Name: Ruchir M. Adnaik

Srn: 202200282

Roll: 08

Batch: B1

DWDM ASSIGNMENT 08

Code:

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import scipy.cluster.hierarchy as shc

from sklearn.cluster import AgglomerativeClustering

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

# Load the dataset

customer\_data = pd.read\_csv('student\_performance\_data.csv')

# Check the shape of the dataset (rows, columns)

print(customer\_data.shape)

# Display the first few rows of the dataset

print(customer\_data.head())

print('---------------------------------------------')

# Select the columns for clustering (ensure they are numeric)

# Here we're including more features for better clustering (e.g., GPA, StudyHoursPerWeek, AttendanceRate)

data = customer\_data[['StudyHoursPerWeek', 'AttendanceRate', 'GPA']].values

print("Selected Data for Clustering:")

print(data)

# Step 1: Standardize the data

scaler = StandardScaler()

data\_scaled = scaler.fit\_transform(data)

# Step 2: Plot the Dendrogram

plt.figure(figsize=(10, 7))

plt.title("Student Dendrograms")

dend = shc.dendrogram(shc.linkage(data\_scaled, method='ward'))

# Show where we might cut the dendrogram for cluster assignment

plt.axhline(y=5, color='r', linestyle='--')  # Adjust this line to fine-tune the number of clusters

plt.show()

# Step 3: Perform Agglomerative Clustering

cluster = AgglomerativeClustering(n\_clusters=5, metric='euclidean', linkage='ward')

labels\_ = cluster.fit\_predict(data\_scaled)

print('---------------------------------------------')

print('Cluster Labels:')

print(labels\_)

# Step 4: Use PCA for Dimensionality Reduction (2D for visualization)

pca = PCA(n\_components=2)

data\_pca = pca.fit\_transform(data\_scaled)

# Step 5: Plot the clustered data using the first two principal components

plt.figure(figsize=(10, 7))

plt.scatter(data\_pca[:, 0], data\_pca[:, 1], c=cluster.labels\_, cmap='rainbow')

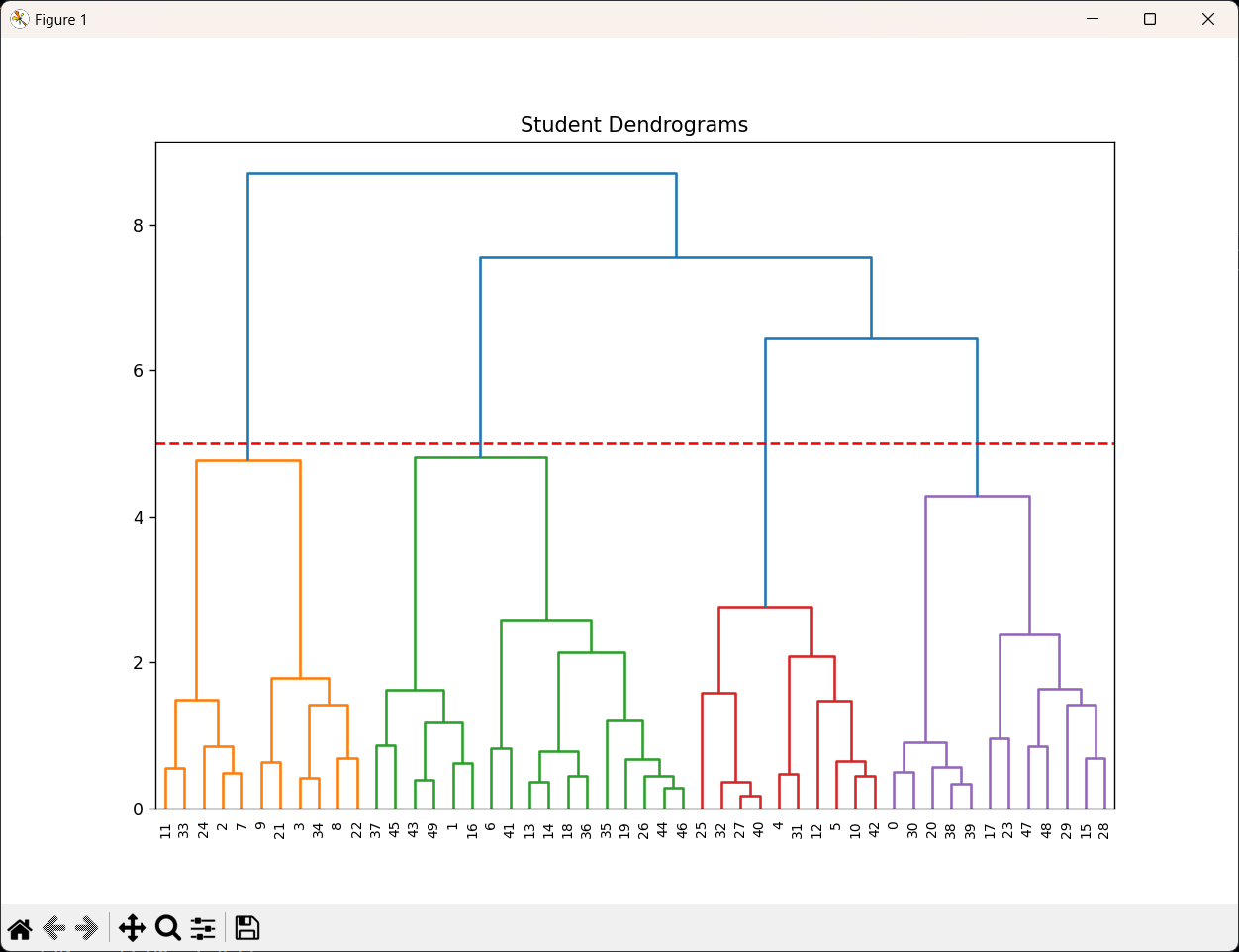
plt.title('Clustered Data (PCA Reduced)')

plt.xlabel('PCA Component 1')

plt.ylabel('PCA Component 2')

plt.show()

OUTPUTS:



A screen shot of a graph

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A number on a black background

Description automatically generated