

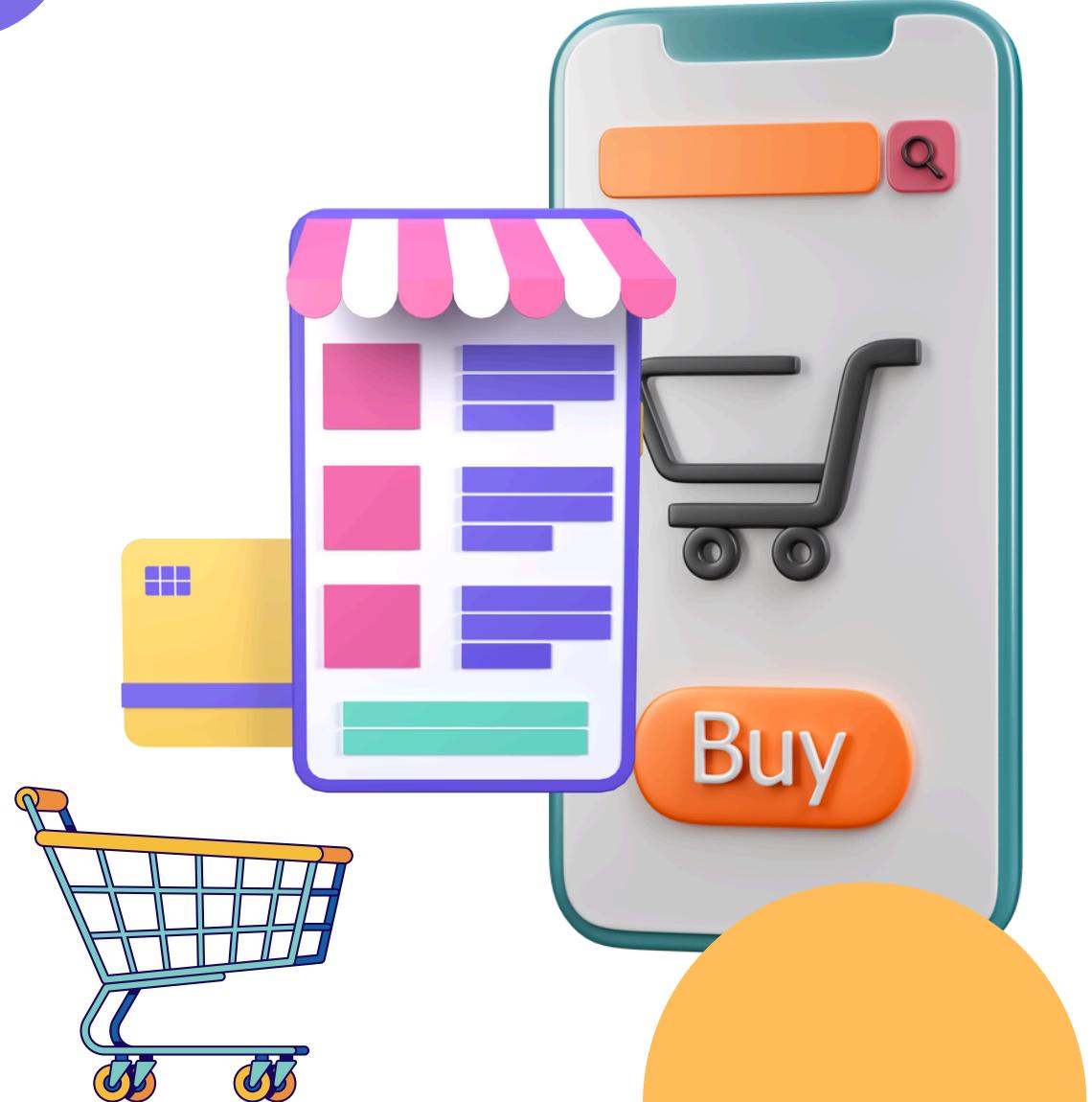


e-commerce

Diwali Sales project

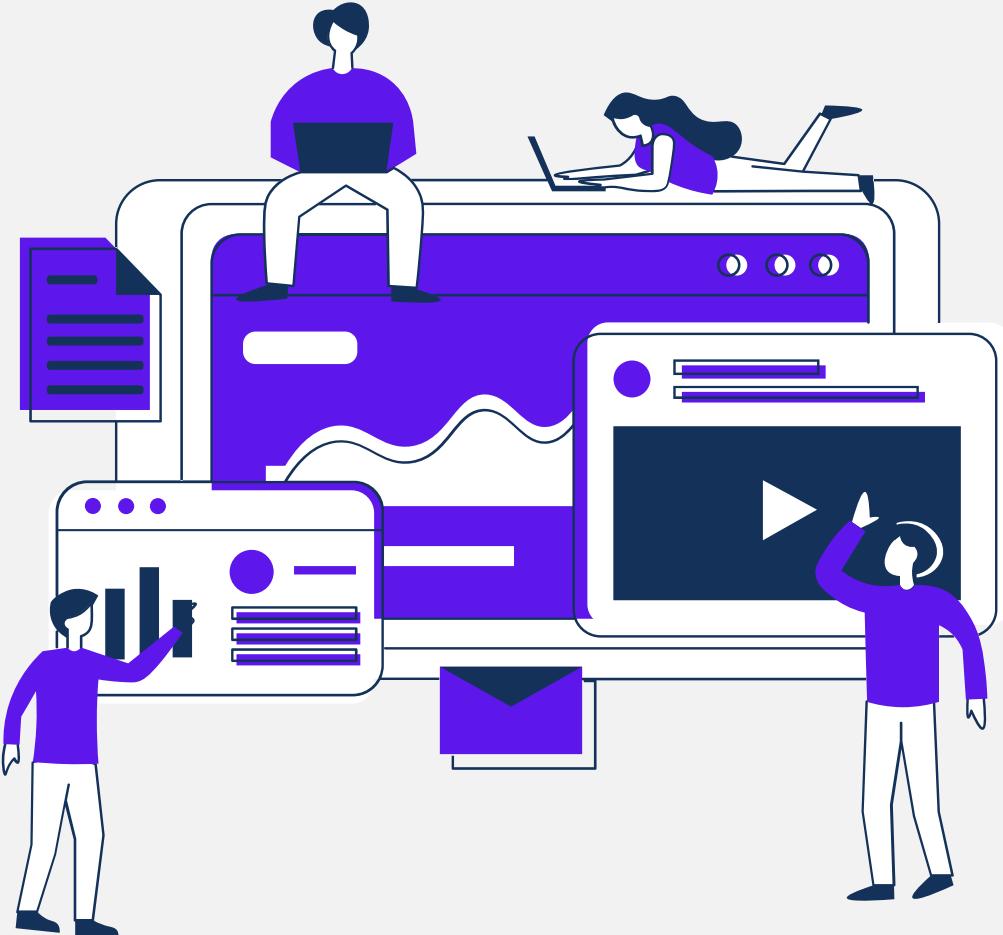
using Python

Created By Kamal



HELLO EVERYONE

I AM KAMAL AND IN THIS PROJECT
I UTILIZE PYTHON CODE TO
PREDICT DIWALI SALES



Step
01

Installing all required
libraries



Diwali_Sales

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```



Step 02

Importing CSV File to Jupyter Notebook



```
In [3]: df = pd.read_csv('E:\Downloads\Diwali Sales Data.csv',encoding = 'unicode_escape')
```

Step 03

Viewing the Dataset

```
In [6]: df
```

```
Out[6]:
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0

Step 04

Viewing columns Info & Finding Null values in Columns



```
In [4]: df.info()  
df.isnull().sum()
```

```
Out[4]: User_ID          0  
Cust_name        0  
Product_ID        0  
Gender            0  
Age Group         0  
Age               0  
Marital_Status    0  
State              0  
Zone               0  
Occupation        0  
Product_Category   0  
Orders             0  
Amount            12  
Status           11251  
unnamed1          11251  
dtype: int64
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 11251 entries, 0 to 11250  
Data columns (total 15 columns):  
 #   Column      Non-Null Count  Dtype     
---  --          --          --  
 0   User_ID     11251 non-null  int64    
 1   Cust_name   11251 non-null  object    
 2   Product_ID  11251 non-null  object    
 3   Gender       11251 non-null  object    
 4   Age Group   11251 non-null  object    
 5   Age          11251 non-null  int64    
 6   Marital_Status  11251 non-null  int64    
 7   State        11251 non-null  object    
 8   Zone         11251 non-null  object    
 9   Occupation   11251 non-null  object    
 10  Product_Category  11251 non-null  object    
 11  Orders       11251 non-null  int64    
 12  Amount        11239 non-null  float64   
 13  Status        0 non-null    float64   
 14  unnamed1      0 non-null    float64  
dtypes: float64(3), int64(4), object(8)  
memory usage: 1.3+ MB
```

Step 05

Drop the Columns Which have Null values



```
In [5]: df.drop(columns=['Status','unnamed1'],axis =1,inplace= True)
```

```
In [6]: df
```

Out[6]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0
...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0

11251 rows × 13 columns

Step 06

Permanently drooping the columns from dataset

```
In [13]: df.dropna(inplace = True)
```



Step 07

checking Null values again

```
In [14]: df.isnull().sum()
```

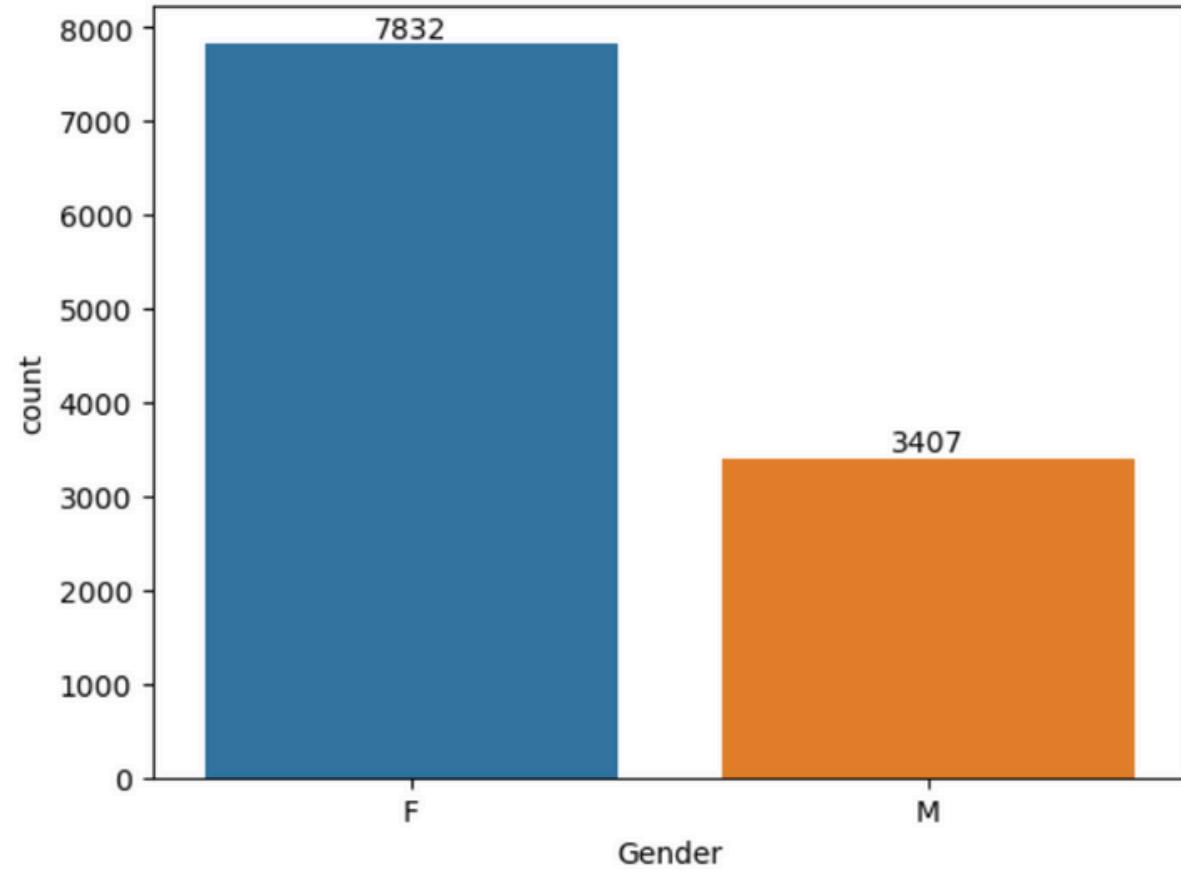
```
Out[14]: User_ID          0
Cust_name         0
Product_ID        0
Gender            0
Age_Group         0
Age               0
Marital_Status    0
State             0
Zone              0
Occupation        0
Product_Category  0
Orders            0
Amount            0
dtype: int64
```

Step 08

Plotting Countplot Graph to know which Gender is most active in Sales



```
In [20]: ax = sns.countplot(data=df,x='Gender')
ax.bar_label(ax.containers[0])
```

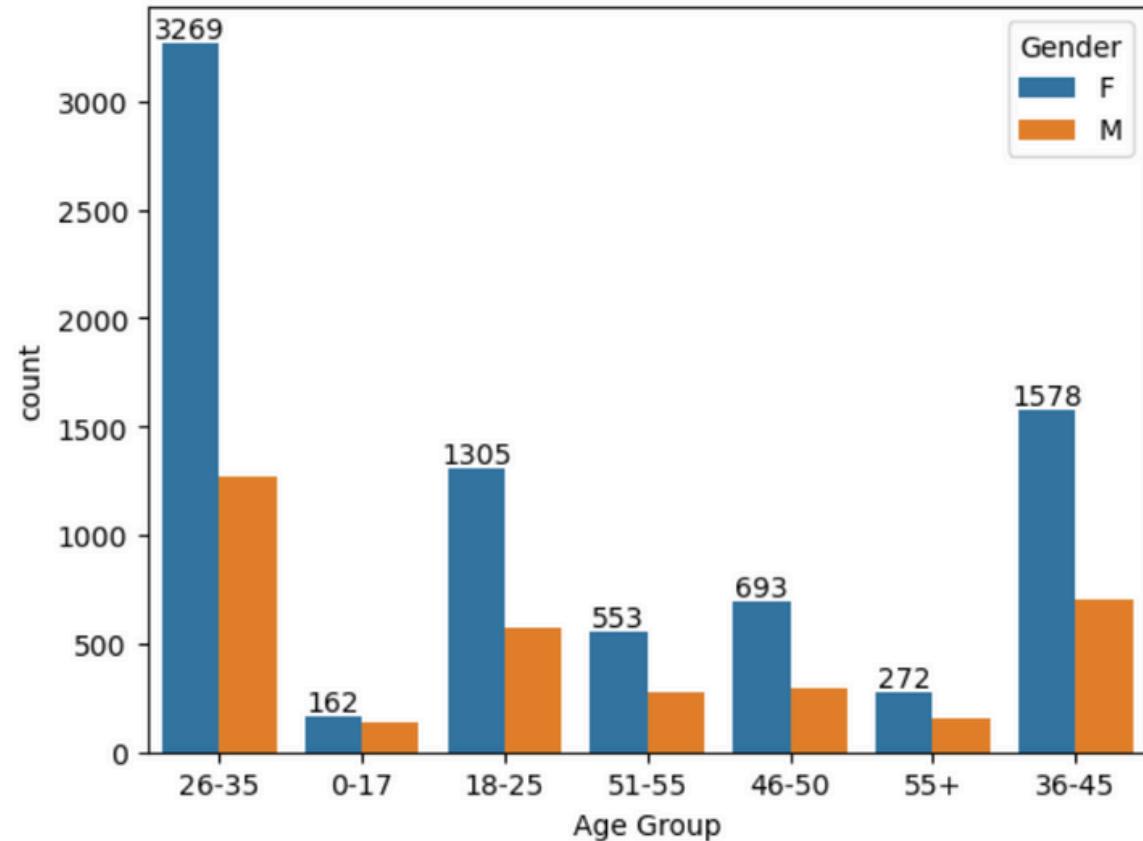


Step 09

Plotting Countplot to know which Age group & Gender is most active in Sales



```
In [21]: a = sns.countplot(data=df,x='Age Group', hue='Gender')
a.bar_label(a.containers[0])
```



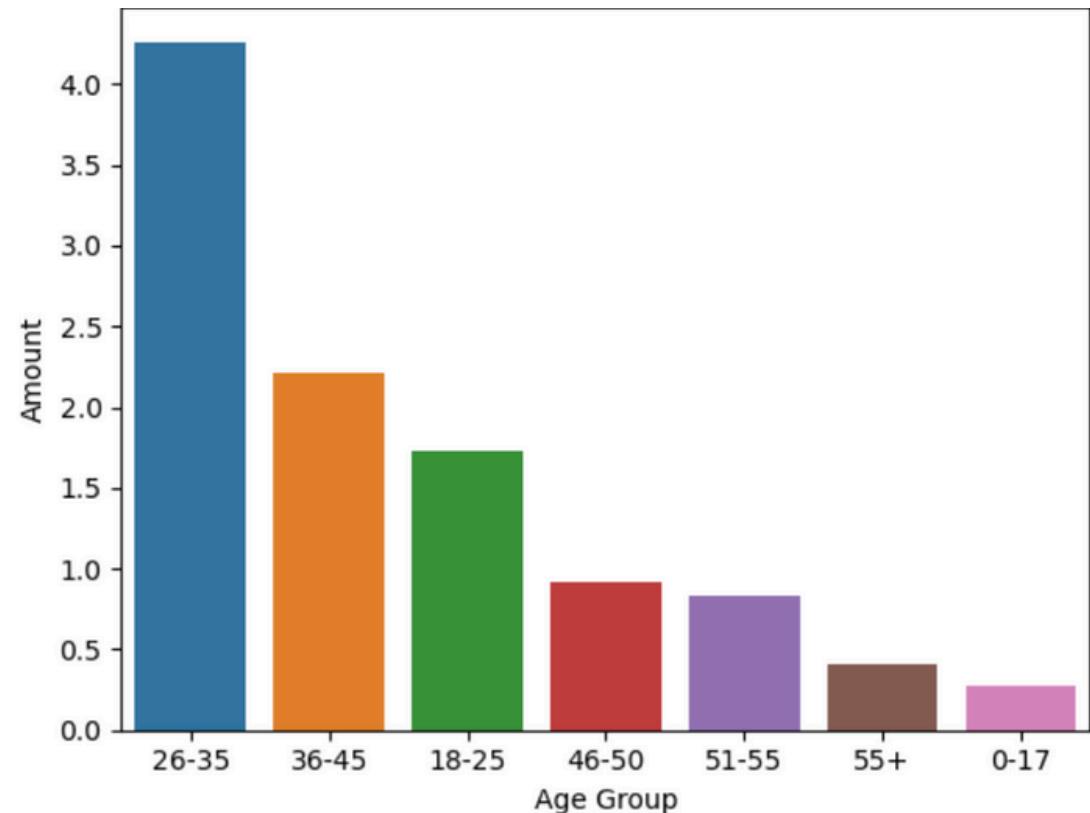
Step 10

Plotting BarPlot to know which Age group people spend the most of money



```
In [23]: sales_age = df.groupby(['Age Group'], as_index=False)[ 'Amount'].sum().sort_values(by='Amount', ascending = False)
sns.barplot(data=sales_age,x='Age Group',y='Amount')
```

```
Out[23]: <Axes: xlabel='Age Group', ylabel='Amount'>
```

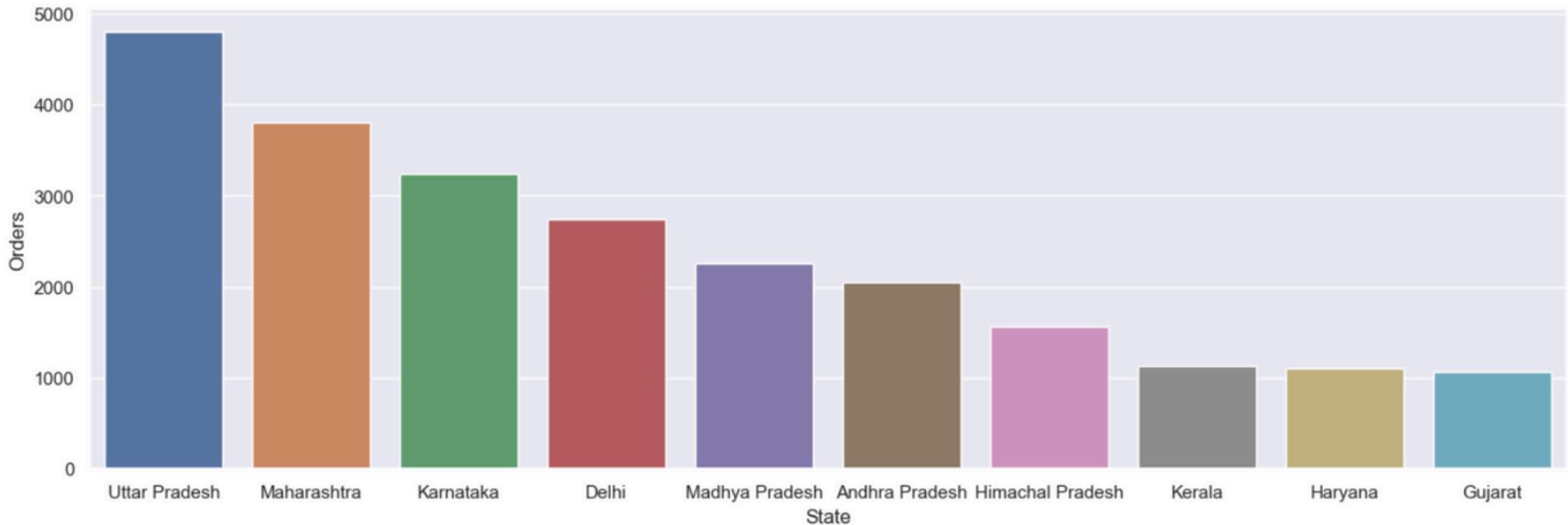


Step 11

Plotting BarPlot to know which state people order most things during sales



```
In [32]: sales_state = df.groupby(['State'], as_index=False)[ 'Orders' ].sum().sort_values(by='Orders', ascending = False).head(10)
sns.set(rc={'figure.figsize':(16,5)})
x1=sns.barplot(data=sales_state,x='State',y='Orders')
x1.bar_label(x1.containers[0])
```

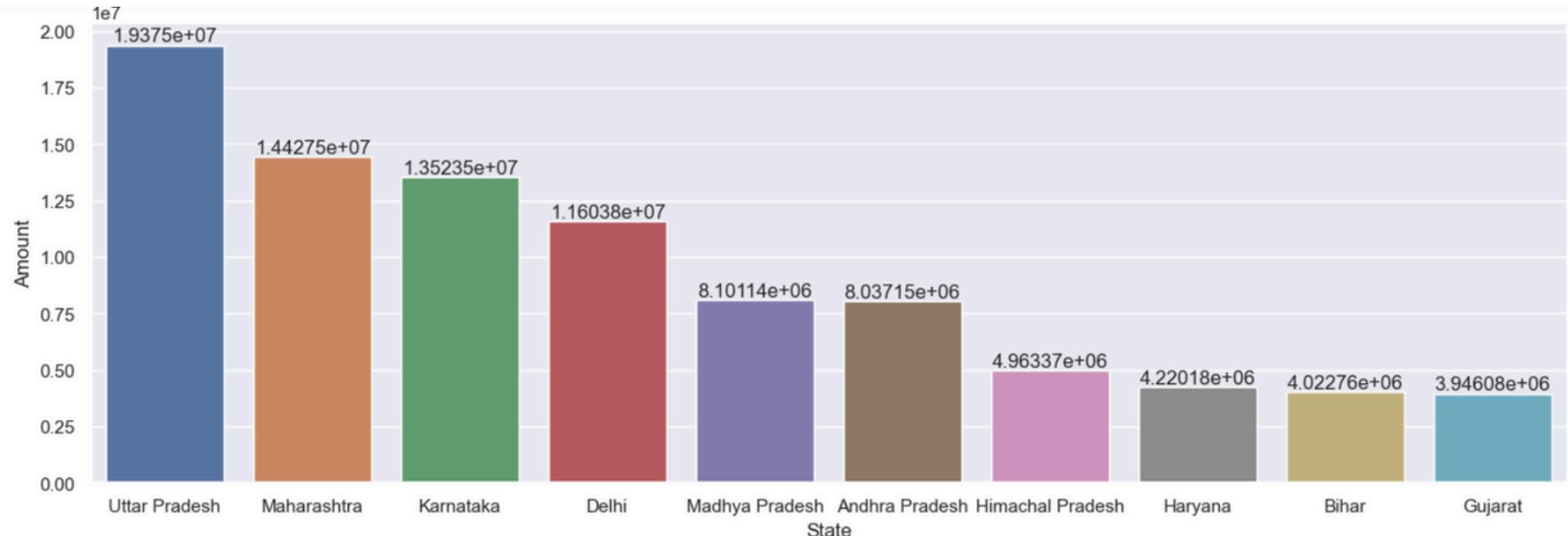


Step 12

Plotting BarPlot to know which state people spend large amount during sales



```
In [31]: sales_state = df.groupby(['State'], as_index=False)[ 'Amount'].sum().sort_values(by='Amount', ascending = False).head(10)
sns.set(rc={'figure.figsize':(16,5)})
x=sns.barplot(data=sales_state,x='State',y='Amount')
x.bar_label(x.containers[0])
```

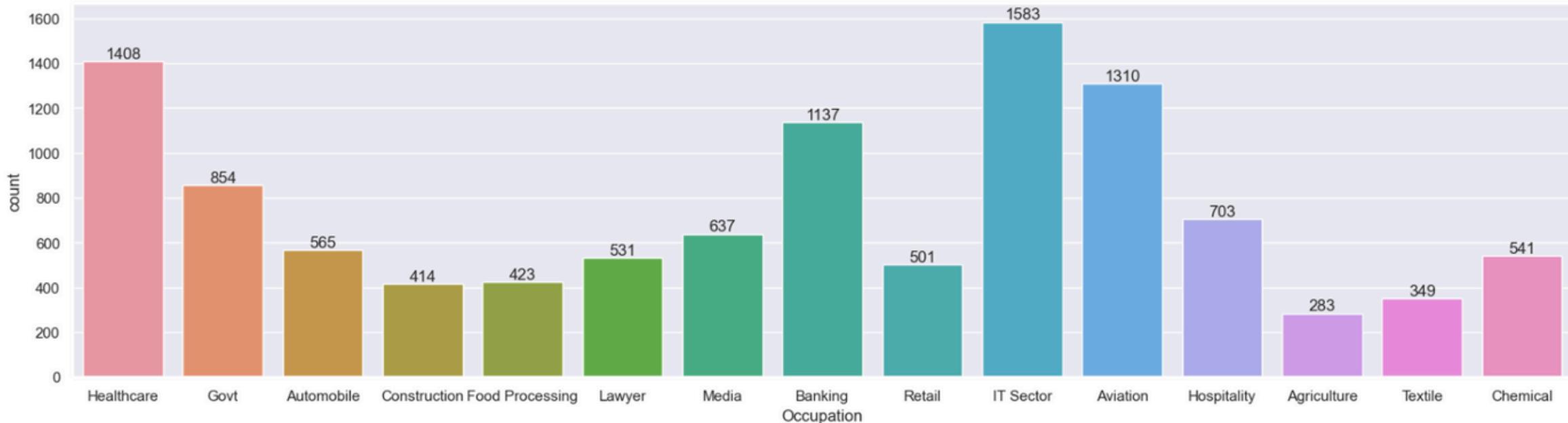


Step 13

Plotting countplot to know which Occupation people ordered most in sales



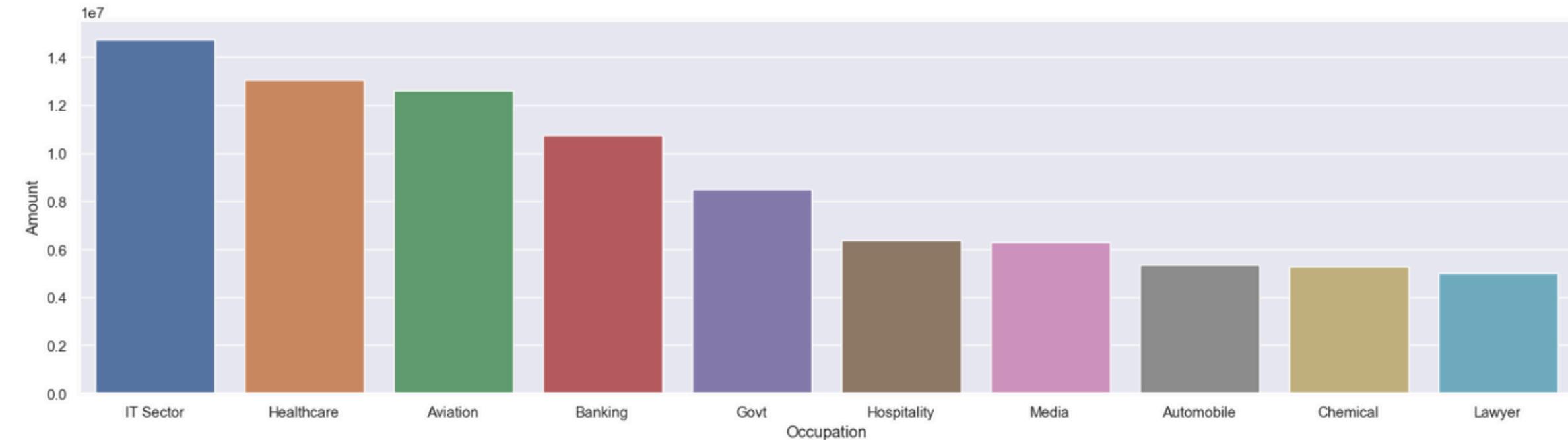
```
In [53]: a1=sns.countplot(data=df,x='Occupation')
a1.bar_label(a1.containers[0])
sns.set(rc={'figure.figsize':(20,5)})
```



Step 14

Plotting Barplot to know which Occupation people spend large amount in sales

```
[54]: sales_state = df.groupby(['Occupation'], as_index=False)[ 'Amount' ].sum().sort_values(by='Amount', ascending = False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
x=sns.barplot(data=sales_state,x='Occupation',y='Amount')
```

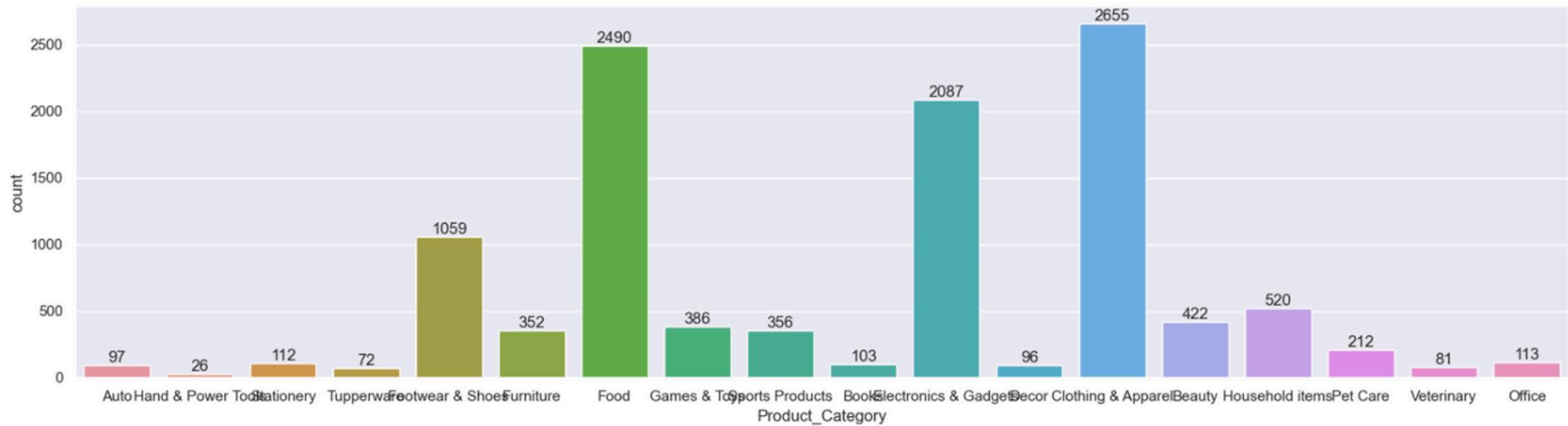


Step 15

Plotting countplot to know which product ordered most in sales



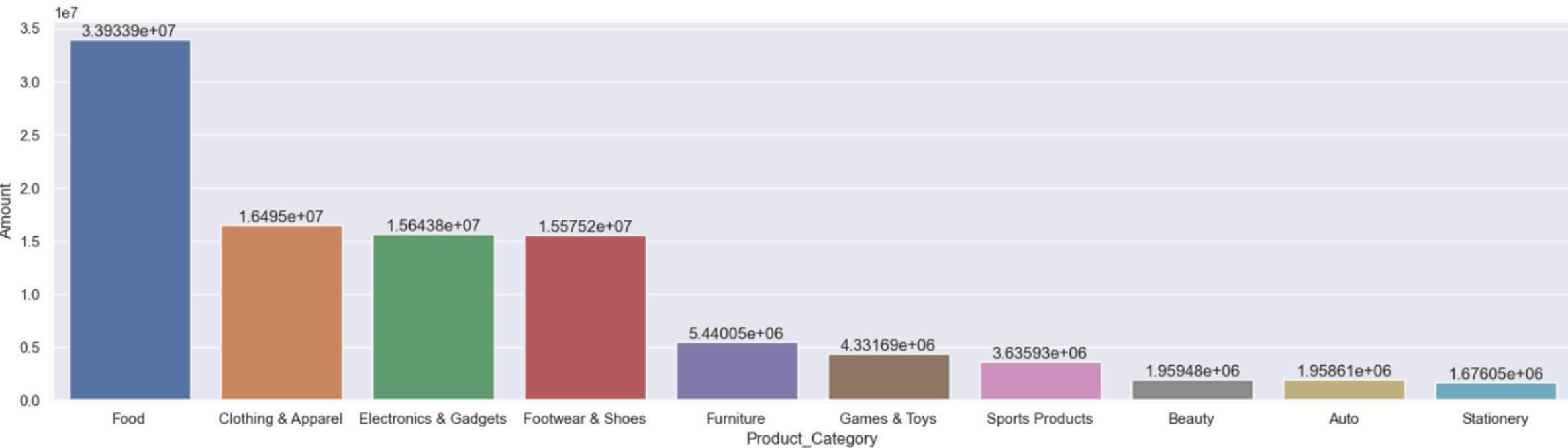
```
[56]: a1=sns.countplot(data=df,x='Product_Category')
a1.bar_label(a1.containers[0])
sns.set(rc={'figure.figsize':(23,5)})
```



Step 16

Plotting Barplot to know in which product people spend most in sales

```
In [58]: sales_state = df.groupby(['Product_Category'], as_index=False)[['Amount']].sum().sort_values(by='Amount', ascending = False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
x=sns.barplot(data=sales_state,x='Product_Category',y='Amount')
x.bar_label(x.containers[0])
```



THANK YOU!

