Data Collection and Preprocessing Phase

Date	June 2024
Team ID	740295
Project Title	Ecommerce shipping prediction using Machine Learning
Maximum Marks	6 Marks

PreparationTemplate

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc. In this project we have used .csv data.
Data Preparation	These are the general steps of pre-processing the data before using it for machine learning
Handling missing values	We use Handling missing values For checking the null values
Handling categorical data	As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding
Handling Outliers in Data	With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of numerical features with some mathematical formula.

Data Preparation

Collect the dataset	Please refer to the link given below to download the dataset. Link: https://www.kaggle.com/datasets/prachi13/customer-analytics?select=Train.csv

Importing the libraries	`import pandas as pd`
	`import numpy as np`
	`import seaborn as sns`
	`import matplotlib.pyplot as plt`
	`from imblearn.over_sampling import SMOTE`
	`from sklearn.model_selection import train_test_split`
	`from sklearn.metrics import classification_report`
	`from sklearn.metrics import confusion_matrix, precision_sc`
	`from sklearn.ensemble import RandomForestClassifier`
	`import warnings`
	`warnings.filterwarnings("ignore")`

Loading Data	We use the code `df=pd.read_csv("/content/Train.csv")` For reading the dataset

Handling missing values	`df.isnull().sum()` ID 0 Warehouse_block 0 Mode_of_Shipment 0 Customer_care_calls 0 Customer_rating 0 Cost_of_the_Product 0 Prior_purchases 0 Product_importance 0 Gender 0 Discount_offered 0 Weight_in_gms 0 Reached.on.Time_Y.N 0 dtype: int64

Handling Categorical values	`label_map={}`
	`for i in df.columns:`
	`  if str(df[i].dtype) == 'object':`
	`  temp={}`
	`  cats=df[i].unique()`
	`  for index in range(len(cats)):`
	`  temp[cats [index]]=index`
	`  label_map[i]=temp`
	`  #Labeling`
	`  df[i]=df[i].map(temp)`
	
	`print(label_map)`
	
	`{'Warehouse_block': {'D': 0, 'F': 1, 'A': 2, 'B': 3, 'C': 4}`

Handling Outliers	`c=0`
	`plt.figure(figsize=(18, 10))`
	`for i in df.drop(columns=['Warehouse_block', 'Mode_of_Shipment','Product_importar`
	`  if str(df[i].dtype)=='object':`
	`  continue`
	`  plt.subplot(2, 3, c+1)`
	`  plt.boxplot(df[i])`
	`  plt.title(i)`
	`  c+=1`
	`plt.show()`