TP 03: Clustering with K-means (scikit-learn)

Objective

The objective of this practical session is to introduce **unsupervised learning** through the **K-means clustering algorithm**. Students will learn how to:

- Understand clustering principles
- Apply K-means on a dataset
- Choose the number of clusters
- Visualize and interpret results

Part 1: Theoretical Questions

1. What is **unsupervised learning**?
2. Define **clustering**.
3. What is the objective of the **K-means algorithm**?
4. Explain the concept of a **centroid**.
5. What is the role of the parameter **K**?

Part 2: Practical Exercise – Customer Segmentation

Problem Description

We want to segment customers based on their:

- Annual income
- Spending score

The goal is to group similar customers into clusters without predefined labels.

Step 1: Import Libraries

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

Step 2: Create the Dataset

```
# Example data: [Income, Spending Score]
X = np.array([
    [15, 39], [16, 81], [17, 6], [18, 77], [19, 40],
    [20, 76], [21, 6], [22, 94], [23, 3], [24, 72],
    [25, 14], [26, 82], [27, 32], [28, 61], [29, 15]
])
```

Step 3: Visualize the Data

```
plt.scatter(X[:, 0], X[:, 1])
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Customer Data")
plt.show()
```

Step 4: Apply K-means

```
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(X)

labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

Step 5: Visualize Clusters

```
plt.scatter(X[:, 0], X[:, 1], c=labels)
plt.scatter(centroids[:, 0], centroids[:, 1], marker='X')
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("K-means Clustering")
plt.show()
```

Step 6: Choosing the Optimal K (Elbow Method)

```
inertia = []

for k in range(1, 10):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)
```

```
plt.plot(range(1, 10), inertia, marker='o')
plt.xlabel("Number of Clusters (K)")
plt.ylabel("Inertia")
plt.title("Elbow Method")
plt.show()
```