Bank Notes Dataset Machine Learning

Pengxi Chen
McMaster University
April 5, 2024

[31]:
```
#(1)
import pandas as pd
file_path = "banknote.csv"
banknote_df = pd.read_csv(file_path)
banknote_df.head()
```

[31]:

	Status	Length	Left	Right	Bottom	Top	Diagonal
0	genuine	214.8	131.0	131.1	9.0	9.7	141.0
1	genuine	214.6	129.7	129.7	8.1	9.5	141.7
2	genuine	214.8	129.7	129.7	8.7	9.6	142.2
3	genuine	214.8	129.7	129.6	7.5	10.4	142.0
4	genuine	215.0	129.6	129.7	10.4	7.7	141.8

[32]:
```
#(2)
status_labels = banknote_df["Status"].copy()
variables = banknote_df.columns.tolist()
banknote_df_modified = banknote_df.drop("Status", axis=1)
status_labels.head(), variables, banknote_df_modified.head()
```

[32]:
```
(0    genuine
 1    genuine
 2    genuine
 3    genuine
 4    genuine
 Name: Status, dtype: object,
 ['Status', 'Length', 'Left', 'Right', 'Bottom', 'Top', 'Diagonal'],
    Length   Left   Right   Bottom   Top   Diagonal
 0   214.8  131.0  131.1      9.0   9.7      141.0
 1   214.6  129.7  129.7      8.1   9.5      141.7
 2   214.8  129.7  129.7      8.7   9.6      142.2
 3   214.8  129.7  129.6      7.5  10.4      142.0
 4   215.0  129.6  129.7     10.4   7.7      141.8)
```

[33]:
```
#(4)
mean_values = banknote_df_modified.mean()
variance_values = banknote_df_modified.var()
max_variance_variable = variance_values.idxmax()
```

```
mean_values, variance_values, max_variance_variable
```

[33]:
```
(Length      214.8960
 Left        130.1215
 Right       129.9565
 Bottom        9.4175
 Top          10.6505
 Diagonal    140.4835
 dtype: float64,
 Length      0.141793
 Left        0.130339
 Right       0.163274
 Bottom      2.086878
 Top         0.644723
 Diagonal    1.327716
 dtype: float64,
 'Bottom')
```

[34]:
```
#(5)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
banknote_df_normalized = scaler.fit_transform(banknote_df_modified)
banknote_df_normalized = pd.DataFrame(banknote_df_normalized,␣
↪columns=banknote_df_modified.columns)
banknote_df_normalized.head()
```

[34]:

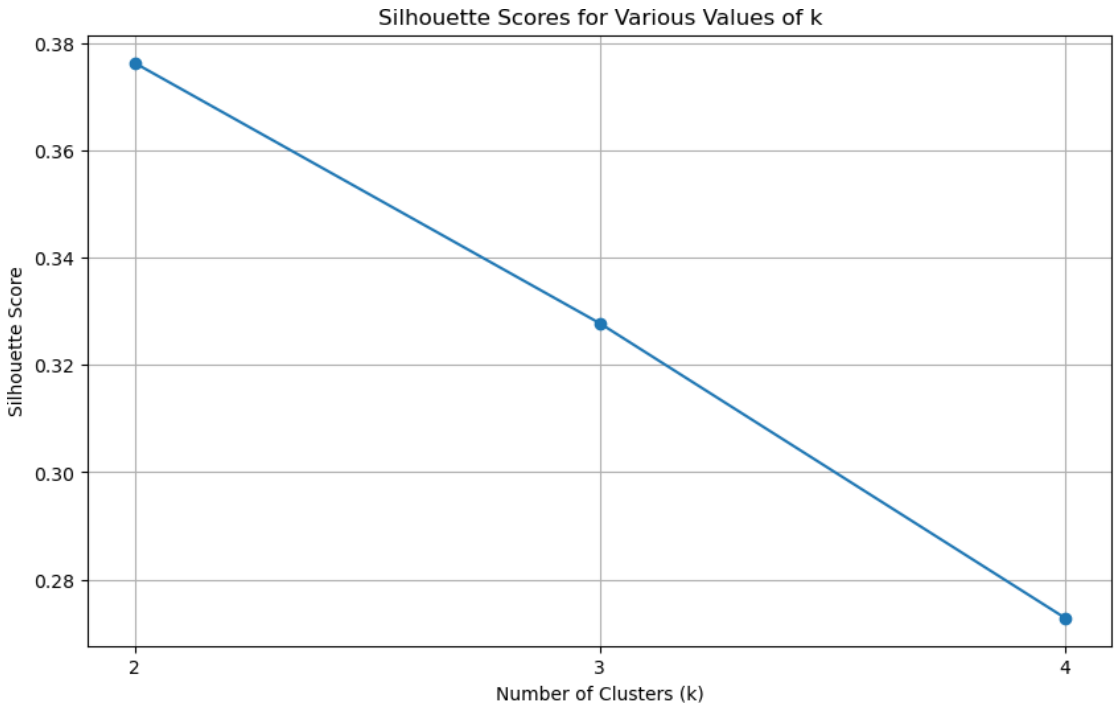
	Length	Left	Right	Bottom	Top	Diagonal
0	-0.255583	2.439452	2.837043	-0.289732	-1.186735	0.449372
1	-0.788048	-1.170437	-0.636381	-0.914304	-1.436443	1.058395
2	-0.255583	-1.170437	-0.636381	-0.497923	-1.311589	1.493412
3	-0.255583	-1.170437	-0.884483	-1.330685	-0.312759	1.319405
4	0.276882	-1.448121	-0.636381	0.681824	-3.683811	1.145399

[35]:
```
#(6)
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
k = 3
kmeans = KMeans(n_clusters=k, n_init=20, random_state=42)
cluster_labels = kmeans.fit_predict(banknote_df_normalized)
silhouette_avg = silhouette_score(banknote_df_normalized, cluster_labels)

silhouette_avg
```

[35]: 0.32777262691591696

[36]:
```
#(7)
import matplotlib.pyplot as plt
```

```
import numpy as np

k_values = range(2, 5)

silhouette_scores = []

for k in k_values:
    kmeans = KMeans(n_clusters=k, n_init=20, random_state=42)
    cluster_labels = kmeans.fit_predict(banknote_df_normalized)
    silhouette_avg = silhouette_score(banknote_df_normalized, cluster_labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(10, 6))
plt.plot(k_values, silhouette_scores, marker="o")
plt.title("Silhouette Scores for Various Values of k")
plt.xlabel("Number of Clusters (k)")
plt.ylabel("Silhouette Score")
plt.grid(True)
plt.xticks(k_values)

plt.show()
```

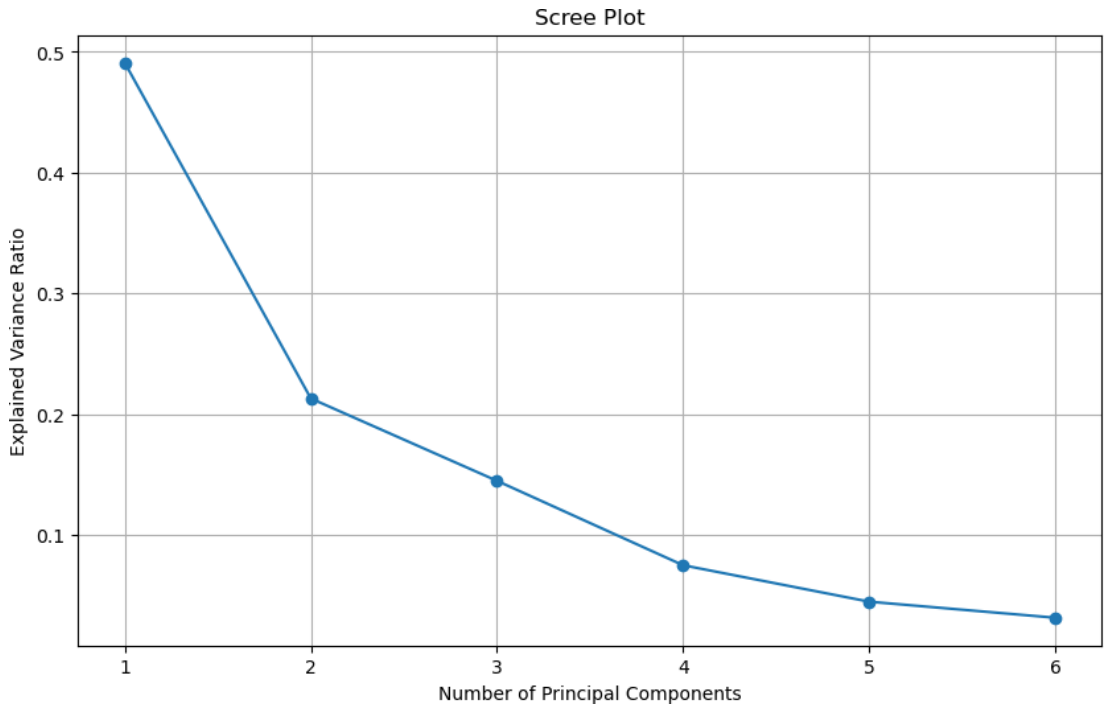


[37]:
```
# 8
k_optimal = 2
```

```
kmeans_optimal = KMeans(n_clusters=k_optimal, n_init=20, random_state=42)
kmeans_optimal.fit(banknote_df_normalized)

cluster_assignments = kmeans_optimal.labels_

clusters, counts = np.unique(cluster_assignments, return_counts=True)
cluster_distribution = dict(zip(clusters, counts))
cluster_distribution
```

[37]: {0: 92, 1: 108}

[38]:
```
#(9)
from sklearn.decomposition import PCA

pca = PCA().fit(banknote_df_normalized)

explained_variance_ratio = pca.explained_variance_ratio_

plt.figure(figsize=(10, 6))
plt.plot(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio,␣
↪marker='o')
plt.title('Scree Plot')
plt.xlabel('Number of Principal Components')
plt.ylabel('Explained Variance Ratio')
plt.grid(True)
plt.xticks(range(1, len(explained_variance_ratio) + 1))
plt.show()

cumulative_explained_variance = np.cumsum(explained_variance_ratio)

n_components = np.argmax(cumulative_explained_variance >= 0.8) + 1

explained_variance_ratio, cumulative_explained_variance, n_components
```



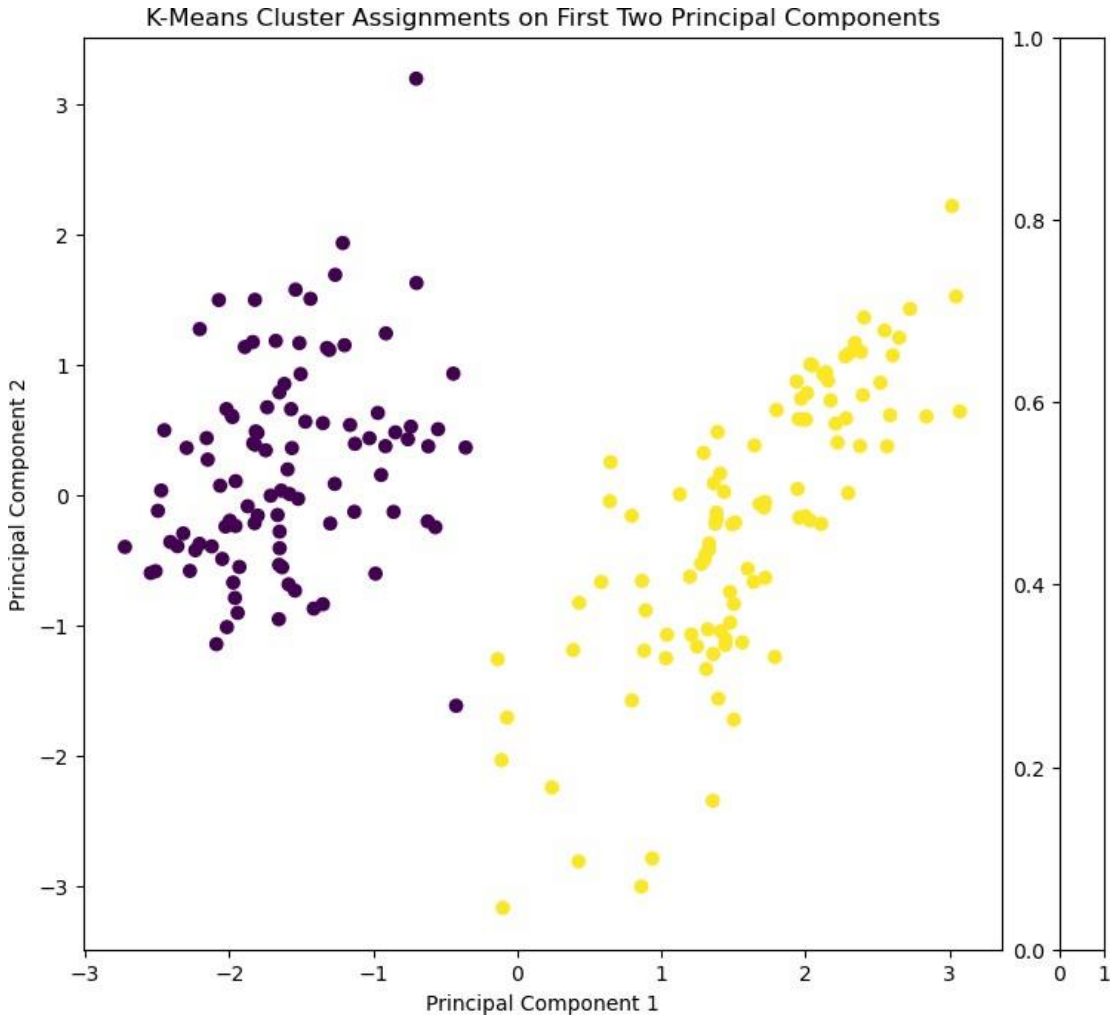
[38]:
```
(array([0.49092637, 0.21301396, 0.14483876, 0.07496145, 0.04477948,
        0.03147998]),
 array([0.49092637, 0.70394033, 0.84877909, 0.92374054, 0.96852002,
        1.        ]),
 3)
```

[39]:
```
#(11)

explained_variance_ratio = [0.49092637, 0.21301396, 0.14483876, 0.07496145, 0.
↪04477948, 0.03147998]
proportion_variance_first_two = sum(explained_variance_ratio[:2])
print(proportion_variance_first_two)
```

```
0.70394033
```

[40]:
```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

pca = PCA(n_components=2)
principal_components = pca.fit_transform(banknote_df_modified)

kmeans = KMeans(n_clusters=2, n_init=20)
```

```
kmeans.fit(banknote_df_modified)
cluster_assignments = kmeans.labels_

plt.figure(figsize=(10, 8))
plt.scatter(principal_components[:, 0], principal_components[:, 1],␣
↪c=cluster_assignments, cmap='viridis')
plt.title('K-Means Cluster Assignments on First Two Principal Components')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.colorbar(title='Cluster')

# Show the plot
plt.show()
```



[30]:
```
from sklearn.metrics import adjusted_rand_score
status_labels = banknote_df['Status']

scaler = StandardScaler()
banknote_df_normalized = scaler.fit_transform(banknote_df.drop('Status',␣
↪axis=1))
```

```
pca = PCA(n_components=2)
principal_components = pca.fit_transform(banknote_df_normalized)

kmeans = KMeans(n_clusters=2, n_init=20, random_state=42)
kmeans.fit(principal_components)

cluster_assignments = kmeans.labels_

ari_score = adjusted_rand_score(status_labels, cluster_assignments)

ari_score
```

[30]: 0.8456292321864344

[ ]: