Data Collection and Preprocessing Phase

Date	10 July 2024
Team ID	SWTID1720369851
Project Title	Ecommerce Shipping Prediction
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	

```
data.shape
```

```
(10999, 12)
```

```
data.describe()
```

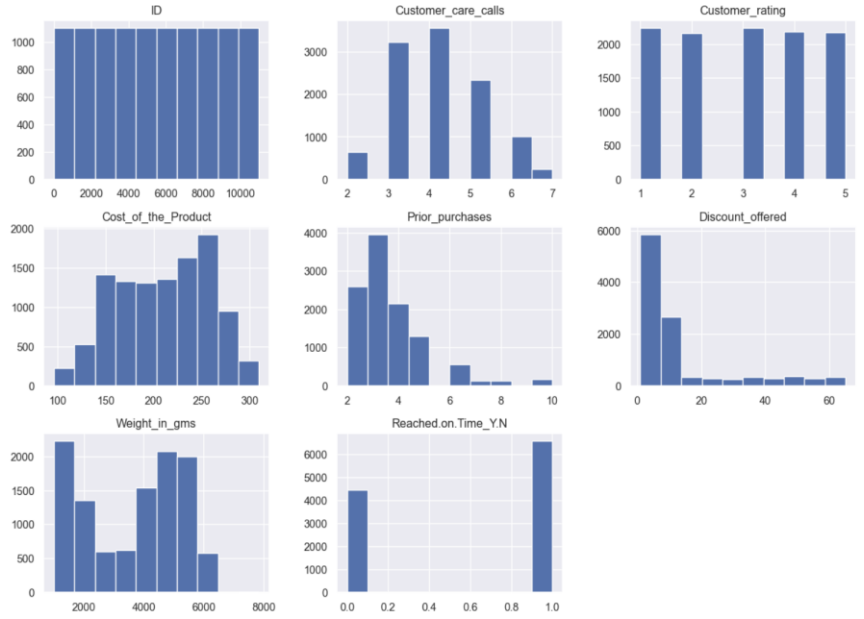
	ID	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
count	10999.00000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000
mean	5500.00000	4.054459	2.990545	210.196836	3.567597	13.373216	3634.016729	0.596691
std	3175.28214	1.141490	1.413603	48.063272	1.522860	16.205527	1635.377251	0.490584
min	1.00000	2.000000	1.000000	96.000000	2.000000	1.000000	1001.000000	0.000000
25%	2750.50000	3.000000	2.000000	169.000000	3.000000	4.000000	1839.500000	0.000000
50%	5500.00000	4.000000	3.000000	214.000000	3.000000	7.000000	4149.000000	1.000000
75%	8249.50000	5.000000	4.000000	251.000000	4.000000	10.000000	5050.000000	1.000000
max	10999.00000	7.000000	5.000000	310.000000	10.000000	65.000000	7846.000000	1.000000

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   10999 non-null  int64
 1   Warehouse_block      10999 non-null  object
 2   Mode_of_Shipment     10999 non-null  object
 3   Customer_care_calls  10999 non-null  int64
 4   Customer_rating      10999 non-null  int64
 5   Cost_of_the_Product  10999 non-null  int64
 6   Prior_purchases      10999 non-null  int64
 7   Product_importance   10999 non-null  object
 8   Gender               10999 non-null  object
 9   Discount_offered     10999 non-null  int64
 10  Weight_in_gms        10999 non-null  int64
 11  Reached.on.Time_Y.N  10999 non-null  int64
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```

Univariate Analysis

```
data.select_dtypes(include=[np.number]).hist(bins=10, figsize=(15,10))
plt.show()
```



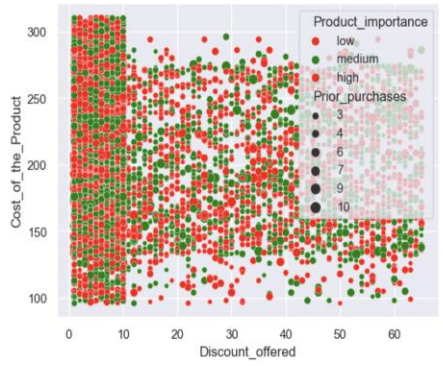
Bivariate Analysis

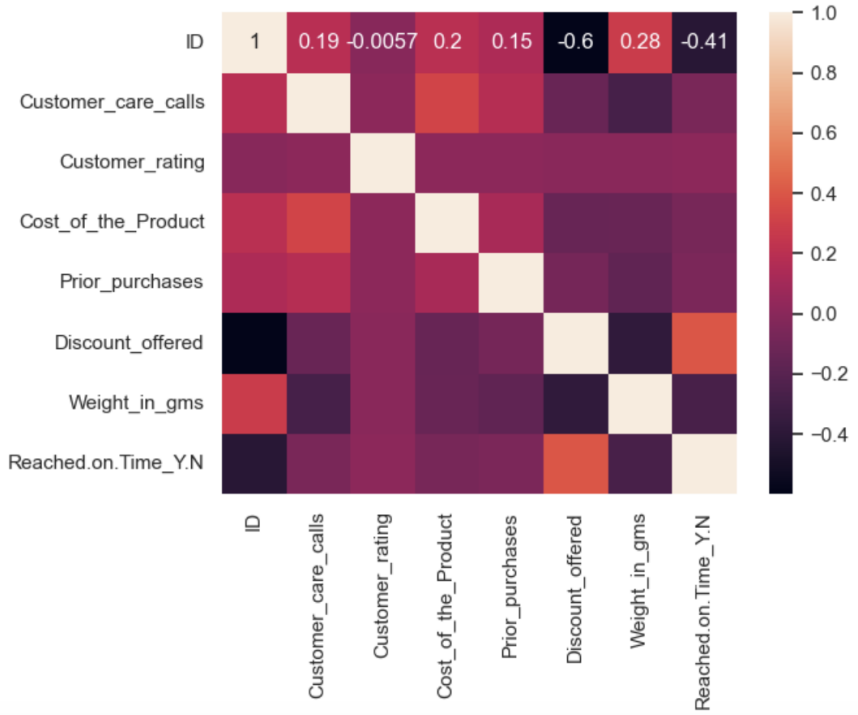
```
data.corr(numeric_only=True)
```

	ID	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Discount_offered	Weight_in_gms	Reached.on.Tin
ID	1.000000	0.188998	-0.005722	0.196791	0.145369	-0.598278	0.278312	-0.
Customer_care_calls	0.188998	1.000000	0.012209	0.323182	0.180771	-0.130750	-0.276615	-0.
Customer_rating	-0.005722	0.012209	1.000000	0.009270	0.013179	-0.003124	-0.001897	0.
Cost_of_the_Product	0.196791	0.323182	0.009270	1.000000	0.123676	-0.138312	-0.132604	-0.
Prior_purchases	0.145369	0.180771	0.013179	0.123676	1.000000	-0.082769	-0.168213	-0.
Discount_offered	-0.598278	-0.130750	-0.003124	-0.138312	-0.082769	1.000000	-0.376067	0.
Weight_in_gms	0.278312	-0.276615	-0.001897	-0.132604	-0.168213	-0.376067	1.000000	-0.2
Reached.on.Time_Y.N	-0.411822	-0.067126	0.013119	-0.073587	-0.055515	0.397108	-0.268793	1.0

```
# Scatterplot
sns.scatterplot(x='Discount_offered',y='Cost_of_the_Product',data=data, hue='Product_importance', size='Prior_purchases', palette=['red','gree
```

```
<Axes: xlabel='Discount_offered', ylabel='Cost_of_the_Product'>
```

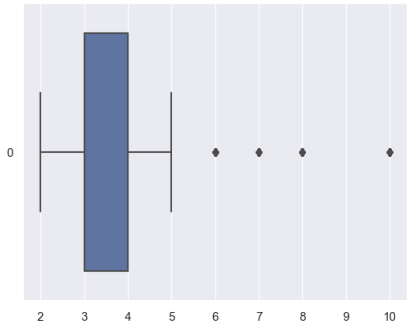


Multivariate Analysis	```# Heat Map
	hm=data.corr(numeric_only=True)
	sns.heatmap(hm, annot=True)```
	```<Axes: >```
	 

Outliers and Anomalies

```
sns.boxplot(data['Prior_purchases'], orient='h')
```

```
<Axes: >
```




```
Q1=data['Prior_purchases'].quantile(0.25)
Q3=data['Prior_purchases'].quantile(0.75)
IQR=Q3-Q1
lower_limit=Q1-1.5*IQR
upper_limit=Q3-1.5*IQR
data[(data.Prior_purchases<lower_limit)|(data.Prior_purchases>upper_limit)]
data=data[(data.Prior_purchases>lower_limit)&(data.Prior_purchases<upper_limit)]
data
```

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Gender
1	2	F	Flight	4	5	216	2	low	M
7	8	F	Flight	4	1	233	2	low	F
10	11	C	Flight	3	4	189	2	medium	M
16	17	C	Flight	3	4	143	2	medium	F
20	21	A	Ship	3	3	161	2	medium	F
...	...	...	...	...	...	...	...	...	...
10892	10893	A	Ship	4	5	167	2	low	F
10897	10898	F	Ship	4	3	242	2	low	M
10898	10899	A	Ship	4	1	222	2	high	M
10901	10902	F	Ship	4	4	201	2	medium	M
10904	10905	A	Ship	4	2	218	2	low	M

2599 rows × 12 columns

Data Preprocessing Code Screenshots

Loading Data

```
# Load the data

data = pd.read_csv("Train.csv")
```

Handling Missing Data	`data.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
	
	`data.duplicated().sum()`
	
	0

Data Transformation	**Encoding the categorical variables**
	
	`le = LabelEncoder()`
	
	`def Label_Enc(col):`
	`  Categorical_col[col] = le.fit_transform(Categorical_col[col])`
	
	`for i in ['Warehouse_block', 'Mode_of_Shipment', 'Product_importance', 'Gender']:`
	`  Label_Enc(i)`
	
	`Categorical_col.head()`
	
	Warehouse_block Mode_of_Shipment Product_importance Gender
	0 3 0 1 0
	1 4 0 1 1
	2 0 0 1 1
	3 1 0 2 1
	4 2 0 2 0

Feature Engineering	`Numerical_col.drop(columns = ["ID"],axis = 1,inplace = True)`
	`Numerical_col.head()`
	
	Customer_care_calls Customer_rating Cost_of_the_Product Prior_purchases Discount_offered Weight_in_gms Reached.on.Time_Y.N
	0 4 2 177 3 44 1233 1
	1 4 5 216 2 59 3088 1
	2 2 2 183 4 48 3374 1
	3 3 3 176 4 10 1177 1
	4 2 2 184 3 46 2484 1

Save Processed Data	`# Save to a CSV file` `data.to_csv('Train(new).csv', index=False)`