import pandas as pd sales

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

--------------------------------------------------------------------------------------------------------------------------------------------------

df = pd.read\_csv('sales.csv', encoding='ISO-8859-1')

--------------------------------------------------------------------------------------------------------------------------------------------------

df.head()

--------------------------------------------------------------------------------------------------------------------------------------------------

df.isnull().sum()

--------------------------------------------------------------------------------------------------------------------------------------------------

df = df.select\_dtypes(include=[np.number])

--------------------------------------------------------------------------------------------------------------------------------------------------

df.head()

--------------------------------------------------------------------------------------------------------------------------------------------------

df.fillna(df.mean(), inplace=True)

--------------------------------------------------------------------------------------------------------------------------------------------------

# Normalize the data using StandardScaler scaler = StandardScaler()

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

--------------------------------------------------------------------------------------------------------------------------------------------------

wcss = []

--------------------------------------------------------------------------------------------------------------------------------------------------

for i in range(1, 11): # Test k from 1 to 10

kmeans = KMeans(n\_clusters=i, init='k-means++', max\_iter=300, n\_init=10, random\_state=42) kmeans.fit(df\_scaled)

wcss.append(kmeans.inertia\_)

--------------------------------------------------------------------------------------------------------------------------------------------------

plt.figure(figsize=(10, 6))

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method for Optimal K') plt.xlabel('Number of Clusters')

plt.ylabel('WCSS (Within-Cluster Sum of Squares)') plt.show()

------------------------------------------------------------------------------------------------------------------------------------------------

optimal\_k = 3

------------------------------------------------------------------------------------------------------------------------------------------------

kmeans = KMeans(n\_clusters=optimal\_k, init='k-means++', max\_iter=300, n\_init=10, random\_state=42) y\_kmeans = kmeans.fit\_predict(df\_scaled)

--------------------------------------------------------------------------------------------------------------------------------------------------

df['Cluster'] = y\_kmeans

------------------------------------------------------------------------------------------------------------------------------------------------

df.head()

-----------------------------------------------------------------------------------------------------------------------------------------------

from sklearn.decomposition import PCA

------------------------------------------------------------------------------------------------------------------------------------------------

pca = PCA(n\_components=2)

pca\_components = pca.fit\_transform(df\_scaled)

------------------------------------------------------------------------------------------------------------------------------------------------

plt.figure(figsize=(10, 6))

plt.scatter(pca\_components[:, 0], pca\_components[:, 1], c=y\_kmeans, cmap='viridis') plt.title('K-Means Clustering (2D PCA)')

plt.xlabel('PCA Component 1')

plt.ylabel('PCA Component 2') plt.colorbar()

plt.show()

------------------------------------------------------------------------------------------------------------------------------------------------