Day 12 – Hierarchical Clustering (Unsupervised Learning)

📓 Introduction

On Day 12 of the training, we focused on **Hierarchical Clustering**, one of the core techniques in **unsupervised learning**. Clustering is a method used to group data points such that items in the same group (or cluster) are more similar to each other than to those in other groups.

Hierarchical clustering is especially useful in datasets where we want to discover **nested structures**, such as customer segmentation, gene expression patterns, document classification, etc.

We implemented hierarchical clustering on the **Mall Customer dataset** and visualized customer clusters based on their **Annual Income** and **Spending Score**.

□ Understanding Hierarchical Clustering

Hierarchical Clustering builds a **tree-like structure** of clusters known as a **dendrogram**, which shows how data points can be merged into clusters.

There are two types of hierarchical clustering:

- **Agglomerative Clustering (Bottom-Up):**
  Each data point starts as its own cluster and merges with the closest clusters step by step.
- **Divisive Clustering (Top-Down):**
  Starts with one cluster and splits it recursively.

In our session, we focused on **Agglomerative Clustering**, which is more commonly used.

📌 Important Concepts:

- **Linkage Methods:** Determine how distance is calculated between clusters (e.g., single, complete, average, ward).
- **Dendrogram:** A visual representation showing how clusters merge at each step.
- **Affinity:** The metric used to compute the distance (e.g., Euclidean, Manhattan).

💼 Dataset Overview: Mall_Customers.csv

The dataset contains information about 200 customers such as:

- Customer ID
- Gender
- Age

- Annual Income (in thousand dollars)
- Spending Score (1–100)

For clustering, we selected **Annual Income** and **Spending Score** to discover groups of similar customers.

🛠️ Step-by-Step Implementation

✅ 1. Importing Required Libraries

We began by importing the essential libraries for data processing, clustering, and visualization.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
```

✅ 2. Loading and Preparing Data

```
data = pd.read_csv("Mall_Customers.csv")
X = data.iloc[:, [3, 4]].values  # Selecting Annual Income and Spending Score
```

We used only **Annual Income** and **Spending Score** columns to perform clustering. These features allow us to understand spending behavior relative to income.

🌳 Creating a Dendrogram

A **dendrogram** helps in determining the optimal number of clusters before applying the algorithm.

```
plt.figure(figsize=(10, 7))
dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean Distance")
plt.show()
```

The **elbow point** in the dendrogram suggested using **5 clusters**.

The **Ward method** minimizes the total within-cluster variance and provides well-formed hierarchical structures.

🔎 Agglomerative Clustering

After analyzing the dendrogram, we applied Agglomerative Clustering with `n_clusters=5`.

```
hc = AgglomerativeClustering(n_clusters=5, affinity='euclidean',
linkage='ward')
y_hc = hc.fit_predict(X)
```

Here,

- `n_clusters` = 5 based on dendrogram
- `affinity='euclidean'` is the metric used to compute the linkage
- `linkage='ward'` minimizes variance between clusters

🎨 Cluster Visualization

Each cluster was plotted using a different color to understand their distribution.

```
plt.figure(figsize=(8,6))
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label
= 'Cluster 1')
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label
= 'Cluster 2')
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green',
label = 'Cluster 3')
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label
= 'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta',
label = 'Cluster 5')

plt.title('Clusters of Mall Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Each cluster showed a different grouping of customers based on spending behavior and income levels.

📊 Observations from Clustering

- **Cluster 1**: High income, high spending → Likely premium customers
- **Cluster 2**: Low income, low spending → Budget-sensitive customers
- **Cluster 3**: Moderate income and spending → Average buyers
- **Cluster 4**: High income, low spending → Possibly conservative spenders
- **Cluster 5**: Low income, high spending → Young or impulsive buyers

These insights help businesses with **targeted marketing**, **loyalty programs**, and **personalized services**.

✅ Advantages of Hierarchical Clustering

- No need to specify the number of clusters in advance
- Produces interpretable tree structure (dendrogram)
- Works well on small to medium-sized datasets

📝 Conclusion

- Learned how to implement **Hierarchical Clustering** using **Agglomerative approach**.
- Used the **Dendrogram** to determine optimal cluster count.
- Visualized customer segmentation based on **spending score and income**.
- Understood how hierarchical clustering provides deeper insights compared to K-Means when **nested grouping or tree-based understanding** is required.

This session strengthened our understanding of **unsupervised learning** and how clustering can be applied in **real-world business analytics**.