Lesson Plan: Clustering and Its Types

⏱️ **Total Time: 60 minutes**

🎯 **Objective:**

By the end of the lesson, students will be able to:

- Define clustering and its purpose
- Understand the key types of clustering
- Differentiate between K-means, Hierarchical, and DBSCAN clustering
- Apply K-means clustering using a simple dataset

⬛ **Lesson Breakdown**

0:00 – 0:10 (10 mins) — Introduction to Clustering

Objectives:

- Define clustering
- Understand its role in unsupervised learning

Activities:

- Begin with a real-world analogy: e.g., grouping news articles or customer segmentation
- Define clustering:
    - *"Clustering is the task of grouping a set of objects such that objects in the same group (cluster) are more similar to each other than to those in other groups."*
- Show example applications (marketing, image compression, etc.)

Materials:

- Slides with visuals
- Whiteboard or virtual board for brainstorming ideas

0:10 – 0:20 (10 mins) — Types of Clustering Algorithms

Objectives:

- Introduce the main categories of clustering

Types to Cover:

1. **Partitioning Methods** (e.g., K-means)
2. **Hierarchical Clustering**

3. **Density-Based Clustering** (e.g., DBSCAN)
4. (Optional mention: Grid-based, Model-based for awareness)

Activities:

- Quick summary of each with diagrams
- Discuss when each method is useful

Visuals:

- Cluster shapes and behavior for each type

0:20 – 0:35 (15 mins) — Deep Dive: K-Means Clustering

Objectives:

- Understand how K-means works
- Learn the algorithm's steps

Topics:

- Initialization (random centroids)
- Assignment step
- Update step
- Convergence

Activities:

- Step-by-step animation or visual walkthrough
- Discuss pros and cons

0:35 – 0:50 (15 mins) — Practical Demonstration (K-Means)

Objectives:

- Apply K-means on a dataset
- Visualize clusters

Tools:

- Python (Jupyter Notebook or Google Colab)
- Libraries: sklearn, matplotlib, pandas

Example Dataset:

- Iris dataset or a 2D synthetic dataset

Code Snippet (Optional):

python

CopyEdit

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

X, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.6, random_state=0)
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red')
plt.title('K-Means Clustering')
plt.show()
```

0:50 – 0:55 (5 mins) — Overview of Other Methods

Quick Comparisons:

- **Hierarchical:**
    - Agglomerative vs Divisive
    - Produces dendrograms
- **DBSCAN:**
    - Density-based
    - Finds arbitrarily shaped clusters
    - Can detect noise/outliers

When to Use What:

Method	**Best For**	**Drawback**
K-means	Spherical clusters	Requires k
Hierarchical	Small datasets	Memory intensive
DBSCAN	Irregular shapes, noise	Sensitive to parameters

0:55 – 1:00 (5 mins) — Q&A + Recap

Activities:

- Recap main points
- Ask: “What type of clustering would you use for [scenario]?”
- Quick quiz (3 questions) or interactive poll