Abhishek Murthy
21BDS0064
Fall Sem 2024-2025
DA-4
Data Mining Lab
30-09-2024

In [1]:
```
!pip install scikit-learn-extra
```

```
Collecting scikit-learn-extra
  Downloading scikit_learn_extra-0.3.0-cp310-cp310-manylinux_2_17_x86_64.manylinux
2014_x86_64.whl.metadata (3.6 kB)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.10/dist-pac
kages (from scikit-learn-extra) (1.26.4)
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.10/dist-pac
kages (from scikit-learn-extra) (1.13.1)
Requirement already satisfied: scikit-learn>=0.23.0 in /usr/local/lib/python3.10/d
ist-packages (from scikit-learn-extra) (1.5.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-pac
kages (from scikit-learn>=0.23.0->scikit-learn-extra) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/d
ist-packages (from scikit-learn>=0.23.0->scikit-learn-extra) (3.5.0)
Downloading scikit_learn_extra-0.3.0-cp310-cp310-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl (2.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 14.8 MB/s eta 0:00:00
Installing collected packages: scikit-learn-extra
Successfully installed scikit-learn-extra-0.3.0
```

In [4]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn_extra.cluster import KMedoids
from scipy.cluster import hierarchy
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt
```

In [5]:
```
def perform_kmedoids(X, k):
  kmedoids = KMedoids(n_clusters=k, random_state=42)
  clusters = kmedoids.fit_predict(X)
  return clusters
def perform_hierarchical(X, k):
  linkage_matrix = linkage(X, method='ward')
  clusters = hierarchy.fcluster(linkage_matrix, k, criterion='maxclust')
  return clusters
def plot_clusters(X, clusters, title):
  plt.figure(figsize=(10, 8))
  plt.scatter(X[:, 0], X[:, 1], c=clusters, cmap='viridis')
  plt.title(title)
  plt.xlabel('Feature 1')
  plt.ylabel('Feature 2')
  plt.colorbar(label='Cluster')
  plt.show()
```

In [6]:
```
df = pd.read_csv('CC GENERAL.csv')
features = ['BALANCE', 'PURCHASES', 'CASH_ADVANCE', 'CREDIT_LIMIT', 'PAYMENTS', 'MI
df.isna().sum()
print(df[features].dtypes)
```

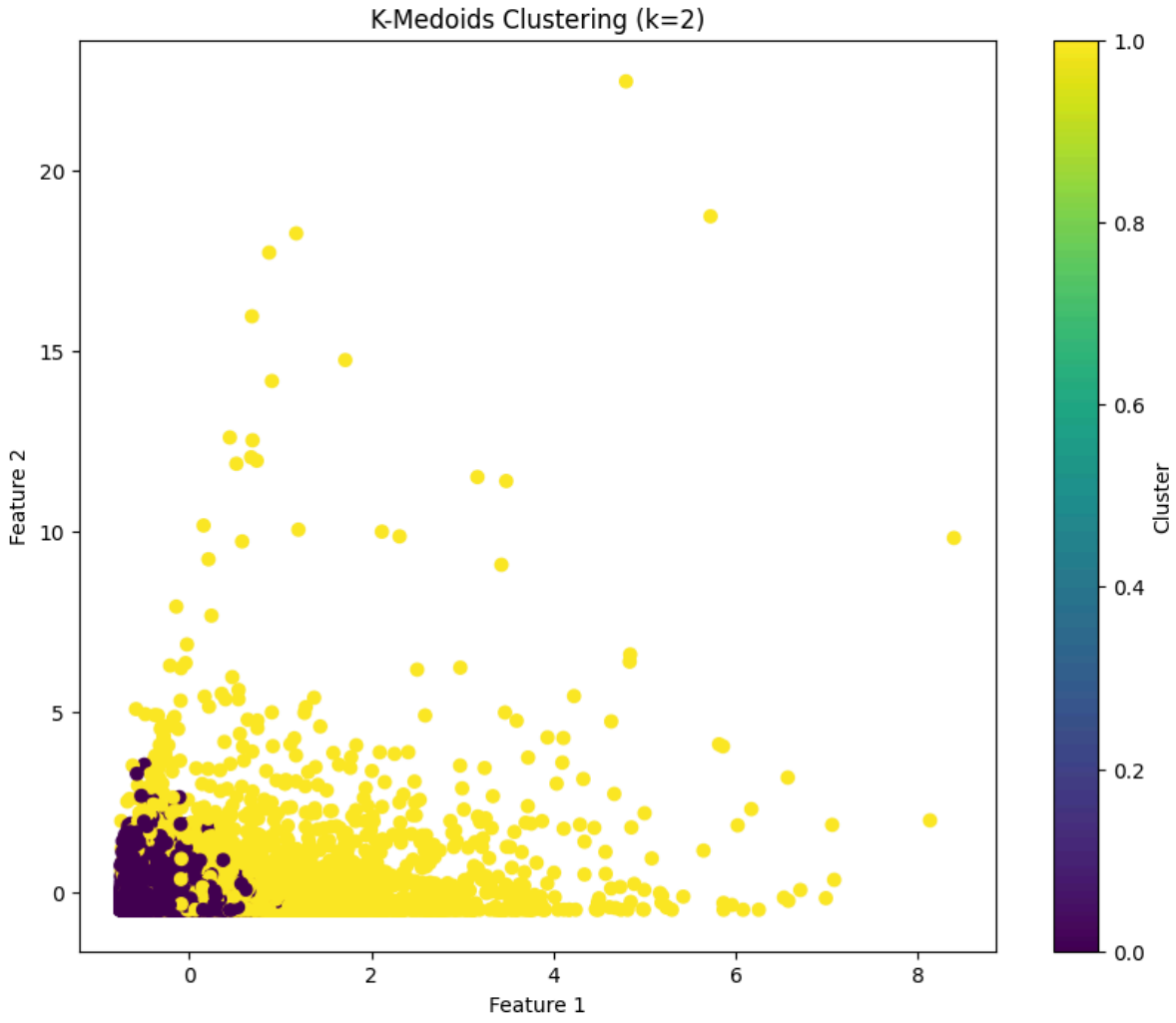
```
BALANCE             float64
PURCHASES           float64
CASH_ADVANCE        float64
CREDIT_LIMIT        float64
PAYMENTS            float64
MINIMUM_PAYMENTS    float64
dtype: object
```

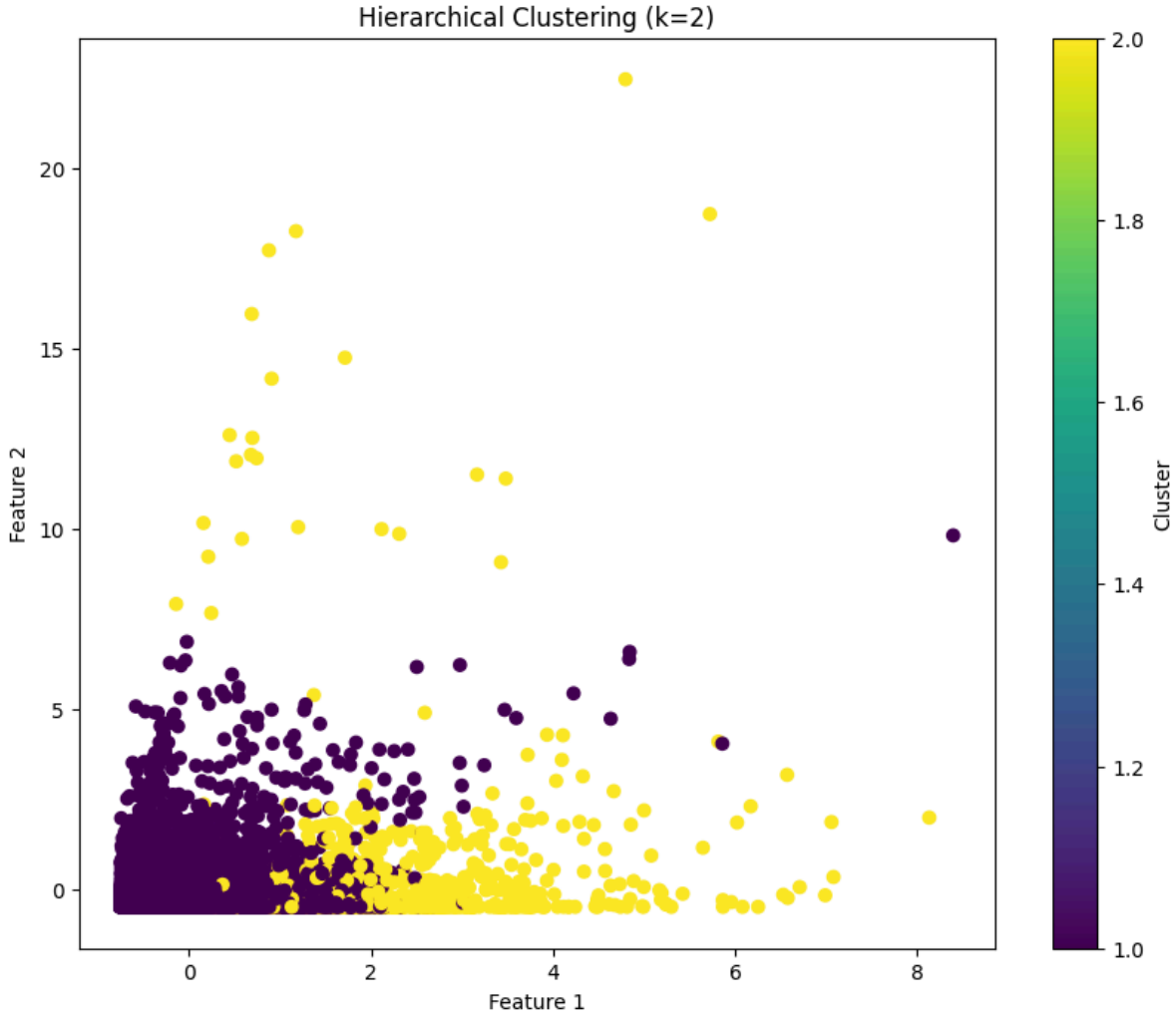
In [7]:
```
for col in features:
 df[col] = pd.to_numeric(df[col], errors='coerce')
```

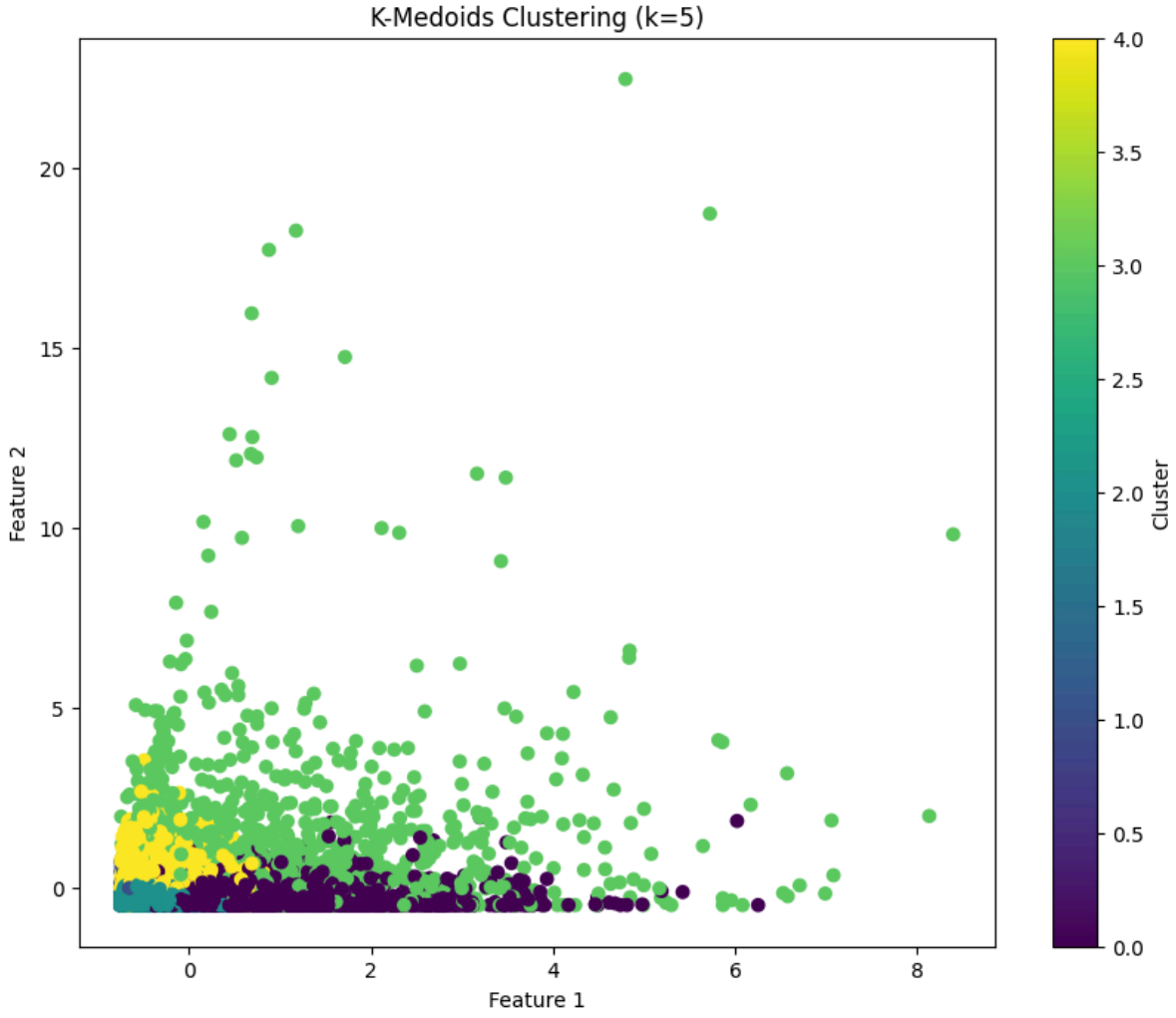
```
for col in features:
 median_val = df[col].median()
 df[col].fillna(median_val, inplace=True)

print(df[features].dtypes)
```

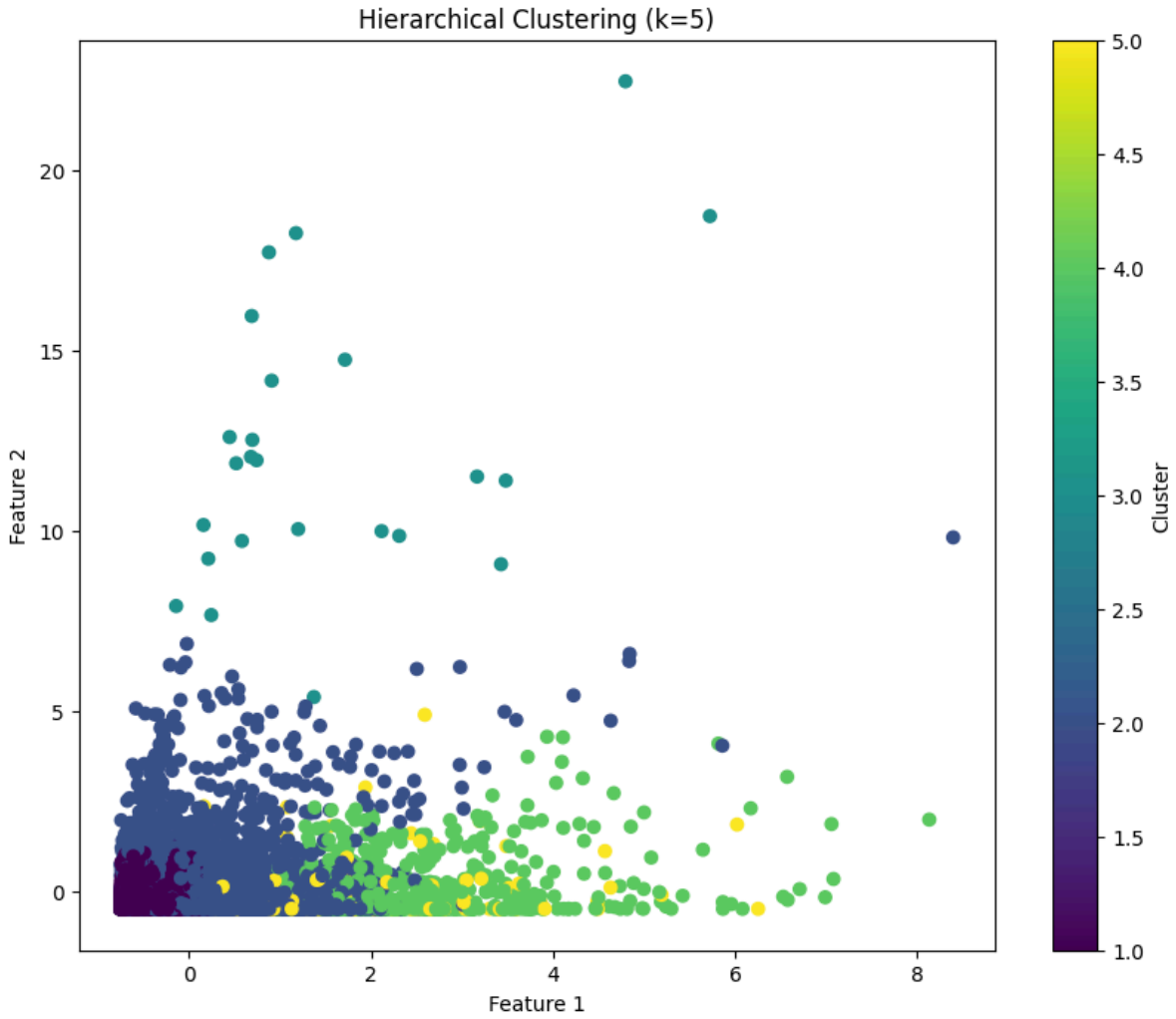
```
BALANCE             float64
PURCHASES           float64
CASH_ADVANCE        float64
CREDIT_LIMIT        float64
PAYMENTS            float64
MINIMUM_PAYMENTS    float64
dtype: object
```

In [8]:
```
features = ['BALANCE', 'PURCHASES', 'CASH_ADVANCE', 'CREDIT_LIMIT', 'PAYMENTS']
scaler = StandardScaler()
X = scaler.fit_transform(df[features])
for k in [2, 5]:

 kmedoids_clusters = perform_kmedoids(X, k)
 plot_clusters(X, kmedoids_clusters, f'K-Medoids Clustering (k={k})')

 hierarchical_clusters = perform_hierarchical(X, k)
 plot_clusters(X, hierarchical_clusters, f'Hierarchical Clustering (k={k})')
```

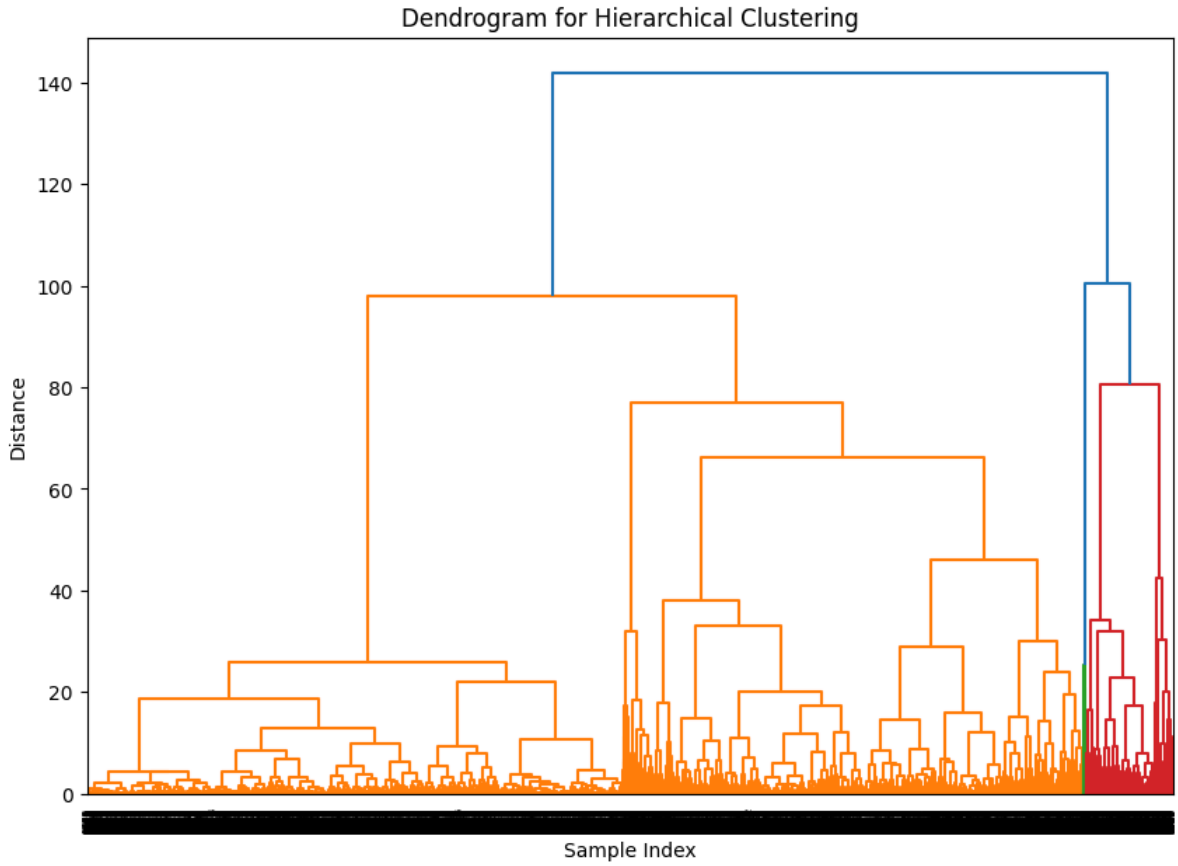



Hierarchical Clustering (k=2)
Feature 2
Feature 1
Cluster
0
5
10
15
20
0
2
4
6
8
1.0
1.2
1.4
1.6
1.8
2.0


K-Medoids Clustering (k=5)
Feature 2
Feature 1
Cluster
0
5
10
15
20
0
2
4
6
8
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0



In [9]:
```
plt.figure(figsize=(10, 7))
dendrogram(linkage(X, method='ward'))
plt.title('Dendrogram for Hierarchical Clustering')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
print(f"Dataset shape: {df.shape}")
```




```
Dataset shape: (8950, 18)
```

In [ ]: