**Implementation of Clustering algorithm (K-means/K-medoids)**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn\_extra.cluster import KMedoids

np.random.seed(42)

user\_data = pd.DataFrame({

'age': np.random.randint(18, 70, 100),

'income': np.random.normal(50000, 20000, 100),

'spending\_score': np.random.randint(1, 100, 100)

})

user\_data['income'] = np.abs(user\_data['income'])

print("Original Data:")

print(user\_data.head())

print(f"\nIncome Stats - Min: {user\_data['income'].min():.2f}, Max: {user\_data['income'].max():.2f}")

user\_data['income\_bins'] = pd.cut(user\_data['income'], bins=5, labels=['Very Low', 'Low', 'Medium', 'High', 'Very High'])

print("\nAfter Discretization:")

print(user\_data[['income', 'income\_bins']].head(10))

bin\_counts = user\_data['income\_bins'].value\_counts().sort\_index()

print(f"\nBin Counts:\n{bin\_counts}")

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)

plt.hist(user\_data['income'], bins=30, alpha=0.7, color='skyblue', edgecolor='black')

plt.title('Original Income Distribution')

plt.xlabel('Income')

plt.ylabel('Frequency')

plt.subplot(1, 3, 2)

bin\_counts.plot(kind='bar', color='lightcoral', edgecolor='black')

plt.title('Discretized Income Categories')

plt.xlabel('Income Bins')

plt.ylabel('Count')

plt.xticks(rotation=45)

X = user\_data[['age', 'income', 'spending\_score']]

kmeans = KMeans(n\_clusters=4, random\_state=42)

kmeans\_labels = kmeans.fit\_predict(X)

kmedoids = KMedoids(n\_clusters=4, random\_state=42)

kmedoids\_labels = kmedoids.fit\_predict(X)

plt.subplot(1, 3, 3)

plt.scatter(user\_data['age'], user\_data['spending\_score'], c=kmeans\_labels, cmap='viridis')

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=200, c='red', marker='X', label='Centroids')

plt.title('K-means Clustering')

plt.xlabel('Age')

plt.ylabel('Spending Score')

plt.legend()

plt.tight\_layout()

plt.show()

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)

plt.scatter(user\_data['age'], user\_data['spending\_score'], c=kmedoids\_labels, cmap='plasma')

plt.scatter(kmedoids.cluster\_centers\_[:, 0], kmedoids.cluster\_centers\_[:, 1], s=200, c='red', marker='X', label='Medoids')

plt.title('K-medoids Clustering')

plt.xlabel('Age')

plt.ylabel('Spending Score')

plt.legend()

plt.subplot(1, 3, 2)

plt.scatter(user\_data['income'], user\_data['spending\_score'], c=kmeans\_labels, cmap='viridis')

plt.title('K-means: Income vs Spending')

plt.xlabel('Income')

plt.ylabel('Spending Score')

plt.subplot(1, 3, 3)

plt.scatter(user\_data['income'], user\_data['spending\_score'], c=kmedoids\_labels, cmap='plasma')

plt.title('K-medoids: Income vs Spending')

plt.xlabel('Income')

plt.ylabel('Spending Score')

plt.tight\_layout()

plt.show()

print(f"\nDiscretization Summary:")

for bin\_name in ['Very Low', 'Low', 'Medium', 'High', 'Very High']:

bin\_data = user\_data[user\_data['income\_bins'] == bin\_name]['income']

print(f"{bin\_name}: ${bin\_data.min():.0f} - ${bin\_data.max():.0f} ({len(bin\_data)} people)")

print(f"\nClustering Results:")

print(f"K-means cluster sizes: {np.bincount(kmeans\_labels)}")

print(f"K-medoids cluster sizes: {np.bincount(kmedoids\_labels)}")

print(f"K-means centers:\n{kmeans.cluster\_centers\_}")

print(f"K-medoids centers:\n{kmedoids.cluster\_centers\_}")