Data Analyst

Python Project

“Diwali Sales Analysis”

By – Sandesh Patidar

GitHub Link - https://github.com/Sandeshpatidar99/Diwali-Sales-Analysis

Under the Guidance of – “Rishabh Mishra”

(https://youtu.be/KgCgpClOkIs)

Objective –

To analyze the sales of a store during Diwali to generate meaningful insights and to help the store to understand the trends and their customers for better decision making in future.

Steps Followed –

- Importing required Libraries
- Loading the dataset
- Data cleaning and analyzing
- Performing EDA
- Final Conclusion

1. Importing Libraries

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

2. Loading the dataset

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

[5]:
```
df.shape
```

[5]:
```
(11251, 15)
```

[6]:
```
df.head()
```

[6]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN

3. Data Cleaning and Analyzing

Deleting empty columns

[4]:
```
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
```

[7]:
```
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 13 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
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

Checking for null values

```
[9]: pd.isnull(df).sum()
```

```
[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
```

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

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

```
[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
```

Changing Data Type

```
[12]: df['Amount'] = df['Amount'].astype('int')
```

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

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

Calculating max, min, mean etc values

```
[15]: df.describe()
```

[15]:

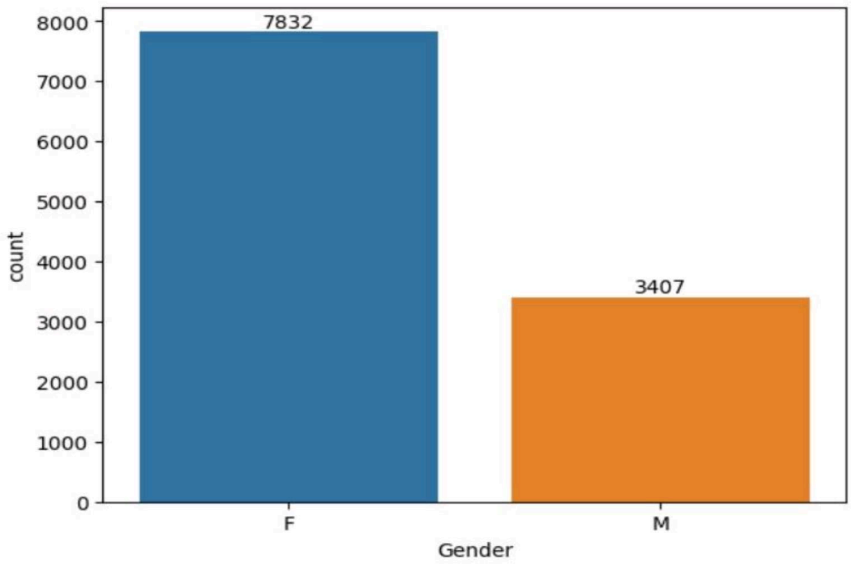
	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	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

4. Performing EDA

Gender Count

[16]:
```
ax = sns.countplot(x = 'Gender',data = df)

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

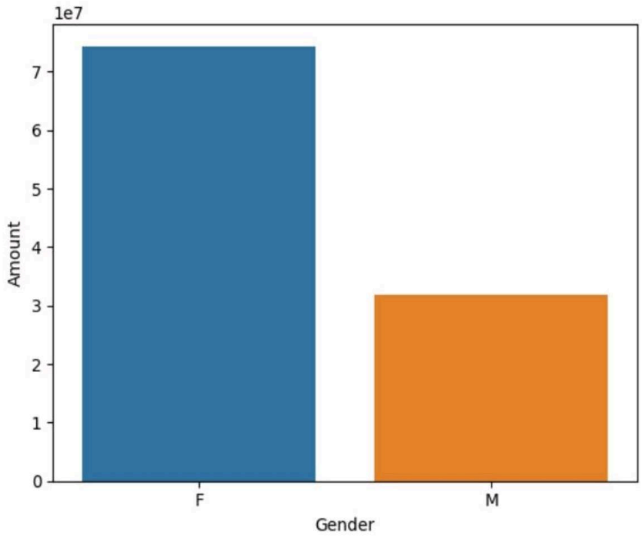


Amount spending by Gender

[17]:
```
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)
```

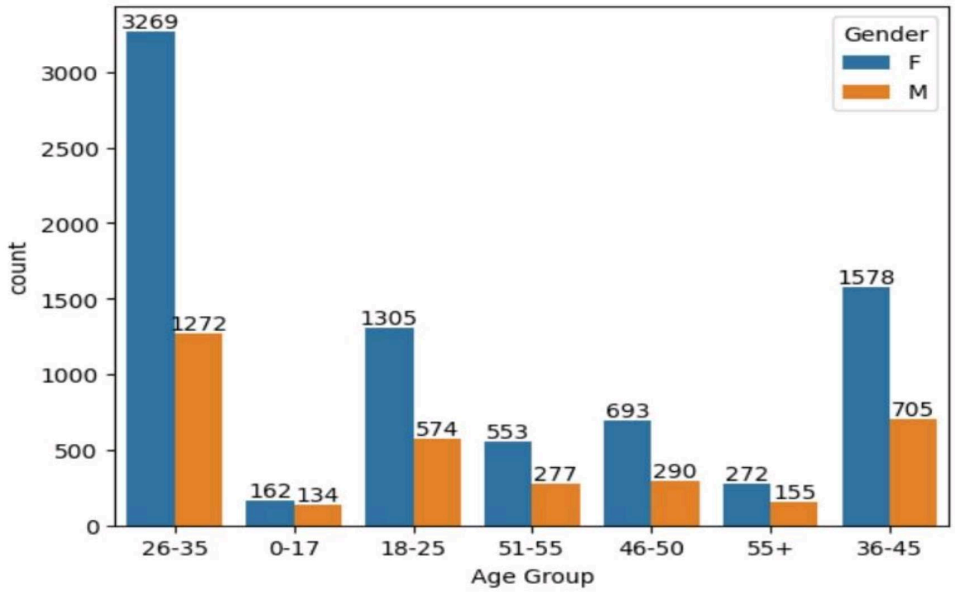
[17]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



Count by Age-group

[18]:
```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

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

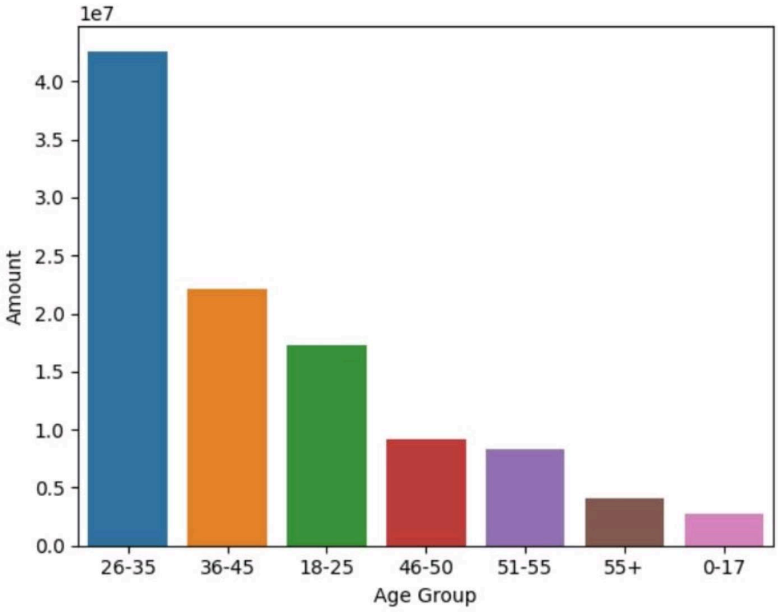


Amount spend by Age-group

[19]:
```
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)
```

[19]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

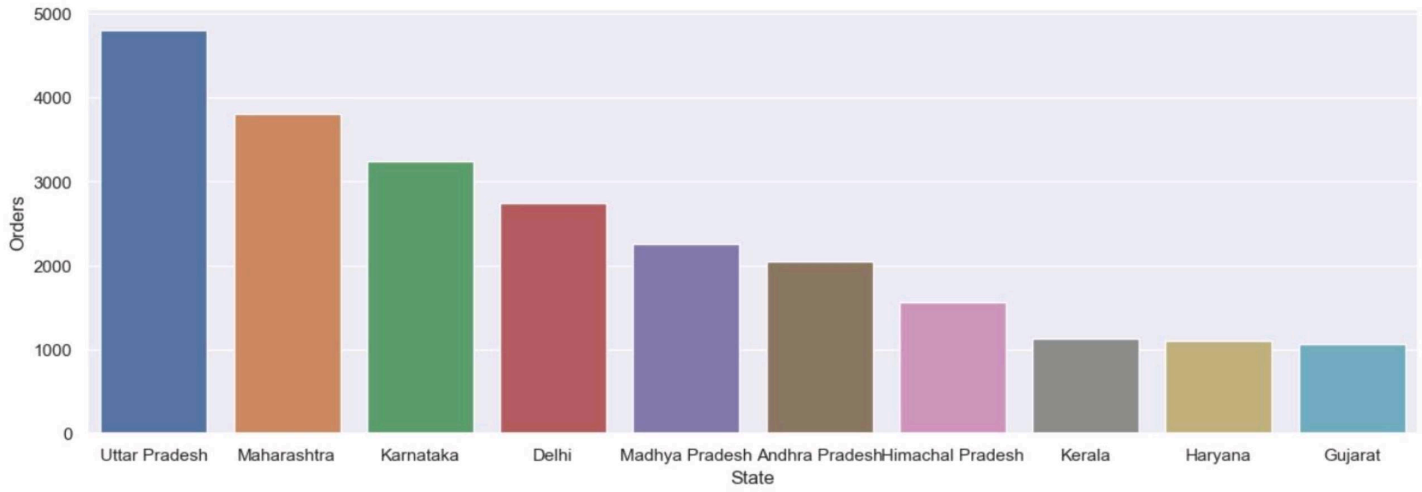


No. of Orders by State

```
[20]: sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

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

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

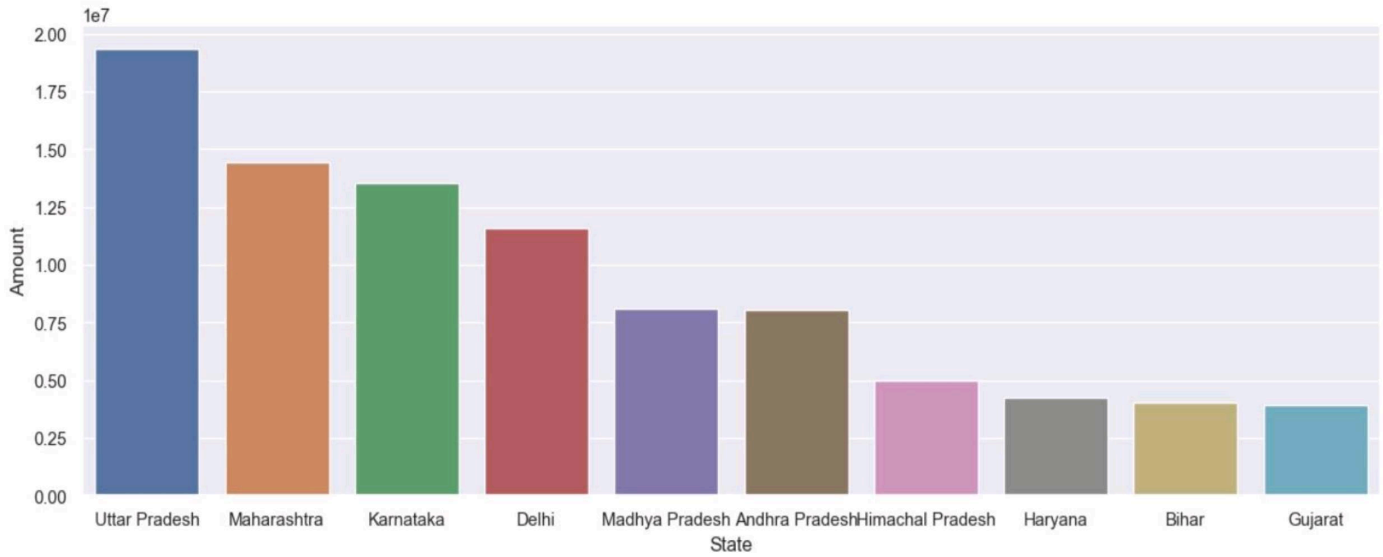


Amount spending by States

```
[21]: sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

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

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

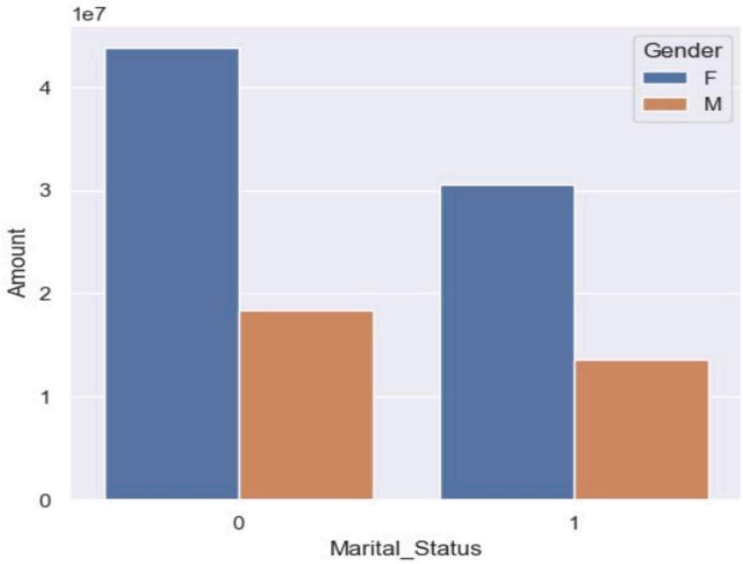


Amount spending by Marital Status

[23]:
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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

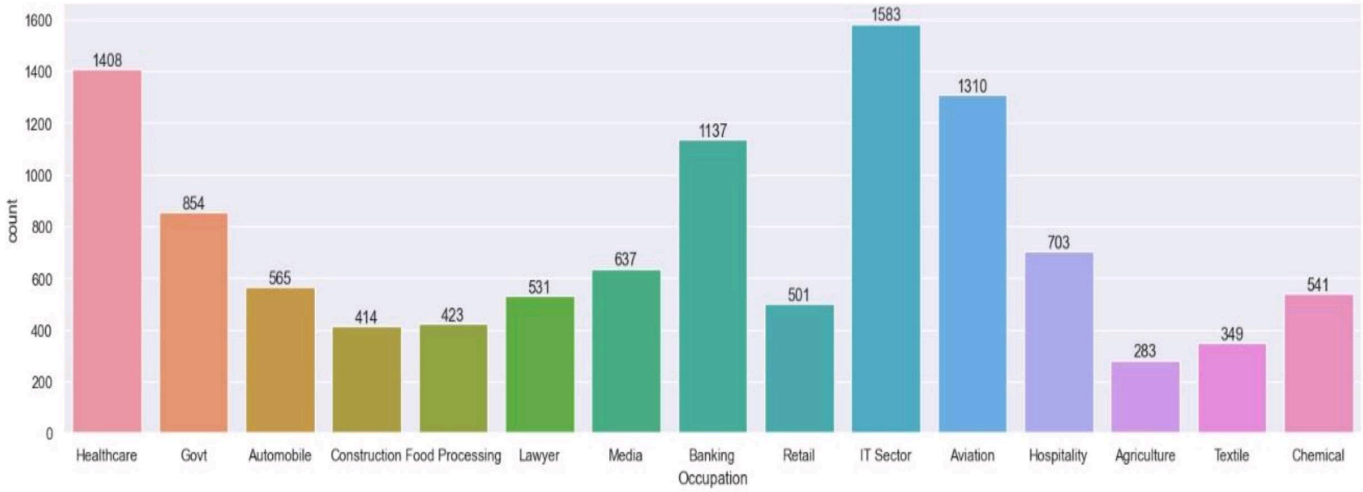
[23]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



Count by Occupation

[24]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

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

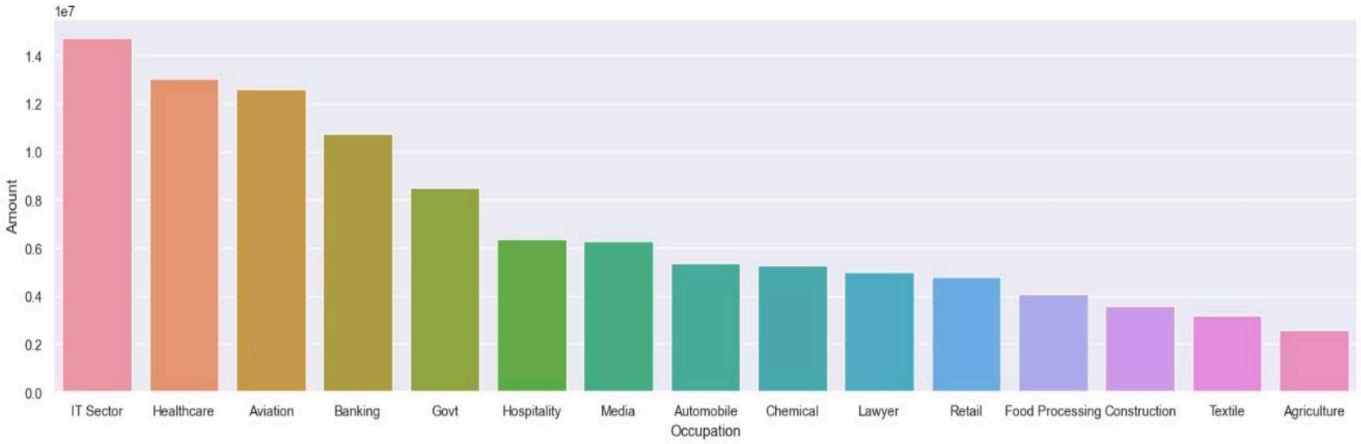


Amount spending by Occupation

[25]:
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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

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

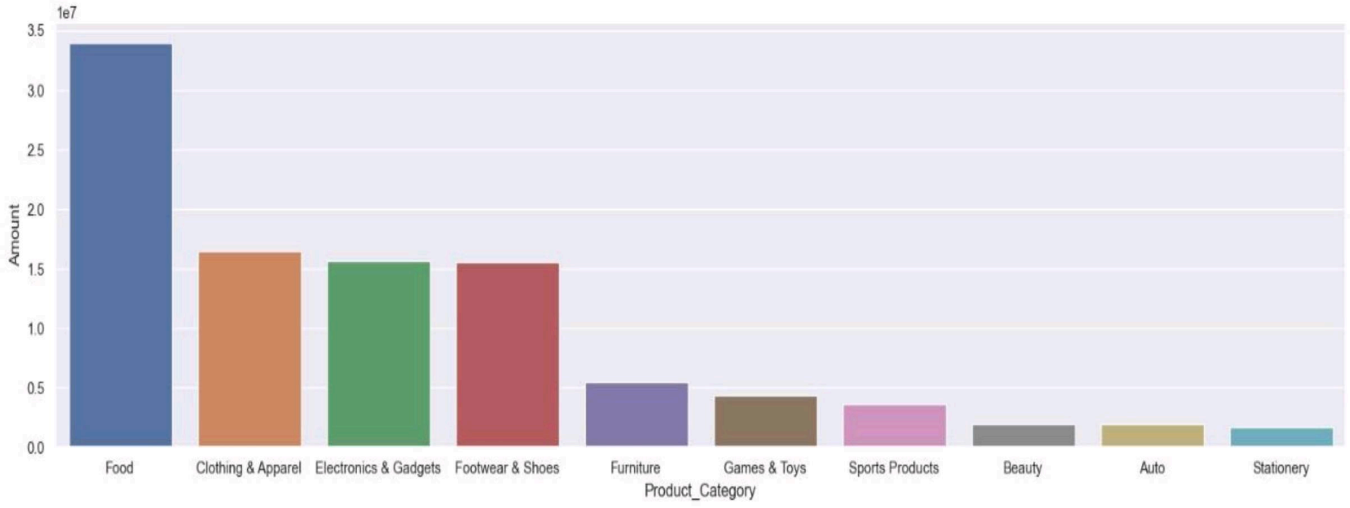


Amount by product category

[27]:
```
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)})
sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount')
```

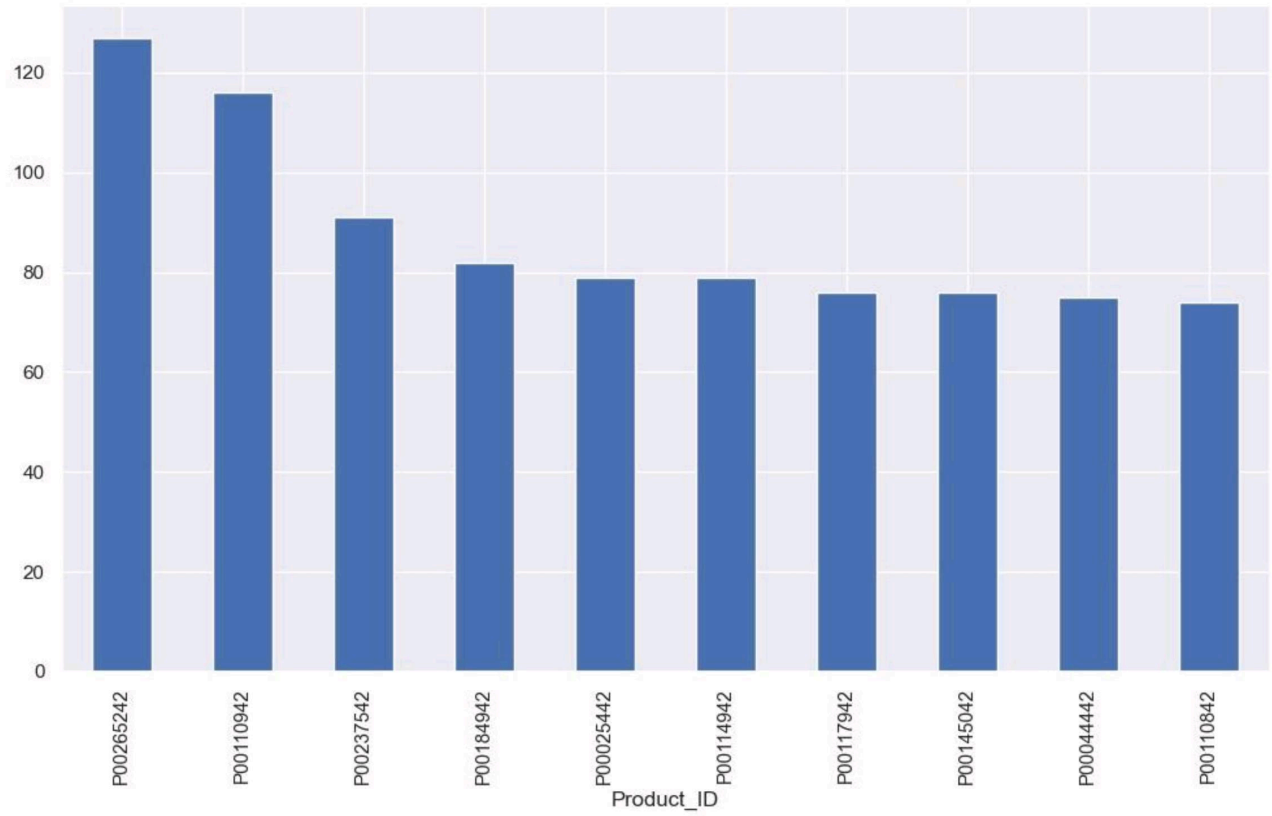
[27]: <Axes: xlabel='Product_Category', ylabel='Amount'>



Orders by Product id

```
[29]: fig1, ax1 = plt.subplots(figsize=(12,7))
      df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')

[29]: <Axes: xlabel='Product_ID'>
```



5. Conclusion

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

End