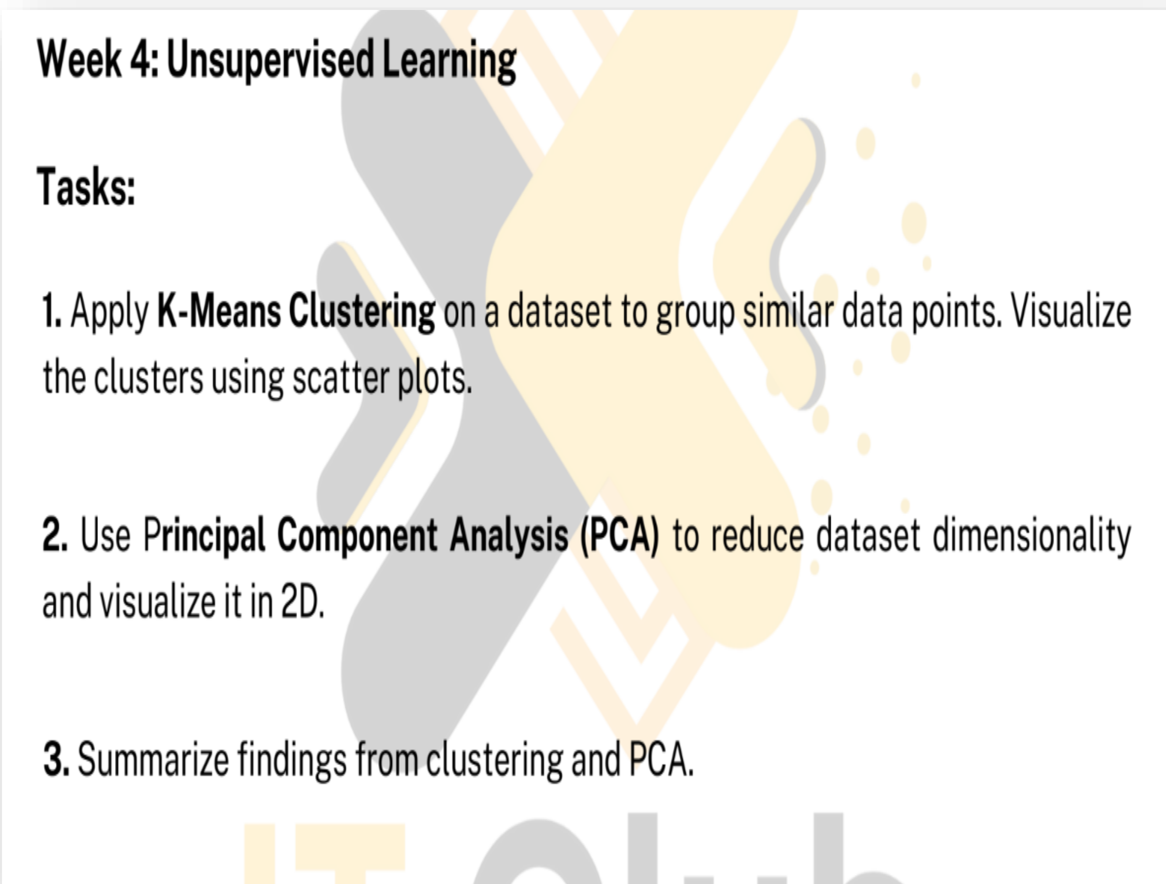
# ITCP-25/AI-003 (Noor Fatima)

## Week 4: Unsupervised Learning

Tasks:

I was assigned the task of implementing **K-Means Clustering** and **Principal Component Analysis (PCA)** for unsupervised learning. This task involves:

1. Applying **K-Means Clustering** on a dataset to group similar data points and visualizing the clusters using scatter plots.
2. Using **Principal Component Analysis (PCA)** to reduce dataset dimensionality and visualizing it in 2D.
3. Summarizing findings from clustering and PCA.



# Task 1:

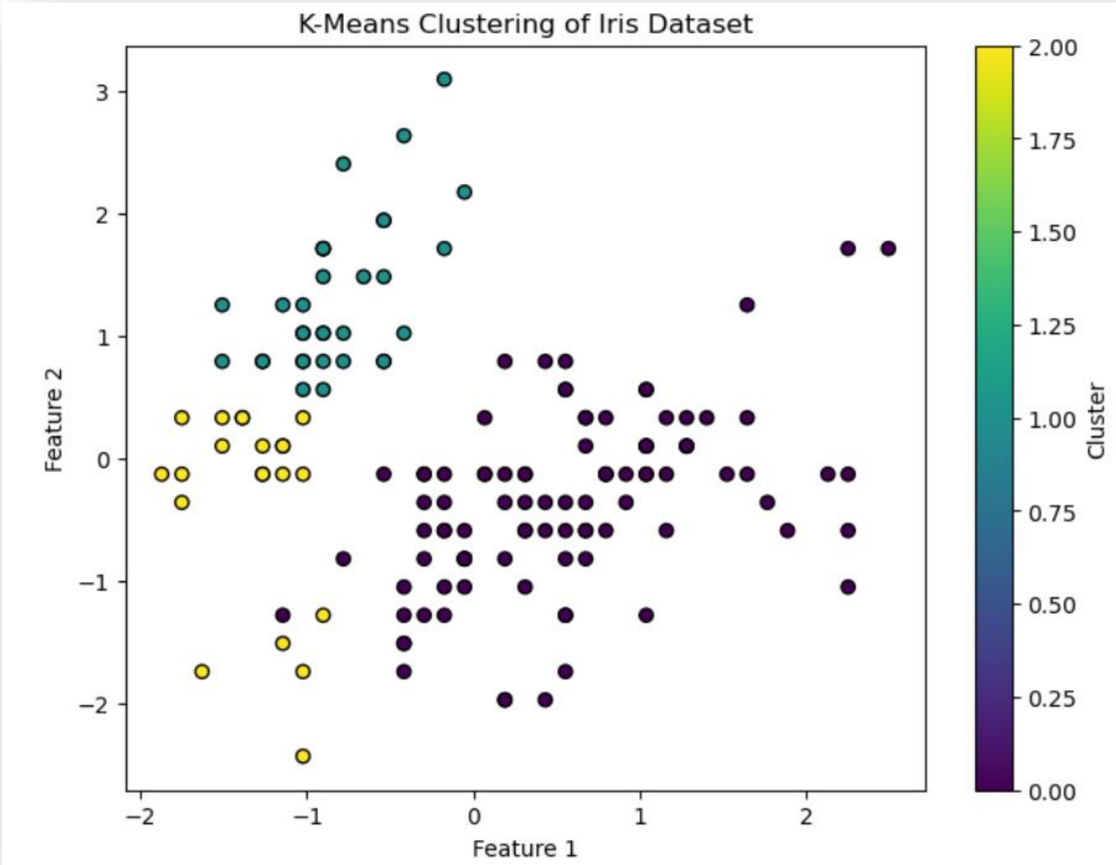
Splitting the Dataset into Training and Testing Subsets

1. K-Means Clustering is used to group similar data points into K clusters. The number of clusters K is chosen based on the Elbow Method.

## Code:

```
# Load the Iris dataset

iris = sns.load_dataset("iris")

# Selecting only numerical features for clustering

features = iris.drop(columns=["species"])

# Standardizing the data

scaler = StandardScaler()

scaled_features = scaler.fit_transform(features)

# Applying K-Means clustering

kmeans = KMeans(n_clusters=3, random_state=42)

iris["cluster"] = kmeans.fit_predict(scaled_features)

# Visualizing clusters using a scatter plot

plt.figure(figsize=(8, 6))

plt.scatter(scaled_features[:, 0], scaled_features[:, 1], c=iris["cluster"], cmap="viridis", edgecolor="k")

plt.xlabel("Feature 1")

plt.ylabel("Feature 2")

plt.title("K-Means Clustering of Iris Dataset")

plt.colorbar(label="Cluster")

plt.show()
```

## Output:




# Task 2:

Applying Principal Component Analysis (PCA)

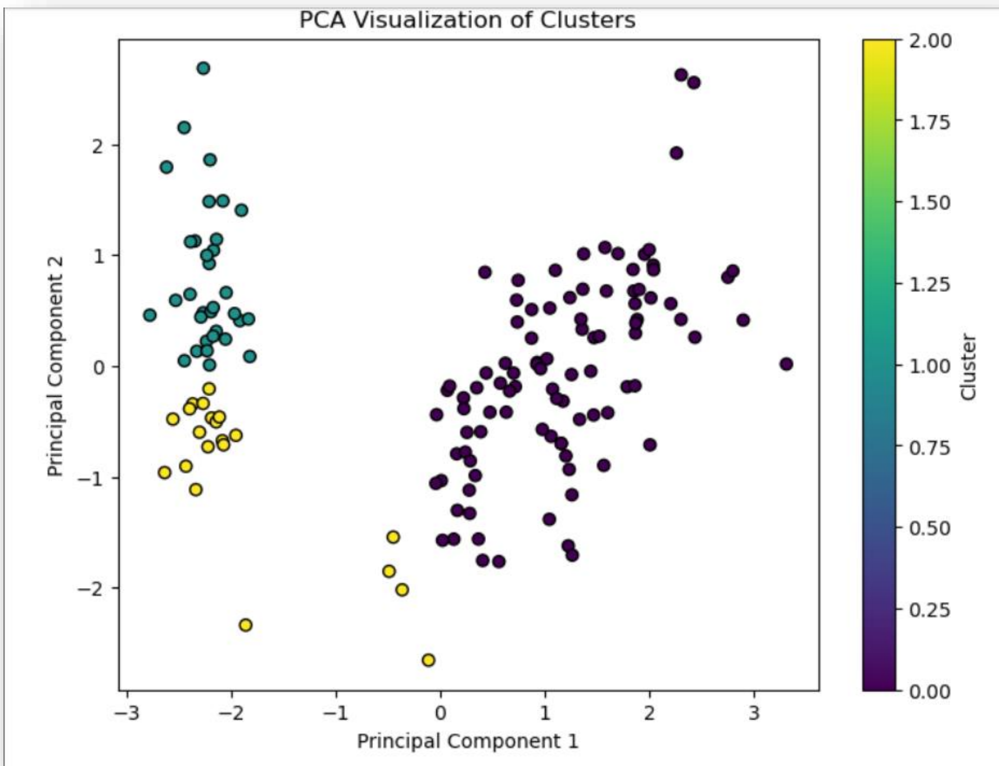
PCA is used to reduce the dataset's dimensionality while preserving the variance in the data. This helps in visualizing high-dimensional data in **2D**.

## Code:

```
# Applying PCA to reduce to 2 dimensions

pca = PCA(n_components=2)

pca_features = pca.fit_transform(scaled_features)

# Creating a DataFrame with PCA results

iris_pca = pd.DataFrame(pca_features, columns=["PC1", "PC2"])
```

```
iris_pca["cluster"] = iris["cluster"]
# Visualizing PCA results
plt.figure(figsize=(8, 6))
plt.scatter(iris_pca["PC1"], iris_pca["PC2"], c=iris_pca["cluster"], cmap="viridis", edgecolor="k")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.title("PCA Visualization of Clusters")
plt.colorbar(label="Cluster")
plt.show()
```

## Output:



# Task 3:

Summary of Findings

- **K-Means Clustering** successfully grouped the dataset into **three distinct clusters**, representing different species in the Iris dataset.
- The **PCA visualization** reduced the high-dimensional dataset to **2D**, while still maintaining clear cluster separations.
- The **clusters formed in PCA visualization matched** the clusters from K-Means, confirming that **dimensionality reduction** preserved the essential structure of the data.

## Conclusion

- Successfully applied **K-Means Clustering** to classify similar data points.
- Used **PCA** to reduce dataset dimensionality for better visualization.
- The clustering results were **consistent** across both methods, demonstrating the effectiveness of unsupervised learning in identifying patterns in data.