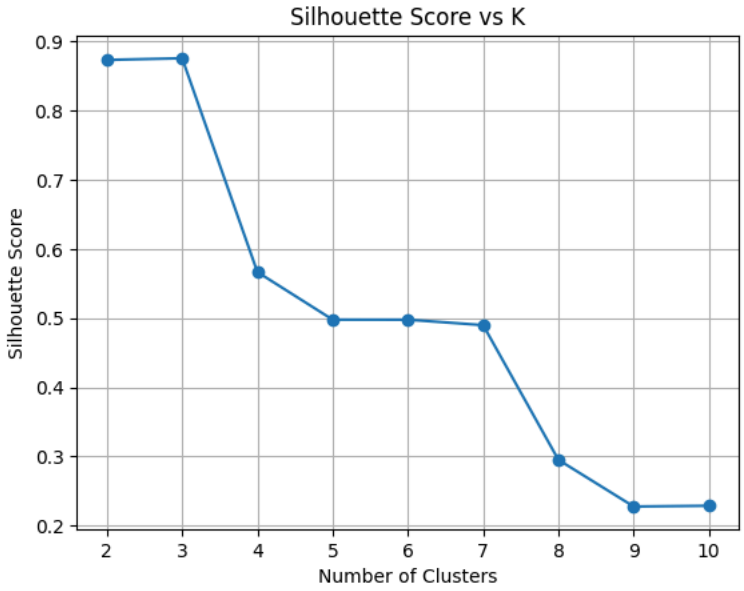
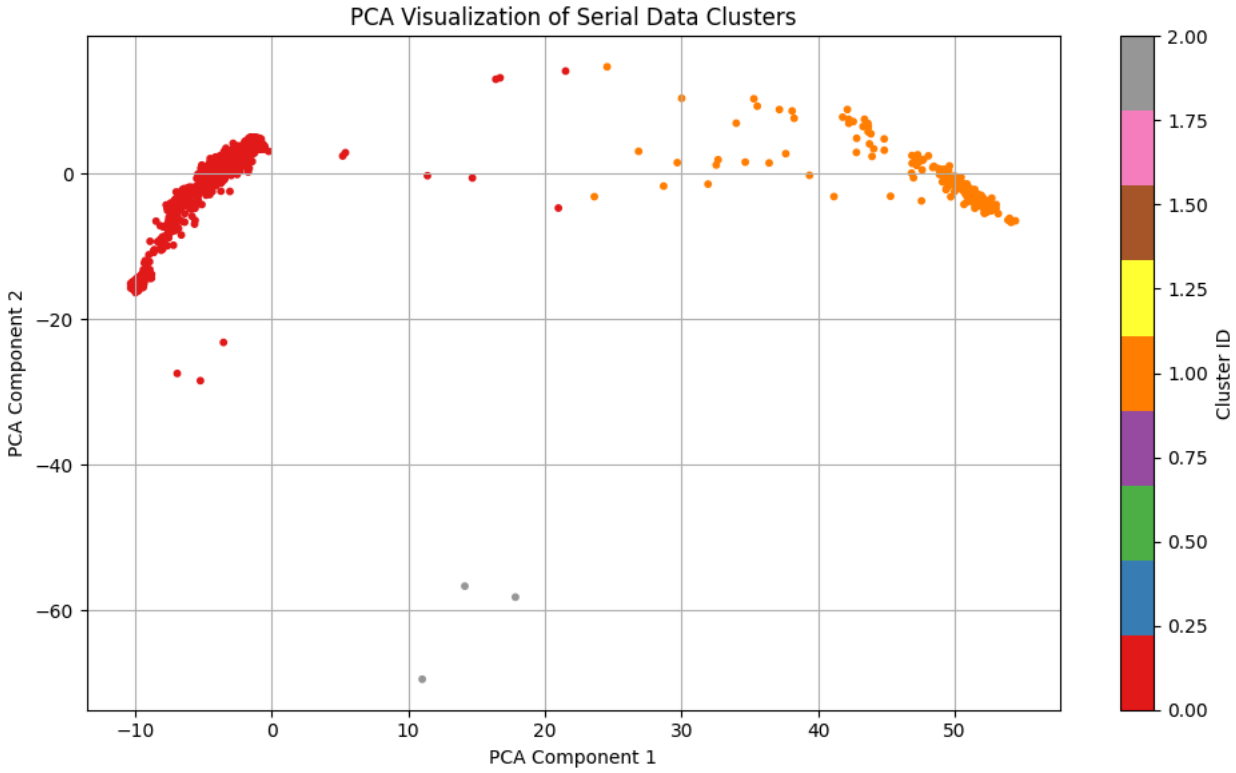
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Step 1: Load and parse 'Serial Data' column
df = pd.read_csv("Serial_data_On_metal_table.csv")
serial_data_numeric = df['Serial Data'].apply(lambda x: list(map(int, x.split(','))))
X = np.array(serial_data_numeric.tolist())

# Step 2: Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 3: Apply PCA (retain 95% variance)
pca = PCA(n_components=0.95)
X_pca = pca.fit_transform(X_scaled)
print("Reduced shape after PCA:", X_pca.shape)

# Step 4: Identify important features
importance_scores = np.abs(pca.components_).sum(axis=0)
important_feature_indices = np.argsort(importance_scores)[::-1]
print("Top 5 important feature indices:", important_feature_indices[:5])

# Step 5: Tune K using silhouette scores
silhouette_scores = []
K_range = range(2, 11)

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(X_pca)
    score = silhouette_score(X_pca, labels)
    silhouette_scores.append(score)

# Plot silhouette scores
plt.plot(K_range, silhouette_scores, marker='o')
plt.title("Silhouette Score vs K")
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Score")
plt.grid(True)
plt.show()

# Step 6: Final clustering with best K
optimal_k = K_range[np.argmax(silhouette_scores)]
print("Optimal number of clusters:", optimal_k)

kmeans_final = KMeans(n_clusters=optimal_k, random_state=42)
final_labels = kmeans_final.fit_predict(X_pca)
df['Cluster'] = final_labels

# Step 7: Visualize in 2D PCA space
pca_2d = PCA(n_components=2)
X_2d = pca_2d.fit_transform(X_scaled)

plt.figure(figsize=(10, 6))
scatter = plt.scatter(X_2d[:, 0], X_2d[:, 1], c=final_labels, cmap='Set1', s=10)
plt.title('PCA Visualization of Serial Data Clusters')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.colorbar(scatter, label='Cluster ID')
plt.grid(True)
plt.tight_layout()
plt.show()
```

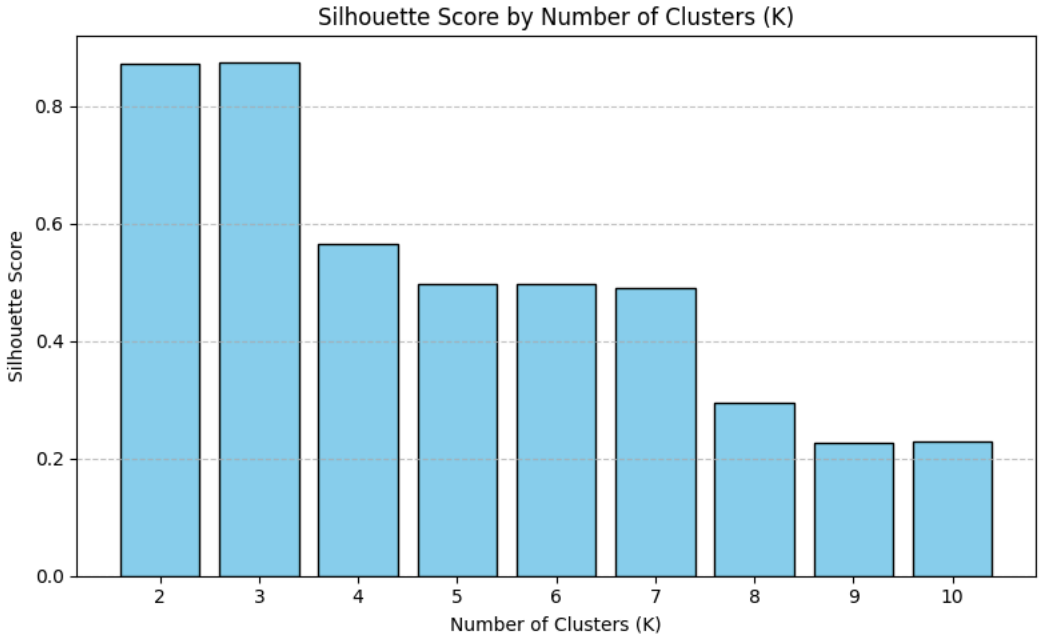
```
Reduced shape after PCA: (3208, 15)
Top 5 important feature indices: [249 230 250 240 245]
```




```
Optimal number of clusters: 3
```




```
import matplotlib.pyplot as plt

# Assuming we already have:
# - silhouette_scores
# - K_range

plt.figure(figsize=(8, 5))
plt.bar(K_range, silhouette_scores, color='skyblue', edgecolor='black')
plt.title("Silhouette Score by Number of Clusters (K)")
plt.xlabel("Number of Clusters (K)")
plt.ylabel("Silhouette Score")
plt.xticks(K_range)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```


Silhouette Score by Number of Clusters (K)
Silhouette Score
0.0
0.2
0.4
0.6
0.8
2
3
4
5
6
7
8
9
10
Number of Clusters (K)