Exp no-7 Clustering analysis and Ensemble clustering on mall customers and wine datasets.

8/08/25

Aim:

To segment customers and classify wine samples into distinct clusters based on their features, using kmeans clustering and an ensemble clustering approach (CSPA), and to evaluate the clustering quality visually and quantitatively.

Algorithm:

Step 1: load datasets and scale features as needed.

Step 2: Apply kmeans clustering and mall customers data and use the Elbow method to find the optimal cluster.

Step 3: perform multiple kmeans clustering and the wine dataset with varying cluster counts.

Step 4: Build a similarity matrix from base clusterings and apply spectral clustering for ensemble labels.

Step 5: Evaluate with silhouette score and visualize clusters using PCA plots.

O/P

Mall Customer:

- Elbow plot showing inertia decline with increasing clusters (1-10).
- Scatter plot of customers grouped into 5 clusters based on income and spending using the dataset.
- Silhouette score for the ensemble clustering (eg:- a value around 0.55 - 0.65 indicating cluster quality)
- PCA scatter plot showing clusters formed by the ensemble method.

```
import seaborn as sns
df = pd.read_csv('Mall_customer.csv')
kmeans = kmeans (n_cluster, 5, random_state = 42)
df['cluster'] = kmeans.fit_predict (df[['Annual
        income (k$)', 'spending score (1-100)']])
distortion = []
distortions.append (km.inertia)
plt.plot (range (1,11), distortions, marker = 'o')
plt.title ("Elbow method')
plt.ylabel ('Inertia')
plt.show()
sns.scatterplot (data = df, x = 'Annual Income
(k)', y = 'spending score (1-100)', hue = 'cluster',
 palette = 'Set2')
from sklearn.clusters using Kmeans
import matplotlib.pyplot as plt
import pandas as pd
wine = load_wine()
X = pd.dataframe (wine data, columns = wine.features
                    - names)
wine.feature_names)
X_scaled = standard scaler().fit_transform(x)
base_clustering = []
for k in [3,4,5]
    Km = kmeans (n_clusters = k, random_state = 72)
```

```
base_clusterings.append(km.fit_predict(X_scaled))
def cspa_ensemble(clusterings):
for i in range(n_samples):
for j in range(n_samples):
    if clusterings[i] == clusterings[j]:
    similarity_matrix[i][j] += 1
similarity_matrix = similarity_matrix / len(clusterings)
ensemble_labels = spectral_clustering(n_clusters=3,
    affinity='precomputed', random_state=42).fit_predict
    (similarity_matrix)
return ensemble_labels
print("Silhouette score:", silhouette_score
    (X_scaled, ensemble_labels))

pca = PCA(n_components=2)
plt.figure(figsize=(10,6))
plt.scatter(X_pca[:,0], X_pca[:,1], c=ensemble_labels,
cmap='viridis', s=50, edgecolors='k')
plt.title("CSPA ensemble clustering on wine
    Dataset (PCA-reduced)")
plt.xlabel("PCA component 1")
plt.colorbar(label='cluster label')
plt.grid(True)
plt.show()
```

Result:

PCA visualization shows clear cluster grouping in reduced dimensions, hence successfully completed and O/P verified.