Data Collection and Preprocessing Phase

Date	6 JULY 2024
Team ID	SWTID1720097765
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	6 Marks

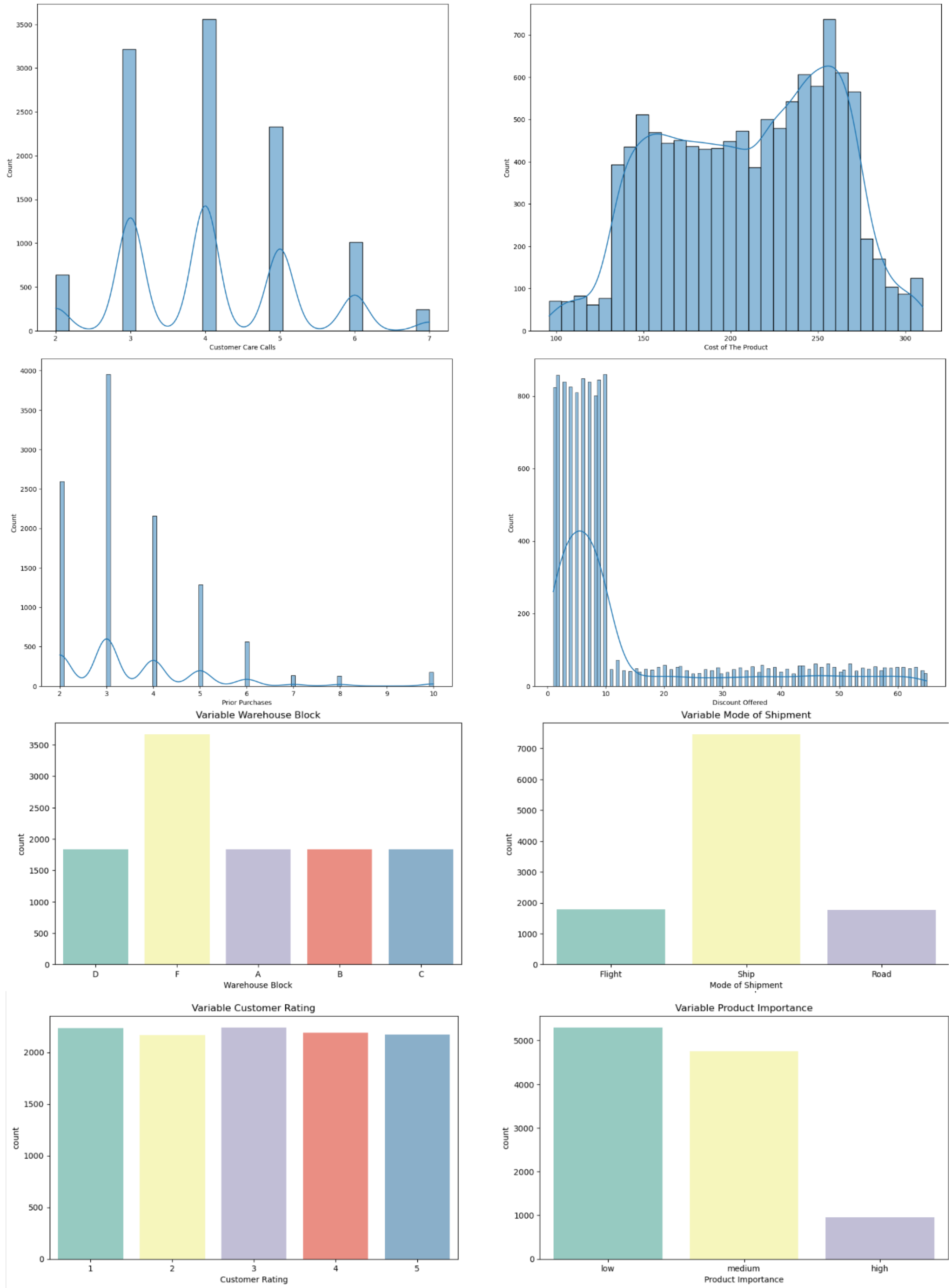
Data Exploration and Preprocessing Template

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employedforpreprocessingtaskslikenormalizationandfeatureengineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

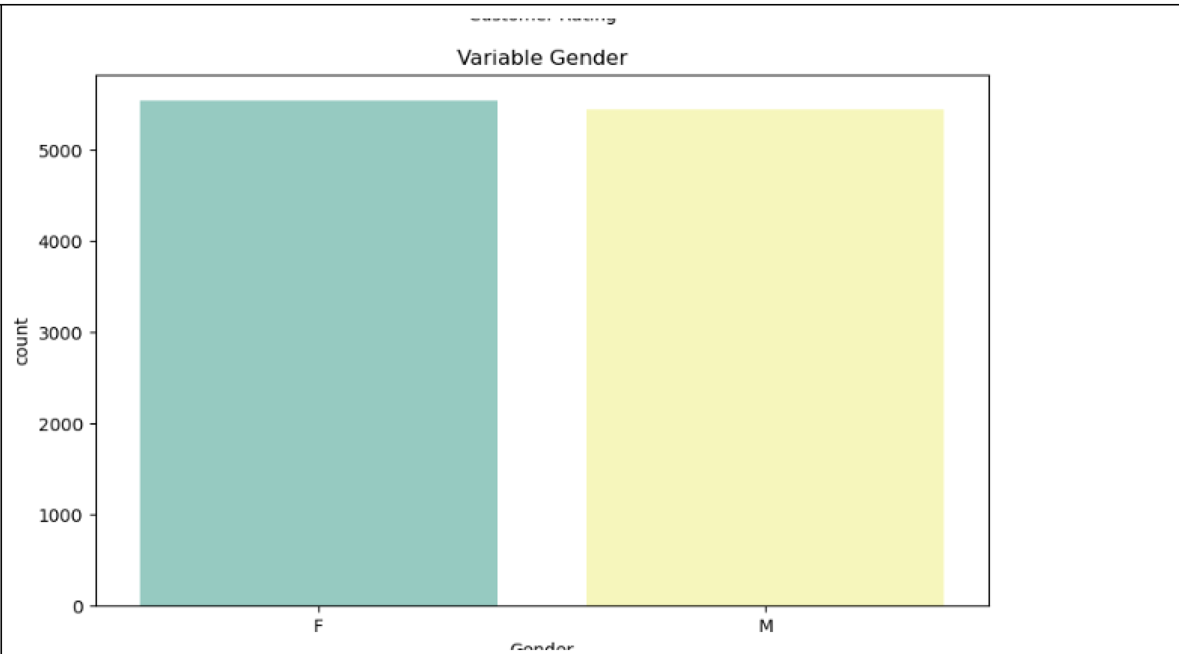
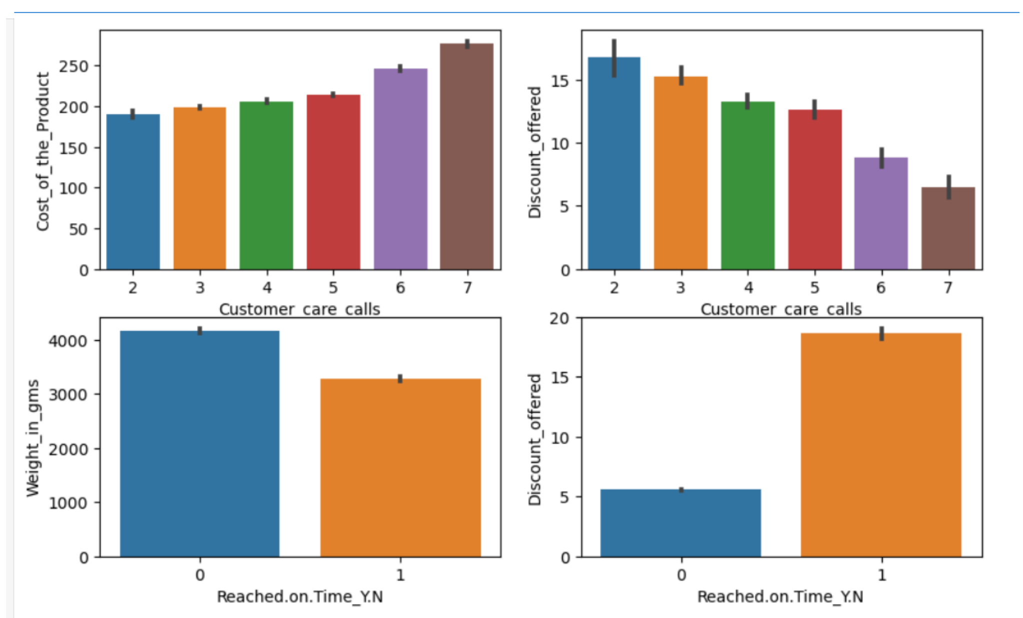
<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>
Dimension:
[4]: <code>print(data.shape)</code>
<code>(10999, 12)</code>
Descriptive analysis:
<code>data.describe()</code>
<table>
<tr><th></th><th>ID</th><th>Customer_care_calls</th><th>Customer_rating</th><th>Cost_of_the_Product</th><th>Prior_purchases</th><th>Discount_offered</th><th>Weight_in_gms</th><th>Reached.on.Time_Y.N</th></tr>
<tr><td>count</td><td>10999.00000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td></tr>
<tr><td>mean</td><td>5500.00000</td><td>4.054459</td><td>2.990545</td><td>210.196836</td><td>3.567597</td><td>13.373216</td><td>3634.016729</td><td>0.596691</td></tr>
<tr><td>std</td><td>3175.28214</td><td>1.141490</td><td>1.413603</td><td>48.063272</td><td>1.522860</td><td>16.205527</td><td>1635.377251</td><td>0.490584</td></tr>
<tr><td>min</td><td>1.00000</td><td>2.000000</td><td>1.000000</td><td>96.000000</td><td>2.000000</td><td>1.000000</td><td>1001.000000</td><td>0.000000</td></tr>
<tr><td>25%</td><td>2750.50000</td><td>3.000000</td><td>2.000000</td><td>169.000000</td><td>3.000000</td><td>4.000000</td><td>1839.500000</td><td>0.000000</td></tr>
<tr><td>50%</td><td>5500.00000</td><td>4.000000</td><td>3.000000</td><td>214.000000</td><td>3.000000</td><td>7.000000</td><td>4149.000000</td><td>1.000000</td></tr>
<tr><td>75%</td><td>8249.50000</td><td>5.000000</td><td>4.000000</td><td>251.000000</td><td>4.000000</td><td>10.000000</td><td>5050.000000</td><td>1.000000</td></tr>
<tr><td>max</td><td>10999.00000</td><td>7.000000</td><td>5.000000</td><td>310.000000</td><td>10.000000</td><td>65.000000</td><td>7846.000000</td><td>1.000000</td></tr>
</table>
</td></tr>
</table>

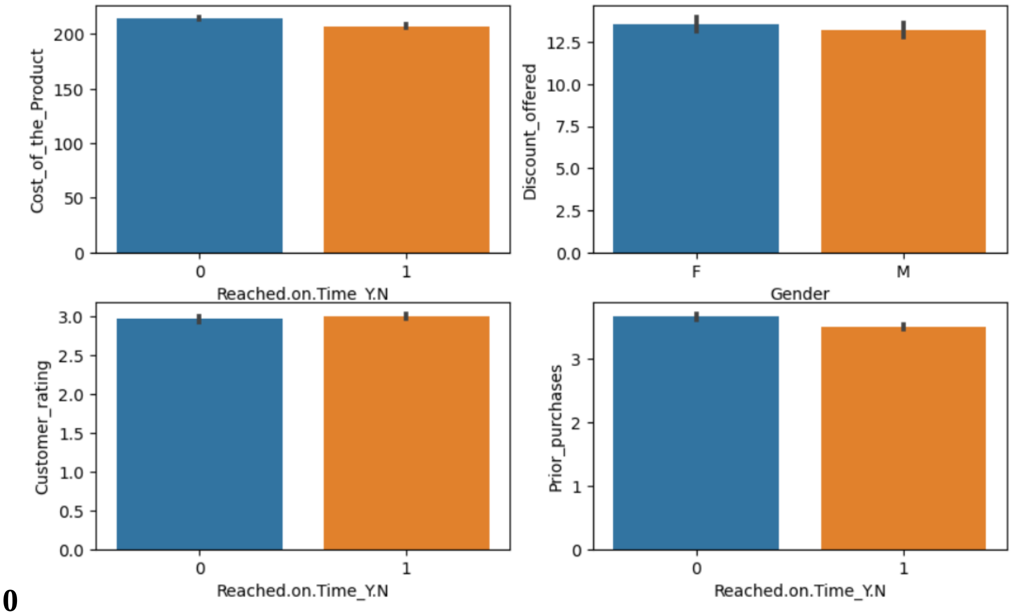
<table>
<tr>
<td>Univariate Analysis</td>
<td>

Text(0.5, 0, 'Discount Offered')

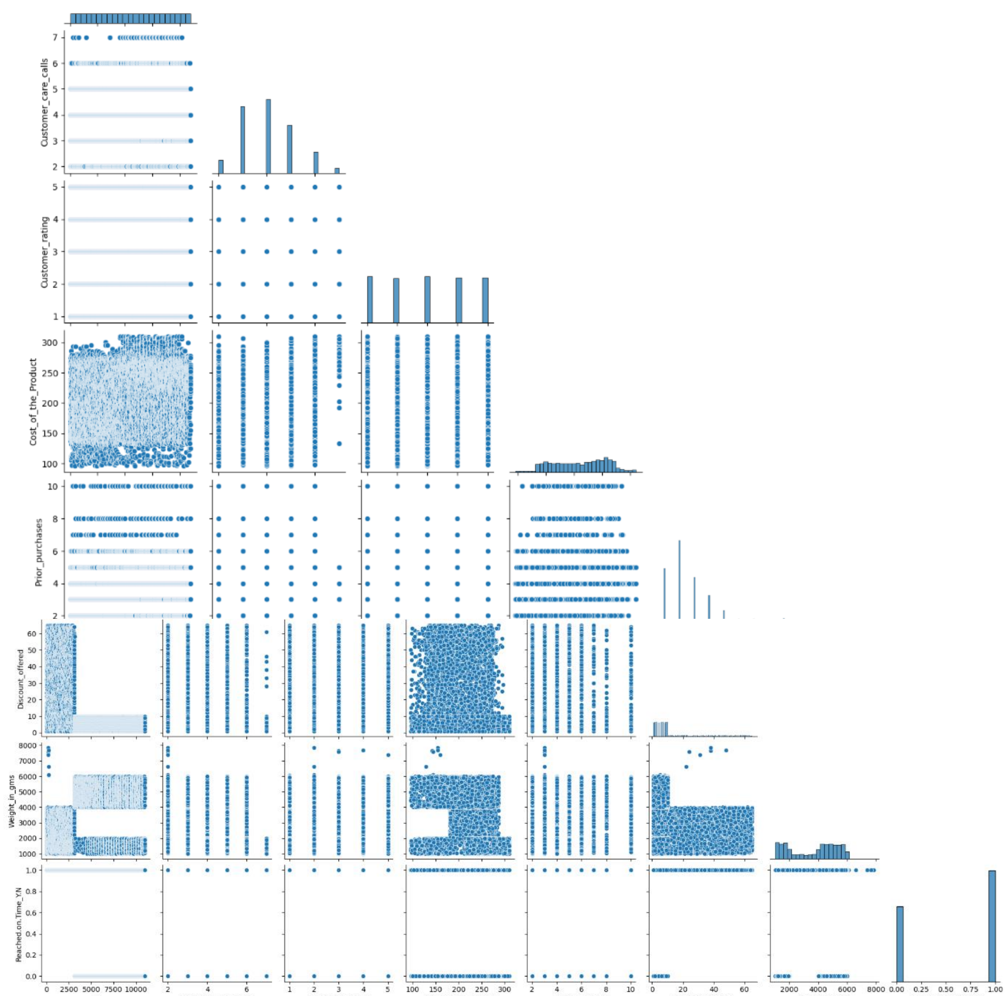


</td>
</tr>
</table>

<table>
<tr><td></td><td>
</td></tr>
<tr><td>Bivariate Analysis</td><td>
</td></tr>
</table>


Cost_of_the_Product
Reached.on.Time_Y.N
Discount_offered
Gender
F
M
Customer_rating
Prior_purchases
Reached.on.Time_Y.N
Reached.on.Time_Y.N

0


Customer_care_calls
Customer_rating
Cost_of_the_Product
Prior_purchases
Discount_offered
Weight_in_gms
Reached.on.Time_Y.N
ID
Customer_care_calls
Customer_rating
Cost_of_the_Product
Prior_purchases
Discount_offered
Weight_in_gms
Reached.on.Time_Y.N

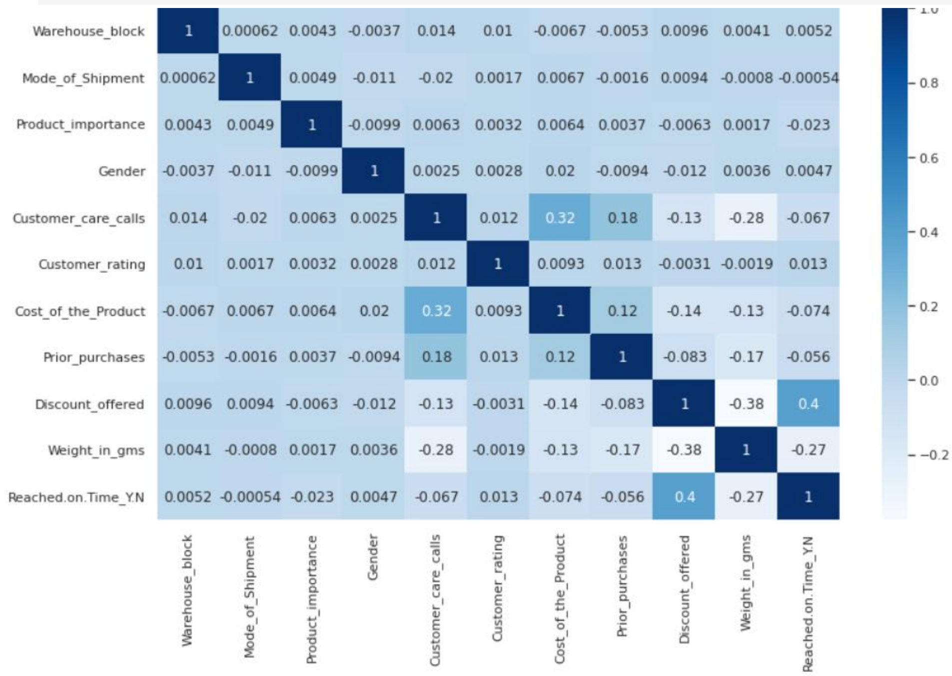
Multivariate Analysis

Multivariate analysis

```
[16]: hm=data.corr(numeric_only=True)
hm
```

[16]:

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance
ID	1.000000	0.000070	-0.002459	0.188998	-0.005722	0.196791	0.145369	0.029081
Warehouse_block	0.000070	1.000000	0.000617	0.014496	0.010169	-0.006679	-0.005262	0.004260
Mode_of_Shipment	-0.002459	0.000617	1.000000	-0.020164	0.001679	0.006681	-0.001640	0.004911
Customer_care_calls	0.188998	0.014496	-0.020164	1.000000	0.012209	0.323182	0.180771	0.006273
Customer_rating	-0.005722	0.010169	0.001679	0.012209	1.000000	0.009270	0.013179	0.003157
Cost_of_the_Product	0.196791	-0.006679	0.006681	0.323182	0.009270	1.000000	0.123676	0.006366
Prior_purchases	0.145369	-0.005262	-0.001640	0.180771	0.013179	0.123676	1.000000	0.003662
Product_importance	0.029081	0.004260	0.004911	0.006273	0.003157	0.006366	0.003662	1.000000
Gender	-0.001695	-0.003700	-0.011288	0.002545	0.002775	0.019759	-0.009395	-0.009865
Discount_offered	-0.598278	0.009569	0.009364	-0.130750	-0.003124	-0.138312	-0.082769	-0.006251
Weight_in_gms	0.278312	0.004086	-0.000797	-0.276615	-0.001897	-0.132604	-0.168213	0.001652
Reached.on.Time_Y.N	-0.411822	0.005214	-0.000535	-0.067126	0.013119	-0.073587	-0.055515	-0.023483



Outliers and Anomalies

Checking Outliers

```python
def check_outliers(arr):
    Q1 = np.percentile(arr, 25,interpolation = 'midpoint')
    Q3 = np.percentile(arr, 75,interpolation = 'midpoint')
    IQR = Q3 - Q1

    #Above Upper bound
    upper=Q3+1.5*IQR
    upper_array=np.array(arr>=upper)
    print(' '*3,len(upper_array[upper_array == True]),'are over the upper bound:',upper)

    #Below Lower bound
    lower=Q1-1.5*IQR
    lower_array=np.array(arr<=lower)
    print(' '*3,len(lower_array[lower_array == True]),'are less than the lower bound:',lower,'\n')

for i in data.drop(columns=[
                            'Warehouse_block','Mode_of_Shipment','Product_importance','Gender','Reached.on.Time_Y.N','ID'
                           ]).columns:
    if str(data[i].dtype)=='object':
        continue
    print(i)
    check_outliers(data[i])
```

```
Customer_care_calls
    0 are over the upper bound: 8.0
    0 are less than the lower bound: 0.0

Customer_rating
    0 are over the upper bound: 7.0
    0 are less than the lower bound: -1.0

Cost_of_the_Product
    0 are over the upper bound: 374.0
    0 are less than the lower bound: 46.0

Prior_purchases
    1003 are over the upper bound: 5.5
    0 are less than the lower bound: 1.5

Discount_offered
    2262 are over the upper bound: 19.0
    0 are less than the lower bound: -5.0

Weight_in_gms
    0 are over the upper bound: 9865.75
    0 are less than the lower bound: -2976.25
```

Data Preprocessing Code Screenshots

Loading Data

```python
data=pd.read_csv("train.csv")
data
```

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Gender	Discount_
0	1	D	Flight	4	2	177	3	low	F	
1	2	F	Flight	4	5	216	2	low	M	
2	3	A	Flight	2	2	183	4	low	M	
3	4	B	Flight	3	3	176	4	medium	M	
4	5	C	Flight	2	2	184	3	medium	F	
...	...	...	...	...	...	...	...	...	...	
10994	10995	A	Ship	4	1	252	5	medium	F	
10995	10996	B	Ship	4	1	232	5	medium	F	
10996	10997	C	Ship	5	4	242	5	low	F	
10997	10998	F	Ship	5	2	223	6	medium	M	
10998	10999	D	Ship	2	5	155	5	low	F	

10999 rows × 12 columns

<table>
<tr>
<td>Handling Missing Data</td>
<td>

```
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
```

</td>
</tr>
<tr>
<td>Data Transformation</td>
<td>

Encoding

```
le = LabelEncoder()
data['Warehouse_block']=le.fit_transform(data['Warehouse_block'])
data['Mode_of_Shipment']=le.fit_transform(data['Mode_of_Shipment'])
data['Product_importance']=le.fit_transform(data['Product_importance'])
data['Gender']=le.fit_transform(data['Gender'])
data['Reached.on.Time_Y.N']=le.fit_transform(data['Reached.on.Time_Y.N'])
```

Scaling

```
sc=StandardScaler()
names=x.columns
x=sc.fit_transform(x)
x=pd.DataFrame(x,columns=names)
```

</td>
</tr>
</table>