Name: Varun Sudhir

Reg No: 21BDS0040

Data Mining Lab Digital Assignment – IV

Consider the credit card dataset (uploaded in moodle). Apply K – Medoids clustering and hierarchical clustering in the dataset.

Show the clustered data if k = 2, k = 5

In [13]:
```
# Varun Sudhir 21BDS0040
# Importing Libraries

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,fcluster
import matplotlib.pyplot as plt
```

In [14]:
```
# Varun Sudhir 21BDS0040

# Perform K-Medoids clustering
def perform_k_medoids(data, num_clusters):
    k_medoids_instance = KMedoids(n_clusters=num_clusters, random_state=42)
    cluster_labels = k_medoids_instance.fit_predict(data)
    return cluster_labels

# Perform Hierarchical clustering
def perform_hierarchical_clustering(data, num_clusters):
    linkage_matrix = linkage(data, method='ward')
    cluster_labels = fcluster(linkage_matrix, num_clusters, criterion='maxclust')
    return cluster_labels

# Visualize clusters
def visualize_clusters(data, cluster_labels, plot_title):
    plt.figure(figsize=(10, 8))
    plt.scatter(data[:, 0], data[:, 1], c=cluster_labels, cmap='viridis')
    plt.title(plot_title)
    plt.xlabel('Feature 1')
    plt.ylabel('Feature 2')
    plt.colorbar(label='Cluster')
    plt.show()
```

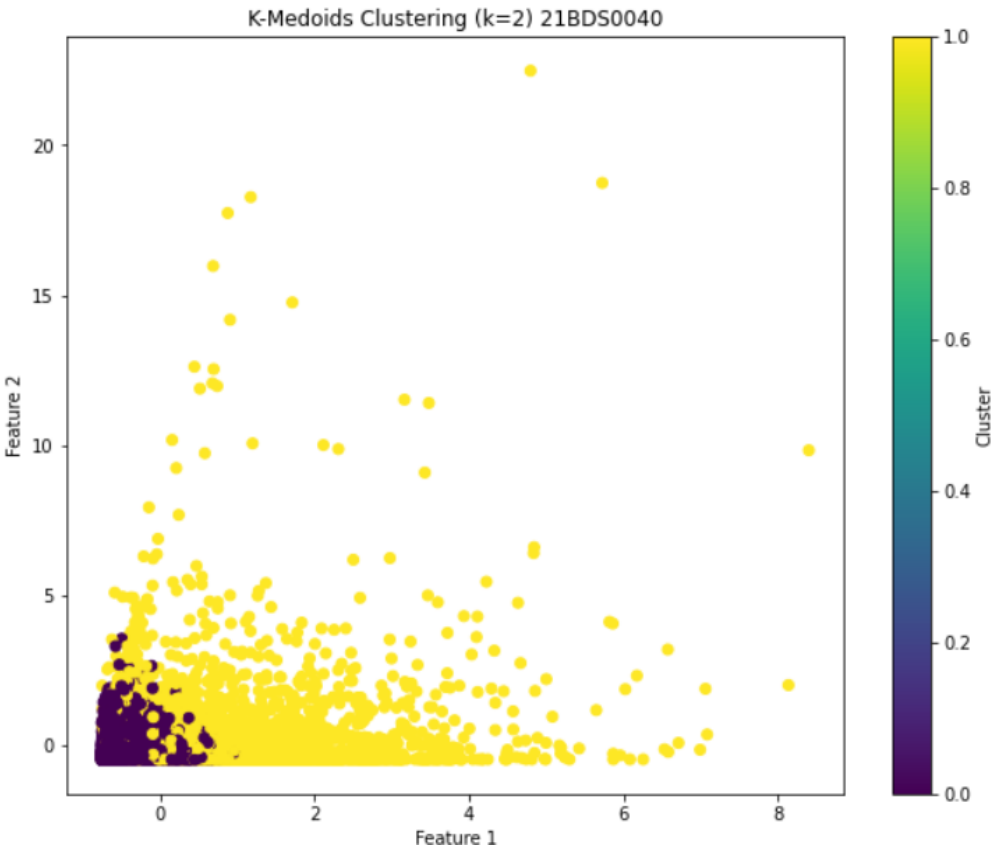
In [15]:
```python
# Varun Sudhir 21BDS0040

# Load the dataset
credit_card_data = pd.read_csv('CC GENERAL.csv')

# Select features for clustering
selected_features = ['BALANCE', 'PURCHASES', 'CASH_ADVANCE', 'CREDIT_LIMIT', 'PAYMENTS']

# Check for missing values in the dataset
missing_values = credit_card_data.isna().sum()

# Print the data types of the selected features
print(credit_card_data[selected_features].dtypes)
```

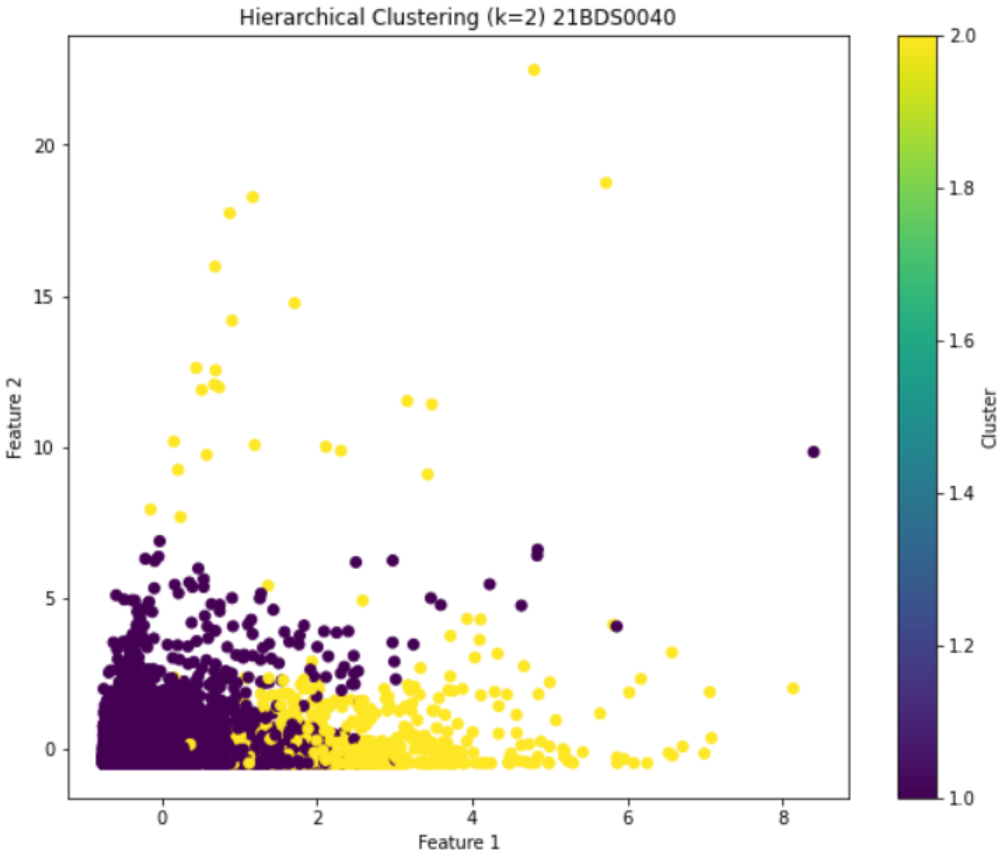
```
BALANCE          float64
PURCHASES        float64
CASH_ADVANCE     float64
CREDIT_LIMIT     float64
PAYMENTS         float64
dtype: object
```

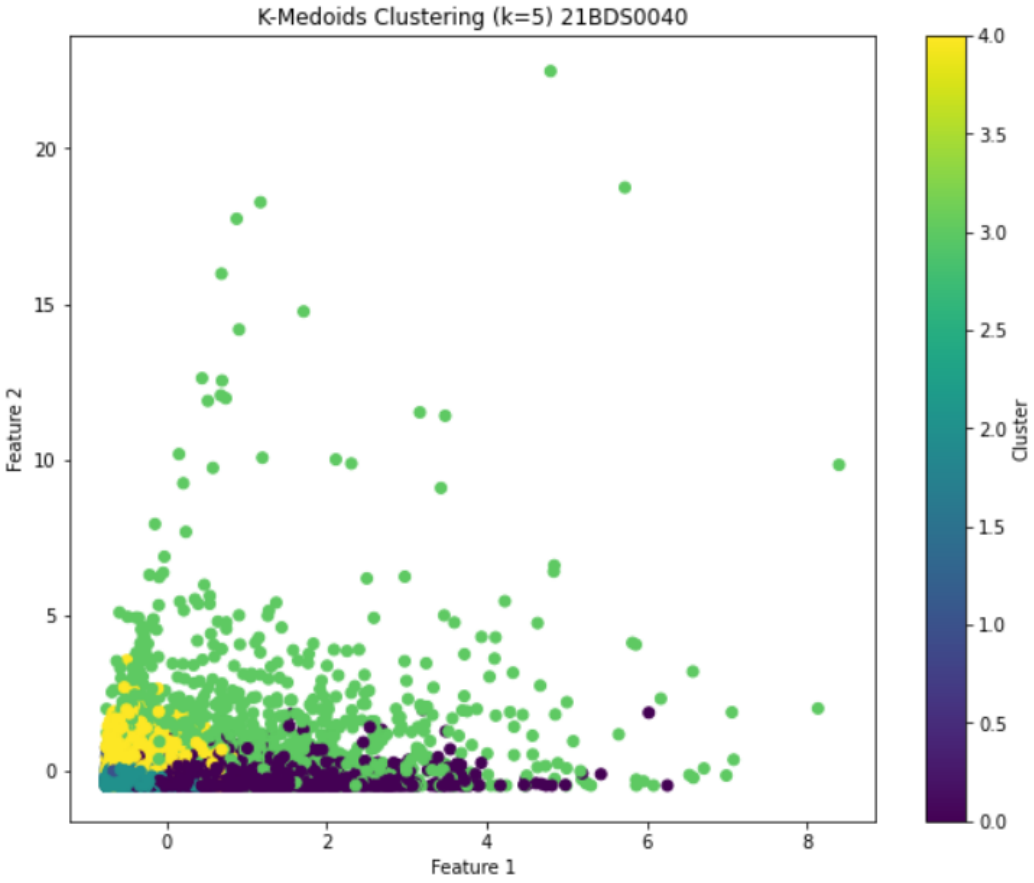
In [16]:
```python
# Varun Sudhir 21BDS0040

# Convert columns to numeric, coercing errors to NaN
for column in selected_features:
    credit_card_data[column] = pd.to_numeric(credit_card_data[column], errors='coerce')

# Impute missing values with the median
for column in selected_features:
    median_value = credit_card_data[column].median()
    credit_card_data[column].fillna(median_value, inplace=True)

# Verify data types after imputation
print(credit_card_data[selected_features].dtypes)
```

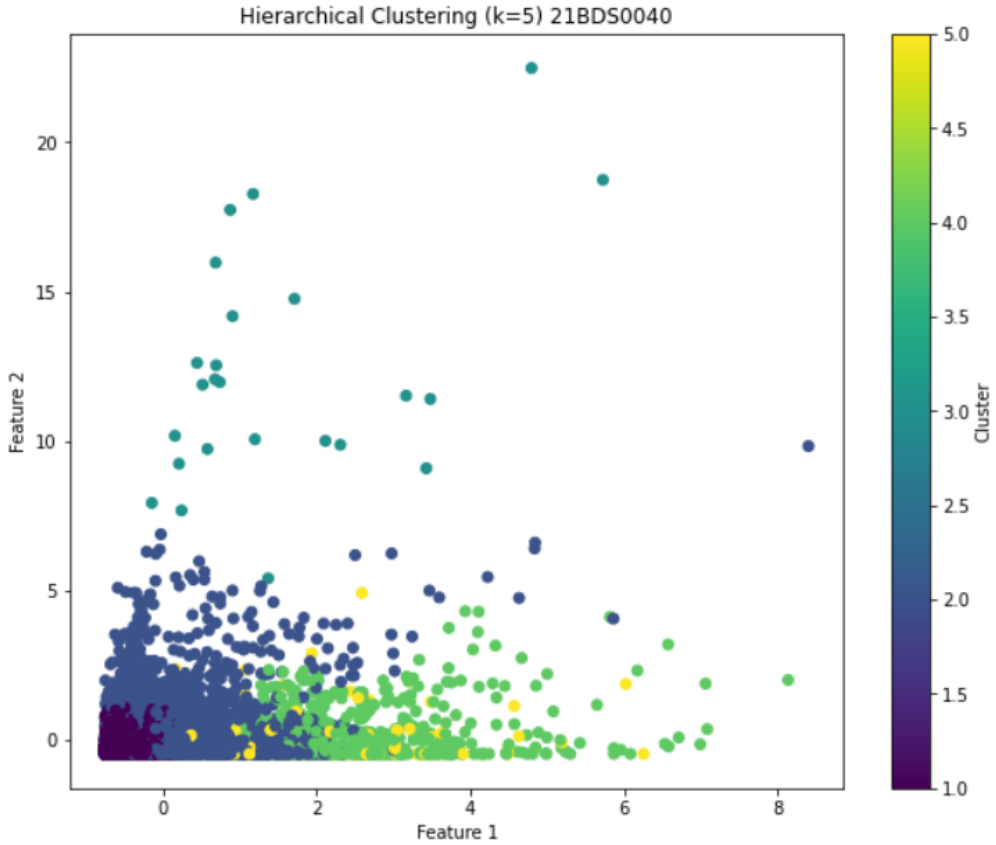
```
BALANCE          float64
PURCHASES        float64
CASH_ADVANCE     float64
CREDIT_LIMIT     float64
PAYMENTS         float64
dtype: object
```

In [17]:
```python
# Varun Sudhir 21BDS0040

# Define the features for clustering
selected_features = ['BALANCE', 'PURCHASES', 'CASH_ADVANCE', 'CREDIT_LIMIT', 'PAYMENTS']

# Normalize the features
scaler = StandardScaler()
normalized_data = scaler.fit_transform(credit_card_data[selected_features])

for num_clusters in [2, 5]:
    # K-Medoids Clustering
    kmedoids_labels = perform_k_medoids(normalized_data, num_clusters)
    visualize_clusters(normalized_data, kmedoids_labels, f'K-Medoids Clustering (k={num_clusters}) 21BDS0040')

    # Hierarchical Clustering
    hierarchical_labels = perform_hierarchical_clustering(normalized_data, num_clusters)
    visualize_clusters(normalized_data, hierarchical_labels, f'Hierarchical Clustering (k={num_clusters}) 21BDS0040')
```


K-Medoids Clustering (k=2) 21BDS0040
Feature 1
Feature 2
Cluster
0
2
4
6
8
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0


Hierarchical Clustering (k=2) 21BDS0040
Feature 1
Feature 2
Cluster
0
2
4
6
8
0
5
10
15
20
1.0
1.2
1.4
1.6
1.8
2.0


K-Medoids Clustering (k=5) 21BDS0040
Feature 1
Feature 2
Cluster
0
2
4
6
8
0
5
10
15
20
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Hierarchical Clustering (k=5) 21BDS0040
Feature 1
Feature 2
Cluster
0
2
4
6
8
0
5
10
15
20
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

In [20]:
```
# Varun Sudhir 21BDS0040

plt.figure(figsize=(10, 7))
dendrogram(linkage(normalized_data, method='ward'))
plt.title('Dendrogram for Hierarchical Clustering ( 21BDS0040 )')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```