Lab No 11

Unsupervised Learning & Clustering

Name Maaz Ahmad

Reg No B23F0722AI170

Section Ai Yellow

Submitted to Sir Abdullah Sajid

Task no 1

Data Loading & Preprocessing

Goal: Load the dataset (Iris) and scale the features. Scaling is mandatory for clustering.

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler

# 1. Load Dataset
data = load_iris()
X = data.data
# We don't load 'y' (target) because this is Unsupervised Learning

# 2. Scale Features
# Clustering relies on distance, so features must be on the same scale
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

print("Data Loaded and Scaled.")
print("Shape:", X_scaled.shape)
```

```
Data Loaded and Scaled.
Shape: (150, 4)
```

Description

Hum ne load_iris se data import kiya. Kyun ke yeh Unsupervised Learning hai, humein target labels (y) ki zaroorat nahi hai. Phir hum ne StandardScaler use kiya taake features ko normalize karein.
Agar hum scale nahi karein ge, toh baray numbers (jaise 1000) chotay numbers (jaise 0.1) par haavi ho jayen ge aur distance calculations ghalat ho jayen gi.

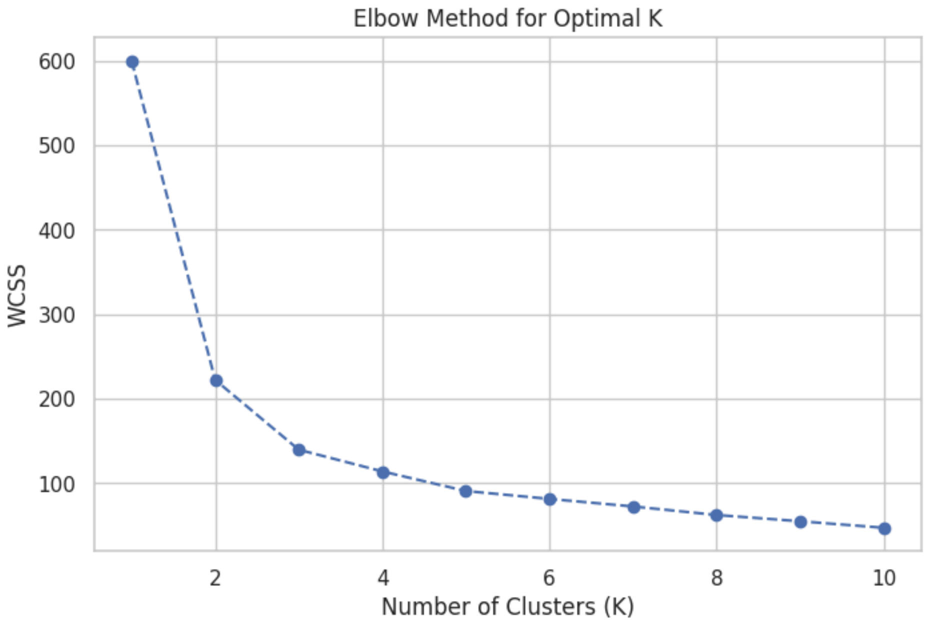
Task no 2

K-Means - The Elbow Method

Goal: Determine the optimal number of clusters (k) by plotting the WCSS (Within-Cluster Sum of Squares).

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

# 1. Calculate WCSS for k=1 to k=10
wcss = []
```

```
K_range = range(1, 11)

for k in K_range:
    # n_init=10 means the algorithm will run 10 times with different centroids
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# 2. Plot the Elbow Graph
plt.figure(figsize=(8, 5))
plt.plot(K_range, wcss, marker='o', linestyle='--', color='b')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()
```



Description

Hum ne 1 se 10 tak loops chala kar WCSS (Error) calculate kiya. Har K ke liye model train kiya aur check kiya ke points apnay cluster center se kitnay door hain. Graph main jahan curve "Kohni" (Elbow) ki tarah mudta hai, wo optimal K hota hai. Is graph main K=3 best nazar aa raha hai.

Task no 3

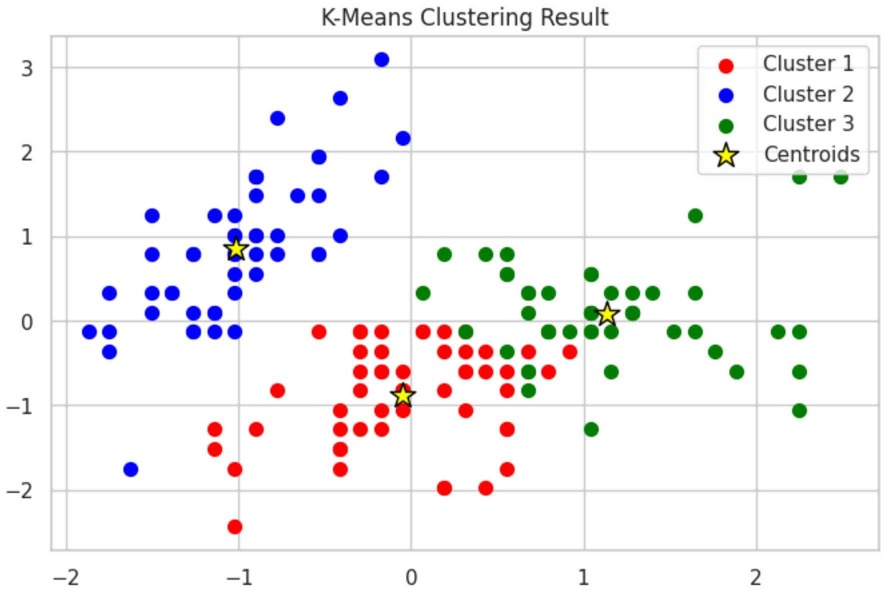
Train K-Means Model & Visualization

Goal: Train the model with the chosen k (3) and visualize the clusters.

```
import seaborn as sns
from sklearn.metrics import silhouette_score

# 1. Train Model with K=3
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
y_kmeans = kmeans.fit_predict(X_scaled)

# 2. Evaluation
score = silhouette_score(X_scaled, y_kmeans)
print(f"Silhouette Score: {score:.3f}")

# 3. Visualization (Using first 2 features)
plt.figure(figsize=(8, 5))
# Plot points for each cluster
```

```
plt.scatter(X_scaled[y_kmeans == 0, 0], X_scaled[y_kmeans == 0, 1], s=50, c='red', label='Cluster 1')
plt.scatter(X_scaled[y_kmeans == 1, 0], X_scaled[y_kmeans == 1, 1], s=50, c='blue', label='Cluster 2')
plt.scatter(X_scaled[y_kmeans == 2, 0], X_scaled[y_kmeans == 2, 1], s=50, c='green', label='Cluster 3')

# Plot Centroids
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
            s=200, c='yellow', marker='*', label='Centroids', edgecolor='black')
plt.title('K-Means Clustering Result')
plt.legend()
plt.show()
```

```
Silhouette Score: 0.460
```



Description

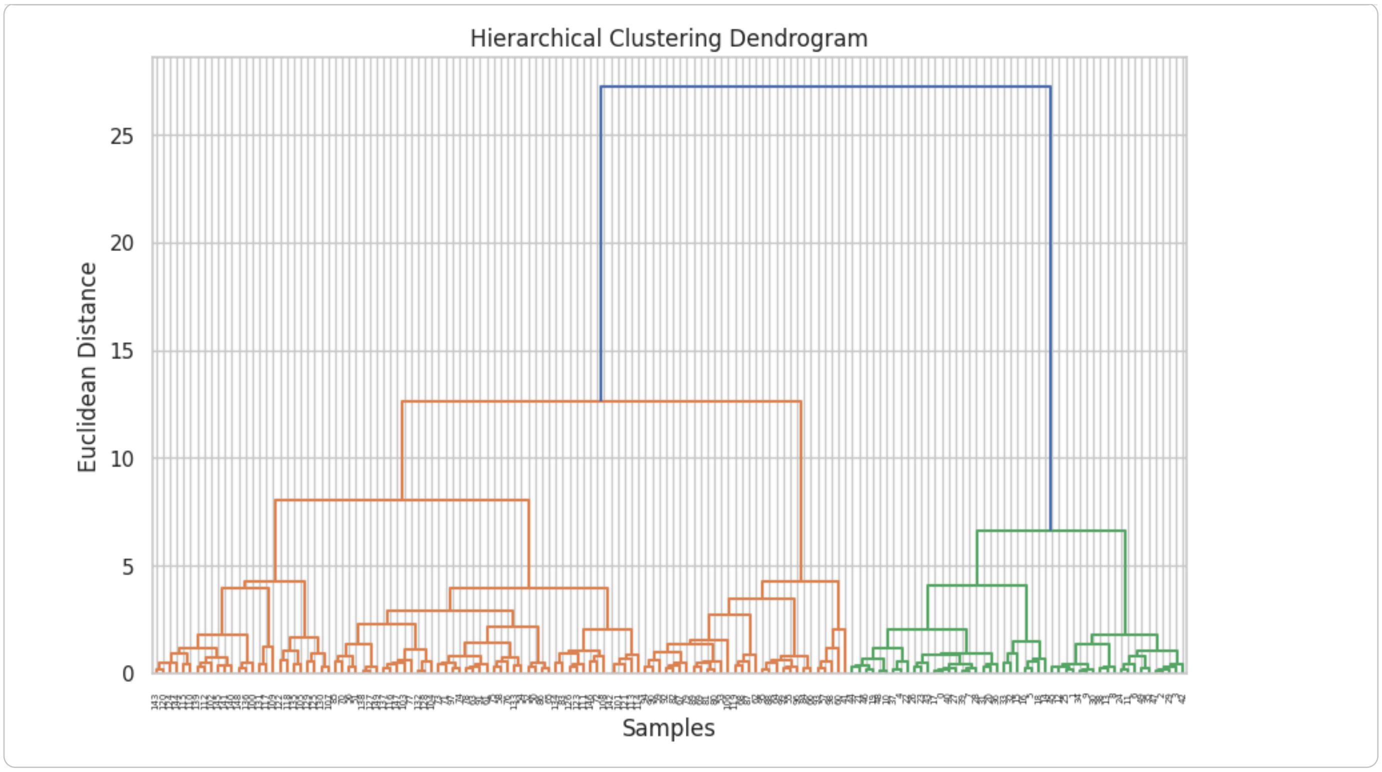
Hum ne K=3 select kar ke final K-Means model train kiya. fit_predict function ne har data point ko aik cluster assign kar diya. Phir hum ne Scatter Plot banaya jis main Yellow Stars Centroids ko show kar rahay hain. Silhouette Score humein batata hai ke clusters kitnay distinct (alag) hain; score jitna 1 ke qareeb ho utna acha hai.

Task no 4

Hierarchical Clustering (Dendrogram)

Goal: Plot a Dendrogram to understand the hierarchical relationship between data points.

```
from scipy.cluster.hierarchy import dendrogram, linkage

# 1. Calculate Linkage Matrix (Ward method minimizes variance)
linked = linkage(X_scaled, method='ward')

# 2. Plot Dendrogram
plt.figure(figsize=(10, 6))
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=True)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Samples')
plt.ylabel('Euclidean Distance')
plt.show()
```



Description

Is task main hum ne linkage function use kar ke data points ka hierarchy (shajra) banaya. Dendrogram aik tree diagram hai jo dikhata hai ke points kis tarah merge ho rahay hain. Vertical lines ki lambai (height) batati hai ke clusters aik dusray se kitnay different hain. Agar hum tree ko aik specific height par cut karein toh humein clusters mil jatay hain.

Task no 5

Agglomerative Clustering

Goal: Train the Hierarchical model using Agglomerative Clustering.

```
from sklearn.cluster import AgglomerativeClustering

# 1. Train Hierarchical Model
# We verify K=3 from dendrogram
hc = AgglomerativeClustering(n_clusters=3, linkage='ward')
y_hc = hc.fit_predict(X_scaled)

# 2. Check Labels
print("First 20 Labels from Hierarchical Clustering:")
print(y_hc[:20])

# Optional: You can plot these labels similarly to Task 3 to compare.
```

```
First 20 Labels from Hierarchical Clustering:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

Description:

Hum ne AgglomerativeClustering use kiya jo ke "Bottom-Up" approach hai. Shuru main har point aik alag cluster hota hai, phir qareebi points merge hotay jatay hain jab tak humein 3 final clusters nahi mil jatay. Yeh K-Means se mukhtalif hai kyun ke is main random centroids ka masla nahi hota.

Lab No 11 Summary

1. **Lab Objective:** The main goal of this lab was to understand Unsupervised Learning, where we find hidden patterns and structures in data without having any labeled target variable (y).
2. **K-Means Algorithm:** We implemented K-Means Clustering, which groups data points into K distinct clusters by assigning each point to the nearest centroid (center of the cluster).
3. **The Elbow Method:** To avoid guessing the number of clusters, we used the Elbow Method. By plotting the WCSS (error) against different values of K, we found that the optimal number of clusters for the Iris dataset is 3.
4. **Evaluation Metrics:** We used the Silhouette Score to evaluate the quality of our clustering. A score closer to 1 indicates that the clusters are well-separated and distinct.
5. **Hierarchical Clustering:** We also explored Agglomerative Clustering (a type of Hierarchical Clustering), which builds clusters from the bottom up by merging similar points step-by-step.
6. **Dendrogram Analysis:** A Dendrogram (tree diagram) was plotted to visualize the hierarchical relationship between data points. Cutting this tree at a specific height allows us to determine the number of clusters visually.
7. **Feature Scaling:** A critical step in both algorithms was using StandardScaler. Since clustering relies on distance calculations (like Euclidean distance), scaling ensures that features with larger numeric ranges do not bias the results.