PHASE 3

CUSTOMER SEGMENTATION USING DATA SCIENCE

Development Part- I

Thoufeeq A - 71772117146
Sudharsan M - 71772117144
Sandhiya S - 71772117138
Srikanth B - 71772117143

Description:

This document outlines the loading and preprocessing steps to develop a data science model for customer segmentation. The main goal of this initiative is to assist businesses in gaining deeper insights into their customer base and tailoring marketing strategies to match each segment's unique characteristics.

Loading and Preprocessing the 'Mall_Customers' Dataset:

INTRODUCTION:

In this document, we will provide a detailed guide on how to load and preprocess the 'Mall_Customers' dataset for further analysis. The dataset contains information about customers of a mall and is stored in a CSV file named 'Mall_Customers.csv.'

STEP 1: Importing Necessary Libraries

Before diving into the dataset, let's make sure we have the essential libraries installed. In this guide, we'll utilise the following Python libraries:

- **Pandas:** This library is a powerhouse for data manipulation and analysis.
- **NumPy:** It's indispensable for various numerical operations.
- **Scikit-Learn (sklearn)**: This library is a cornerstone for data preprocessing.

Code:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
```

STEP 2: Loading the Dataset

We load the dataset into a Pandas DataFrame as 'Mall_Customers' dataset from the CSV file.

Code:

```
data = pd.read_csv('Mall_Customers.csv')
```

STEP 3: Data Extraction and Selection

In this step, we extract and select the relevant columns from the dataset. Assuming that the first column in the dataset contains '**CustomerID**' ,we extract this column for reference:

Code:

```
customer_ids = data['CustomerID']
```

This is used for the extraction of data from the CustomerID column.

STEP 4: Handling Missing Values

It's common to encounter missing values in datasets.To overcome these difficulties and errors, we impute missing values with the mean of each feature using the Simple Imputer from scikit-learn.

Code:

```
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)
```

STEP 5: Standardizing the Data

Standardization is crucial for many machine learning techniques, including t-SNE. It ensures that all features have a mean of 0 and a standard deviation of 1. We use the StandardScaler for this purpose.

Code:

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)
```

At this point, the 'X_scaled' variable contains the standardized data, which is now ready for use in further analysis, such as t-SNE visualization and clustering.

CONCLUSION:

In this document, we have provided a step-by-step guide on how to load and preprocess the '**Mall_Customers**' dataset for data analysis. Following these steps, you can ensure that your data is ready for various machine learning and data visualization techniques.