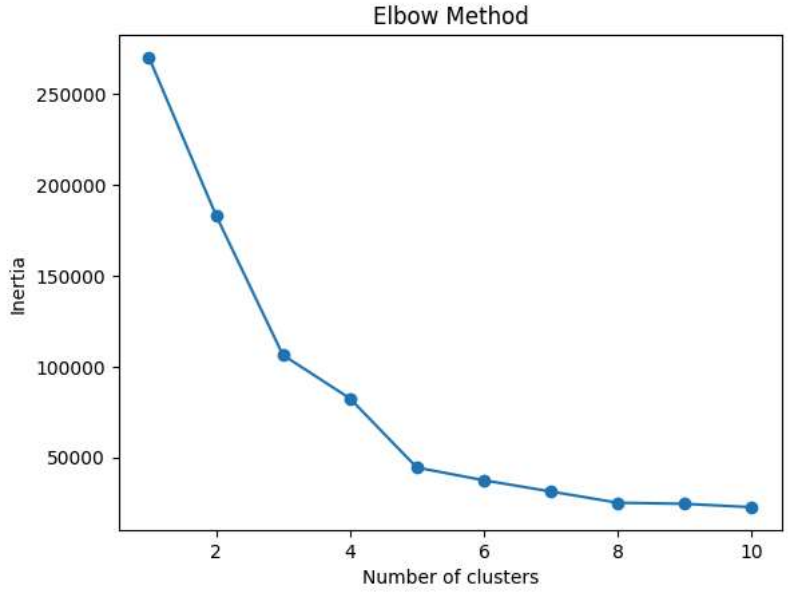
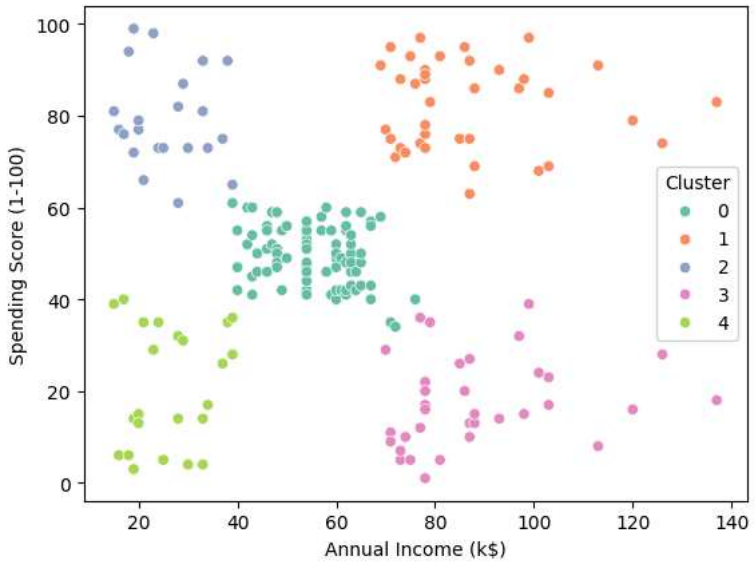
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import seaborn as sns
```

```
df = pd.read_csv("Mall_Customers.csv")
kmeans = KMeans(n_clusters=5, random_state=42)
df["Cluster"] = kmeans.fit_predict(df[["Annual Income (k$)", "Spending Score (1-100)"]])
distortions = []
for i in range(1, 11):
    km = KMeans(n_clusters=i)
    km.fit(df[["Annual Income (k$)", "Spending Score (1-100)"]])
    distortions.append(km.inertia_)
plt.plot(range(1, 11), distortions, marker="o")
plt.title("Elbow Method")
plt.xlabel("Number of clusters")
plt.ylabel("Inertia")
plt.show()
```




```
sns.scatterplot(data=df, x="Annual Income (k$)", y="Spending Score (1-100)",hue="Cluster", palette="Set2")
```

```
<Axes: xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```




```
from sklearn.datasets import load_wine
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
from sklearn.cluster import SpectralClustering
import numpy as np
```

```
wine = load_wine()
X = pd.DataFrame(wine.data, columns=wine.feature_names)
X_scaled = StandardScaler().fit_transform(X)
```

```
base_clusterings = []
for k in [3, 4, 5]:
    km = KMeans(n_clusters=k, random_state=42)
    base_clusterings.append(km.fit_predict(X_scaled))
```

```
def cspa_ensemble(clusterings):
    n_samples = len(clusterings[0])
    similarity_matrix = np.zeros((n_samples, n_samples))
    for clustering in clusterings:
        for i in range(n_samples):
            for j in range(n_samples):
                if clustering[i] == clustering[j]:
                    similarity_matrix[i][j] += 1
                    similarity_matrix = similarity_matrix / len(clusterings)
                    ensemble_labels = SpectralClustering(n_clusters=3, affinity="precomputed",random_state=42).fit_predict(similarity_matrix)
    return ensemble_labels
```

```
ensemble_labels = cspa_ensemble(base_clusterings)
print("Silhouette Score:", silhouette_score(X_scaled, ensemble_labels))
```

```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
plt.figure(figsize=(10, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=ensemble_labels, cmap="viridis", s=50,edgecolor="k")
plt.title("CSPA Ensemble Clustering on Wine Dataset (PCA-reduced)")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.colorbar(label="Cluster Label")
plt.grid(True)
plt.show()
```