PART A

(PART A : TO BE REFFERED BY STUDENTS)

# Experiment No.08

**Aim**: Implementation of Clustering algorithm (K-means/K-medoids) using Java/Python.

**Outcome:**

**After successful completion of this experiment students will be able to**

1. Demonstrate an understanding of the importance of data mining
2. Organize and prepare the data needed for data mining using pre preprocessing techniques
3. Perform exploratory analysis of the data to be used for mining.
4. Implement the appropriate data mining methods like clustering on large data sets.

**Theory:**

**K-Means Clustering Algorithm**

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.

**What is K-Means Algorithm?**

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

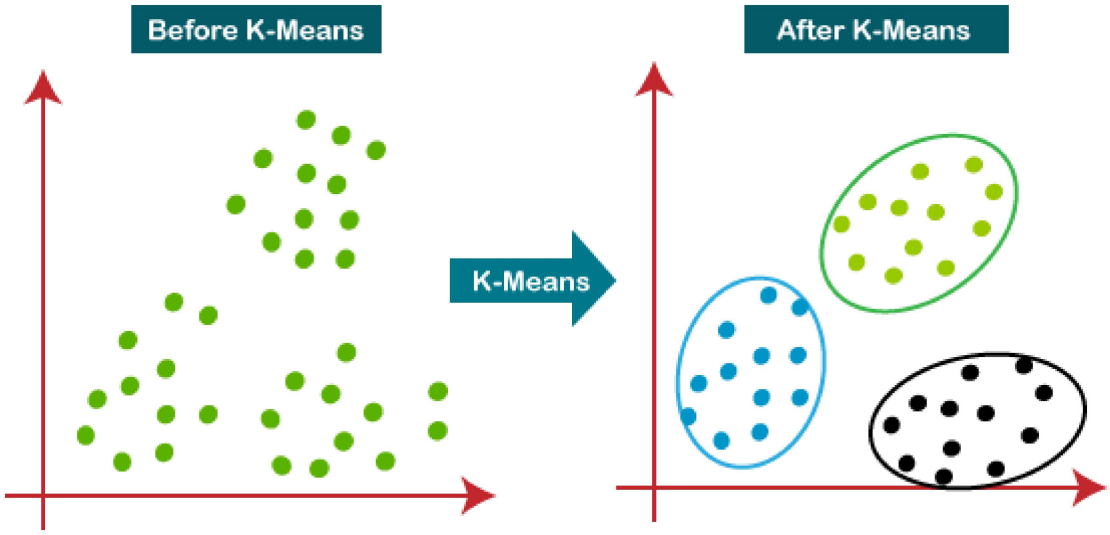
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:




**How does the K-Means Algorithm Work?**

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other from the input dataset).

**Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.

**Step-4:** Calculate the variance and place a new centroid of each cluster.

**Step-5:** Repeat the third steps, which mean reassign each datapoint to the new closest centroid of each cluster.

**Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

**Step-7**: The model is ready.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

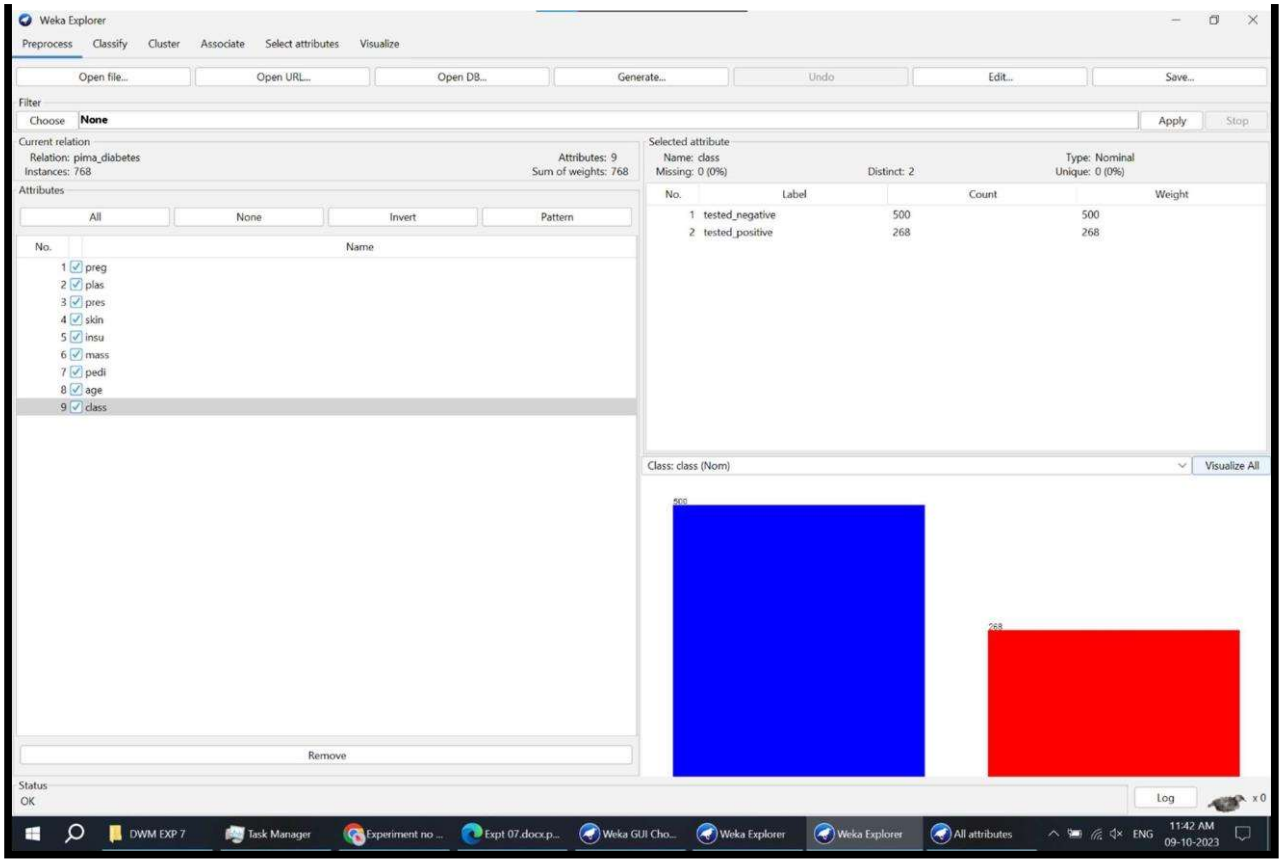
| | |
|---|---|
| Roll. No: A07 | Name: Niharika Seth |
| Class: TE AI&DS | Batch: A1 |
| Date of Experiment: 02/10/2024 | Date of Submission: 8/10/2024 |
| Grade: | |

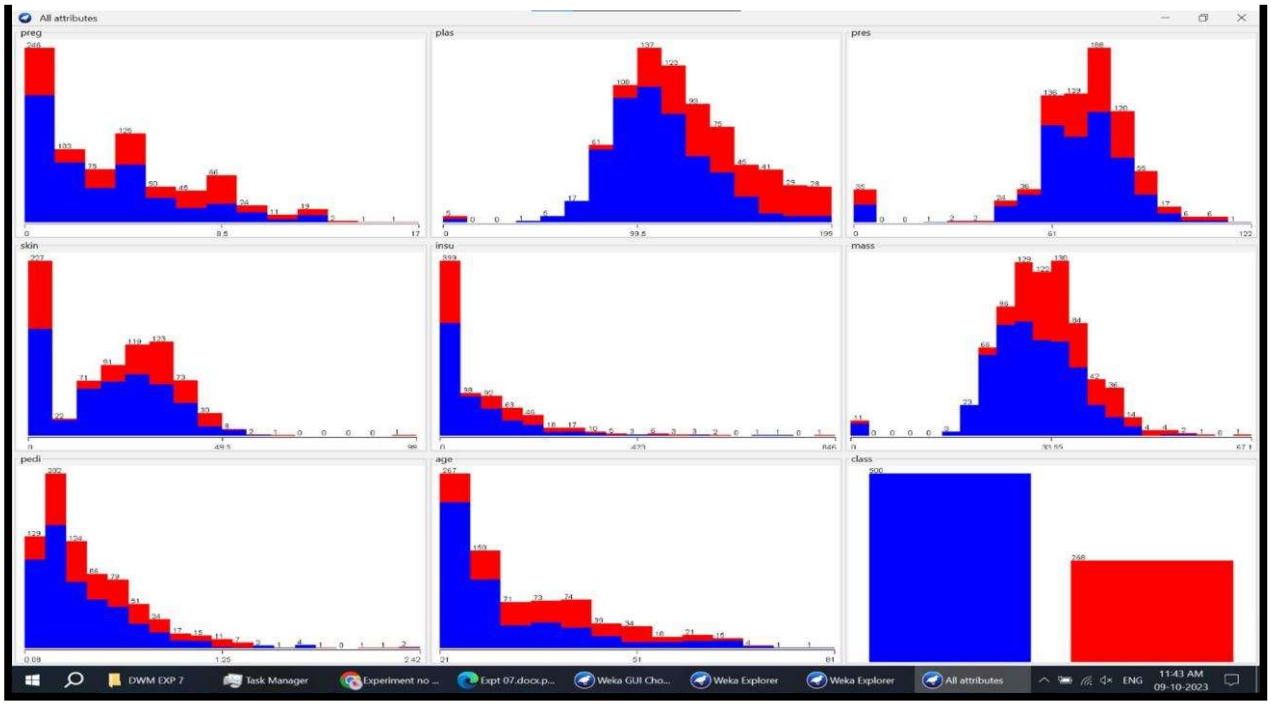
## B.1. Software Code written by student:

Implementation of K-means clustering using WEKA.

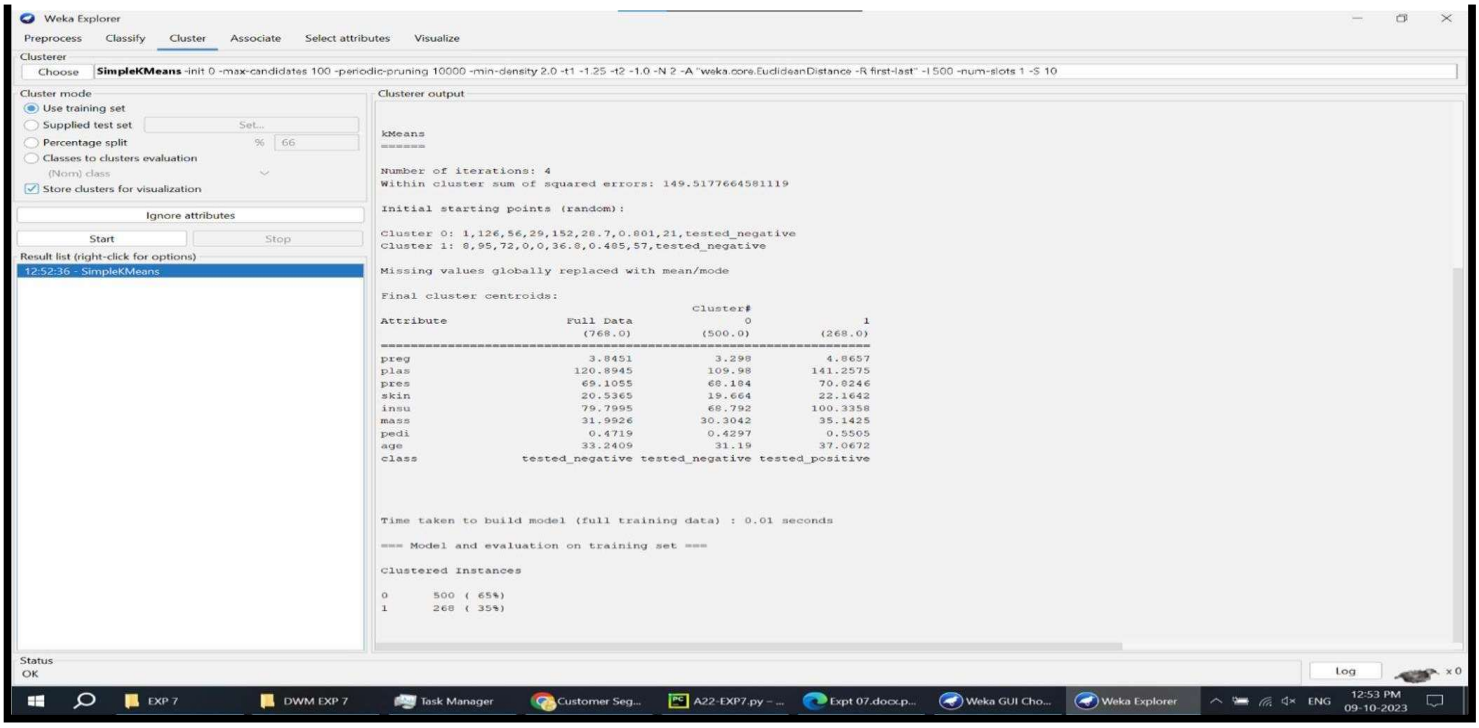
## B.2. Input and Output:

Step 1: Selecting diabities.arff file from database
Step 2: Select Cluster from above, and choose SimpleKMeans. Click on Start button.
Step 3: Right click on SimpleKMeans and select Visualize cluster assignments.

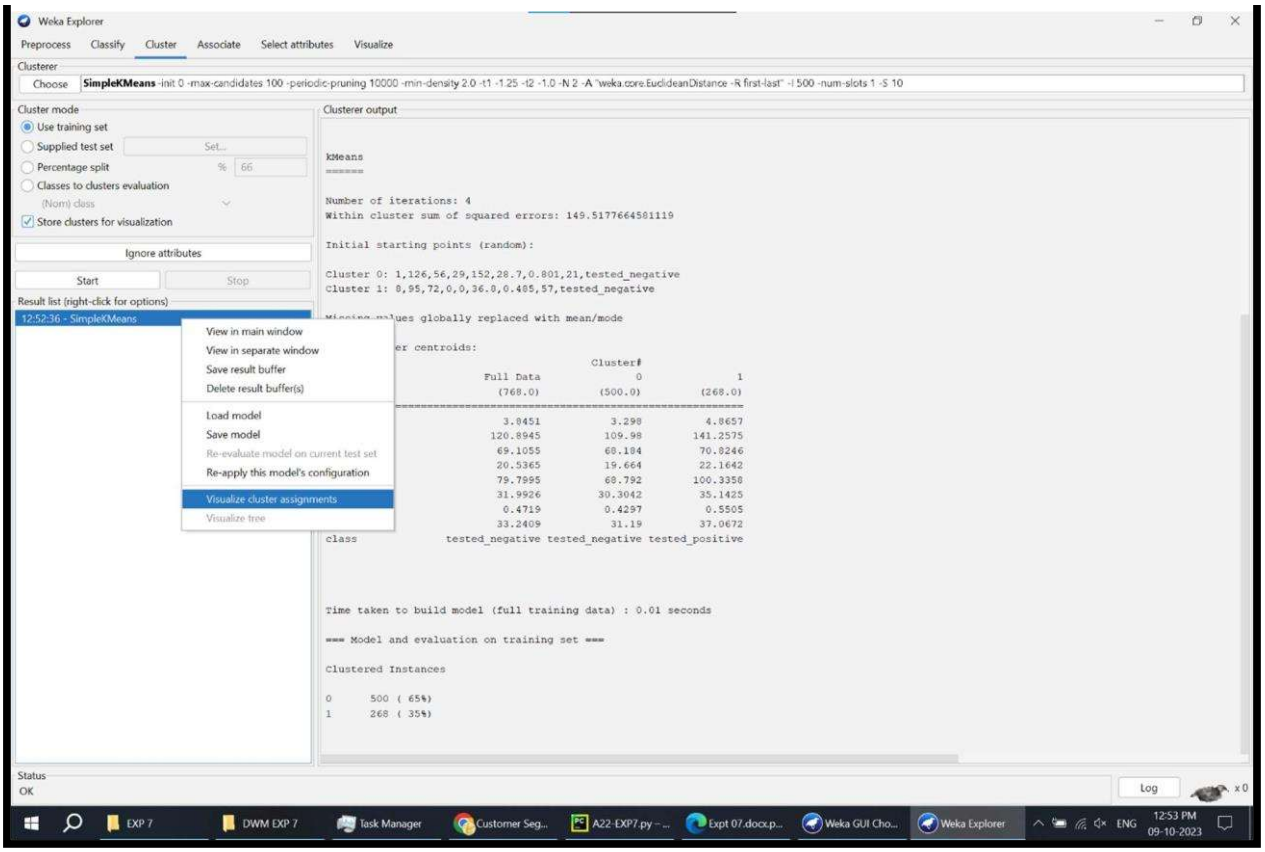


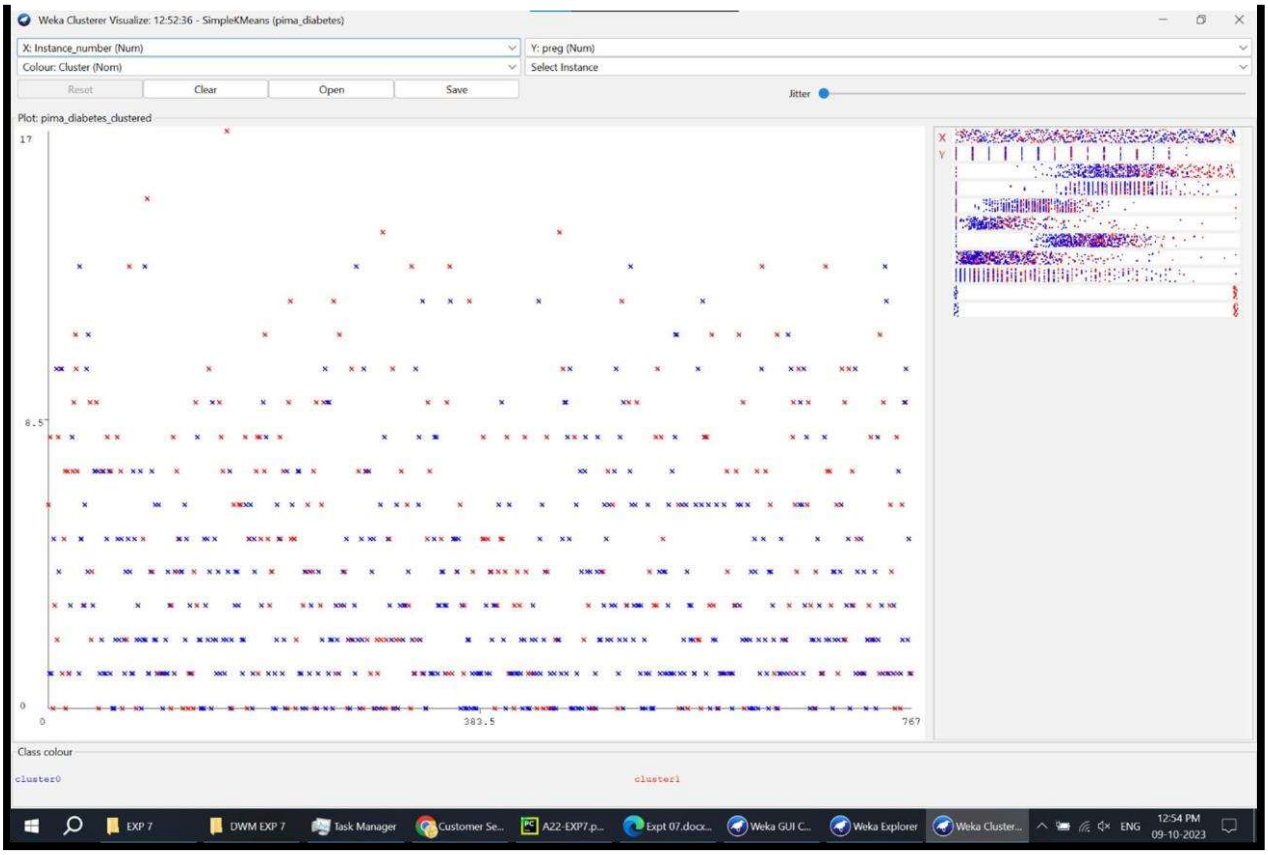



Step 2: Select Cluster from above, and choose SimpleKMeans. Click on Start button.




Step 3: Right click on SimpleKMeans and select Visualize cluster assignments.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
12:52:36 - SimpleKMeans
View in main window
View in separate window
Save result buffer
Delete result buffer(s)
Load model
Save model
Re-evaluate model on current test set
Re-apply this model's configuration
Visualize cluster assignments
Visualize tree
Clusterer output
kMeans
======
Number of iterations: 4
Within cluster sum of squared errors: 149.5177664581119
Initial starting points (random):
Cluster 0: 1,126,56,29,152,28.7,0.801,21,tested_negative
Cluster 1: 8,95,72,0,0,36.0,0.485,57,tested_negative
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Full Data
0
1
(768.0)
(500.0)
(268.0)
3.8451
3.298
4.8657
120.8945
109.98
141.2575
69.1055
68.184
70.8246
20.5365
19.664
22.1642
79.7995
68.792
100.3358
31.9926
30.3042
35.1425
0.4719
0.4297
0.5505
33.2409
31.19
37.0672
class
tested_negative
tested_negative
tested_positive
Time taken to build model (full training data) : 0.01 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 500 ( 65%)
1 268 ( 35%)
Status
OK
Log
x 0
EXP 7
DWM EXP 7
Task Manager
Customer Seg...
A22-EXP7.py – ...
Expt 07.docx.p...
Weka GUI Cho...
Weka Explorer
ENG
12:53 PM
09-10-2023


Weka Clusterer Visualize: 12:52:36 - SimpleKMeans (pima_diabetes)
X: Instance_number (Num)
Y: preg (Num)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: pima_diabetes_clustered
17
8.5
0
0
383.5
767
Class colour
cluster0
cluster1
EXP 7
DWM EXP 7
Task Manager
Customer Se...
A22-EXP7.p...
Expt 07.docx...
Weka GUI C...
Weka Explorer
Weka Cluster...
ENG
12:54 PM
09-10-2023

## Observations and learning:

Observation: Implementing K-means in WEKA involves loading data, configuring K and distance metrics, running the algorithm, and interpreting results.
Learning: Carefully choose K, preprocess data, experiment with distance measures, and be open to trying other clustering algorithms in WEKA as per your data's characteristics and objectives.

## Conclusion:

We have successfully conducted the K-means clustering implementation with WEKA and have gained a solid understanding of the K-means clustering concepts through this experiment.

## Question of Curiosity

**Q1: How is the number of cluster chosen?**

Ans: Choosing the right number of clusters (often denoted as "K") in a clustering algorithm is a critical step, as it can significantly impact the quality and interpretability of the results. Several methods can help determine the optimal number of clusters:
1. Elbow Method:

In this method, you plot the number of clusters (K) on the x-axis and the sum of squared distances (inertia or within-cluster variation) on the y-axis.
As you increase K, the sum of squared distances generally decreases because the data points are closer to their cluster centroids. However, the rate of decrease slows down after a certain point.
Look for the "elbow" point on the graph, where the rate of decrease sharply changes.
This point represents a good estimate of the optimal number of clusters.
Keep in mind that the elbow method can be subjective, and the elbow may not always be clearly defined.

2. Silhouette Score:

The silhouette score measures how similar each data point is to its own cluster compared to other clusters. It ranges from -1 (a poor clustering) to +1 (a perfect clustering).
Compute the silhouette score for different values of K and choose the K that yields the highest silhouette score.
A higher silhouette score indicates better separation between clusters.

3. Gap Statistics:

Gap statistics compare the quality of your clustering to that of a random clustering. Generate random data with the same distribution as your dataset and perform clustering on it.
Compare the clustering performance (e.g., sum of squared distances) of your actual data to that of the random data for various values of K.
Choose K where the gap between the clustering quality of actual data and random data is the largest.

4. Dendrogram from Hierarchical Clustering:
If you're using hierarchical clustering, you can create a dendrogram, which shows how clusters merge as you increase the number of clusters.

Look for a point in the dendrogram where the clusters start merging less abruptly, indicating an appropriate number of clusters.

5. Domain Knowledge:
Sometimes, domain knowledge or business objectives can provide insights into the appropriate number of clusters. For example, if you're clustering customer data, you might know that there are three distinct customer segments based on your business needs.

6. Visual Inspection:
Visualize the data in 2D or 3D space and try to identify natural groupings. While this method is subjective, it can provide valuable insights.

7. Cross-Validation:
If you plan to use clustering as a preprocessing step for another task (e.g., classification), you can perform cross-validation on that downstream task with different values of K to determine which K leads to the best performance.

**Q2: What is Clustering? Types of clustering? Explain advantages and disadvantages of clustering.**

Ans: Clustering is a fundamental technique in data analysis and machine learning used to group similar data points together based on their inherent characteristics or similarities. The goal of clustering is to divide a dataset into clusters or groups so that data points within the same cluster are more similar to each other than to those in other clusters. This allows for data exploration, pattern recognition, and often serves as a preprocessing step for various machine learning tasks.

Types of Clustering:

1. K-Means Clustering: This is one of the most popular clustering algorithms. It partitions the dataset into K clusters, where K is predefined. It assigns each data point to the cluster with the nearest mean, minimizing the within-cluster sum of squared distances. K-Means is simple, efficient, and works well when clusters are spherical and have roughly equal sizes.

2. Hierarchical Clustering: This method builds a hierarchical representation of data, creating a tree-like structure of clusters. It can be either agglomerative (starting with individual data points and merging them) or divisive (starting with one big cluster and splitting it). The result is a dendrogram that can be cut at different levels to obtain clusters of different sizes.

3. Density-Based Clustering (DBSCAN): DBSCAN identifies clusters based on regions of high data point density. It forms clusters by connecting data points that are close to each other and have a sufficient number of nearby neighbors. Unlike K-Means, it can discover clusters of arbitrary shapes and does not require specifying the number of clusters in advance.

4. Mean-Shift Clustering: Mean-shift is a non-parametric clustering algorithm that seeks modes (peaks) in the data distribution. It starts with a set of data points and iteratively shifts them towards the mode of the nearest data points. The modes represent cluster centers.

5. DBSCAN Clustering: Density-based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm. It groups together data points that are closely packed and separates regions of lower density.

Advantages of Clustering:

1. Data Exploration: Clustering helps in understanding the underlying structure in data, making it easier to identify patterns and relationships between data points.

2. Anomaly Detection: It can be used to detect outliers or anomalies by considering data points that do not belong to any cluster as outliers.

3. Feature Engineering: Clustering can assist in feature engineering by creating new features based on cluster assignments, which can improve the performance of machine learning models.

4. Customer Segmentation: In business applications, clustering can be used for customer segmentation, helping companies target specific customer groups with tailored marketing strategies.

5. Image and Document Organization: Clustering is used in image and document analysis to organize and group similar images or documents together for easier retrieval and analysis.

Disadvantages of Clustering:

1. Sensitivity to Initial Conditions: Some clustering algorithms, like K-Means, are sensitive to initial cluster centers, which can lead to different results with different initializations.

2. Choosing the Right Number of Clusters: Selecting the optimal number of clusters (K) can be challenging, and a poor choice can lead to suboptimal results.

3. Scalability: Some clustering algorithms may not scale well to large datasets or highdimensional data due to computational complexity.

4. Assumption of Cluster Shape: Many clustering algorithms assume that clusters have certain shapes (e.g., spherical), which may not always hold in real-world data.

5. Handling Noisy Data: Clustering algorithms can be sensitive to noisy data or outliers, which may lead to suboptimal cluster assignments.

**Q3: Give the advantages and disadvantages of K- means clustering.**

Ans: Advantages of K-Means Clustering are as follows:

1. Simple and Easy to Implement: K-Means is relatively simple to understand and implement, making it a good choice for beginners and for quick exploratory data analysis.

2. Efficiency: It is computationally efficient and can handle large datasets with a moderate number of clusters.

3. Scalability: K-Means can work well in high-dimensional spaces, although the curse of dimensionality can affect its performance as dimensionality increases.

4. Cluster Interpretability: The resulting clusters are often easy to interpret and visualize, especially when the data is well-separated and roughly spherical in shape.

5. Deterministic Results: With the same initial conditions (random seeds), K-Means will produce the same clusters, making the results reproducible.

Disadvantages of K-Means Clustering:

1. Sensitive to Initializations: The choice of initial cluster centers can significantly impact the results. K-Means may converge to suboptimal solutions if initialized poorly.

2. Requires Specifying K: You need to specify the number of clusters (K) in advance, which can be challenging and

may require trial and error.

3. Assumes Spherical Clusters: K-Means assumes that clusters are spherical, equally sized, and have roughly similar densities, which may not hold for all types of data.

4. Outliers Affect Results: Outliers can heavily influence the position and size of clusters in K-Means, potentially leading to suboptimal results.

5. May Not Handle Non-Globular Shapes: K-Means tends to perform poorly on datasets with non-globular or irregularly shaped clusters. It may split such clusters into smaller spherical clusters.

6. Doesn't Handle Unequal Cluster Sizes: K-Means struggles when dealing with clusters of significantly different sizes because it minimizes the sum of squared distances, favoring larger clusters.

7. Hard Assignment: K-Means uses hard assignment, meaning each data point belongs to one and only one cluster. This can be limiting when data points have mixed memberships.

**********