unsuper

January 20, 2025

[31]:
```python
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans, DBSCAN
from sklearn.decomposition import PCA
from sklearn import datasets
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
sns.set()
```

[32]:
```python
df = pd.read_csv("./Country-data.csv")
df.head()
```

[32]:
```
               country  child_mort  exports  health  imports  income  \
0          Afghanistan        90.2     10.0    7.58     44.9    1610
1              Albania        16.6     28.0    6.55     48.6    9930
2              Algeria        27.3     38.4    4.17     31.4   12900
3               Angola       119.0     62.3    2.85     42.9    5900
4  Antigua and Barbuda        10.3     45.5    6.03     58.9   19100

   inflation  life_expec  total_fer   gdpp
0       9.44        56.2       5.82    553
1       4.49        76.3       1.65   4090
2      16.10        76.5       2.89   4460
3      22.40        60.1       6.16   3530
4       1.44        76.8       2.13  12200
```

[35]:
```python
label = LabelEncoder()
df['country'] = label.fit_transform(df['country'])
```

[36]:
```python
X = df[['country', 'child_mort', 'exports', 'health', 'imports', 'income',
       'inflation', 'life_expec', 'total_fer', 'gdpp']]
```

[37]:
```python
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
scaler = StandardScaler()
```

```python
for colm in X:
    df[colm] = scaler.fit_transform(df[colm].values.reshape(-1,1))
df
```

[37]:

```
      country  child_mort   exports    health   imports    income  inflation  \
0   -1.721710    1.291532 -1.138280  0.279088 -0.082455 -0.808245   0.157336
1   -1.700967   -0.538949 -0.479658 -0.097016  0.070837 -0.375369  -0.312347
2   -1.680223   -0.272833 -0.099122 -0.966073 -0.641762 -0.220844   0.789274
3   -1.659480    2.007808  0.775381 -1.448071 -0.165315 -0.585043   1.387054
4   -1.638736   -0.695634  0.160668 -0.286894  0.497568  0.101732  -0.601749
..        ...         ...       ...       ...       ...       ...        ...
162  1.638736   -0.225578  0.200917 -0.571711  0.240700 -0.738527  -0.489784
163  1.659480   -0.526514 -0.461363 -0.695862 -1.213499 -0.033542   3.616865
164  1.680223   -0.372315  1.130305  0.008877  1.380030 -0.658404   0.409732
165  1.700967    0.448417 -0.406478 -0.597272 -0.517472 -0.658924   1.500916
166  1.721710    1.114951 -0.150348 -0.338015 -0.662477 -0.721358   0.590015

     life_expec  total_fer      gdpp
0     -1.619092   1.902882 -0.679180
1      0.647866  -0.859973 -0.485623
2      0.670423  -0.038404 -0.465376
3     -1.179234   2.128151 -0.516268
4      0.704258  -0.541946 -0.041817
..          ...        ...       ...
162   -0.852161   0.365754 -0.546913
163    0.546361  -0.316678  0.029323
164    0.286958  -0.661206 -0.637754
165   -0.344633   1.140944 -0.637754
166   -2.092785   1.624609 -0.629546

[167 rows x 10 columns]
```

[38]:

```python
inertia = []
k_range = range(1, 11)
```

[42]:

```python
for k in k_range:
    kmeans = KMeans(n_clusters=3, random_state=42)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)
    labels_kmeans = kmeans.labels_
    print("K-Means cluster labels:", labels_kmeans)
    print("Cluster centroids:\n", kmeans.cluster_centers_)
```

```
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
```

```
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
```

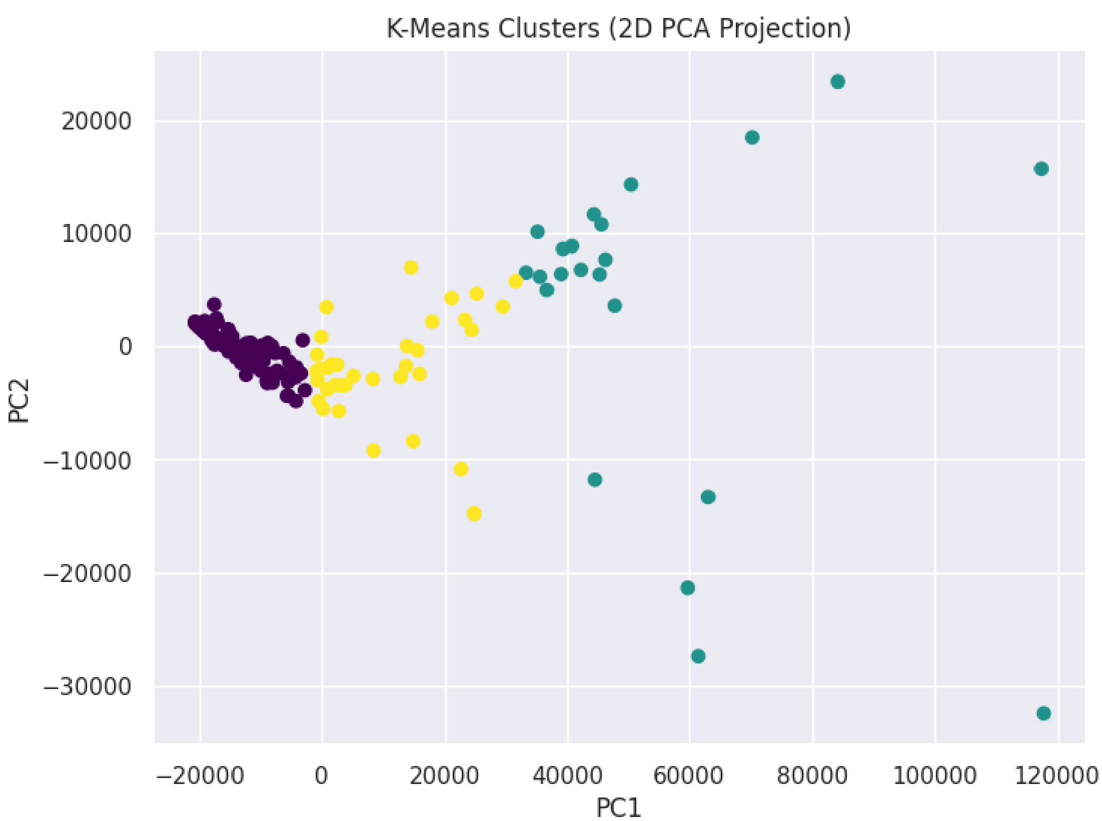
```
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
```

```
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
K-Means cluster labels: [0 0 0 0 2 2 0 1 1 0 2 2 0 2 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 2 0 0 0
 0 0 0 0 2 2 2 1 0 0 0 0 2 0 2 0 1 1 0 0 0 1 0 2 0 0 0 0 0 0 2 1 0 0 0 0 1
 2 2 0 1 0 2 0 0 1 0 0 2 0 0 0 2 2 1 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1
```

```
 2 0 0 1 2 0 0 0 0 0 2 2 1 0 2 0 0 2 0 0 2 0 1 2 2 0 0 2 2 0 0 0 0 1 1 0 0
 0 0 0 0 0 2 0 0 0 1 2 1 2 0 0 2 0 0 0]
Cluster centroids:
 [[8.20825688e+01 5.37440367e+01 3.32513670e+01 6.31477064e+00
  4.54767514e+01 6.48589908e+03 9.15205505e+00 6.66321101e+01
  3.50211009e+00 3.07499083e+03]
 [8.03181818e+01 5.07727273e+00 6.31681818e+01 8.94272727e+00
  5.12590909e+01 5.48000000e+04 3.30872727e+00 8.04272727e+01
  1.80000000e+00 5.25545455e+04]
 [8.74166667e+01 1.17027778e+01 5.14194444e+01 7.03250000e+00
  4.85000000e+01 2.64055556e+04 6.36666667e+00 7.64027778e+01
  1.97166667e+00 1.87122222e+04]]
```

[43]:
```python
pca_kmeans = PCA(n_components=2)
X_2d_kmeans = pca_kmeans.fit_transform(X)

plt.figure(figsize=(8, 6))
plt.scatter(X_2d_kmeans[:, 0], X_2d_kmeans[:, 1], c=labels_kmeans,
  cmap='viridis')
plt.title("K-Means Clusters (2D PCA Projection)")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.show()
```



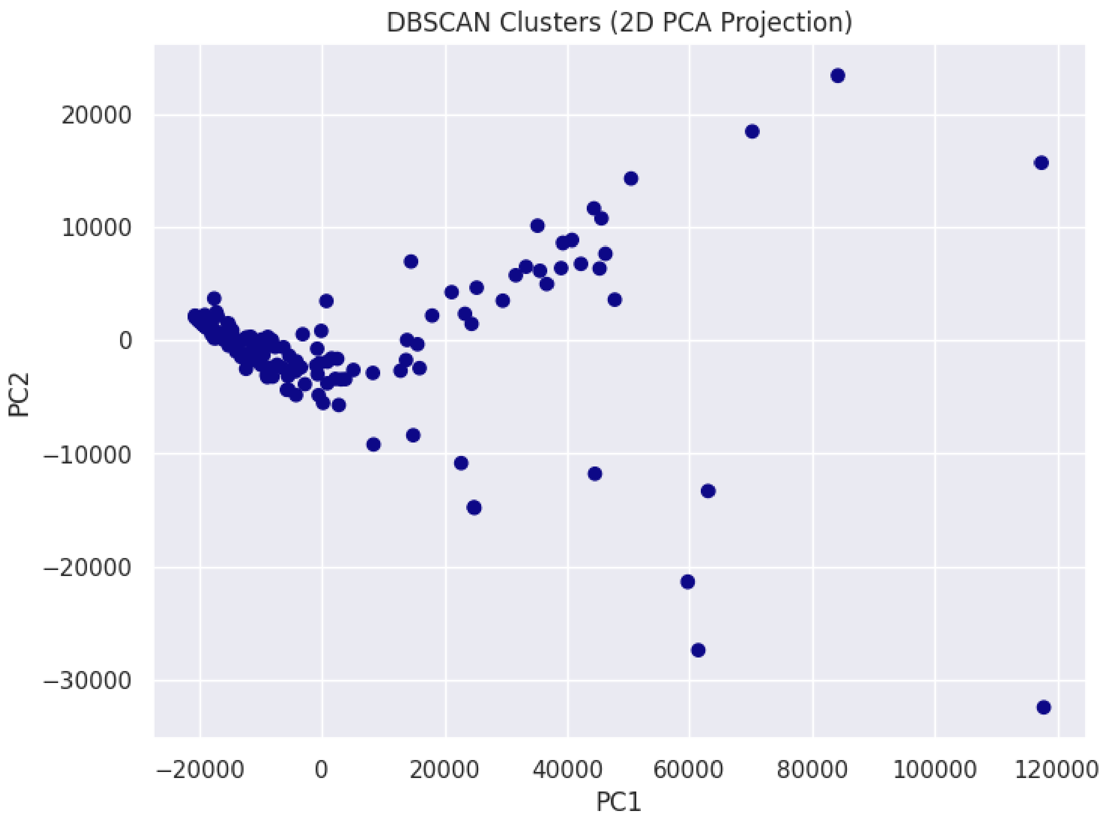
[44]:
```
dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan.fit(X)

labels_dbscan = dbscan.labels_
print("DBSCAN labels:", labels_dbscan)
```

```
DBSCAN labels: [-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
```

[45]:
```
pca_dbscan = PCA(n_components=2)
X_2d_dbscan = pca_dbscan.fit_transform(X)

plt.figure(figsize=(8, 6))
plt.scatter(X_2d_dbscan[:, 0], X_2d_dbscan[:, 1], c=labels_dbscan,␣
 ↪cmap='plasma')
plt.title("DBSCAN Clusters (2D PCA Projection)")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.show()
```

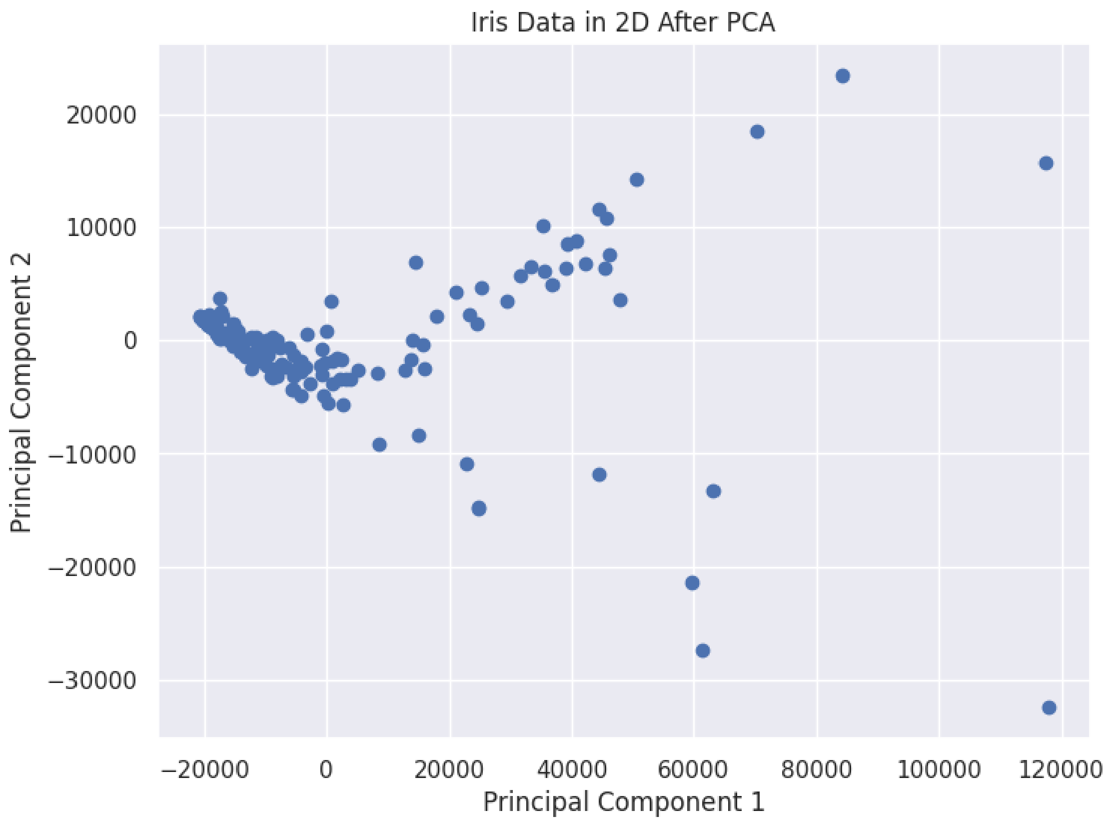


[46]:
```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

print("Explained variance ratio:", pca.explained_variance_ratio_)
```

```
Explained variance ratio: [0.94791994 0.05207329]
```

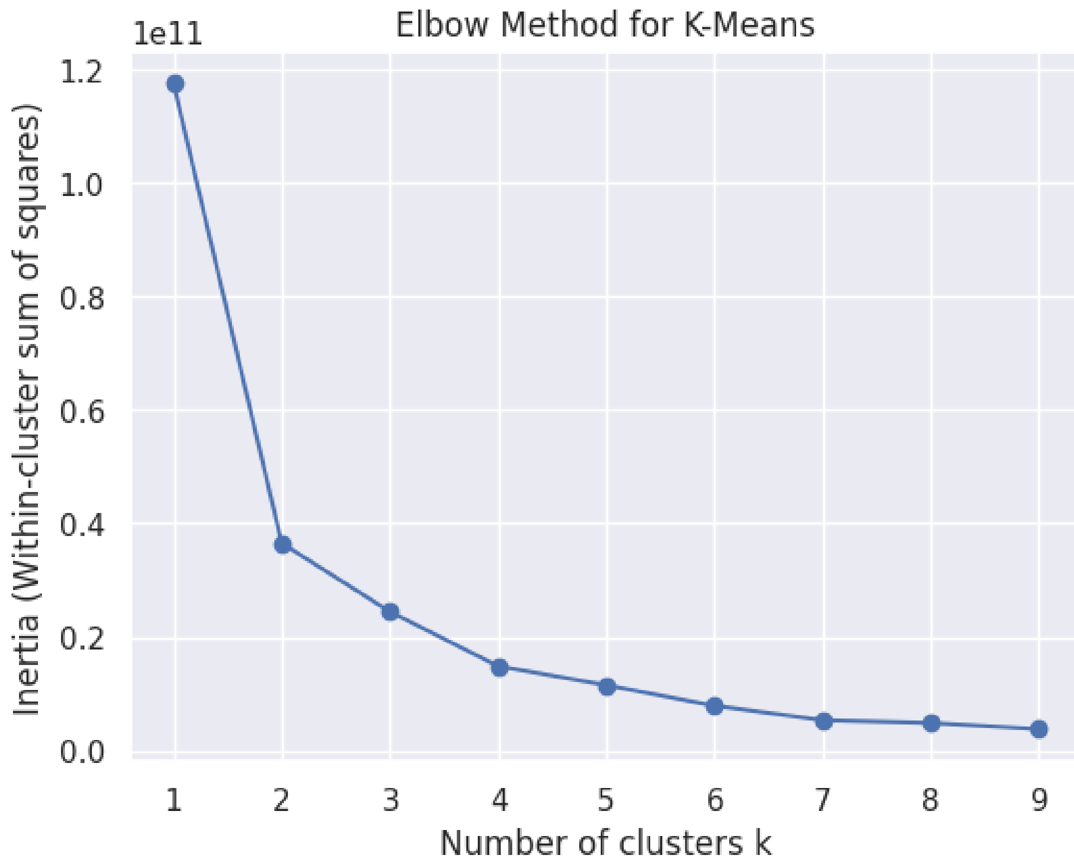
[47]:
```
plt.figure(figsize=(8, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1])
plt.title("Iris Data in 2D After PCA")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()
```



[48]:
```python
inertias = []
K_range = range(1, 10)  # Example: testing k = 1 to 9

for k in K_range:
    kmeans_temp = KMeans(n_clusters=k, random_state=42)
    kmeans_temp.fit(X)
    inertias.append(kmeans_temp.inertia_)

plt.plot(K_range, inertias, marker='o')
plt.xlabel("Number of clusters k")
plt.ylabel("Inertia (Within-cluster sum of squares)")
plt.title("Elbow Method for K-Means")
plt.show()
```



[53]:
```python
from sklearn.metrics import silhouette_score

kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X)
labels_kmeans = kmeans.labels_

silhouette_kmeans = silhouette_score(X, labels_kmeans)
print("Silhouette Score (K-Means):", silhouette_kmeans)

dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan.fit(X)
labels_dbscan = dbscan.labels_
if len(set(labels_dbscan) - {-1}) > 1:  # Exclude noise (-1)
    silhouette_dbscan = silhouette_score(X, labels_dbscan)
    print("Silhouette Score (DBSCAN):", silhouette_dbscan)
else:
    print("DBSCAN did not produce enough clusters to compute a silhouette score.
 ↪")
```

```
Silhouette Score (K-Means): 0.6004224256794997
```

```
DBSCAN did not produce enough clusters to compute a silhouette score.
```

[50]:
```python
from sklearn.decomposition import PCA

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

print("Explained variance ratio:", pca.explained_variance_ratio_)
```

```
Explained variance ratio: [0.94791994 0.05207329]
```

[51]:
```python
from sklearn.decomposition import PCA
from sklearn.metrics import mean_squared_error
```

[52]:
```python
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
print("Explained variance ratio:", pca.explained_variance_ratio_)
```

```
Explained variance ratio: [0.94791994 0.05207329]
```