Project Name-Diwali sales Analysis

```
Project Name -Diwali Sales Analysis
Project Type - EDA
Contribution - Individual
Team Member 1 -Jyoti Ghaytadak
```

In [1]:
```
#importing python libraries
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt #visulazing data
%matplotlib inline
import seaborn as sns
```

In [3]:
```
 #import csv file",
df=pd.read_csv(r'C:\Users\C ZONE\Downloads\Diwali Sales Data.csv',
               encoding='unicode_escape')
```

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	Zon
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Wester
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Souther
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Centr
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Souther
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Wester

In [6]:

```
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]:

```
#drop unrelated/bank columns
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

In [8]:

```
#check for null values
pd.isnull(df).sum()
```

Out[8]:

```
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 [9]:

```
#drop null values
df.dropna(inplace=True)
```

In [10]:

```
#chgane the data type
df['Amount']=df['Amount'].astype('int')
```

In [11]:

```
df['Amount'].dtypes
```

Out[11]:

```
dtype('int32')
```

In [12]:

```
df.columns
```

Out[12]:

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [13]:

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

Out[13]:

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

11239 rows × 13 columns

In [14]:

```
#describe() method returns description of the DataFrame(i.e.count,mean.std,etc)
df.describe()
```

Out[14]:

	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 [15]:
```
#use describe() for specific columns
df[['Age','Orders','Amount']].describe()
```

Out[15]:

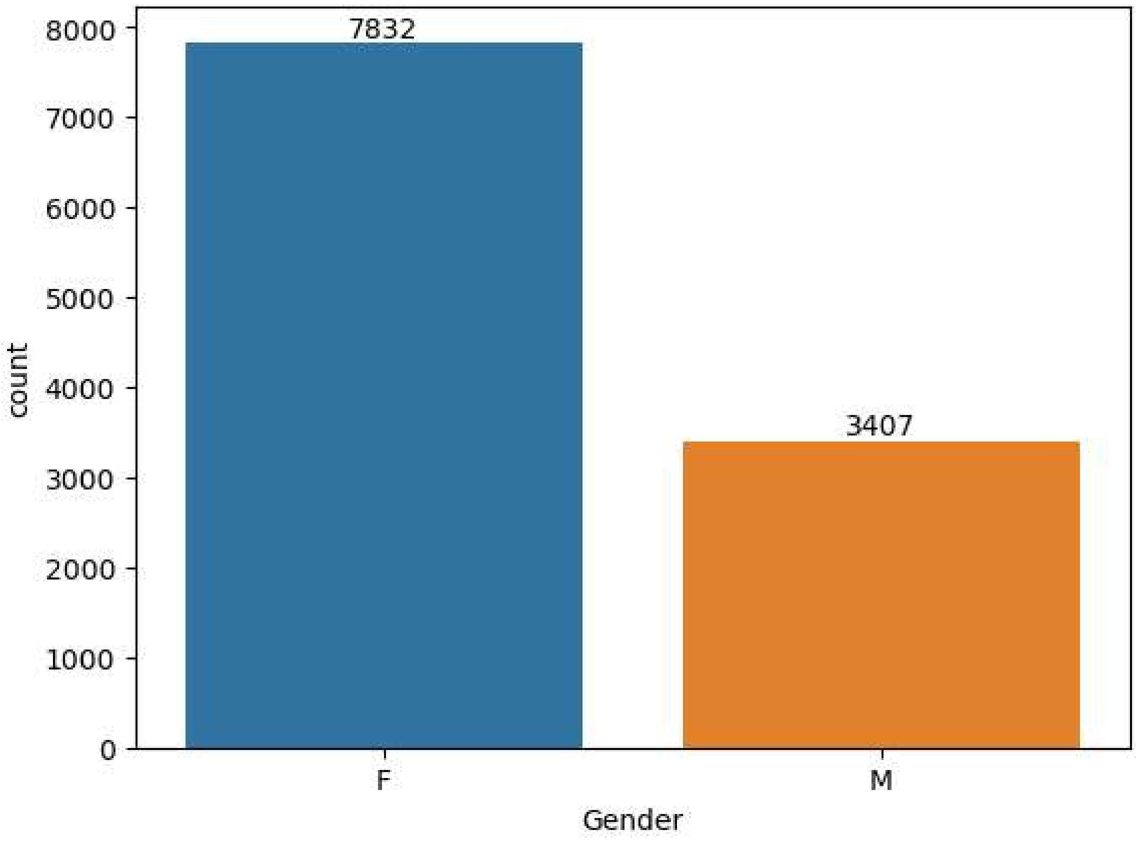
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis

In [16]:
```
#gender
```

In [17]:
```
#plotting a bar chart for gender and it's count
```

In [18]:
```
ax=sns.countplot(x='Gender',data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```

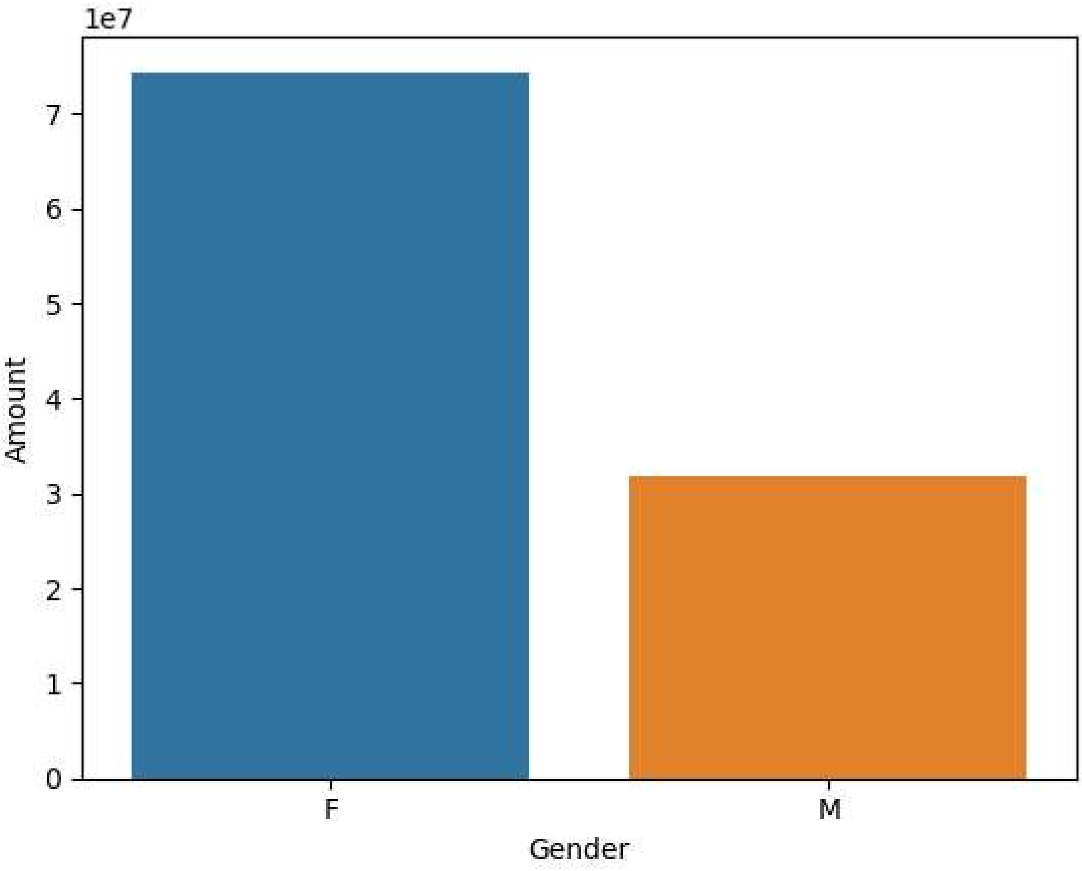


In [19]:

```
#plotting a bar chart for gender vs amount
sales_gen=df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values
(by='Amount',ascending=False)
sns.barplot(x='Gender',y='Amount',data=sales_gen)
```

Out[19]:

```
<Axes: xlabel='Gender', ylabel='Amount'>
```

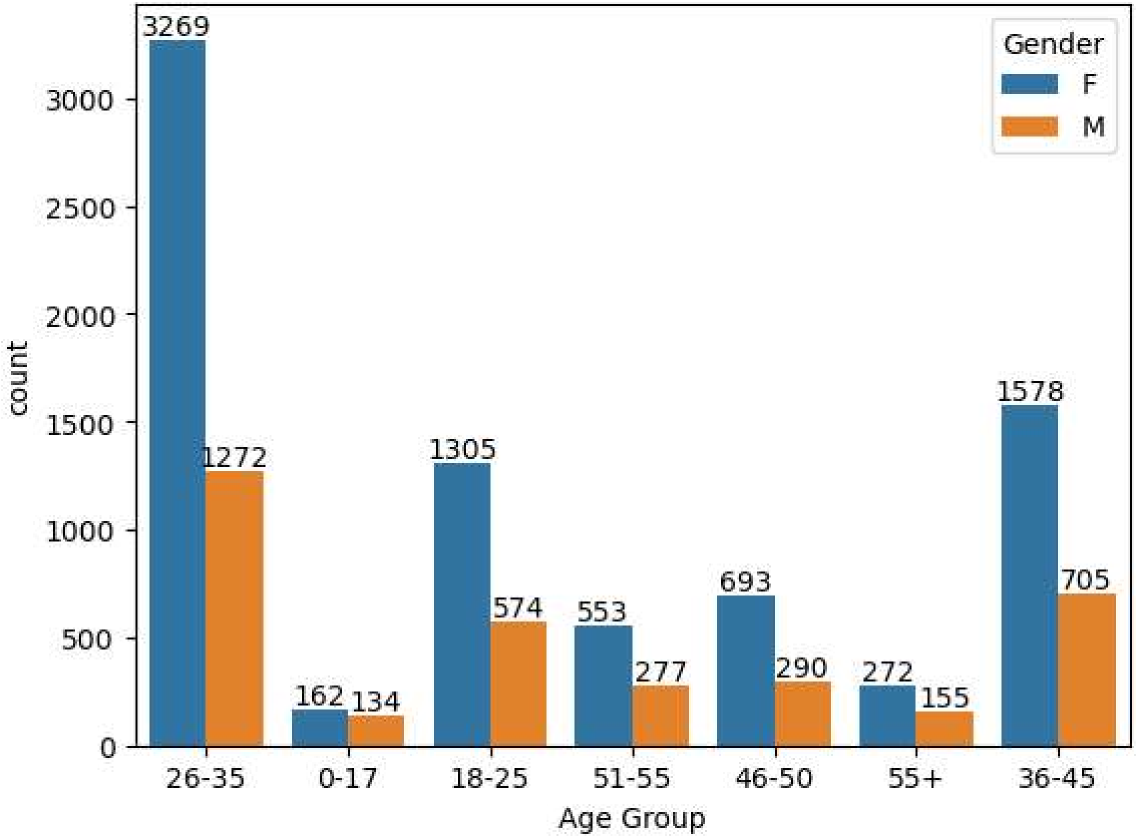


from above graphs we see that most of the buyers are females and even the purchasing power of females are greater than men

Age

In [20]:

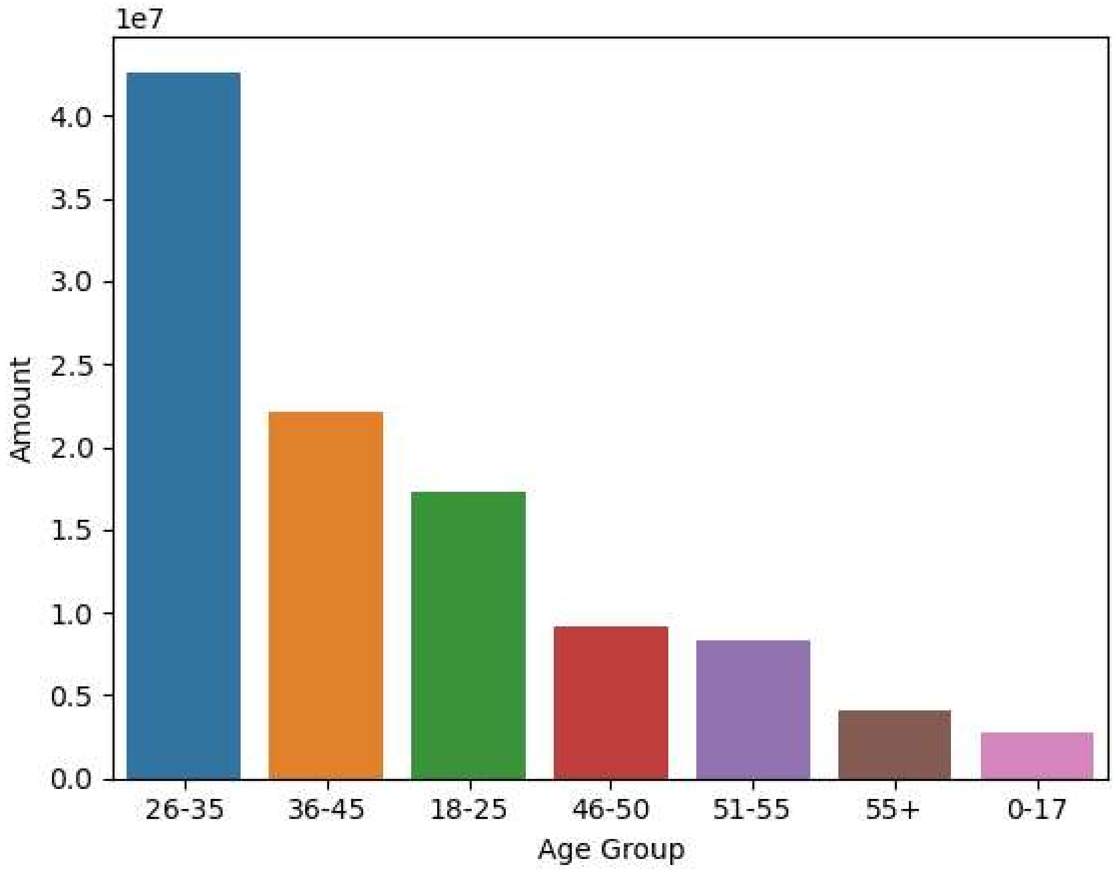
```
ax=sns.countplot(data=df,x='Age Group',hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



Age Group

In [21]:
```
#total amount vs age group
sales_age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values
(by='Amount',ascending=False)
sns.barplot(x='Age Group',y='Amount',data=sales_age)
```

Out[21]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



```
from above graphs we can that most of the buyer are of age group between 26-
35yrs female
```

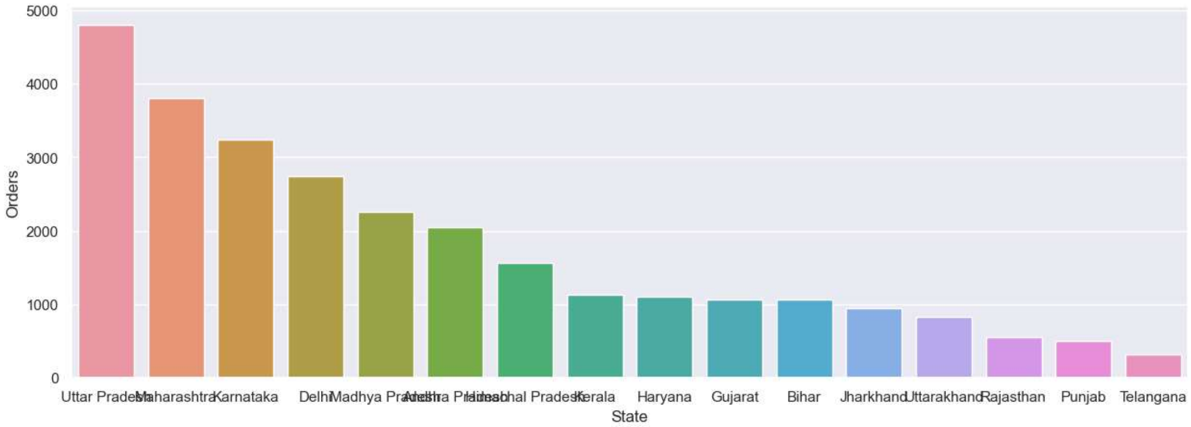
State

In [49]:

```
#total number of orders from top 10  states
sales_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values
(by='Orders',ascending=False)
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state,x='State',y='Orders')
```

Out[49]:

```
<Axes: xlabel='State', ylabel='Orders'>
```

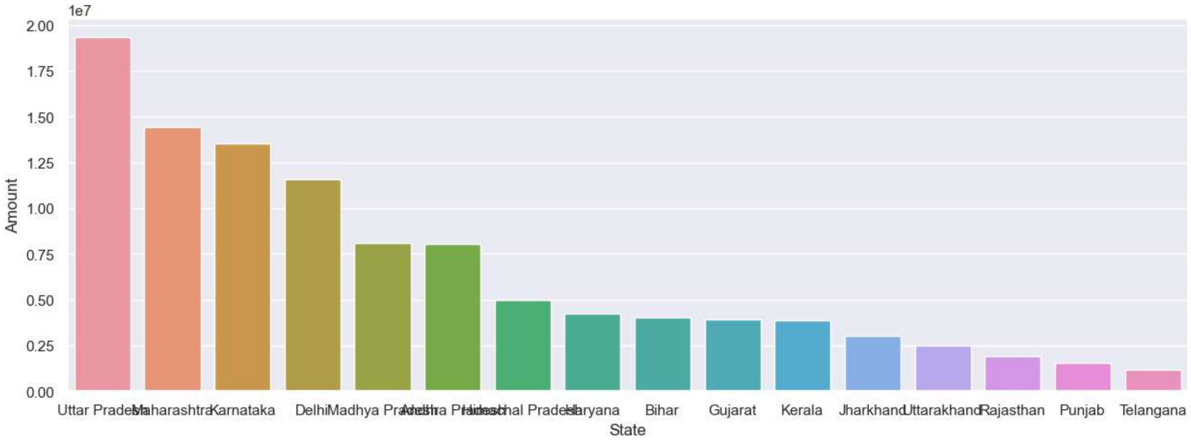


In [28]:

```
sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values
(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state,x='State',y='Amount')
```

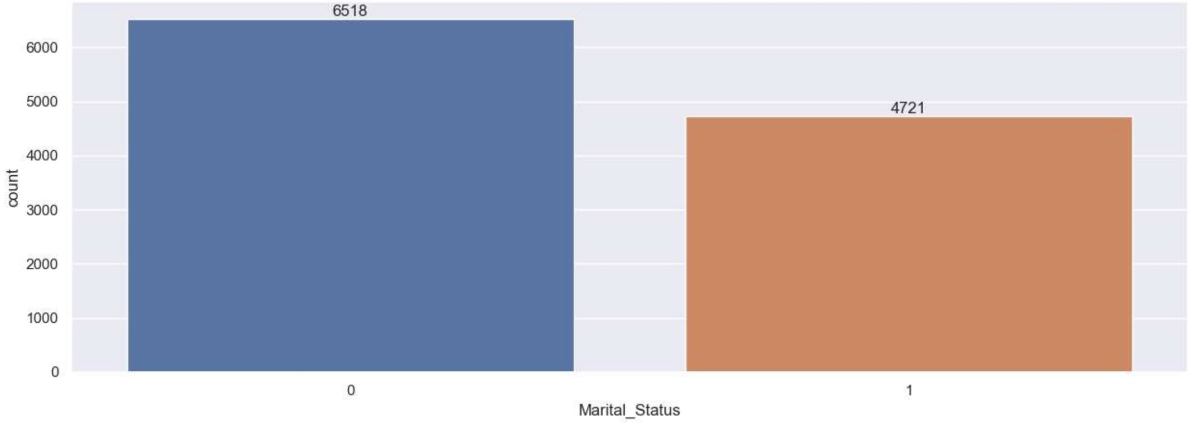
Out[28]:

```
<Axes: xlabel='State', ylabel='Amount'>
```



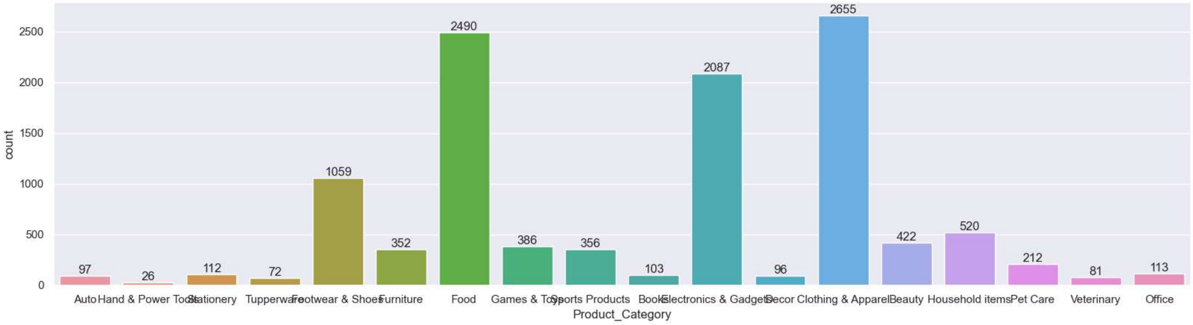
Marital Status

In [29]:
```
ax=sns.countplot(data=df,x='Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in  ax.containers:
    ax.bar_label(bars)
```



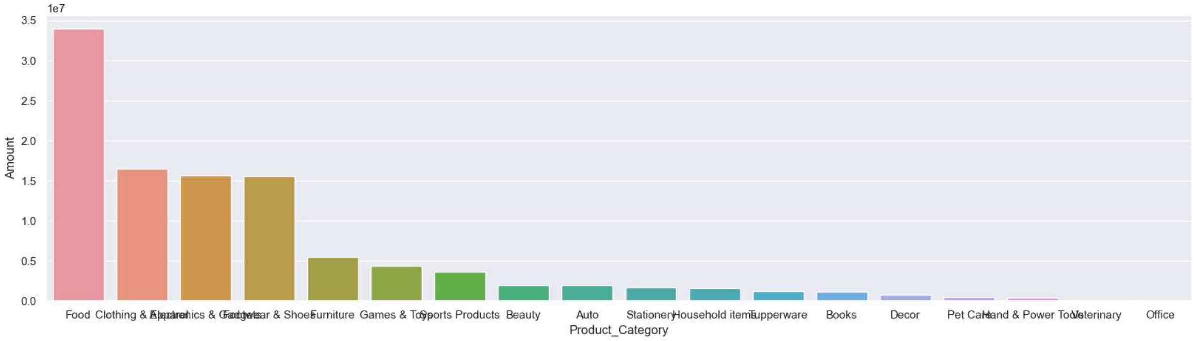
Product Category

In [23]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data=df,x='Product_Category')
for bars in ax.containers:
    ax.bar_label(bars)
```



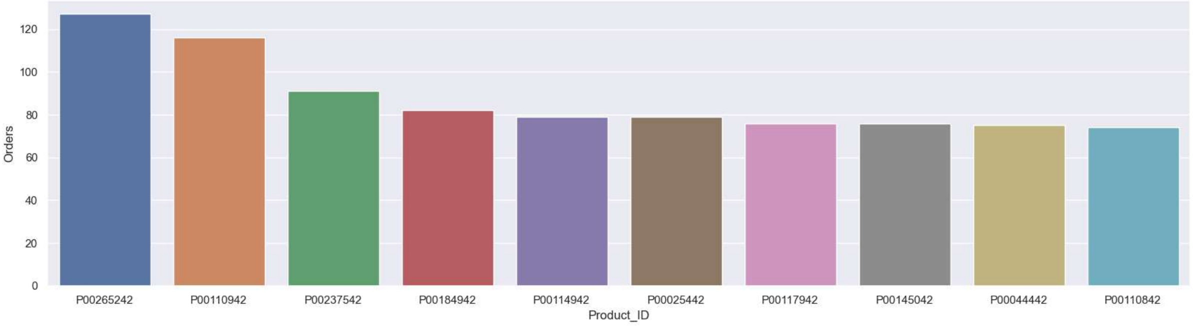
In [26]:
```
sales_state=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sor
(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state,x='Product_Category',y='Amount')
```

Out[26]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



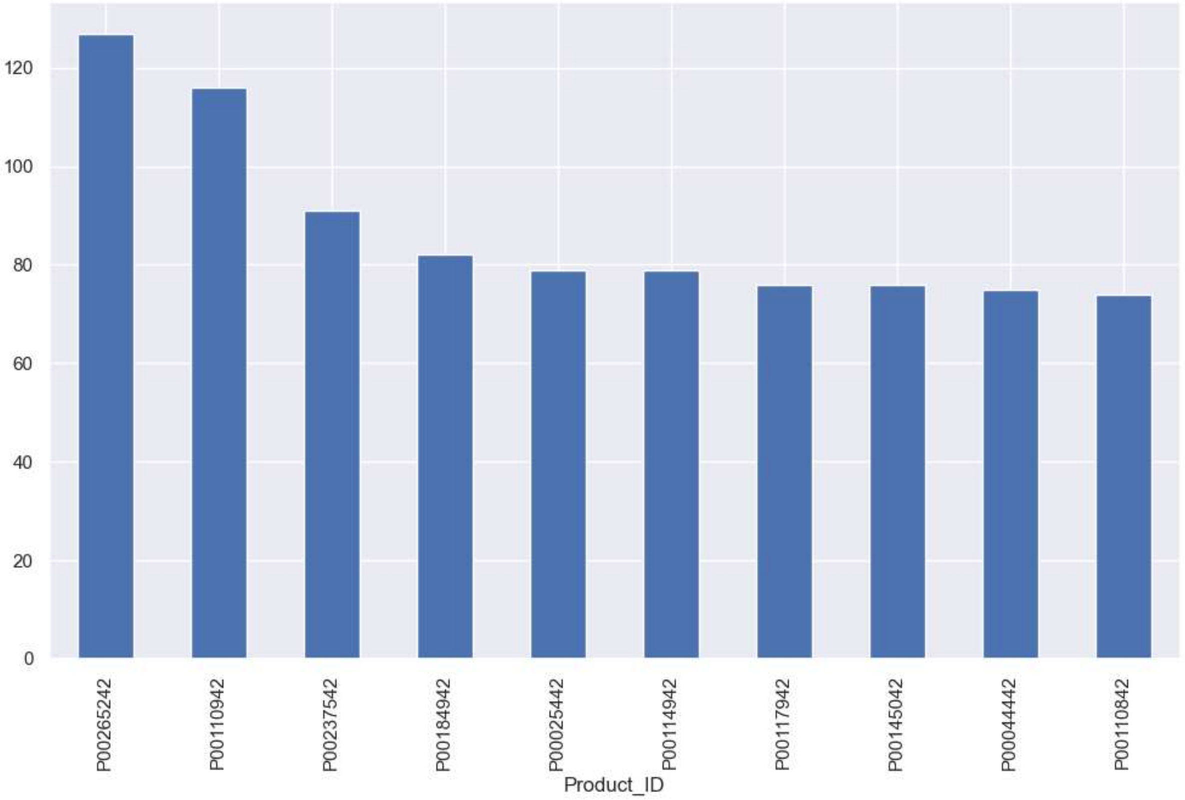
In [48]:
```
sales_state=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_valu
(by='Orders',ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state,x='Product_ID',y='Orders')
```

Out[48]:
```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```



In [47]:
```
#top 10 most sold product (same thing as above)
fig1=ax1=plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values
(ascending=False).plot(kind='bar')
```

Out[47]:
```
<Axes: xlabel='Product_ID'>
```



conclusion

Married women age group 26-27 years from UP,Maharashtra,Karnataka working in IT, Healthcare and Avviation are more likely to buy product from food clothing and electrnics category