Unsupervised Machine Learning

UNIT I:

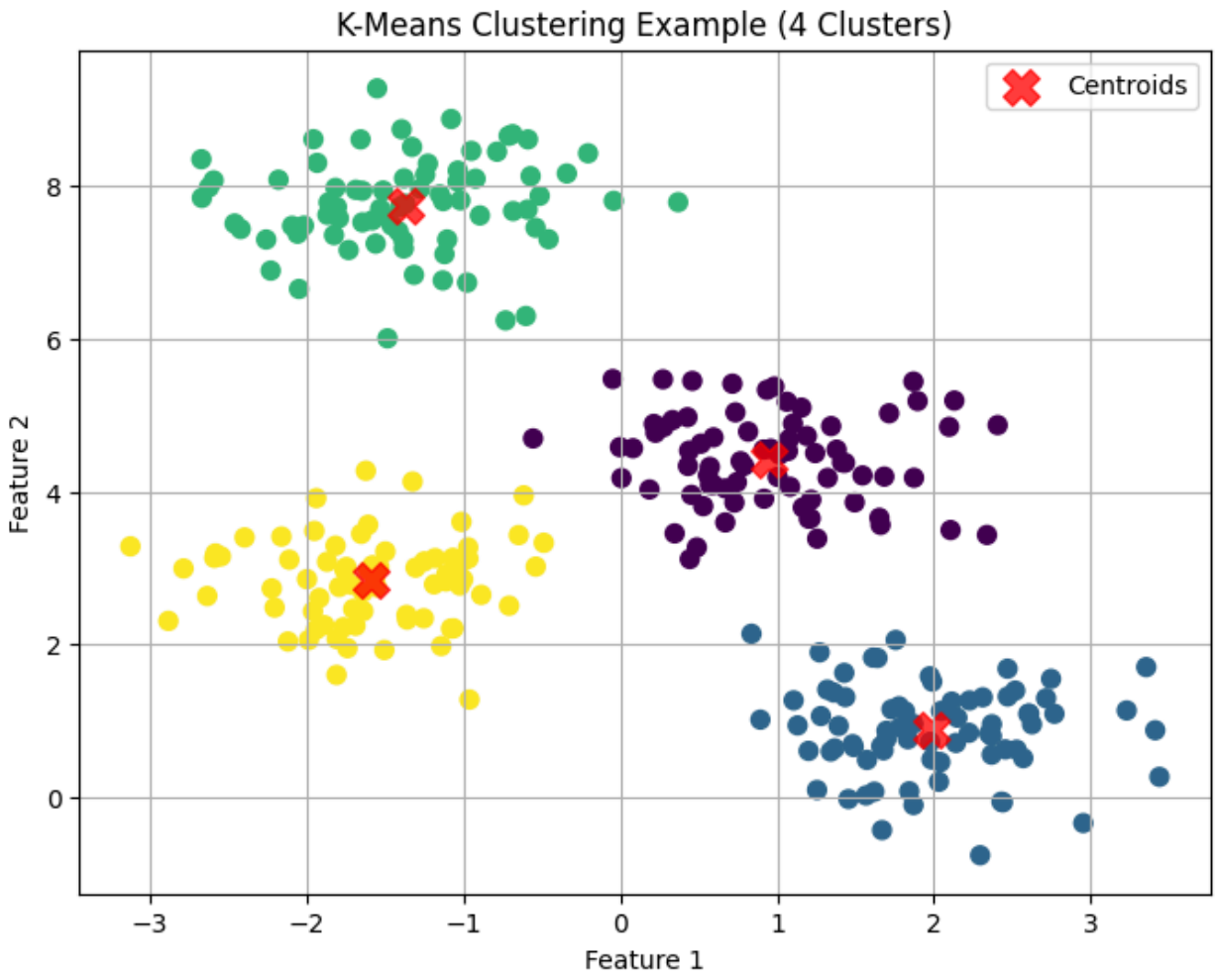
Unsupervised Learning: Clustering: k-means clustering algorithm, Improving cluster performance with post processing, Bisecting k-means, Example: clustering points on a map.

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans

# Generate synthetic data
X, y_true = make_blobs(n_samples=300, centers=4, cluster_std=0.60,
random_state=0)

# Fit KMeans model
kmeans = KMeans(n_clusters=4, random_state=0)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

# Plot the clusters and centroids
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75,
marker='X', label='Centroids')
plt.title('K-Means Clustering Example (4 Clusters)')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.grid(True)
plt.show()
```



1. Introduction to Unsupervised Learning

Definition: Unsupervised learning is a type of machine learning that deals with data without labeled responses. The algorithm tries to find patterns, groupings, or structures in the data.

Applications: Market segmentation, document categorization, customer clustering, image compression.

2. Clustering

Definition: Clustering is the task of dividing a set of data points into groups (clusters) such that points in the same group are more similar to each other than to those in other groups.

Types of Clustering Algorithms:

Partition-based (e.g., K-Means)

Hierarchical

Density-based

Model-based

3. K-Means Clustering Algorithm

Goal: Partition data into k clusters, where k is a user-defined parameter.

Steps:

1. Initialize: Choose k initial centroids randomly.
2. Assign: Assign each point to the nearest centroid based on Euclidean distance.
3. Update: Calculate new centroids by taking the mean of points in each cluster.
4. Repeat steps 2 and 3 until centroids do not change or maximum iterations are reached.

Distance Metric: Most commonly, Euclidean distance is used.

Output: A set of k clusters with minimized intra-cluster distances.

4. Limitations of K-Means

A. Requires the number of clusters (k) to be specified in advance.
B. Sensitive to the initial placement of centroids.
C. Assumes spherical clusters and similar cluster sizes.

D. Poor performance with non-globular shapes or outliers.

5. Improving Cluster Performance with Post-Processing

Techniques:

Re-run K-means multiple times with different initial centroids and choose the best result (lowest inertia).

Silhouette Analysis: Evaluate how well each point lies within its cluster.

Elbow Method: Helps to determine the optimal number of clusters.

Principal Component Analysis (PCA): Reduces dimensionality to improve visualization and performance.

Outlier Removal: Helps to improve cluster purity.

Cluster Evaluation Metrics:

Inertia (within-cluster sum-of-squares)

Silhouette score

Davies–Bouldin index

6. Bisecting K-Means

Definition: A hierarchical variant of k-means that improves clustering quality.

Algorithm:

1. Start with all data in one cluster.
2. Split the cluster into two using basic K-means (k=2).
3. Choose the split that gives the best separation (e.g., lowest total SSE – sum of squared errors).
4. Repeat step 2 on the cluster with the highest SSE until the desired number of clusters is reached.

Advantages:

Better clustering than standard K-means in many scenarios.

Automatically determines better initial centroids by recursive division.

7. Example: Clustering Points on a Map

Scenario:
Suppose you have the GPS coordinates of several restaurants spread across a city.

Objective:
To group these restaurants into **four distinct geographic regions (k = 4)** based on their location.

Approach:

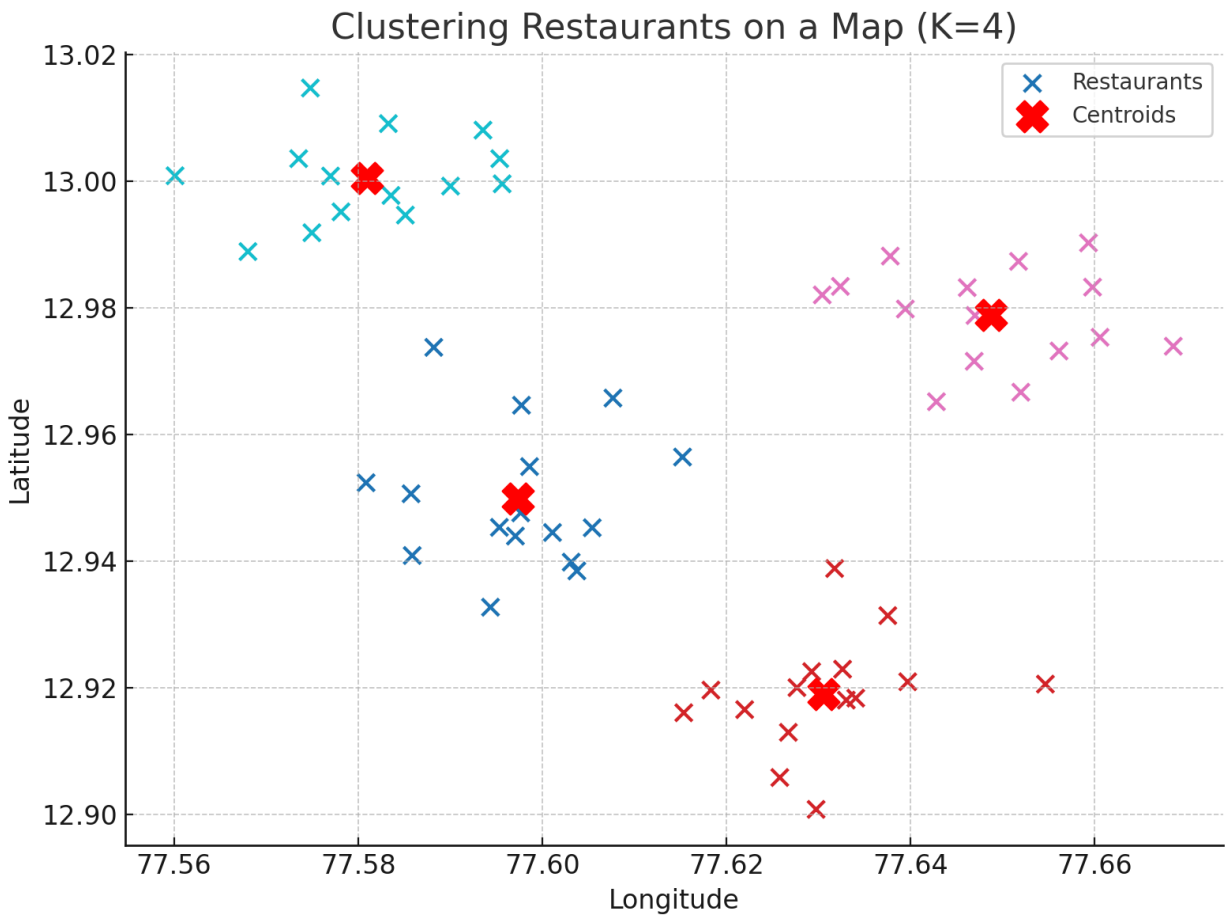
- Apply the **K-Means clustering algorithm** to the latitude and longitude data of the restaurants.
- The algorithm will automatically partition the data into four clusters by minimizing the distance of each point to the cluster centroid.

Applications:

- **Delivery optimization**: Assign delivery agents or hubs by zone.
- **Customer segmentation**: Understand regional customer preferences.
- **Market analysis**: Identify potential areas for new restaurant outlets based on clustering patterns.

Visualization:

- Use a **2D scatter plot or map** where each point (restaurant) is colored based on its assigned cluster.
- Cluster centroids can be shown as larger or distinct markers (e.g., red 'X') to indicate regional centers.



clustering GPS coordinates of restaurants into **4 geographic zones** using the K-Means algorithm. Each color represents a different cluster, and the red **'X'** markers indicate the cluster centroids.

Use Case Recap: Clustering Points on a Map

- **Dataset**: Simulated GPS-like data of 60 restaurants.
- **Clusters**: 4 distinct regions grouped based on spatial proximity.
- **Applications**:
    - Designing delivery zones
    - Planning regional marketing strategies
    - Visualizing density and spread of food outlets

📊 Data Points

We have the following 6 points in 2D space:

Point	x	y
A	1	2
B	1	4
C	1	0
D	10	2
E	10	4
F	10	0

Let's apply **K-means** with **k = 2** (we want to find 2 clusters).

Let's apply **K-means** with **k = 2** (we want to find 2 clusters).

🔁 Step 1: Initialization

Choose any two points as **initial centroids** (randomly or manually). Let's pick:

- **Centroid 1 (C1)** = A (1, 2)
- **Centroid 2 (C2)** = D (10, 2)

📐 Step 2: Assign Points to the Nearest Centroid (Using Euclidean Distance)

We compute the distance of each point to C1 and C2:

Distance Formula:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Point	Distance to C1 (1,2)	Distance to C2 (10,2)	Nearest
A (1,2)	0.0	9.0	C1
B (1,4)	2.0	9.22	C1
C (1,0)	2.0	9.22	C1
D (10,2)	9.0	0.0	C2
E (10,4)	9.22	2.0	C2
F (10,0)	9.22	2.0	C2

✅ **Cluster 1**: A, B, C

✅ **Cluster 2**: D, E, F

🔄 Step 3: Compute New Centroids

Take the mean of points in each cluster:

Cluster 1 (A, B, C):

- $\bar{x}$ = (1+1+1)/3 = 1

- $\bar{y}$ = (2+4+0)/3 = 2
  ➡️ New C1 = (1, 2)

Cluster 2 (D, E, F):

- $\bar{x}$ = (10+10+10)/3 = 10
- $\bar{y}$ = (2+4+0)/3 = 2
  ➡️ New C2 = (10, 2)

💡 New centroids are the same as before → **Convergence achieved**!

Final Clusters

- **Cluster 1**: A, B, C
- **Cluster 2**: D, E, F