APPLIED DATA SCIENCE

**CUSTOMER SEGMENTATION USING DATA SCIENCE-PHASE 3**

**TEAM MEMBERS:**

**1.MOUSIKA.S-TEAM LEADER (731221104025)  
 2. YAMUNA.K (731221104040)  
 3. DURGA.K(731221104012)  
 4. PRIYA DHARSHINI.S(731221104027)  
 5. NANDINI.M(731221104305)**

**BEGIN THE CUSTOMER SEGMENTATION PROJECT BY LOADING AND PREPROCESSING THE CUSTOMER DATA.**

**INTRODUCTION:**

In this part you will begin building your project by loading and preprocessing the dataset. Begin the customer segmentation project by loading and preprocessing the customer data. Collect and preprocess the customer data for analysis.

**STEP 1: DATA COLLECTION**

First, you need to collect the customer data from a reliable source. Ensure the data is in a structured format, such as a CSV file. You can use libraries like Pandas to load the data into a Data Frame.

***CODING:***

**import pandas as pd**

**# Load the customer data**

**data = pd.read\_csv('** **C:\Users\ASUS\Downloads\mall dataset.csv') # Replace 'customer\_data.csv' with the actual data file path**

**STEP 2: DATA CLEANING**

Check for missing values in the dataset and handle them appropriately. For simplicity, we will drop rows with missing values. You can choose other strategies like imputation based on your project requirements.

***CODING:***

**# Drop rows with missing values**

**data.dropna(inplace=True)**

**STEP 3: FEATURE SELECTION**

PCA inherently reduces dimensionality, but you can perform feature selection before PCA if needed. Here, we'll use all available features.

**STEP 4: STANDARDIZATION**

Standardize the data to make it suitable for PCA. Standardization ensures that all features have a mean of 0 and a standard deviation of 1.

***CODING:***

**from sklearn.preprocessing import StandardScaler**

**# Separate features from the target variable (if applicable)**

**X = data.drop('target\_variable', axis=1) # Replace 'target\_variable' with the actual target column name**

**# Standardize the data**

**scaler = StandardScaler()**

**X\_scaled = scaler.fit\_transform(X)**

**STEP 5: PCA (PRINCIPAL COMPONENT ANALYSIS)**

Apply PCA to reduce the dimensionality of the data. You can choose the number of principal components to retain based on your project's goals.

***CODING:***

**from sklearn.decomposition import PCA**

**# Choose the number of components you want to keep**

**n\_components = 2**

**# Apply PCA**

**pca = PCA(n\_components=n\_components)**

**X\_pca = pca.fit\_transform(X\_scaled)**

Now, X\_pca contains the reduced dimensionality data, and you can use it for customer segmentation. To get insights into the explained variance and principal components, you canaccess them as follows

***CODING:***

**explained\_variance = pca.explained\_variance\_ratio\_**

**print("Explained Variance:", explained\_variance)**

**STEP 6: CUSTOMER SEGMENTATION USING CLUSTERING (K-MEANS)**

Now that you have the reduced features in X\_pca, you can apply a clustering algorithm. Here, we'll use K-Means as an example:

***CODING:***

**from sklearn.cluster import KMeans**

**# Choose the number of clusters (you may need to determine the optimal number using methods like the Elbow method or Silhouette analysis)**

**n\_clusters = 3**

**# Apply K-Means clustering**

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

**kmeans.fit(X\_pca)**

**# Assign cluster labels to the original data**

**data['Cluster'] = kmeans.labels\_**

You can choose the number of clusters (n\_clusters) based on your specific use case and the results of your analysis. You may need to experiment with different values to find the optimal number.

**STEP 7: VISUALIZE THE CUSTOMER SEGMENTATION**

To visualize the customer segmentation, you can use various plotting libraries such as Matplotlib or Seaborn. In this example, we'll create a scatter plot to visualize how customers are grouped into clusters based on the first two principal components:

***CODING:***

**import matplotlib.pyplot as plt**

**# Plot the customer segmentation**

**plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=data['Cluster'], cmap='viridis')**

**plt.xlabel('Principal Component 1')**

**plt.ylabel('Principal Component 2')**

**plt.title('Customer Segmentation using PCA and K-Means')**

**plt.show()**

This scatter plot will show how customers are distributed in the reduced feature space, which was obtained through PCA, and color-coded by their cluster assignments.

**CONCLUSION:**

After this preprocessing, you can proceed with clustering or other segmentation techniques using the reduced-dimension data in X\_pca. Make sure to choose an appropriate clustering algorithm and decide on the number of clusters based on your specific business objectives and the results of your analysis.