Assignment No. 5

Problem Statement: Perform K-Means Clustering algorithm on the **Iris dataset** to group flowers into clusters based on their features.

Objective: To explore unsupervised machine learning using K-Means Clustering. The task involves:

- Performing EDA and preprocessing on the Iris dataset
- Applying K-Means Clustering
- Visualizing clusters
- Analyzing clustering performance and insights

Prerequisite :

1. Python environment with libraries: pandas, numpy, matplotlib, seaborn, and scikit-learn
2. Understanding of unsupervised learning and clustering algorithms
3. Basic statistics, data preprocessing, and visualization techniques

Theory :

What is K-Means Clustering?

K-Means is an **unsupervised learning algorithm** that groups data into *k* clusters based on feature similarity. It iteratively assigns each data point to one of *k* clusters to minimize the **intra-cluster variance**.

Working Steps:

1. Choose the number of clusters (k)
2. Randomly initialize k centroids
3. Assign each point to the nearest centroid
4. Recompute centroids as the mean of points in each cluster
5. Repeat steps 3–4 until centroids no longer change significantly

Mathematical Intuition:

- Objective Function (to minimize):

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

Where:

- Ci = cluster
- μi = centroid of cluster
- xj = data point

Applications of K-Means:

- Customer segmentation in marketing
- Document or image clustering
- Pattern recognition
- Anomaly detection

Dataset Used: Iris Dataset

- Features:
  - Sepal Length
  - Sepal Width
  - Petal Length
  - Petal Width
- Target (for reference only): Species (Setosa, Versicolor, Virginica)

Choosing Optimal K – Elbow Method:

- Plot **Within-Cluster-Sum-of-Squares (WCSS)** against different values of *k*
- Identify the "elbow point" where adding more clusters doesn't significantly reduce WCSS

Code & Output

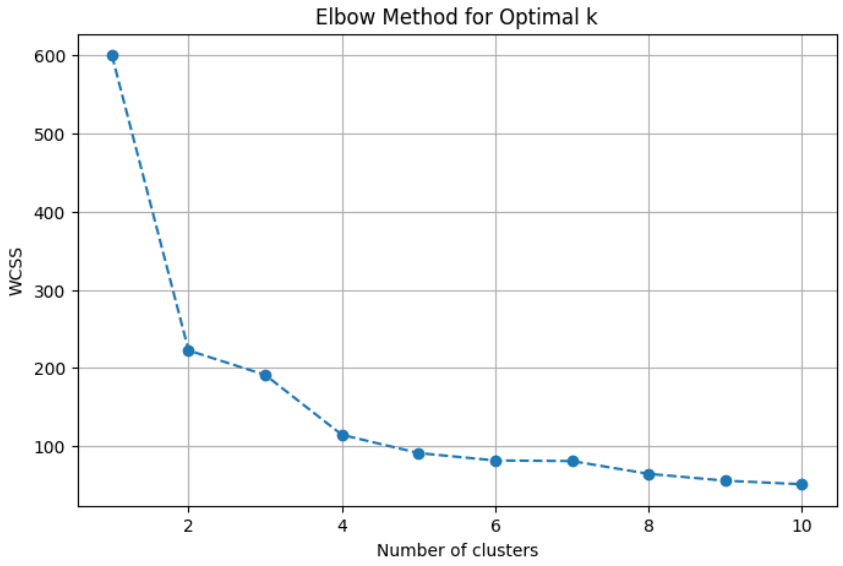
[10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

[11]:
```
# Step 1: Load the inbuilt Iris dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
```

[12]:
```
# Step 2: Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df)
```

[13]:
```
# Step 3: Elbow Method to choose optimal K
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)
```

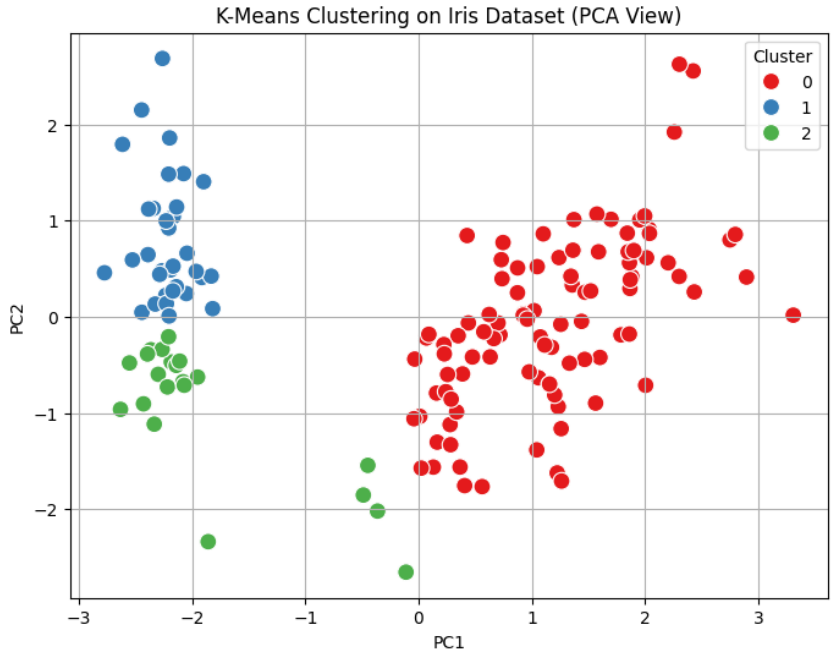
:
```
# Step 4: Plot the Elbow curve
plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title("Elbow Method for Optimal k")
plt.xlabel("Number of clusters")
plt.ylabel("WCSS")
plt.grid(True)
plt.show()
```



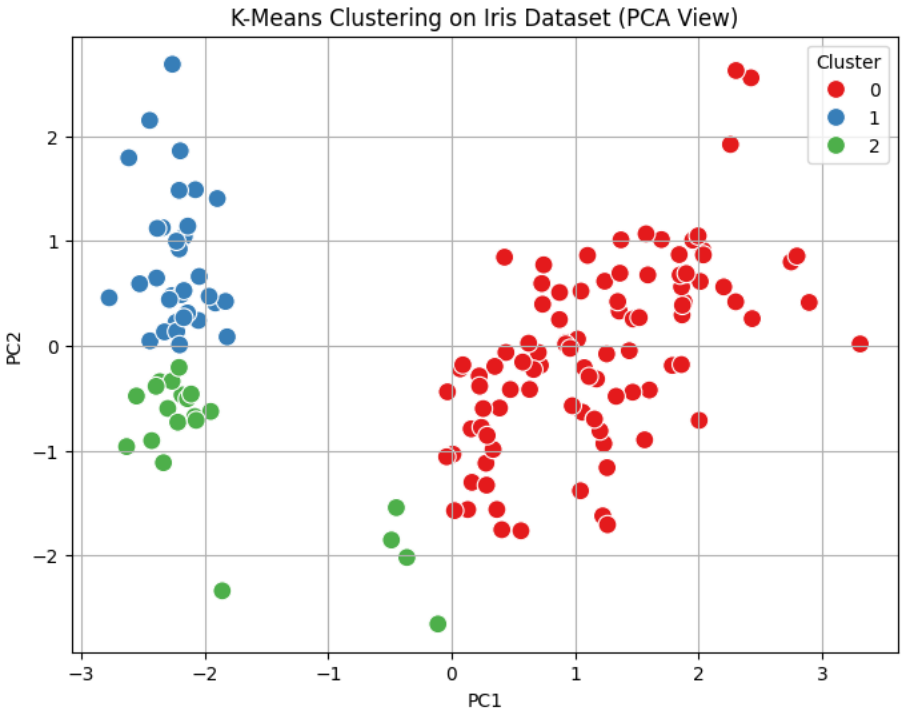
[15]:
```
# Step 5: Apply KMeans with optimal K (e.g., 3 for Iris)
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)
```

[16]:
```
# Step 6: Dimensionality reduction for visualization (PCA)
pca = PCA(n_components=2)
components = pca.fit_transform(X_scaled)
df['PC1'] = components[:, 0]
df['PC2'] = components[:, 1]
```

[17]:
```
# Step 7: Visualize the clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x='PC1', y='PC2', hue='Cluster', data=df, palette='Set1', s=100)
plt.title("K-Means Clustering on Iris Dataset (PCA View)")
plt.grid(True)
plt.show()
```




```
# Step 7: Visualize the clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x='PC1', y='PC2', hue='Cluster', data=df, palette='Set1', s=100)
plt.title("K-Means Clustering on Iris Dataset (PCA View)")
plt.grid(True)
plt.show()
```



Github : https://github.com/Vishalgodalkar/Machine-Learning

Conclusion:

K-Means clustering effectively grouped the Iris flowers into **three distinct clusters**, demonstrating the potential of unsupervised learning in identifying natural groupings within data. Despite not using target labels, the clustering aligned closely with actual species. Petal length and width were the most influential features in forming these clusters. The Elbow Method helped determine the optimal number of clusters (k=3). Visualizations confirmed well-separated clusters, especially for the Setosa class. This experiment demonstrated the strength of unsupervised learning in uncovering natural data patterns.