Prediction Using Unsupervised ML

(Level - Beginner)

Author: M. Roohi Jahan

From the given 'Iris' dataset, predict the optimum number of clusters and represent it visually.

I have utilised Python to perform this task

Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

Load the Iris dataset

```
iris = load_iris()
X = iris.data  # Using the features of the Iris dataset
```

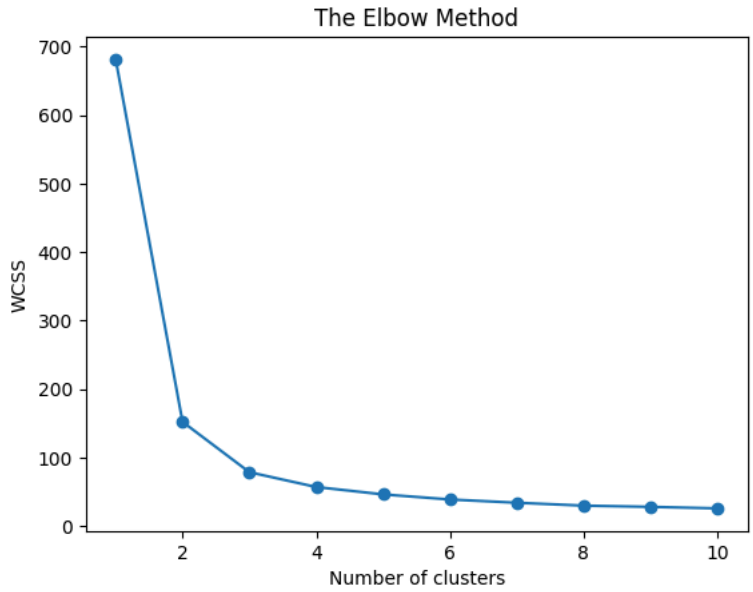
Determine the optimum number of clusters using the Elbow method

```
wcss = []  # within-cluster sum of squares
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
```

Plot the K-Means

```
plt.plot(range(1, 11), wcss, marker='o')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



Choose the optimum number of clusters (3 in this case)

```
optimum_clusters = 3
```

Apply K-Means clustering

```
kmeans = KMeans(n_clusters=optimum_clusters, init='k-means++', random_state=0)
y_kmeans = kmeans.fit_predict(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil
  warnings.warn(
```

Visualize the clusters using PCA

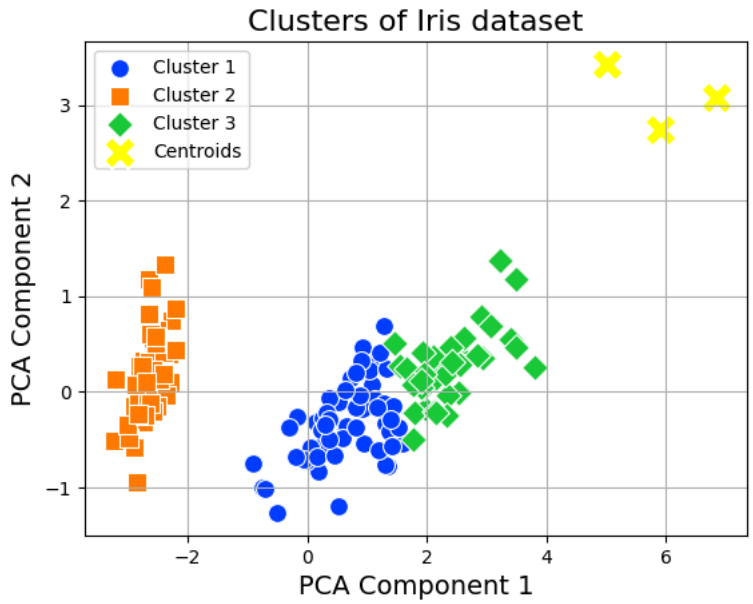
```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
```

Setting up a colour pallete

```
palette = sns.color_palette("bright", 3)

plt.figure(figsize=(10, 6))
```

```
<Figure size 1000x600 with 0 Axes>
<Figure size 1000x600 with 0 Axes>
```

```
sns.scatterplot(x=X_pca[y_kmeans == 0, 0], y=X_pca[y_kmeans == 0, 1],
                s=100, color=palette[0], label='Cluster 1', marker='o')
sns.scatterplot(x=X_pca[y_kmeans == 1, 0], y=X_pca[y_kmeans == 1, 1],
                s=100, color=palette[1], label='Cluster 2', marker='s')
sns.scatterplot(x=X_pca[y_kmeans == 2, 0], y=X_pca[y_kmeans == 2, 1],
                s=100, color=palette[2], label='Cluster 3', marker='D')

# Plotting the centroids
sns.scatterplot(x=kmeans.cluster_centers_[:, 0], y=kmeans.cluster_centers_[:, 1],
                s=300, color='yellow', label='Centroids', marker='X')

plt.title('Clusters of Iris dataset', fontsize=16)
plt.xlabel('PCA Component 1', fontsize=14)
plt.ylabel('PCA Component 2', fontsize=14)
plt.legend()
plt.grid(True)
plt.show()
```



Conclusion

Optimal Number of Clusters:

1. The Elbow plot indicates that 3 is the optimal number of clusters for the Iris dataset.

Cluster Visualization:

1. The scatter plot of the PCA-transformed data shows the three clusters visually separated.
2. The centroids of the clusters are marked, indicating the center of each cluster.

Data Separation:

1. The plot demonstrates how the data points are grouped into clusters based on the K-Means algorithm.
2. It shows that the Iris dataset, which consists of three different species of iris flowers, can be effectively grouped into three clusters.

Key Insights:

1. The Iris dataset can be effectively clustered into three groups.
2. The Elbow method is a useful technique to determine the optimal number of clusters.
3. PCA helps in visualizing high-dimensional data in 2D, making it easier to understand clustering results.
4. The visualization confirms that the K-Means algorithm successfully identified the natural groupings in the dataset.