningful, and useful for analysis.

1. **It ensures that the clustering algorithm produces the best possible grouping** by keeping similar data points together and separating dissimilar points.
2. **The process involves identifying the optimal number of clusters**, as choosing too many or too few clusters affects clustering performance.
3. **An objective function is used for optimization**, such as minimizing the Sum of Squared Errors (SSE) in K-Means clustering.
4. **Techniques like the Elbow Method, Silhouette Score, and Gap Statistics** help evaluate and select the most suitable cluster configuration.
5. **Optimization helps improve compactness and separation of clusters**, which increases the clarity and strength of patterns in the dataset.
6. **It also enhances stability and consistency**, ensuring the same clustering results even with repeated runs of the algorithm.
7. **Optimization reduces noise and improves cluster interpretation**, especially in real-world applications where datasets may contain outliers.
8. It is important because optimized clustering leads to **better analysis, accurate predictions, and improved decision-making** in areas such as customer segmentation, pattern recognition, and anomaly detection.

**b) Compare Hierarchical clustering and K-means clustering. [6]**

<b>k-means Clustering</b>	<b>Hierarchical Clustering</b>
k-means, using a pre-specified number of clusters, the method assigns records to each cluster to find the mutually exclusive cluster of spherical shape based on distance.	Hierarchical methods can be either divisive or agglomerative.
K Means clustering needed advance knowledge of K i.e. no. of clusters one want to divide your data.	In hierarchical clustering one can stop at any number of clusters, one find appropriate by interpreting the dendrogram.

One can use median or mean as a cluster centre to represent each cluster.	Agglomerative methods begin with 'n' clusters and sequentially combine similar clusters until only one cluster is obtained.
Methods used are normally less computationally intensive and are suited with very large datasets.	Divisive methods work in the opposite direction, beginning with one cluster that includes all the records and Hierarchical methods are especially useful when the target is to arrange the clusters into a natural hierarchy.
In K Means clustering, since one start with random choice of clusters, the results produced by running the algorithm many times may differ.	In Hierarchical Clustering, results are reproducible in Hierarchical clustering
K- means clustering a simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset).	A hierarchical clustering is a set of nested clusters that are arranged as a tree.
K Means clustering is found to work well when the structure of the clusters is hyper spherical (like circle in 2D, sphere in 3D).	Hierarchical clustering don't work as well as, k means when the shape of the clusters is hyper spherical.
<b>Advantages:</b> 1. Convergence is guaranteed. 2. Specialized to clusters of different sizes and shapes.	<b>Advantages:</b> 1 .Ease of handling of any forms of similarity or distance. 2. Consequently, applicability to any attributes types.
<b>Disadvantages:</b> 1. K-Value is difficult to predict 2. Didn't work well with global cluster.	<b>Disadvantage:</b> 1. Hierarchical clustering requires the computation and storage of an $n \times n$ distance matrix. For very large datasets, this can be expensive and slow

### c) Explain how a cluster is formed in the density based clustering algorithm.

A cluster in a density-based clustering algorithm (such as DBSCAN) is formed based on the concept of density reachability using two important parameters:

- **Epsilon ( $\epsilon$ ):** Defines the radius of the neighborhood around a point.
- **MinPts:** Minimum number of points required within  $\epsilon$  to form a dense region.

#### Types of Points Used in Cluster Formation:

1. **Core Point:** A point having at least MinPts within its  $\epsilon$ -neighborhood. It lies in a dense area and helps expand the cluster.
2. **Border Point:** A point that lies within the  $\epsilon$ -neighborhood of a Core Point but does not have enough points around it to be a Core Point.
3. **Noise Point:** A point that is neither a Core Point nor a Border Point and lies in sparse regions.

#### Cluster Formation Process:

4. **Start With an Unvisited Point:** If it is a Core Point, a new cluster is created, otherwise it is temporarily labeled as noise.
5. **Expand the Cluster (Density-Reachability):** All points that are directly density-reachable from the Core Point (within  $\epsilon$ ) are added to the cluster. The process continues recursively for every new Core Point found.
6. **Form Final Cluster (Density-Connected Points):** The cluster grows until no more points can be added. The final cluster becomes a **maximal set of density-connected Core Points along with their Border Points**, while Noise Points remain excluded.

### Q6) a) How would you choose the number of clusters when designing a K Medoid clustering algorithm? [6]

Choosing the number of clusters in K-Medoid clustering is important to ensure meaningful grouping and accurate results. Since the algorithm requires the number of clusters ( $k$ ) to be defined beforehand, different evaluation methods are used to identify the most suitable value of  $k$ .

#### Methods to Choose the Number of Clusters:

1. **Elbow Method:** Evaluate the total cost (sum of dissimilarities) for different values of  $k$ . The point where the cost curve bends or forms an "elbow" is selected as the optimal number of clusters.
2. **Silhouette Score Analysis:** Calculate the silhouette score for different values of  $k$ . The value of  $k$  that gives the highest silhouette score represents well-separated and dense clusters.
3. **Gap Statistic Method:** Compare the clustering result with a reference random dataset. The  $k$  value with the largest gap indicates the best cluster separation.
4. **Domain Knowledge and Data Characteristics:** Sometimes the number of clusters is chosen based on prior knowledge of the dataset or practical requirements, such as customer segmentation categories.

5. **Stability Testing:** Run the algorithm multiple times with different  $k$  values and check which value produces stable and consistent clusters.
6. **Visual Representation (for 2D/3D Data):** Scatter plots or dendrogram-based insights may be used to visually inspect natural grouping and estimate a meaningful number of clusters.

**b) Write a short note on outlier analysis with respect to clustering. [6]**

Outlier analysis in clustering refers to identifying data points that do not follow the normal data pattern or do not fit well into any formed cluster. Outliers are usually far from dense regions and may represent noise, rare events, or anomalies.

**Role of Outlier Analysis in Clustering:**

1. **Outliers Affect Clustering Accuracy:** Outliers can distort clustering results, especially in centroid-based methods like K-Means and K-Medoids, by shifting cluster centers and increasing intra-cluster distance.
2. **Preprocessing Use:** Outlier detection is often performed before clustering to remove or reduce the influence of extreme values and improve cluster quality.
3. **Clustering as a Tool for Outlier Detection:** Clustering can also identify outliers by finding points that do not belong to any dense group or cluster.
4. **Density-Based Detection (DBSCAN):** DBSCAN naturally identifies outliers as **Noise Points**, since they are not density-reachable from any Core Point.
5. **Distance-Based Detection in K-Means:** In centroid-based methods, points with unusually large distances from the cluster centroid or belonging to very small clusters are treated as outliers.
6. **Importance:** Outlier analysis improves accuracy in domains like fraud detection, intrusion detection, medical diagnosis, and data cleaning.

**c) Differentiate between K-means and Spectral clustering. [6]**

Point of Comparison	K-Means Clustering	Spectral Clustering
<b>1. Basis of Working</b>	Uses distance-based partitioning and assigns points to nearest centroid.	Uses graph theory and eigenvalues of similarity matrix to form clusters.
<b>2. Shape of Clusters</b>	Works best with spherical and well-separated clusters.	Can detect complex, non-linear, and irregular cluster shapes.
<b>3. Input Required</b>	Requires number of clusters ( $k$ ) and initial centroids.	Requires $k$ and a similarity (affinity) matrix of the data.
<b>4. Suitability for High</b>	May perform poorly with high-	Handles high-dimensional and non-

<b>Dimensions</b>	dimensional or non-convex data.	convex data more effectively.
<b>5. Sensitivity to Noise/Outliers</b>	Highly sensitive to noise and outliers.	Less affected by noise due to graph-based similarity modeling.
<b>6. Computational Complexity</b>	Computationally efficient and faster for large datasets.	More computationally expensive due to eigen decomposition of matrices.

➤ **MAY / JUN 2024**

**Q5) a) Why K-medoid is used? Explain k-medoid algorithm. [5]**

K-Medoid is used instead of K-Means because it is more robust and less sensitive to noise and outliers. It selects an actual data point from the dataset as the cluster center (medoid), making results more stable and interpretable.

**Why K-Medoid Is Used:**

1. **Robust Against Outliers:** Unlike K-Means, the medoid does not shift drastically due to extreme values.
2. **Better Interpretability:** The medoid is an actual dataset point, making the cluster center meaningful.
3. **Flexible Distance Metrics:** Works with various similarity measures (Manhattan, Cosine, Jaccard), unlike K-means which mainly uses Euclidean distance.
4. **Works in Non-Euclidean Space:** Suitable for text, categorical or sequence data where mean cannot be computed.

**K-Medoid Algorithm (PAM – Partitioning Around Medoids) (Steps):**

1. **Initialization:** Select  $k$  random points as medoids and assign points to nearest medoid.
2. **Swapping Phase:** Try replacing current medoids with non-medoids and compute new clustering cost.
3. **Optimization:** If the swap reduces total dissimilarity, accept it; otherwise reject.
4. **Repeat** until no further improvement occurs and clusters stabilize.

**b) Why density based clustering is used? Explain any one. [6]**

Density-based clustering is used because it can discover clusters of arbitrary shapes and can effectively identify noise or outliers in the dataset. It groups data points based on dense regions rather than distance from a centroid, making it suitable for complex and non-linear data distributions.

**Why Density-Based Clustering Is Used:**

1. **Detects Arbitrary Shaped Clusters:** Unlike K-Means, which works best for spherical clusters, density-based methods can detect curved, irregular, or complex shaped clusters.
2. **Handles Noise and Outliers:** These algorithms naturally separate noisy points as outliers rather than forcing them into clusters.
3. **No Need to Predefine Number of Clusters:** Clusters are formed based on density parameters rather than a predefined 'k' value.
4. **Suitable for Real-World Data:** Works well in spatial data, image segmentation, geographic mapping, and anomaly detection where data may not be uniformly distributed.
5. **Detects Varying Density:** It identifies dense regions of points as clusters and sparse regions as gaps or noise.

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

- DBSCAN uses two parameters:
  - ✓ **Epsilon ( $\epsilon$ ):** Radius defining neighborhood of a point
  - ✓ **MinPts:** Minimum points required within  $\epsilon$  to form a dense region
- Points are classified as:
  - ✓ **Core Points:** Have at least MinPts neighbors
  - ✓ **Border Points:** Near core points but don't meet MinPts condition
  - ✓ **Noise Points:** Do not belong to any cluster
- The algorithm begins with a core point and expands a cluster by adding all density-reachable points. Points that cannot be assigned remain as noise.

**c) What is outlier analysis? [6]**

Outlier analysis, also called anomaly detection, is the process of identifying data points that significantly deviate from the normal pattern in a dataset. These unusual observations are called **outliers**, and they may represent errors, rare events, or important hidden insights.

**Key Points:**

1. **Purpose:** Outlier analysis separates normal data behavior from abnormal patterns to improve data quality and model accuracy.
2. **Reason for Occurrence:** Outliers may occur due to measurement errors, system faults, unusual behavior, or unexpected rare events.
3. **Impact on Analysis:** Outliers can **distort mean, variance, and clustering results**, and may mislead machine learning models if not handled properly.
4. **Types of Outliers:**
  - **Point Outlier:** Single abnormal value

- **Contextual Outlier:** Abnormal only in a specific context
  - **Collective Outlier:** Group of abnormal values forming a pattern
5. **Detection Methods:** Outliers can be detected using statistical methods (IQR, Z-score), clustering-based methods (DBSCAN, K-Means), distance-based approaches, or machine learning methods like Isolation Forest and One-Class SVM.
  6. **Applications:** Used in fraud detection, cybersecurity intrusion detection, medical diagnosis, manufacturing fault detection, and financial risk analysis.

#### Q6) a) What is isolation factor model? [5]

The Isolation Forest model is an anomaly or outlier detection technique based on the idea that anomalous data points are easier to isolate from the rest of the dataset. Instead of measuring distance or density, it isolates points by randomly selecting a feature and then randomly selecting a split value to divide the data.

##### Key Concepts:

1. **Isolation Principle:** Outliers lie far from normal data, so they require fewer random splits to be isolated, whereas normal points need more splits to be separated.
2. **Tree Structure:** The algorithm builds multiple random trees called **Isolation Trees (iTrees)**. Each tree isolates points by splitting features at random values.
3. **Path Length:** The number of splits required to isolate a point is called its **path length**.
  - Short path length → likely an outlier
  - Long path length → normal point
4. **Anomaly Score:** The final anomaly score is computed by averaging the path lengths across all trees. Scores closer to **1** indicate strong anomalies, while lower values indicate normal points.
5. **Advantages:** Isolation Forest is efficient, scalable to large datasets, and works well in high-dimensional data without requiring distance measures or predefined thresholds.

#### b) Explain k means algorithm. [6]

K-Means is an unsupervised, centroid-based clustering algorithm used to partition data into  $k$  clusters where each point belongs to the nearest cluster center. The main objective is to minimize the **Within-Cluster Sum of Squares (WCSS)**, which represents the total distance between data points and their assigned centroid.

##### Steps of K-Means Algorithm:

1. **Choose Number of Clusters (k):** The user specifies the number of clusters required.
2. **Initialize Centroids:** Randomly select  $k$  data points as initial centroids.

3. **Assignment Step:** Assign each data point to the nearest centroid using distance measures such as Euclidean distance.
4. **Update Step:** Recalculate each centroid by computing the mean of all points assigned to that cluster.
5. **Repeat Until Convergence:** Continue assigning and updating until either the centroids no longer change, the cluster assignments stabilize, or a maximum number of iterations is reached.

#### Objective Function:

The algorithm minimizes the following cost:

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

Where:

- $C_i$  = cluster  $i$
- $\mu_i$  = centroid of cluster  $i$

#### Advantages:

- Simple to understand and implement
- Computationally efficient, especially for large datasets

#### c) Explain Hierarchical clustering with example [6]

Hierarchical clustering is an unsupervised clustering method that builds a tree-like structure of clusters called a **dendrogram**. Instead of forming a fixed number of clusters initially, it forms clusters step-by-step either by merging smaller clusters or splitting larger ones. The goal is to group similar data points based on distance or similarity.

#### Types of Hierarchical Clustering:

1. **Agglomerative (Bottom-Up):**
  - Starts with each data point as an individual cluster.
  - Clusters are gradually merged based on similarity until one big cluster remains.
2. **Divisive (Top-Down):**
  - Starts with one large cluster.
  - Splits the cluster repeatedly into smaller clusters until each point stands alone.

#### Steps in Agglomerative Hierarchical Clustering (Common Approach):

1. Treat each data point as an individual cluster.



2. Compute the distance between all clusters (using Euclidean, Manhattan, etc.).
3. Merge the **two closest clusters** based on a linkage method such as:
  - **Single Linkage:** Minimum distance between clusters
  - **Complete Linkage:** Maximum distance
  - **Average Linkage:** Average distance
4. Recalculate distances and repeat merging until only one cluster remains.
5. Represent the process visually using a **dendrogram**.

**Example:** Consider 4 points: **A, B, C, D**

Point	Values
A	2
B	3
C	8
D	9

**Step-by-step clustering:**

- Step 1: {A}, {B}, {C}, {D} → all separate.
- Step 2: A and B are closest (distance 1), so merge → {AB}, {C}, {D}.
- Step 3: C and D are closest (distance 1), merge → {AB}, {CD}.
- Step 4: Merge {AB} and {CD} to form final cluster → {ABCD}.

The dendrogram will show A and B merging first, then C and D, and finally both groups joining into one cluster.

➤ **MAY / JUN 2025**

**a) Explain Hierarchical clustering with example [6] → DONE**

**b) What is outlier Analysis? Explain it with importance, advantages & disadvantages. [6]**

Outlier analysis, also called anomaly detection, is the process of identifying data points that significantly deviate from the normal pattern in a dataset. These unusual observations are called **outliers**, and they may represent errors, rare events, or valuable hidden insights.

**Key Points of Outlier Analysis:**

1. **Purpose:** Outlier analysis separates normal behavior from abnormal patterns to improve data quality and increase the accuracy of analytical or machine learning models.

2. **Reason for Occurrence:** Outliers may appear due to measurement errors, equipment faults, unusual behavior, fraud attempts, or rare natural events.
3. **Impact on Analysis:** Outliers can distort the mean, variance, correlations, and clustering structure of a dataset, leading to misleading conclusions if not addressed.
4. **Types of Outliers:**
  - **Point Outlier:** A single abnormal value that is significantly different from others.
  - **Contextual Outlier:** A value that is abnormal only under specific conditions or context (e.g., temperature anomaly in winter).
  - **Collective Outlier:** A group of data points that together form an unusual pattern (e.g., repeated abnormal network traffic indicating an attack).

#### Advantages of Outlier Analysis:

- Helps detect rare events such as fraud, cyber-attacks, machine failures, or medical abnormalities.
- Improves data reliability and model performance by identifying and treating noisy or incorrect data.
- Provides deeper insights that regular clustering or statistical analysis may fail to detect.

#### Disadvantages of Outlier Analysis:

- It may be difficult to distinguish between true anomalies and natural variations, especially in complex datasets.
- Misclassification risk exists — meaningful rare cases may be mistakenly removed as noise.
- Advanced detection techniques (e.g., Isolation Forest, LOF, One-Class SVM) can be computationally expensive for large datasets.

#### c) Write short note on Elbow method used in K-mean clustering. [6]

The Elbow Method is a commonly used technique in K-Means clustering to determine the optimal number of clusters ( $k$ ) for a given dataset. Since K-Means requires the user to specify  $k$  in advance, the Elbow Method helps identify a value where adding more clusters does not significantly improve the clustering quality.

#### Key Idea of the Elbow Method:

- The method measures the **Within-Cluster Sum of Squares (WCSS)**, also known as **inertia**, for different values of  $k$ .
- WCSS represents how tightly the data points in a cluster are grouped around their centroid.

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

Where:

- $C_i$  = cluster
- $\mu_i$  = centroid of cluster

As the number of clusters increases, WCSS decreases because clusters become smaller and points are closer to their centroids.

#### Steps of the Elbow Method:

1. Run K-Means for different values of  $k$  (for example,  $k = 1$  to 10).
2. Compute WCSS for each  $k$ .
3. Plot the values of  $k$  on the x-axis and corresponding WCSS values on the y-axis.
4. Identify the point where the curve starts to flatten or bend — resembling an "elbow" shape.
5. The point where the elbow appears is selected as the **optimal number of clusters**.

#### Importance of the Elbow Method:

- Helps avoid too few clusters (underfitting) or too many clusters (overfitting).
- Provides a visual tool for selecting the best  $k$  value.
- Improves accuracy and interpretability of K-Means clustering.

#### Limitation:

- The elbow point may not always be clearly visible, making interpretation subjective.

#### Q6) a) Compare Intrinsic motivation with extrinsic motivation. [6]

Point of Comparison	Intrinsic Motivation	Extrinsic Motivation
<b>1. Meaning</b>	Motivation driven by internal satisfaction, personal interest, or enjoyment of the task.	Motivation driven by external rewards such as money, grades, recognition, or fear of punishment.
<b>2. Source of Motivation</b>	Comes from within the individual (internal factors).	Comes from outside the individual (external factors).
<b>3. Purpose of Action</b>	Performed because the person genuinely wants to do it.	Performed to achieve a reward or avoid a negative consequence.
<b>4. Example</b>	Studying because you enjoy learning or solving problems.	Studying to get good grades or avoid failing.
<b>5. Impact on Creativity</b>	Encourages higher creativity, deeper learning, and long-term engagement.	May limit creativity because focus is on reward rather than learning.
<b>6. Duration of Motivation</b>	Usually long-lasting and self-sustaining.	Often temporary and lasts only until the reward or pressure exists.

## b) Explain K mean clustering with essential steps used in it. [6]

K-Means clustering is an unsupervised learning algorithm used to partition a dataset into  $k$  clusters based on similarity. Each cluster is represented by a **centroid**, and the algorithm groups data points so that points inside the same cluster are closer to each other than to points in other clusters. The main objective is to minimize the **Within-Cluster Sum of Squares (WCSS)**.

### Essential Steps Used in K-Means Clustering:

1. **Select the Number of Clusters (k):** The user decides how many clusters are needed before starting the algorithm.
2. **Initialize Centroids:** Randomly select  $k$  data points from the dataset as the initial centroids. Methods like **K-Means++** can also be used for better initialization.
3. **Assign Data Points to the Nearest Centroid:** Calculate the distance (commonly Euclidean distance) between each point and all centroids, then assign each point to the cluster with the closest centroid.
4. **Update Centroids:** After all points are assigned, compute the new centroid of each cluster by taking the **average of all points** within that cluster.
5. **Repeat Reassignment and Update:** Recalculate distances and update clusters repeatedly until:
  - Centroids stop changing significantly, or
  - Cluster assignments remain stable.

### Objective Function:

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

Where:

- $C_i$  = cluster
- $\mu_i$  = centroid of that cluster

## c) Write short note on. [6]

### i) Graph Based clustering

### ii) Density Based clustering

#### i) Graph-Based Clustering

Graph-based clustering is a clustering technique where the dataset is represented in the form of a **graph**, where nodes represent data points and edges represent similarity or distance between them. The goal is to group nodes such that there are strong connections (high similarity) within a cluster and weak connections between clusters.

**Key Points:**

1. **Representation:** Data is converted into a graph structure using similarity metrics such as cosine similarity, Jaccard similarity, or Euclidean distance.
2. **Clustering Process:** Clusters are formed by finding groups of highly connected nodes using methods like **Minimum Spanning Tree (MST)** or **Connected Components**.
3. **Community Detection:** Algorithms like **Spectral Clustering**, **Girvan–Newman**, and **Modularity-based clustering** are commonly used.
4. **Advantages:** Handles complex and non-linear cluster shapes and works well with social networks, biological networks, and web data.
5. **Disadvantages:** Computationally expensive for large datasets and sensitive to how similarity thresholds are chosen.

**ii) Density-Based Clustering**

Density-based clustering groups data points based on the idea of region density. A cluster is formed where the data points are tightly packed (high density), and gaps or sparse regions represent boundaries or noise.

**Key Points:**

1. **Core Idea:** Clusters are formed by identifying dense regions instead of distances from centroids.
2. **DBSCAN Example:** Uses parameters **Epsilon ( $\epsilon$ )** and **MinPts** to identify:  
✓ Core Points    ✓ Border Points    ✓ Noise Points
3. **Cluster Formation:** A cluster grows by connecting points that are **density-reachable**, meaning there is a chain of close points within  $\epsilon$  distance.
4. **Advantages:** Automatically detects clusters of arbitrary shapes and effectively identifies noise/outliers.
5. **Disadvantages:** Performance depends on choosing good  $\epsilon$  and MinPts values, and may struggle with data having varying densities.

➤ **NOV / DEC 2023****Q5) a) Explain K-Means clustering in detail with a suitable example. [8]**

K-Means clustering is an unsupervised learning algorithm used to partition a dataset into  $k$  meaningful groups based on similarity. Each cluster is represented by a **centroid**, and the algorithm assigns data points to the nearest centroid so that points within a cluster are similar while points in different clusters are dissimilar. The main objective is to minimize the **Within-Cluster Sum of Squares (WCSS)**, which measures how tightly the data points are grouped around their centroid.

**Objective Function:** 
$$WCSS = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

Where:

- $C_i$  = cluster
- $\mu_i$  = centroid of the cluster
- $x$  = data point

#### Steps in the K-Means Algorithm:

1. **Choose the Number of Clusters (k):** The user decides the value of  $k$  based on requirements or methods like the Elbow Method.
2. **Initialize Centroids:** Randomly select  $k$  data points as initial centroids. Methods like **K-Means++** can improve centroid initialization.
3. **Assign Points to Nearest Centroid:** Calculate the distance (usually Euclidean) from each data point to all centroids and assign the point to the closest cluster.
4. **Recalculate Centroids:** Compute the new centroid for each cluster by taking the mean of all data points assigned to that cluster.
5. **Repeat Assignment and Update:** Continue reassigning points and updating centroids until cluster assignments stabilize or centroids stop changing significantly.
6. **Stop (Convergence):** The algorithm stops when no improvement in WCSS occurs.

**Example of K-Means Clustering:** Suppose we have the following 5 data points representing student marks:

Student	Marks
A	25
B	27
C	32
D	85
E	90

We want to form **k = 2 clusters**.

#### Step 1: Initialize Centroids

Assume initial centroids are:

- Cluster 1: 25
- Cluster 2: 85

#### Step 2: Assignment

Calculate distances:

Point	Distance to 25	Distance to 85	Assigned Cluster
25	0	60	C1
27	2	58	C1
32	7	53	C1
85	60	0	C2
90	65	5	C2

So clusters are:

- $C1 = \{25, 27, 32\}$
- $C2 = \{85, 90\}$

### Step 3: Recalculate Centroids

- New centroid of C1:  $(25 + 27 + 32) / 3 = 28$
- New centroid of C2:  $(85 + 90) / 2 = 87.5$

### Step 4: Repeat Process

Reassign points based on new centroids.

Since assignments do not change in the next iteration, the algorithm **converges**.

### Final Clusters:

Cluster	Members	Centroid
C1	25, 27, 32	28
C2	85, 90	87.5

## b) What is outlier analysis? How is Local Outlier Factor (LOF) detected? [5]

Outlier analysis, also called anomaly detection, is the process of identifying data points that significantly differ from normal patterns in a dataset. These unusual points are known as **outliers**, and they may represent errors, rare events, fraud, or abnormal system behavior. Detecting outliers is important because they can distort analysis, affect model accuracy, or reveal meaningful hidden patterns.

### Local Outlier Factor (LOF) Method for Detecting Outliers:

The **Local Outlier Factor (LOF)** is a density-based method used to detect outliers by comparing the **local density** of a data point with the density of its neighbors.

**Steps Used in LOF:**

1. **Select k Nearest Neighbors:** For each data point, identify its  $k$  nearest neighbors using a distance measure (such as Euclidean distance).
2. **Compute Local Density:** Calculate the **local reachability density (LRD)** of each point, which measures how densely the point is surrounded by its neighbors.
3. **Compare Density with Neighbors:** The LOF score is computed by comparing the density of the point with the density of its neighbors.
4. **Compute LOF Score:**  $LOF(A) = \frac{\text{Average density of neighbors}}{\text{Density of point A}}$
5. **Interpretation of LOF Score:**
  - $LOF \approx 1 \rightarrow$  Normal point
  - $LOF > 1 \rightarrow$  Lower density than neighbors  $\rightarrow$  **Potential Outlier**
  - $LOF \gg 1$  (for example  $>1.5$  or  $2$ )  $\rightarrow$  **Strong Outlier**

✓ Local Outlier Factor detects points that have significantly lower density than surrounding points, making it effective for identifying outliers in datasets with varying density.

**c) Explain Spectral Cluster in algorithm. [5]**

Spectral Clustering is an unsupervised learning algorithm that uses concepts from graph theory and linear algebra to form clusters. Instead of grouping data directly based on distance (like K-Means), it transforms the data into a **graph structure**, computes eigenvalues of a similarity matrix, and then applies clustering to the transformed feature space. Spectral clustering is useful for detecting **non-linear, complex, and arbitrary-shaped clusters**.

**Steps in Spectral Clustering Algorithm:**

1. **Construct a Similarity Matrix:** Create a matrix (such as Gaussian or adjacency matrix) that represents similarity between all data points. If two points are close, their similarity value is high.
2. **Build a Graph Laplacian:** Convert the similarity matrix into a graph Laplacian, which represents connections between points. The Laplacian helps capture cluster structure mathematically.
3. **Compute Eigenvalues and Eigenvectors:** Perform eigen decomposition on the Laplacian matrix. The top  $k$  eigenvectors (lowest eigenvalues) are selected to create a new lower-dimensional space.
4. **Apply K-Means to Transformed Space:** Treat the eigenvector space as new feature representation and apply K-Means to form the final clusters.
5. **Output the Clusters:** The algorithm groups points based on graph connection patterns rather than simple distance.



**Advantages:**

- Detects **non-linear and complex-shaped clusters**.
- Works well in high-dimensional data and graph-based relationships.

**Limitations:**

- Computationally expensive for very large datasets.
- Requires selecting the similarity function and number of clusters in advance.

**Q6) a) Explain Hierarchical and Density-based Clustering approaches.**

Clustering is an unsupervised learning process used to group similar data points. Two commonly used clustering approaches are **Hierarchical Clustering** and **Density-Based Clustering**. Both methods form clusters differently and are suitable for different data patterns.

**1) Hierarchical Clustering:** Hierarchical clustering builds a hierarchy of clusters and represents the process visually through a **dendrogram**. It does not require specifying the number of clusters in advance.

**Types:**

- **Agglomerative (Bottom-Up):**  
Each data point starts as its own cluster, and the closest clusters are merged step-by-step until one large cluster remains.
- **Divisive (Top-Down):**  
All points start in one cluster and are split repeatedly into smaller groups.

**Steps in Agglomerative Approach:**

1. Treat every data point as a separate cluster.
2. Calculate the distance between all clusters.
3. Merge the two closest clusters based on linkage (single, complete, or average).
4. Repeat merging until only one cluster remains.
5. Use the dendrogram to decide the number of clusters by making a horizontal cut.

**Example:** For points A, B, C, D:

- First merge closest pair (A and B  $\rightarrow$  {AB})
- Then merge next closest (C and D  $\rightarrow$  {CD})
- Finally merge both groups to form {ABCD}

Hierarchical clustering is useful in **bioinformatics, text mining, and pattern recognition**.

**2) Density-Based Clustering:** Density-based clustering groups data based on dense regions. High-density areas form clusters, while low-density areas are treated as noise or outliers. It is effective for discovering clusters of **arbitrary shape**.

**Example: DBSCAN (Popular Density-Based Algorithm)**

DBSCAN uses two parameters:

- **$\epsilon$  (Epsilon):** Neighborhood radius
- **MinPts:** Minimum points required to form a dense region

**Point Types:**

- **Core Points:** Have at least MinPts neighbors within  $\epsilon$
- **Border Points:** Near core points but do not meet MinPts rule
- **Noise Points:** Do not belong to any cluster

**Process:**

1. Select an unvisited point.
2. If it is a core point, start a new cluster and expand it by connecting all density-reachable points.
3. If it is not a core point, mark it temporarily as noise.
4. Continue until all points are processed.

Density-based clustering is useful in **geospatial analysis, anomaly detection, and image segmentation**.

**b) Write short note on: [9]**

**i) Optimization of clusters**

**ii) K-Medoids**

**iii) Evaluation metrics**

**i) Optimization of Clusters**

Optimization of clusters refers to improving the quality of formed clusters by ensuring that similar data points are grouped together while dissimilar points remain separated. The goal is to achieve meaningful, compact, and well-separated clusters.

**Key Points:**

1. **Objective:** To find the best number of clusters and improve the structure of clustering results.
2. **Methods Used:** Techniques like **Elbow Method, Silhouette Score, and Gap Statistic** are commonly used to measure cluster quality.
3. **Importance:** Optimization prevents under-clustering (too few clusters) and over-clustering (too many clusters), improves model accuracy, ensures stable results, and enhances decision-making in applications like customer segmentation and anomaly detection.

## ii) K-Medoids

K-Medoids is a partitioning clustering algorithm similar to K-Means but uses an **actual data point called a Medoid** as the cluster center instead of a centroid. It is more robust to noise and outliers.

### Key Points:

1. **Why Used:** It is preferred when datasets contain extreme values because medoids are less sensitive to outliers.
2. **Working (PAM Algorithm):**
  - Select k initial medoids
  - Assign points to nearest medoid
  - Swap medoids with non-medoids if it reduces total clustering cost
  - Repeat until no improvement
3. **Advantages:** Works with various distance metrics (Manhattan, Cosine, Jaccard) and is suitable for text, categorical, and non-Euclidean data.

## iii) Evaluation Metrics

Evaluation metrics are measurement techniques used to assess the performance and quality of clustering results. Since clustering is unsupervised, evaluation relies on internal structure, density, and separation.

### Key Metrics:

1. **Silhouette Score:** Measures how similar a data point is to its own cluster compared to other clusters. Score ranges from **-1 to +1**; higher values indicate better clustering.
2. **Dunn Index / Davies-Bouldin Index:**
  - **Dunn Index:** Higher value = better cluster separation
  - **DB Index:** Lower value = better clustering
3. **WCSS (Within-Cluster Sum of Squares):** Used in K-Means to measure how tightly points cluster around the centroid. Lower WCSS indicates compact and meaningful clusters.

➤ **NOV / DEC 2024**

**Q5) a) Why K-medoid is used? Explain K-medoid algorithm. [5] → DONE**

**b) Why density based clustering is used? Explain any one. [6] → DONE**

**c) Cluster the following eight points (with (x, y) representing locations) into three clusters: P1(1, 3), P2(2, 2), P3(5, 8), P4(8, 5), P5(3, 9), P6(10, 7), P7(3, 3), P8(9, 4), P9(3, 7)**

**Use K-Means Algorithm to find the three cluster**

**Step 1:** We will use **K-Means clustering** to divide the given 9 points into **3 clusters**. Assume initial centroids as:

- **C1 = P1(1, 3)**
- **C2 = P3(5, 8)**
- **C3 = P6(10, 7)**

Point	(1,3)=C1	(5,8)=C2	(10,7)=C3	Assigned Cluster
P1(1,3)	0.00	6.40	9.85	C1
P2(2,2)	1.41	6.71	9.43	C1
P3(5,8)	6.40	0.00	5.10	C2
P4(8,5)	7.28	4.24	2.83	C3
P5(3,9)	6.32	2.24	7.28	C2
P6(10,7)	9.85	5.10	0.00	C3
P7(3,3)	2.00	5.39	8.06	C1
P8(9,4)	8.06	5.66	3.16	C3
P9(3,7)	4.47	2.24	7.00	C2

So after **1st assignment**:

- **Cluster 1 (C1):** P1(1,3), P2(2,2), P7(3,3)
- **Cluster 2 (C2):** P3(5,8), P5(3,9), P9(3,7)
- **Cluster 3 (C3):** P4(8,5), P6(10,7), P8(9,4)

### Step 2: Recalculate New Centroids

**Cluster 1:** P1(1,3), P2(2,2), P7(3,3)

- New centroid  $C1' = \left( \frac{1+2+3}{3}, \frac{3+2+3}{3} \right) = (2, 2.67)$

**Cluster 2:** P3(5,8), P5(3,9), P9(3,7)

- New centroid  $C2' = \left( \frac{5+3+3}{3}, \frac{8+9+7}{3} \right) = (3.67, 8)$

**Cluster 3:** P4(8,5), P6(10,7), P8(9,4)

- New centroid  $C3' = \left( \frac{8+10+9}{3}, \frac{5+7+4}{3} \right) = (9, 5.33)$

### Step 3: Check Next Iteration

If we recompute distances using new centroids, **each point remains in the same cluster** as above.

So the algorithm has **converged**.

**Final Clusters (K-Means Result, k = 3)**

- **Cluster 1:**  
P1(1,3), P2(2,2), P7(3,3)
- **Cluster 2:**  
P3(5,8), P5(3,9), P9(3,7)
- **Cluster 3:**  
P4(8,5), P6(10,7), P8(9,4)

This is the final clustering of the 9 points into **3 clusters using K-Means**.

**Q6) a) What is isolation factor model? [5] → DONE**

**b) Explain Hierarchical Clustering with an example. → DONE**

**c) Micro-Average Precision and Recall, Micro-Average F-score [6]**

Micro-averaging is a method used in multi-class or multi-label classification evaluation to compute Precision, Recall, and F-score by considering the total number of true positives, false positives, and false negatives across all classes. Instead of calculating metrics per class separately, micro-averaging treats each prediction equally and computes a global score.

**Micro-Average Precision:** Micro-average precision measures how many of the total predicted positive instances across all classes are actually correct.

$$\text{Micro Precision} = \frac{\sum TP}{\sum (TP + FP)}$$

Where: **TP:** True Positive & **FP:** False Positive

✓ It gives more weight to larger classes.

**Micro-Average Recall:** Micro-average recall measures how many of the total actual positive instances across all classes were correctly predicted.

$$\text{Micro Recall} = \frac{\sum TP}{\sum (TP + FN)}$$

Where: **FN:** False Negative

✓ In multi-class problems, **micro precision = micro recall**, because total positives equal total actual positives.

**Micro-Average F-score:** The Micro F-score combines micro-precision and micro-recall using the harmonic mean. It provides a single performance score reflecting both precision and recall.

$$\text{Micro F-score} = \frac{2 \times (\text{Micro Precision} \times \text{Micro Recall})}{(\text{Micro Precision} + \text{Micro Recall})}$$

✓ It is useful when all instances are equally important, especially in **balanced datasets**.

**Key Characteristics:**

- Gives equal weight to each instance, not each class
- Works best for datasets where majority and minority classes exist
- Used in applications such as text classification, medical diagnosis, and multi-label problems

➤ **Additional NOV / DEC 2022 Questions:**

**Q5) a) Explain Density Based clustering with reference to DBSCAN, OPTICS and DENCLUE. [6]**

**1. Introduction to Density-Based Clustering**

- Density-based clustering groups data points based on **regions of high density**, separated by regions of **low density**.
- Clusters can be of **arbitrary shapes** (unlike K-means which finds spherical clusters).
- Uses two key concepts:
  - **$\epsilon$  (Epsilon)**: Neighborhood radius
  - **MinPts**: Minimum points required to form a dense region
- Identifies **core points, border points, and noise/outliers** based on density.

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

**2. Key Ideas**

- Forms clusters by expanding from **core points** that have at least MinPts points within  $\epsilon$  radius.
- **Density Reachability**: Points reachable from a core form a cluster.
- Border points are attached to clusters; low-density points become **noise**.

**3. Advantages & Characteristics**

- Finds arbitrary-shaped clusters.
- Handles noise effectively.
- Requires **only  $\epsilon$  and MinPts** as parameters.

**OPTICS (Ordering Points To Identify the Clustering Structure)**

**4. Key Ideas**

- Extension of DBSCAN that removes the need to choose a single  $\epsilon$  value.
- Produces an **ordering of points** based on their density relationship.
- Uses two distances:
  - **Core Distance** – minimum radius needed for MinPts
  - **Reachability Distance** – density connectivity between points

**5. Characteristics**

- Generates a **reachability plot** from which clusters at multiple density levels can be extracted.
- More flexible than DBSCAN in datasets with **varying densities**.
- Does not explicitly produce clusters; instead, provides a structure from which clusters can be derived.

**DENCLUE (DENSity-based CLUstEring)****6. Key Ideas**

- Uses **mathematical density functions** (Kernel Density Estimation) instead of counting points in a radius.
- Each point contributes to a smooth density function using **Gaussian kernels**.
- Clusters are formed around **density attractors** (local maxima in density).
- Uses gradient ascent to find these attractors.

**7. Characteristics**

- Handles noise efficiently due to density threshold.
- Works well for large and high-dimensional datasets.
- Provides a more **theoretical and precise** density model than DBSCAN/OPTICS.

**8. Summary**

- **DBSCAN**: Simple, effective for arbitrary shapes but struggles with varying densities.
- **OPTICS**: Overcomes DBSCAN's fixed  $\epsilon$  limitation by providing a full density-based ordering.
- **DENCLUE**: Uses kernel density estimation and mathematical modeling for cluster formation.

**9. Conclusion:** Density-based clustering discovers clusters as **dense regions** separated by **low-density regions**, and methods like **DBSCAN, OPTICS, and DENCLUE** efficiently identify such clusters, each progressively improving flexibility and accuracy.

**Q6) a) What is LOF? Explain it with it's advantages and disadvantages. [6]****1. Definition of LOF**

- **LOF (Local Outlier Factor)** is a *density-based outlier detection method*.
- It measures how *isolated* a point is with respect to its local neighborhood.
- A point is considered an outlier if its **local density is significantly lower** than the densities of its neighbors.

**2. Key Idea / Concept**

- LOF compares the **local reachability density (LRD)** of a point with the LRD of its *k-nearest neighbors*.
- **Local Reachability Density (LRD):**
  - Inverse of the average distance from a point to its neighbors.
  - Low LRD  $\Rightarrow$  Sparse neighborhood  $\Rightarrow$  Possible outlier.

### 3. LOF Score

- LOF score  $\approx 1$ : point is similar to neighbors (normal).
- LOF score  $> 1$ : point has lower density  $\rightarrow$  potential outlier.
- Higher the LOF score, the stronger the outlier.

### 4. Steps in LOF Algorithm

1. **Choose  $k$**  (number of neighbors).
2. Compute  **$k$ -distance** and identify  **$k$ -nearest neighbors** for each point.
3. Calculate **reachability distance** for each point relative to its neighbors.
4. Compute **LRD (local reachability density)**.
5. Compute **LOF score** by comparing a point's LRD with its neighbors' LRD.
6. Points with **high LOF** are flagged as outliers.

### 5. Advantages

- a) **Detects local outliers:** Works even when outliers are only unusual *relative to their local neighborhood*.
- b) **Handles datasets with varying density:** More flexible than global distance-based methods.
- c) **No assumption of data distribution:** Non-parametric; works on complex real-world data.
- d) **Identifies subtle anomalies:** Very effective for detecting *nearby but low-density* anomalies.

### 6. Disadvantages

- a) **Parameter sensitivity:** Sensitive to choice of  $k$ . Incorrect value affects detection quality.
- b) **Computationally expensive:** Requires finding  $k$ -nearest neighbors  $\rightarrow$  costly for large datasets.
- c) **Hard to set threshold:** LOF scores do not have fixed boundaries; choosing cutoff is tricky.
- d) **Not suitable for very high-dimensional data:** Distance measures become unreliable (curse of dimensionality).

**7. Conclusion:** LOF is a powerful **density-based local outlier detection** technique that identifies anomalies by comparing a point's neighborhood density with that of its neighbors. It works well for datasets with varying density but needs careful parameter selection and is computationally expensive.

**Note: Please verify all answers before referring.**