diwali-sales-analysis

September 10, 2023

[1]:

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

[2]:

```
diwali = pd.read_csv(r'D:
 ↪\DatSets\Python_Diwali_Sales_Analysis\Python_Diwali_Sales_Analysis\Diwali␣
 ↪Sales Data.csv', encoding = 'unicode_escape')
diwali
```

[2]:

```
       User_ID    Cust_name Product_ID Gender Age Group  Age  Marital_Status  \
0      1002903    Sanskriti  P00125942      F     26-35   28               0
1      1000732       Kartik  P00110942      F     26-35   35               1
2      1001990        Bindu  P00118542      F     26-35   35               1
3      1001425       Sudevi  P00237842      M      0-17   16               0
4      1000588         Joni  P00057942      M     26-35   28               1
…        …           …        …    …      …  …           …
11246  1000695      Manning  P00296942      M     18-25   19               1
11247  1004089  Reichenbach  P00171342      M     26-35   33               0
11248  1001209        Oshin  P00201342      F     36-45   40               0
11249  1004023       Noonan  P00059442      M     36-45   37               0
11250  1002744      Brumley  P00281742      F     18-25   19               0

                State      Zone       Occupation Product_Category  Orders  \
0         Maharashtra   Western       Healthcare             Auto       1
1      Andhra Pradesh  Southern             Govt             Auto       3
2       Uttar Pradesh   Central       Automobile             Auto       3
3           Karnataka  Southern     Construction             Auto       2
4             Gujarat   Western  Food Processing             Auto       2
…               …       …            …          …     …
11246     Maharashtra   Western         Chemical           Office       4
11247         Haryana  Northern       Healthcare       Veterinary       3
11248  Madhya Pradesh   Central          Textile           Office       4
11249       Karnataka  Southern      Agriculture           Office       3
11250     Maharashtra   Western       Healthcare           Office       3
```

```
        Amount  Status  unnamed1
0      23952.0     NaN       NaN
1      23934.0     NaN       NaN
2      23924.0     NaN       NaN
3      23912.0     NaN       NaN
4      23877.0     NaN       NaN
…          …     …       …
11246    370.0     NaN       NaN
11247    367.0     NaN       NaN
11248    213.0     NaN       NaN
11249    206.0     NaN       NaN
11250    188.0     NaN       NaN

[11251 rows x 15 columns]
```

```
[3]: diwali.shape # to give toatl rows & columns in dataset
```

```
[3]: (11251, 15)
```

```
[4]: diwali.info() # To give total information about dataset
```

```
<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
```

```
[5]: diwali.columns #To give total columns names in dataset
```

```
[5]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
            'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
            'Orders', 'Amount', 'Status', 'unnamed1'],
           dtype='object')
```

```
[6]: diwali.drop(['Status','unnamed1'], axis = 1, inplace = True) # To drop the␣
       ↪Status& unnamed1 columns from dataset
```

```
[7]: diwali.columns
```

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

```
[8]: diwali.isnull().sum() # To check the total null vales in each columns in data␣
       ↪set
```

```
[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
```

```
[9]: diwali.dropna(inplace=True)
```

```
[10]: diwali.isnull().sum()
```

```
[10]: 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
```

```
[11]: diwali.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           11239 non-null  int64
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender            11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64
 6   Marital_Status    11239 non-null  int64
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64
 12  Amount            11239 non-null  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.2+ MB
```

```
[12]: diwali['Amount'] =diwali['Amount'].astype('int') # To change the --[Amount]␣
        ↪columns float to int format
```

```
[13]: diwali['Amount'].dtypes
```

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

```
[14]: diwali.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           11239 non-null  int64
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender            11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64
 6   Marital_Status    11239 non-null  int64
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64
 12  Amount            11239 non-null  int32
dtypes: int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

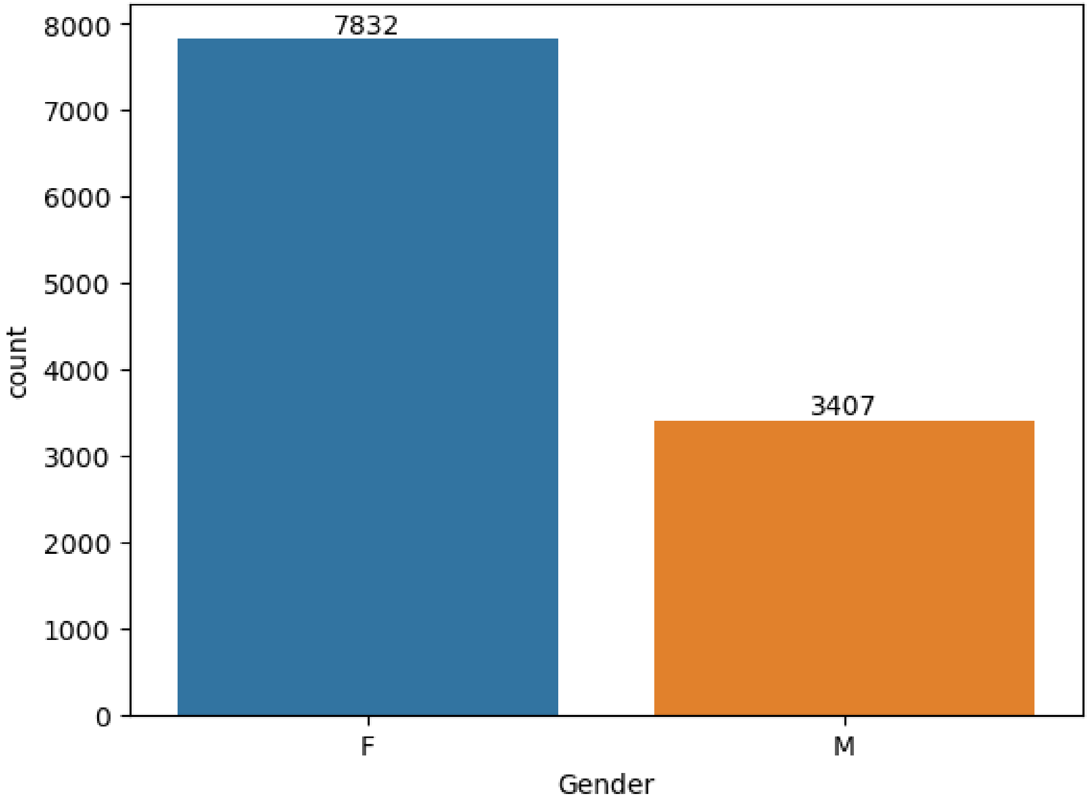
[15]:
```
diwali.describe() #Descriptive statistics of DataSet
```

[15]:
```
            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
```

0.1 Exploratory_Data_Analysis

Witch Gender Purchase the Most ?

[16]:
```
aa =sns.countplot(x='Gender', data=diwali) #In Seaborn, a countplot is a type␣
 ↪of categorical plot used to
for bars in aa.containers:
    aa.bar_label(bars)                                   #display the count␣
 ↪of observations in each category
```



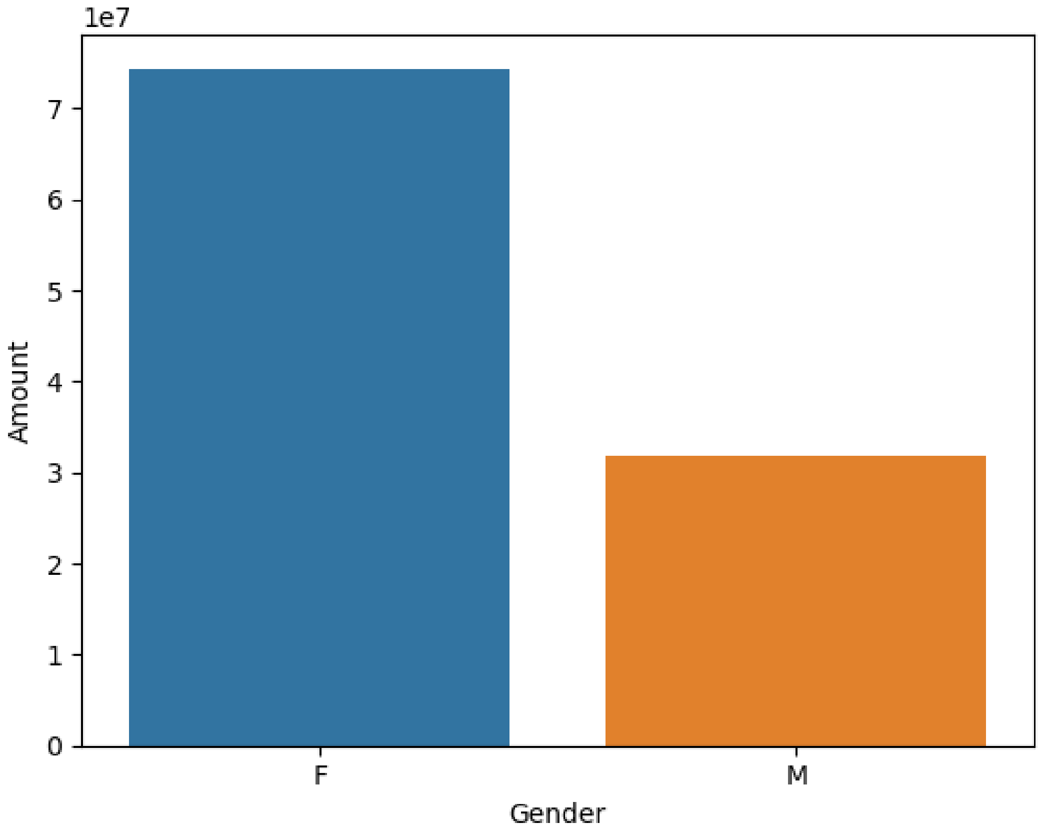
As per Graph Female's are more purchased in Diwali Sales as Compare to Male's

Who have the Most Purchaging Power ?

[17]:
```
#In Seaborn, a barplot is a statistical plot used to visualize the relationship␣
 ↪between
#a categorical variable and a continuous variable'''
```

[18]:
```
sales_Gen = diwali.groupby(['Gender'], as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount', ascending= False)
sns.barplot(x='Gender', y = 'Amount', data=sales_Gen)
```

[18]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



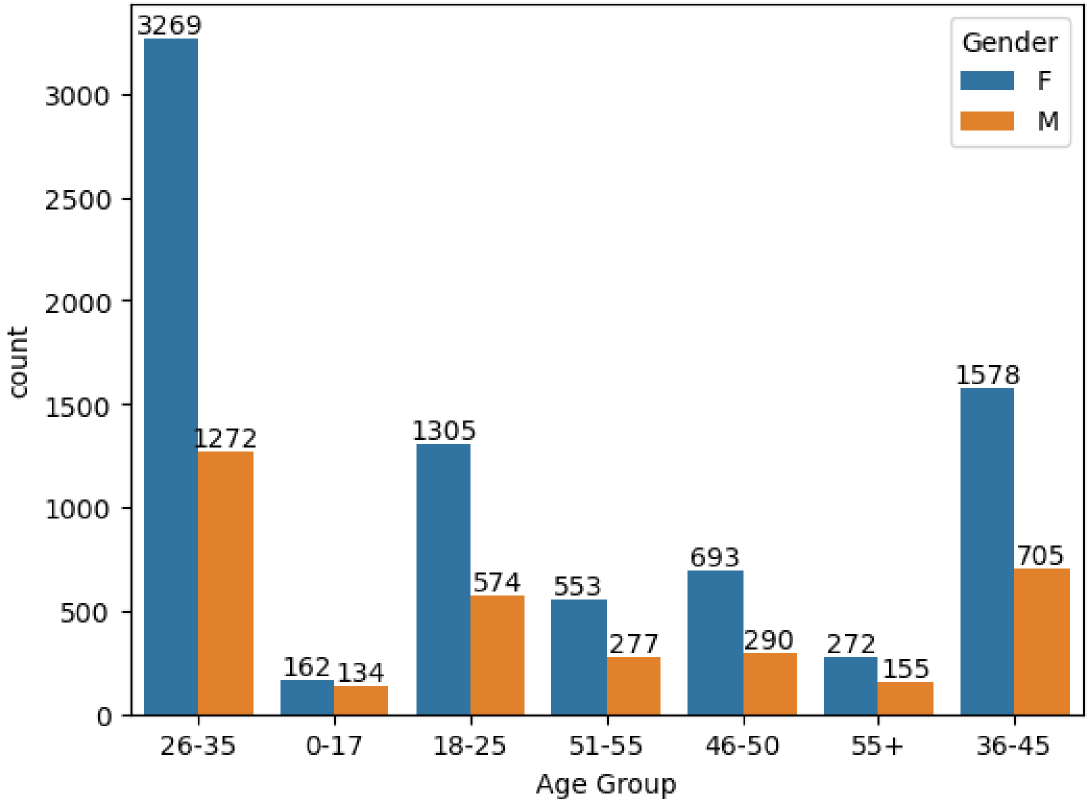
As per above graph we can see Female's Have Most Purchasing Power as compare to Males

0.1.1 Witch Age Group has Most Purchased in Diwali Sales

```
[19]: diwali.columns
```

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

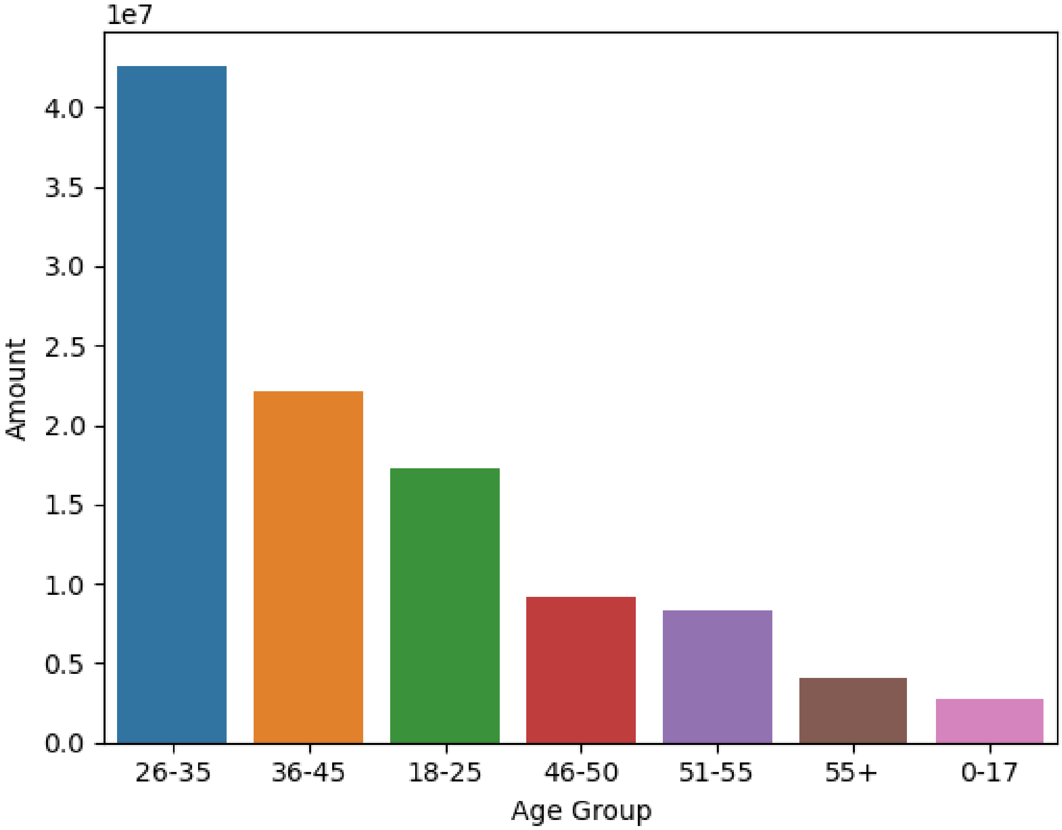
```
[20]: ab = sns.countplot(x='Age Group',data= diwali, hue = 'Gender')
      for bars in ab.containers:
          ab.bar_label(bars)
```



As per AboVe Graph We can conclde the Age_Group 26-35 have more Purchased

[21]:
```
#Total Amount vs AgeGroup
Sales_Age= diwali.groupby(['Age Group'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount', ascending= False)
sns.barplot(x='Age Group', y='Amount', data=Sales_Age)
```

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



0.2 What are the Top 10 States have orders in Diwali Sales

```
[22]: diwali.columns
```

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

```
[23]: Sales_State = diwali.groupby(['State'],as_index=False)['Orders'].sum().
        ↪sort_values(by='Orders',ascending=False).head(10)
```

```
[24]: Sales_State
```

```
[24]:                State   Orders
      14     Uttar Pradesh    4807
      10       Maharashtra    3810
      7          Karnataka    3240
      2              Delhi    2740
      9     Madhya Pradesh    2252
```

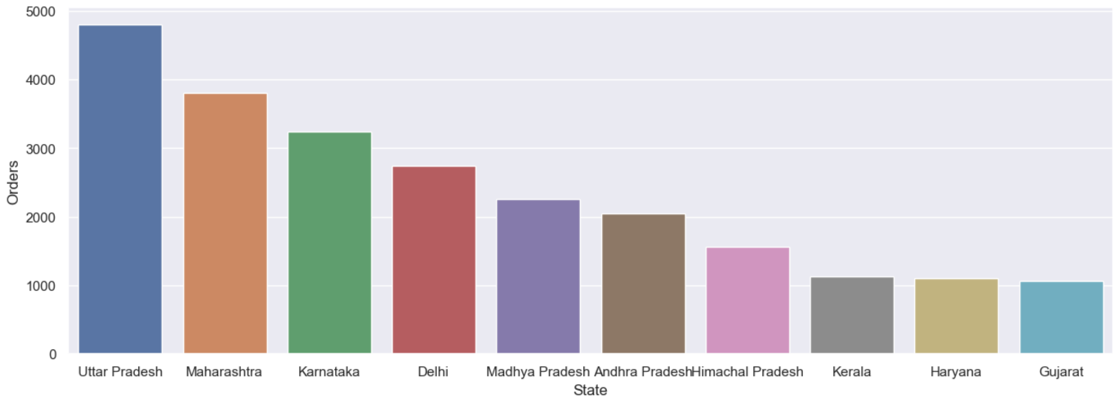
```
0     Andhra Pradesh    2051
5   Himachal Pradesh    1568
8             Kerala    1137
4            Haryana    1109
3            Gujarat    1066
```

[25]:

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

[25]:

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



From Above Graph UttarPradesh have more diwali sales

0.3 What are the Top 10 States have Spending Amount in Diwali Sales

[26]:

```
diwali.columns
```

[26]:

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

[27]:

```
Sales_Amount= diwali.groupby(['State'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False).head(10)
```

[28]:

```
Sales_Amount
```

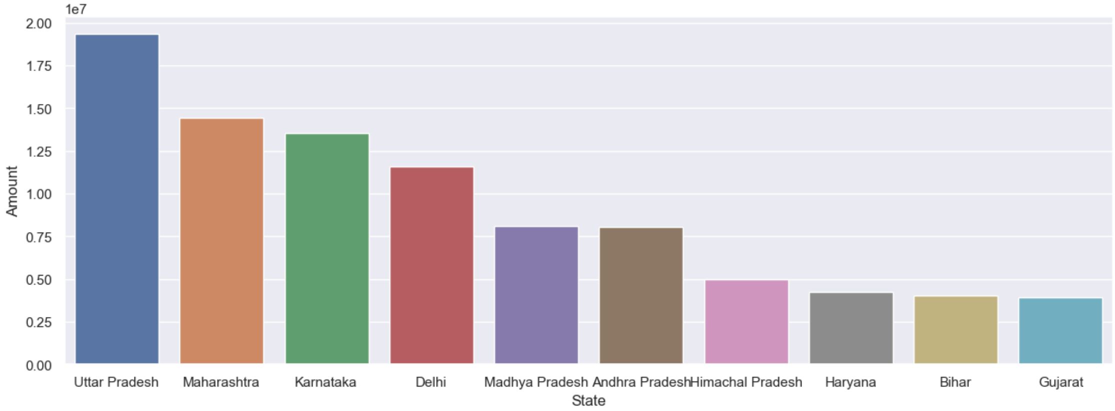
[28]:

```
               State    Amount
14     Uttar Pradesh  19374968
10       Maharashtra  14427543
7          Karnataka  13523540
2              Delhi  11603818
9     Madhya Pradesh   8101142
```

```
0     Andhra Pradesh   8037146
5   Himachal Pradesh   4963368
4            Haryana   4220175
1              Bihar   4022757
3            Gujarat   3946082
```

```
[29]: sns.set(rc={"figure.figsize":(15,5)})
      sns.barplot(x='State',y='Amount',data =Sales_Amount)
```

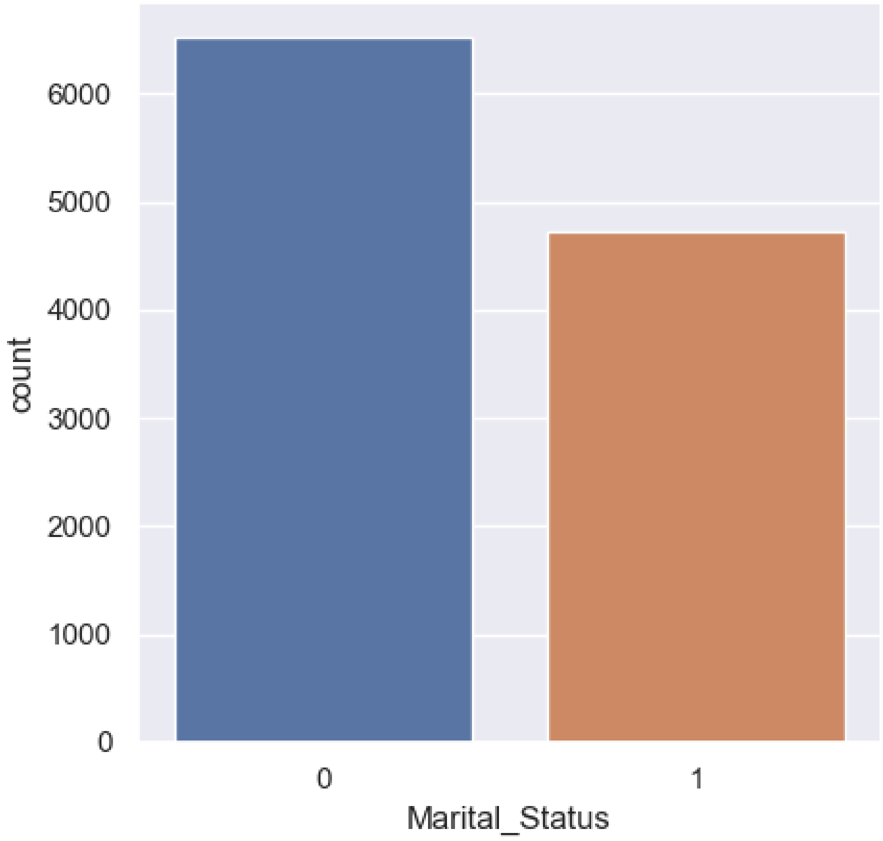
```
[29]: <Axes: xlabel='State', ylabel='Amount'>
```



```
[32]: diwali.columns
```

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

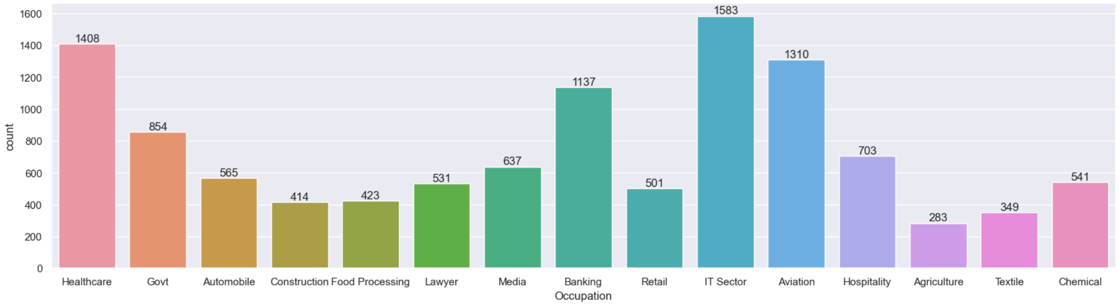
```
[44]: sns.countplot(x='Marital_Status' , data=diwali)
      sns.set(rc={"figure.figsize" : (5,5)})
```



As per above graph Married Peoples are purchased more in Diwali Sales

0.4 Howmany People in Each Sector Purchased More

[59]:
```
sns.set(rc={'figure.figsize':(20,5)})
ac= sns.countplot(x='Occupation', data=diwali)
for bars in ac.containers:
    ac.bar_label(bars)
```



0.5 Witch Occupition Peoples are Spending More Amount in Diwali Sales

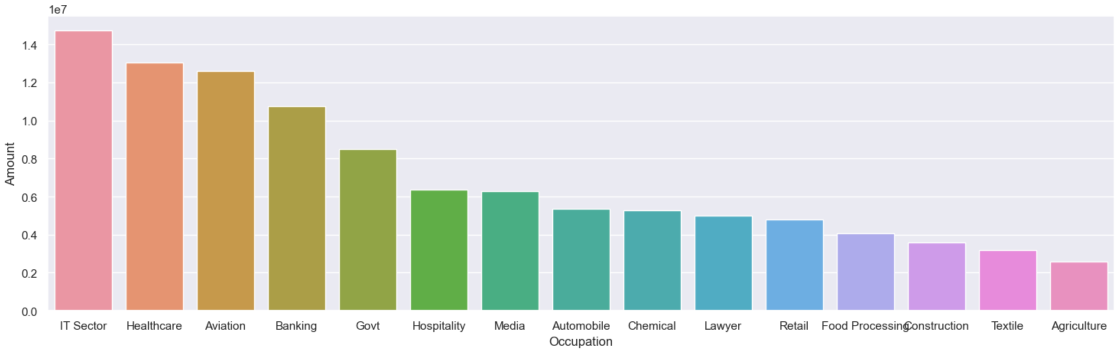
```
[51]: Sales_Pur = diwali.groupby(['Occupation'],as_index=False)['Amount'].sum().
        ↪sort_values(by='Amount', ascending=False)
```

```
[52]: Sales_Pur
```

```
[52]:         Occupation    Amount
      10       IT Sector  14755079
      8       Healthcare  13034586
      2         Aviation  12602298
      3          Banking  10770610
      7             Govt   8517212
      9      Hospitality   6376405
      12           Media   6295832
      1       Automobile   5368596
      4         Chemical   5297436
      11          Lawyer   4981665
      13          Retail   4783170
      6  Food Processing   4070670
      5     Construction   3597511
      14         Textile   3204972
      0      Agriculture   2593087
```

```
[57]: sns.set(rc={'figure.figsize':(18,5)})
      sns.barplot(x='Occupation',y='Amount', data=Sales_Pur)
```

```
[57]: <Axes: xlabel='Occupation', ylabel='Amount'>
```



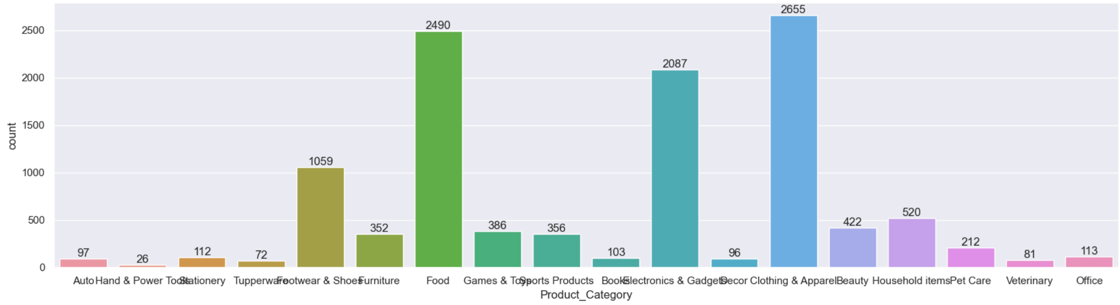
As per above graph IT Sector.Healthcare,Aviation are most spending amount in diwali sales

0.6 Based on Product Category

[60]:
```
diwali.columns
```

[60]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[61]:
```
sns.set(rc={'figure.figsize':(20,5)})
ad= sns.countplot(x='Product_Category', data=diwali)
for bars in ad.containers:
    ad.bar_label(bars)
```



[62]:
```
Sales_Product =diwali.groupby(['Product_Category'],as_index=False)['Amount'].
 ↪sum().sort_values(by='Amount',ascending =False)
```

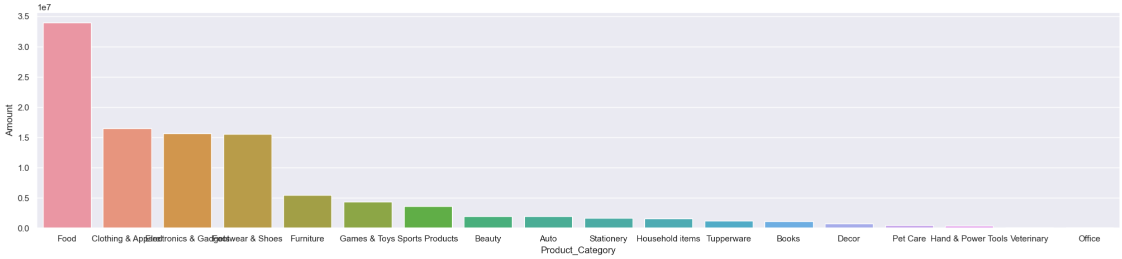
[63]:
```
Sales_Product
```

[63]:
```
         Product_Category    Amount
6                    Food  33933883
3      Clothing & Apparel  16495019
5   Electronics & Gadgets  15643846
7        Footwear & Shoes  15575209
8               Furniture   5440051
9             Games & Toys   4331694
14        Sports Products   3635933
1                  Beauty   1959484
0                    Auto   1958609
15             Stationery   1676051
11        Household items   1569337
16             Tupperware   1155642
2                   Books   1061478
4                   Decor    730360
13               Pet Care    482277
10     Hand & Power Tools    405618
```

```
17              Veterinary    112702
12                  Office     81936
```

[64]:
```
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(x='Product_Category', y='Amount', data=Sales_Product)
```

[64]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



0.7 Based on ProductID

[65]:
```
diwali.columns
```

[65]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[66]:
```
diwali['Product_ID']
```

[66]:
```
0        P00125942
1        P00110942
2        P00118542
3        P00237842
4        P00057942
           …
11246    P00296942
11247    P00171342
11248    P00201342
11249    P00059442
11250    P00281742
Name: Product_ID, Length: 11239, dtype: object
```

0.8 Conclusion

According to the analyzed data it can be determined that Female individuals who are married and aged between 26 and 35 years and who work within the Information Technology, Healthcare, and Aviation sectors in the states of Uttar Pradesh, Maharashtra, and Karnataka, have a higher propensity to purchase products from the categories of Food, Clothing, and Electronics