Diwali Sales Analysis

Project Learning

1. Performed data cleaning and manipulation.
2. Performed exploratory data analysis (EDA) using pandas, matplotlib and seaborn libraries.
3. Improved customer experience by identifying potential customers across different states, occupation, gender and age groups.
4. Improved sales by identifying most selling product categories and products which can help to plan inventry and hence meet the demands.

In [2]:
```
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

In [3]:
```
# import csv file
df = pd.read_csv('C:\\Users\\User\\Desktop\\Diwali Sales Data.csv', enc
```

In [4]:
```
df.shape
```

Out[4]: (11251, 15)

In [5]:
```
df.head()
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	W
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Sc
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	(
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sc
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W

In [6]:
```python
df.info()
```

```
<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
```

In [7]:
```python
#drop unrelated/blank columns
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

In [8]:
```python
pd.isnull(df)
```

Out[8]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

In [9]:
```
#check for null values
pd.isnull(df).sum()
```

Out[9]:
```
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
dtype: int64
```

In [10]:
```
# drop null values
df.dropna(inplace=True)
```

In [ ]:
```

```

In [11]:
```
pd.isnull(df).sum()
```

Out[11]:
```
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
```

In [12]:
```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

In [13]:
```
df['Amount'].dtypes
```

Out[13]:
```
dtype('int32')
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [15]:

```
#rename column
df.rename(columns= {'Marital_Status':'Shaadi'})
```

Out[15]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Z
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Wes
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Sout
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Ce
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sout
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Wes
...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Wes
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Nort
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Ce
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Sout
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Wes

11239 rows × 13 columns

In [16]:

```
# describe() method returns description of the data in the DataFrame (i
df.describe()
```

Out[16]:

	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	2.000000	5443.000000
50%	1.003064e+06	33.000000	0.000000	2.000000	8109.000000
75%	1.004426e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

In [17]:

```python
# use describe() for specific columns
df[['Age', 'Orders', 'Amount']].describe()
```

Out[17]:

	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

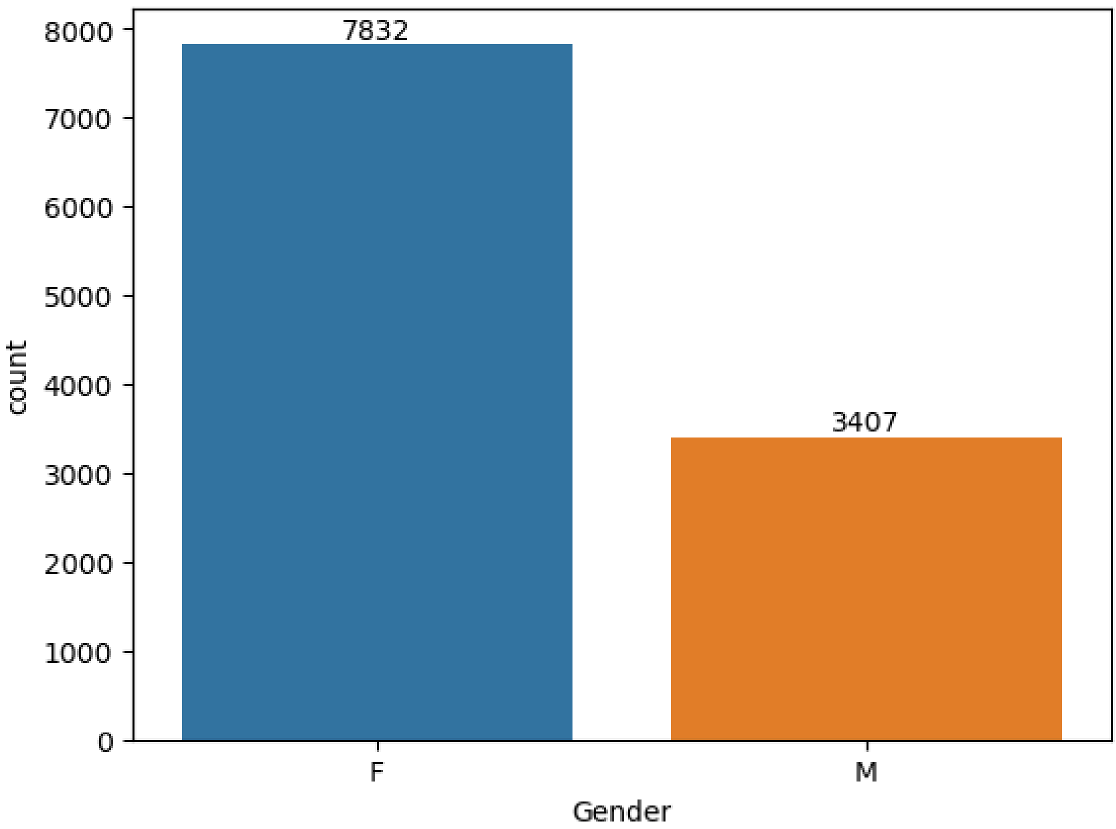
Exploratory Data Analysis

In [18]:

```python
# plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

for bars in ax.containers:
    ax.bar_label(bars)
```



In [19]:
```python
df.groupby(['Gender'],as_index = False)['Amount'].sum().sort_values(by
```

Out[19]:

	Gender	Amount
0	F	74335853
1	M	31913276

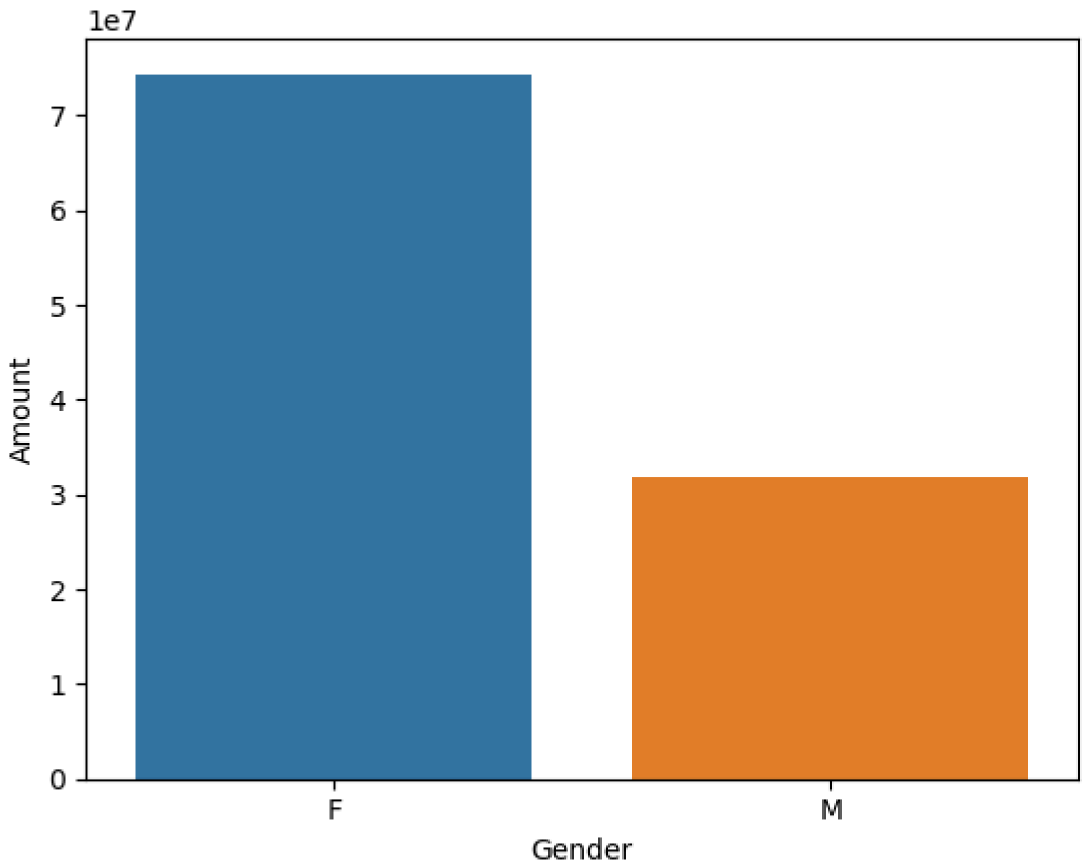
In [20]:
```python
# plotting a bar chart for gender vs total amount

sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort

sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gen)

#From above graphs we can see that most of the buyers are females and e
```

Out[20]:
```
<AxesSubplot:xlabel='Gender', ylabel='Amount'>
```

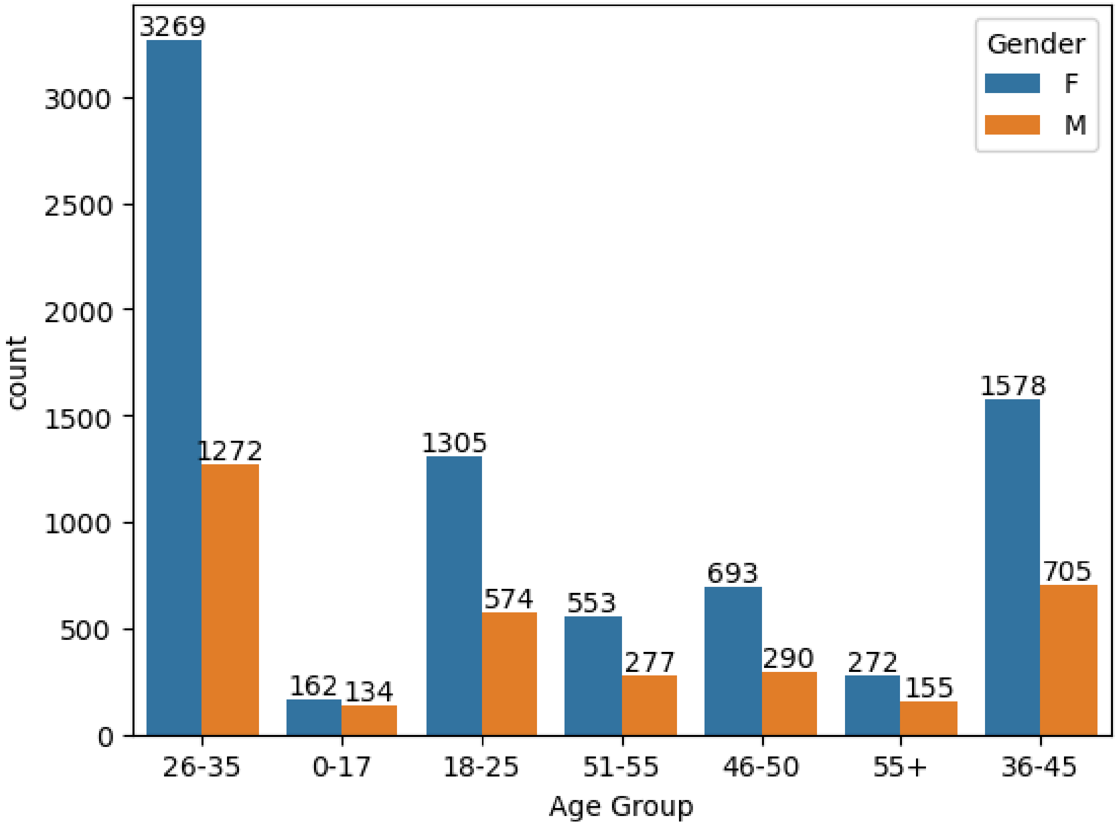


Age

In [21]:

```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

for bars in ax.containers:
    ax.bar_label(bars)
```

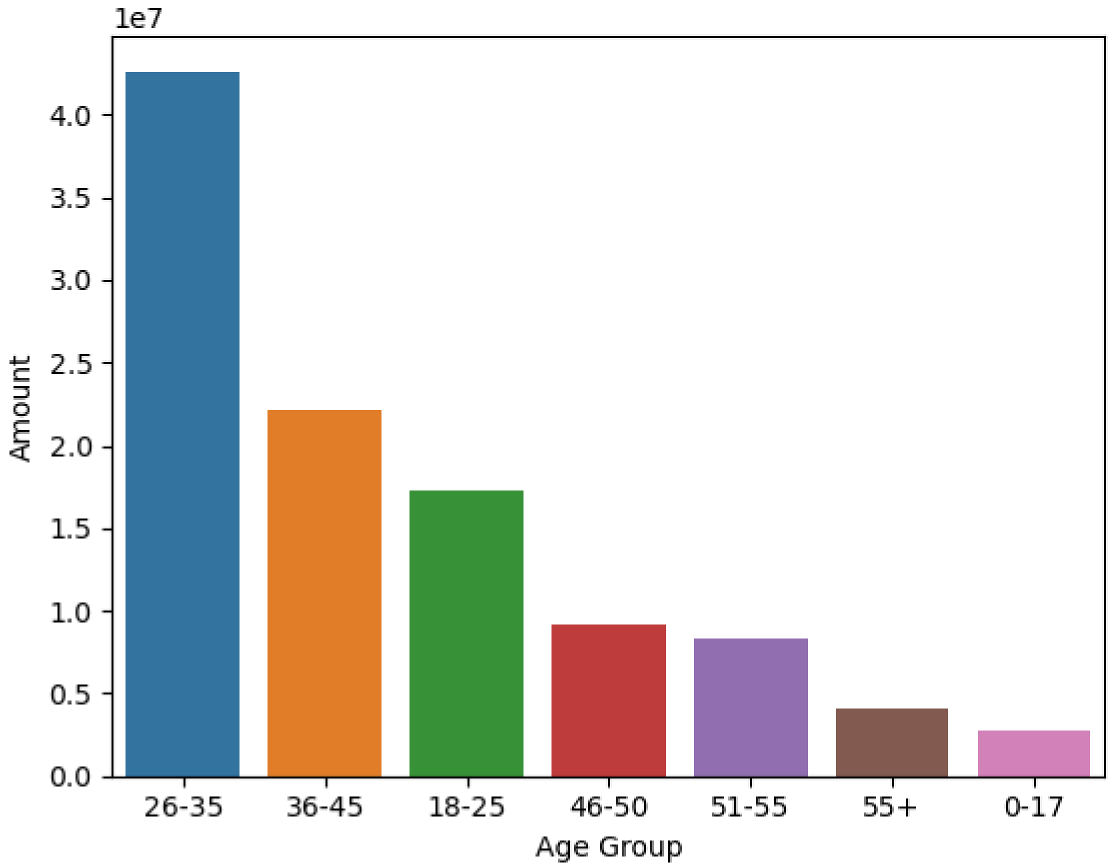


In [22]:

```
# Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().s

sns.barplot(x = 'Age Group',y= 'Amount' ,data = sales_age)
#From above graphs we can see that most of the buyers are of age group
```

Out[22]: <AxesSubplot:xlabel='Age Group', ylabel='Amount'>



State

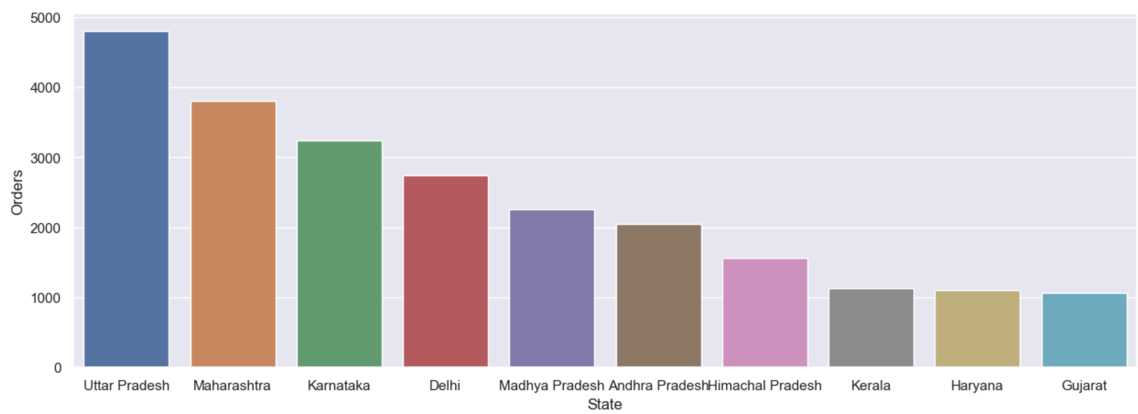
In [23]:

```
# total number of orders from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sor

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Orders')
```

Out[23]: <AxesSubplot:xlabel='State', ylabel='Orders'>



In [24]:

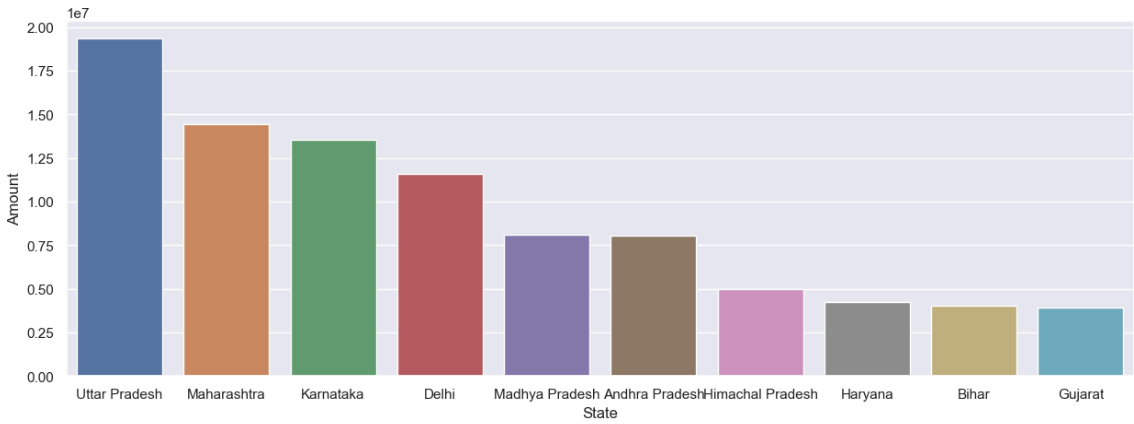
```
# total amount/sales from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sor

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Amount')

#From above graphs we can see that most of the orders & total sales/amo
```

Out[24]: <AxesSubplot:xlabel='State', ylabel='Amount'>

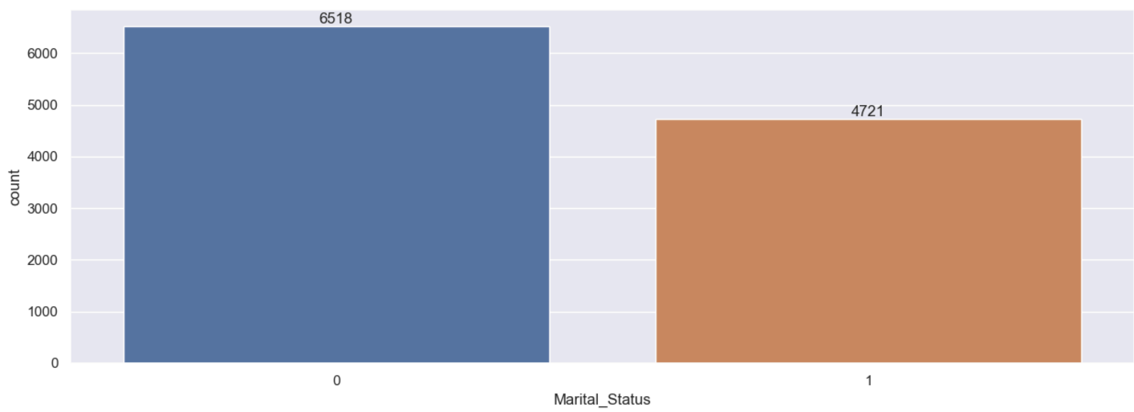


Marital Status

In [25]:

```
ax = sns.countplot(data = df, x = 'Marital_Status')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```

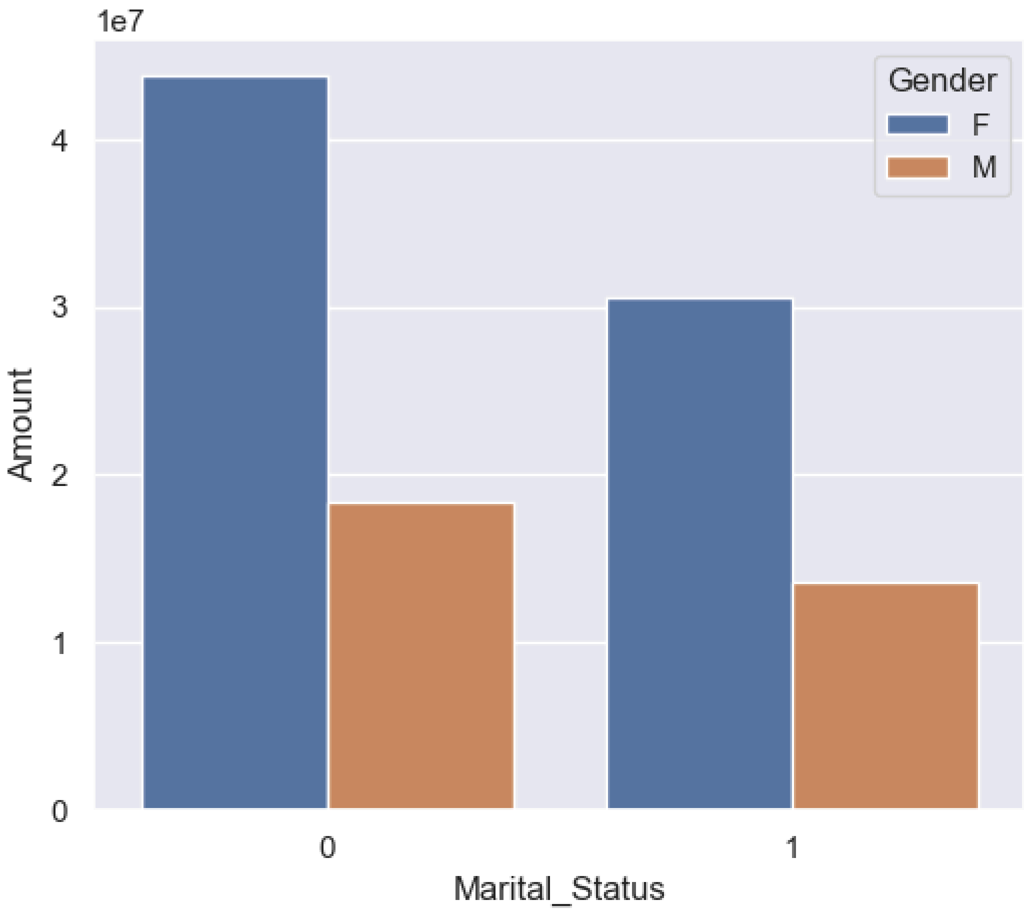


In [26]:
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)[

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='

#From above graphs we can see that most of the buyers are married (wome
```

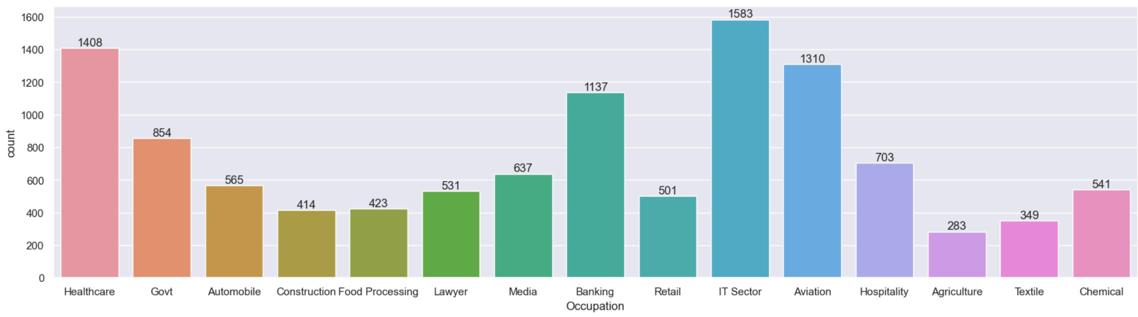
Out[26]: <AxesSubplot:xlabel='Marital_Status', ylabel='Amount'>



Occupation

In [27]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

for bars in ax.containers:
    ax.bar_label(bars)
```

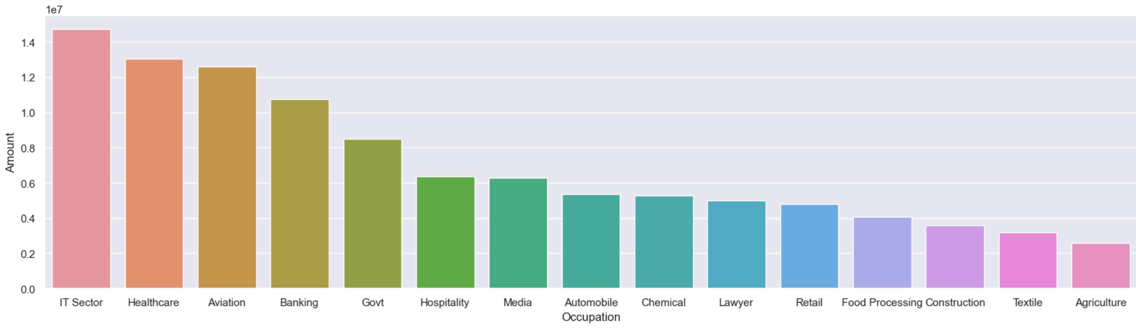


In [28]:

```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum(

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
```

Out[28]: <AxesSubplot:xlabel='Occupation', ylabel='Amount'>

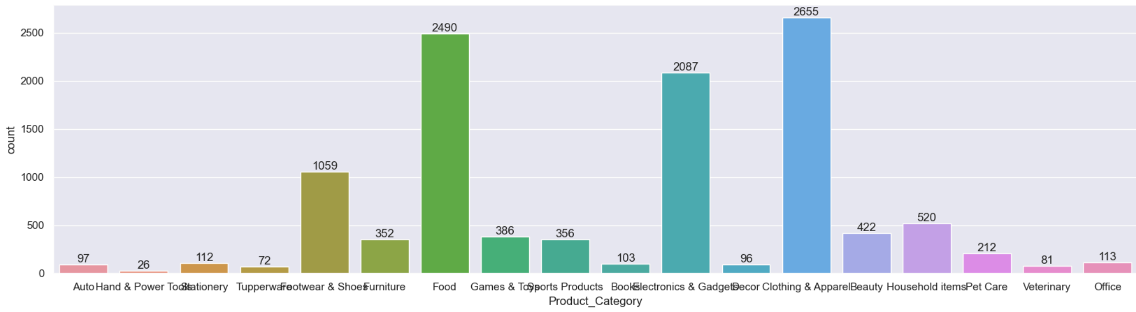


Product Category

In [29]:

```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

for bars in ax.containers:
    ax.bar_label(bars)
```



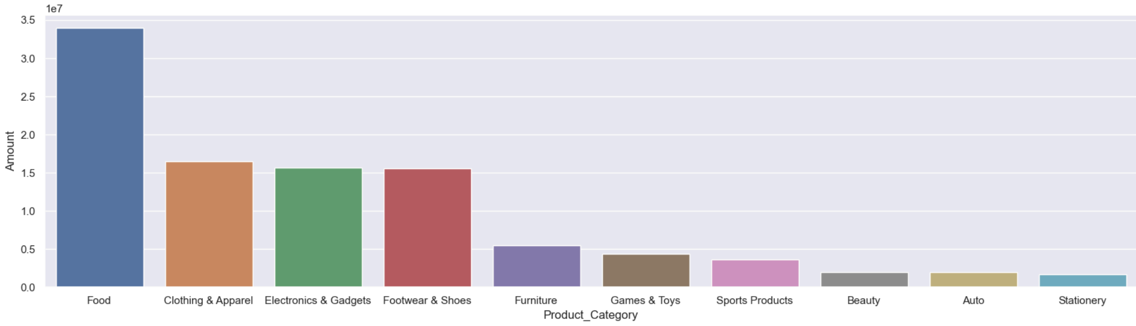
In [30]:

```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount')

#From above graphs we can see that most of the sold products are from F
```

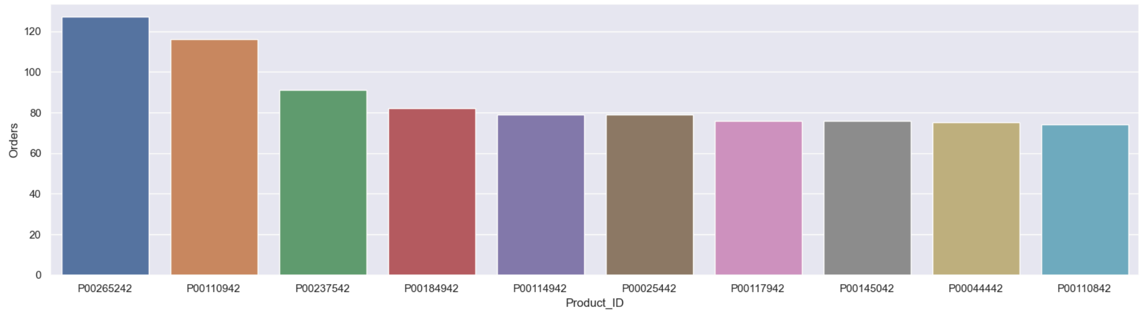
Out[30]: <AxesSubplot:xlabel='Product_Category', ylabel='Amount'>



In [31]:
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum(

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_ID',y= 'Orders')
```

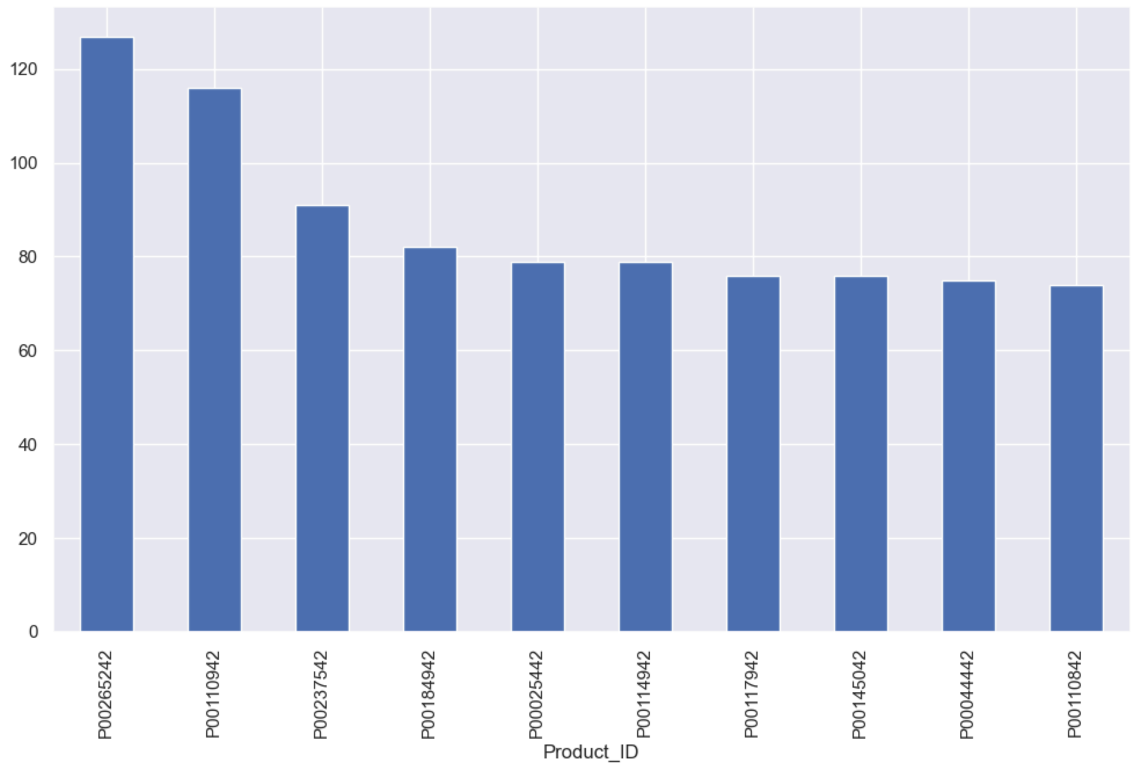
Out[31]: <AxesSubplot:xlabel='Product_ID', ylabel='Orders'>



In [32]:
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascer
```

Out[32]: <AxesSubplot:xlabel='Product_ID'>



Conclusion:

Married women age group 26-35 yrs from UP, Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothing and Electronics category.