```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import norm
```

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv?1641285094
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv?1641285094
To: /content/walmart_data.csv?1641285094
100% 23.0M/23.0M [00:00<00:00, 88.7MB/s]
```

1. (a)Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

```
df=pd.read_csv('walmart_data.csv?1641285094')
df
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	
550064	1006035	P00375436	F	26-35	1	C	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   User_ID                     550068 non-null  int64
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64
 8   Product_Category            550068 non-null  int64
 9   Purchase                    550068 non-null  int64
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

```
df['User_ID']=df['User_ID'].astype('object')
df['Occupation']=df['Occupation'].astype('object')
df['Marital_Status']=df['Marital_Status'].astype('object')
df['Product_Category']=df['Product_Category'].astype('object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
```

```
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype 
---  ------                      --------------   ----- 
 0   User_ID                     550068 non-null  object
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  object
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  object
 8   Product_Category            550068 non-null  object
 9   Purchase                    550068 non-null  int64 
dtypes: int64(1), object(9)
memory usage: 42.0+ MB
```

(b).Non-Graphical Analysis: Value counts and unique attributes

```
df.value_counts()
```

```
User_ID  Product_ID  Gender  Age    Occupation  City_Category  Stay_In_Current_City_Years  Marital_Status  Product_Category  Purchase
1000001  P00000142   F       0-17   10          A              2                           0               3                 13650
1
1004007  P00105342   M       36-45  12          A              1                           1               1                 11668
1
         P00115942   M       36-45  12          A              1                           1               8                 9800
1
         P00115142   M       36-45  12          A              1                           1               1                 11633
1
         P00114942   M       36-45  12          A              1                           1               1                 19148
1
                                                                                                                                       
..
1001973  P00265242   M       26-35  1           A              0                           0               5                 8659
1
         P00226342   M       26-35  1           A              0                           0               11                6112
1
         P00198042   M       26-35  1           A              0                           0               11                5915
1
         P00129842   M       26-35  1           A              0                           0               6                 16101
1
1006040  P00349442   M       26-35  6           B              2                           0               6                 16389
1
Length: 550068, dtype: int64
```

```
df.nunique()
```

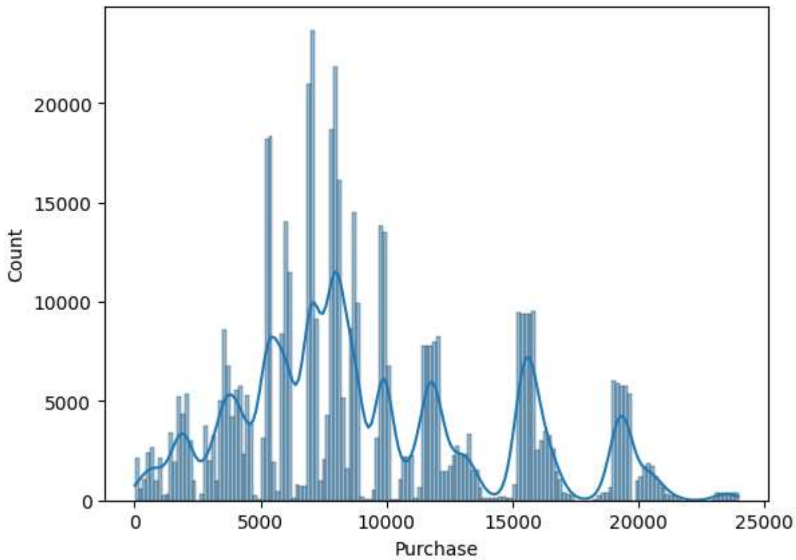
```
User_ID                        5891
Product_ID                     3631
Gender                            2
Age                               7
Occupation                       21
City_Category                     3
Stay_In_Current_City_Years        5
Marital_Status                    2
Product_Category                 20
Purchase                      18105
dtype: int64
```

*(c).Visual Analysis - Univariate & Bivariate *

- For continuous variable(s): Distplot, countplot, histogram for univariate analysis
- For categorical variable(s): Boxplot
- For correlation: Heatmaps, Pairplots

```
sns.countplot(data=df,x='Gender')
plt.show()
```

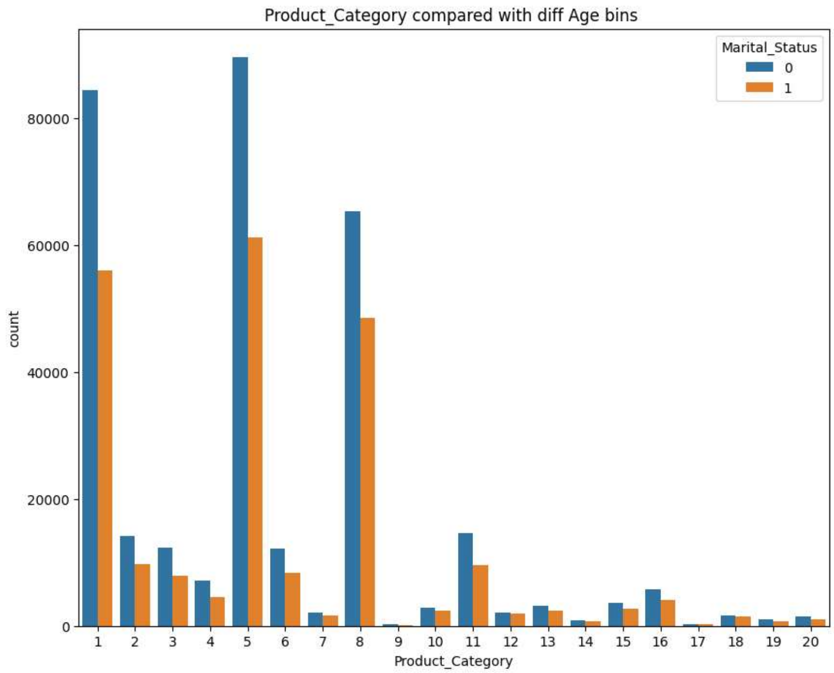

```
sns.histplot(df['Purchase'], kde=True)
plt.show()
```



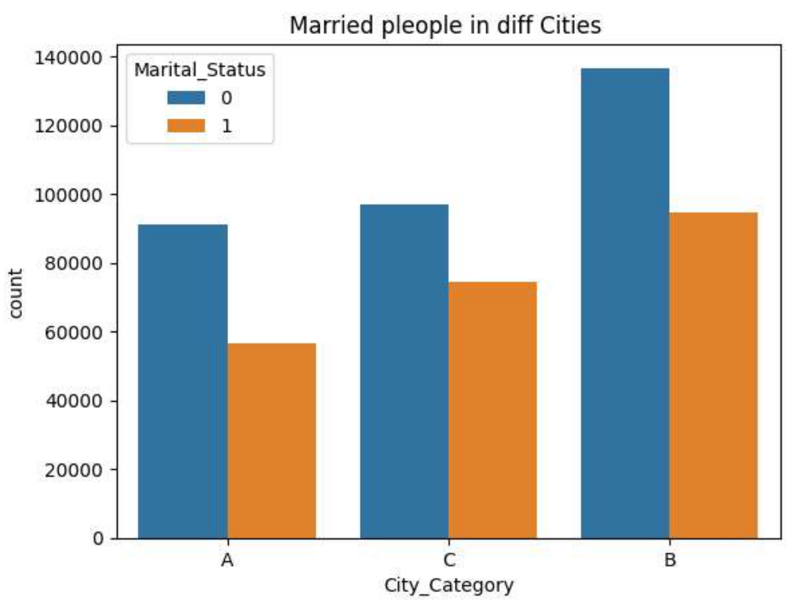

```
df.groupby('Product_Category')['Purchase'].mean()
```

```
Product_Category
1     13606.218596
2     11251.935384
3     10096.705734
4      2329.659491
5      6240.088178
6     15838.478550
7     16365.689600
8      7498.958078
9     15537.375610
10    19675.570927
11     4685.268456
12     1350.859894
13      722.400613
14    13141.625739
15    14780.451828
16    14766.037037
17    10170.759516
18     2972.864320
19       37.041797
20      370.481176
Name: Purchase, dtype: float64
```

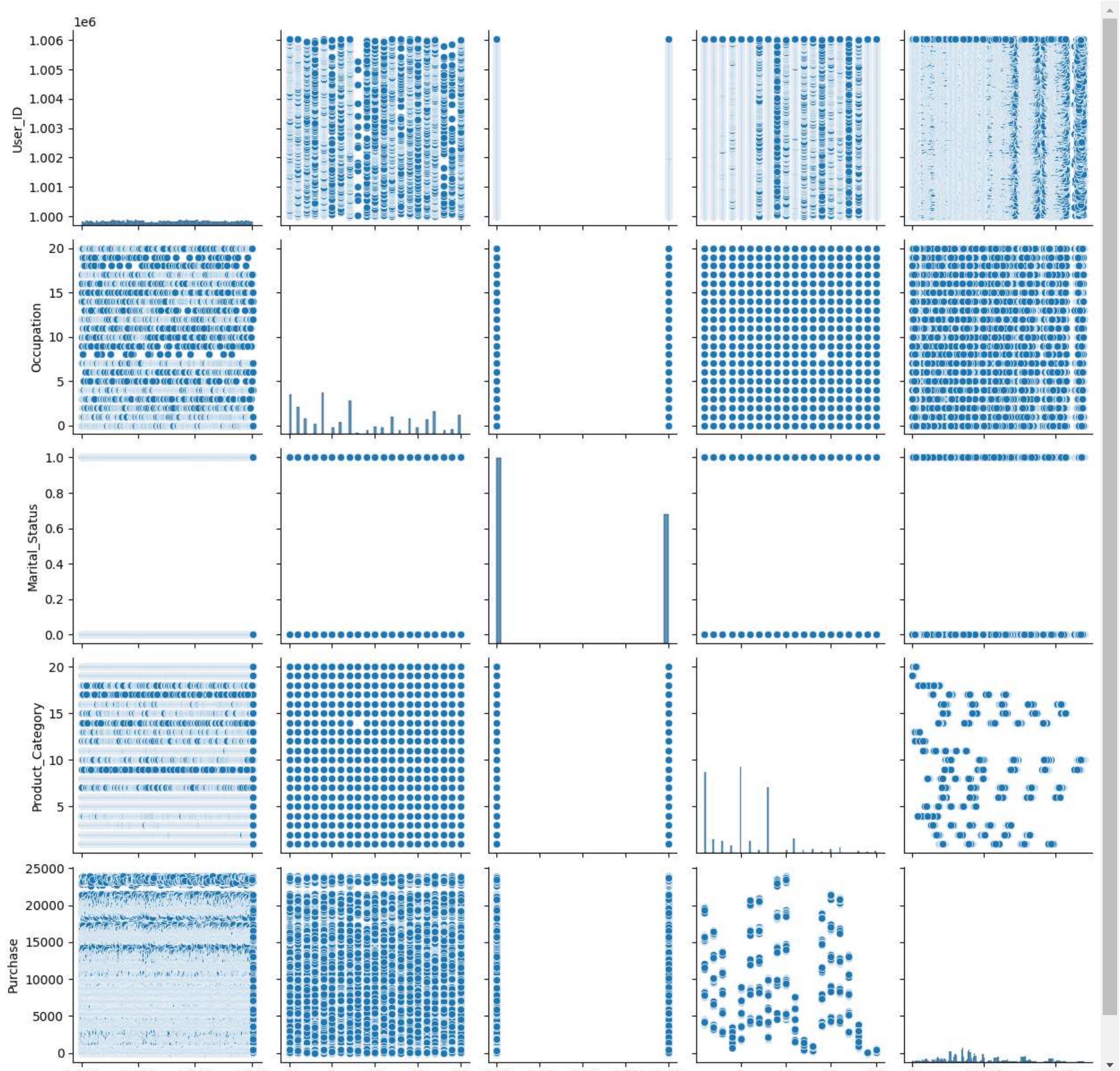
```
plt.figure(figsize=(10,8))
sns.countplot(data=df,x='Product_Category',hue='Marital_Status')
plt.title('Product_Category compared with diff Age bins ')
plt.show()
```




```
sns.countplot(data=df,x='City_Category',hue='Marital_Status')
plt.title('Married pleople in diff Cities ')
plt.show()
```




```
sns.pairplot(data=df)
plt.show()
```



*2. Missing Value & Outlier Detection : *

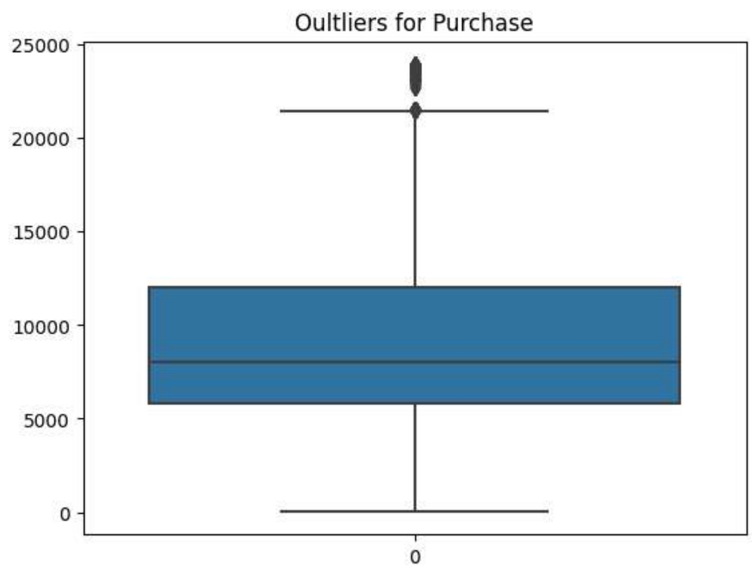
```
df.isnull().sum()
```

```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category              0
Purchase                      0
dtype: int64
```

```
df.describe()
```

	Purchase
count	550068.000000
mean	9263.968713
std	5023.065394
min	12.000000
25%	5823.000000
50%	8047.000000

```
sns.boxplot(data=df['Purchase'])
plt.title('Oultliers for Purchase')
plt.show()
```



3.Business Insights based on Non- Graphical and Visual Analysis

- Comments on the range of attributes
- Comments on the distribution of the variables and relationship between them
- Comments for each univariate and bivariate plot

(explained in the insight section)

```
purchase_max=df['Purchase'].max()
purchase_min=df['Purchase'].min()
```

```
purchase_max
```

```
23961
```

```
purchase_min
```

```
12
```

4.

(a).Are women spending more money per transaction than men? Why or Why not?

```
Men are spending more money than women. Maybe men have lots of stuff to buy other than groceries
```

(b).Confidence intervals and distribution of the mean of the expenses by female and male customers

```
sample_size= 15000
mean_all_women=[]
```

```
mean_all_men=[]
for i in range(500):
  women_mean=df[df['Gender']=='F']['Purchase'].sample(sample_size).mean()
  men_mean=df[df['Gender']=='M']['Purchase'].sample(sample_size).mean()
  mean_all_women.append(women_mean)
  mean_all_men.append(men_mean)
```

(c).Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?

> NO.

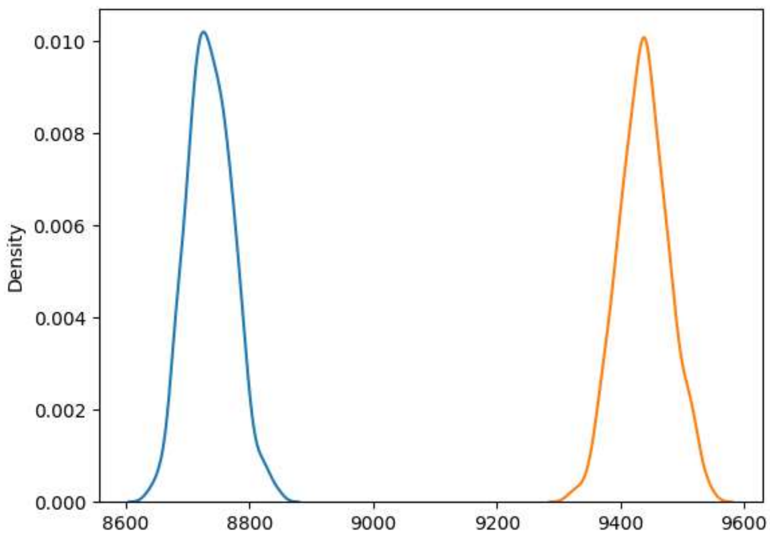
```
np.percentile(mean_all_women,[2.5,97.5])
```

```
array([8671.60943333, 8811.90962167])
```

```
np.percentile(mean_all_men,[2.5,97.5])
```

```
array([9361.71866667, 9517.59311   ])
```

```
sns.kdeplot(mean_all_women)
sns.kdeplot(mean_all_men)
plt.show()
```



(d).Results when the same activity is performed for Married vs Unmarried

```
sample_size= 15000
mean_all_Mar_people=[]
mean_all_UNmar_people=[]
for i in range(500):
  Unmarried_mean=df[df['Marital_Status']==0]['Purchase'].sample(sample_size).mean()
  married_mean=df[df['Marital_Status']==1]['Purchase'].sample(sample_size).mean()
  mean_all_Mar_people.append(married_mean)
  mean_all_UNmar_people.append(Unmarried_mean)
```

```
np.percentile(mean_all_Mar_people,[2.5,97.5])
```

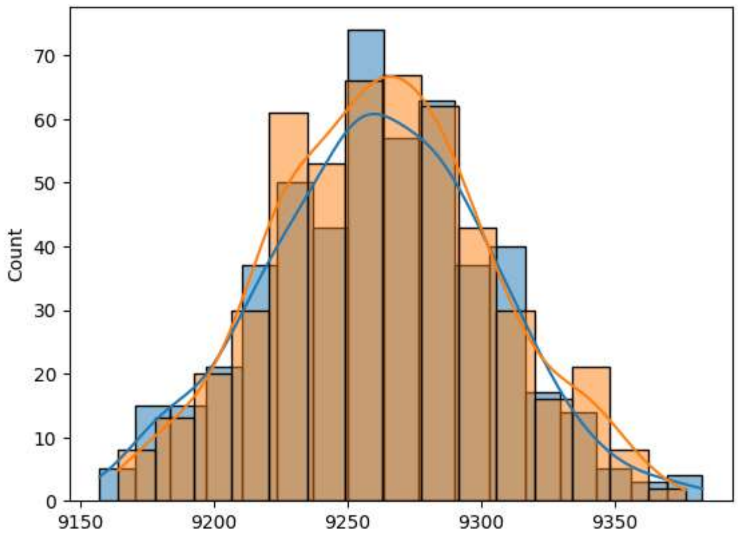
```
array([9178.77339667, 9341.92934   ])
```

```
np.percentile(mean_all_UNmar_people,[2.5,97.5])
```

```
array([9183.256735  , 9344.63152333])
```

```
sns.histplot(mean_all_Mar_people, kde=True,label='Married')
sns.histplot(mean_all_UNmar_people, kde=True, label='Unmarried')
```

```
plt.show()
```



(e).Results when the same activity is performed for Age

```
sample_size= 15000
mean_all_teen_people=[]
mean_all_adult_people=[]
for i in range(500):
  teen_mean=df[(df["Age"]=='0-17') ]['Purchase'].sample(sample_size).mean()
  adult_mean=df[(df['Age']=='26-35')]['Purchase'].sample(sample_size).mean()
  mean_all_teen_people.append(teen_mean)
  mean_all_adult_people.append(adult_mean)
```

```
np.percentile(mean_all_teen_people,[2.5,97.5])
```

```
array([8927.32985833, 8939.49201667])
```

```
np.percentile(mean_all_adult_people,[2.5,97.5])
```

```
array([9168.22450833, 9330.47526333])
```

Population mean of purchases based on City_Category

```
df.groupby('City_Category')['Purchase'].mean()
```

```
City_Category
A    8911.939216
B    9151.300563
C    9719.920993
Name: Purchase, dtype: float64
```

Insights-

1. Except purchase column every in the dataframe is catagorical column.And there are no null values
2. Visual Analysis

> - Men buy more products than women
> - No of products bought from price range(5k-10k) is significantly higher than other products
> - Product_catagory 1,5 and 8 are most purchased products
> - The outliers for the purchase column is above 2100 approx

3. The distribution of the mean of the expenses by female and male customers does not overlap

4. The distribution of the mean of the expenses by married and unmarried customers with 95% confidence interval , it does overlap
5. The distribution of the mean of the expenses with teen and adult customers with 95% confidence interval also does not overlap

Recommendations-

> 1. We can increase the sale of products purchased by women by giving offers and coupons exclusively for women
>
> 2. We can increase the revenue by offering discounts on tools and other household essential products on an occasional basis
>
> 3. We can also target customers with different Age groups based on their preferences and interests