Importing Libraries

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

Importing CSV files

In [2]:
```
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

In [3]:
```
df.shape
```

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

In [4]:
```
df
```

Out[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Ma
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhr
2	1001990	Bindu	P00118542	F	26-35	35	1	Utta
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	Ma
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	Ma

11251 rows × 15 columns

Data Cleaning

In [5]:
```
df.head(10)
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	St
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharasl
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prad
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prad
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnat
4	1000588	Joni	P00057942	M	26-35	28	1	Guj
5	1000588	Joni	P00057942	M	26-35	28	1	Hima Prad
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Prad
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharasl
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Prad
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Prad

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]:

```
df.shape
```

Out[7]:

```
(11251, 15)
```

In [8]:

```
# removing unrelated columns
df.drop(["Status", "unnamed1"] ,axis=1, inplace= True)
```

In [9]:

```
df.shape
```

Out[9]:

```
(11251, 13)
```

In [10]:

```
# Removing Duplicate Values
```

```
df = df.drop_duplicates()
df.shape
```

Out[10]:

```
(11243, 13)
```

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              12
dtype: int64
```

In [28]:

```
# We have 12 null values in Amount columns, which need to be removed
# df.dropna(inplace = True)
df = df.dropna()
df.shape
```

Out[28]:

```
(11231, 13)
```

In [13]:

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

Out[13]:

```
dtype('float64')
```

In [29]:

```
# Change DataType
df['Amount'] = df['Amount'].astype('int')
df['Amount'].dtypes
```

Out[29]:

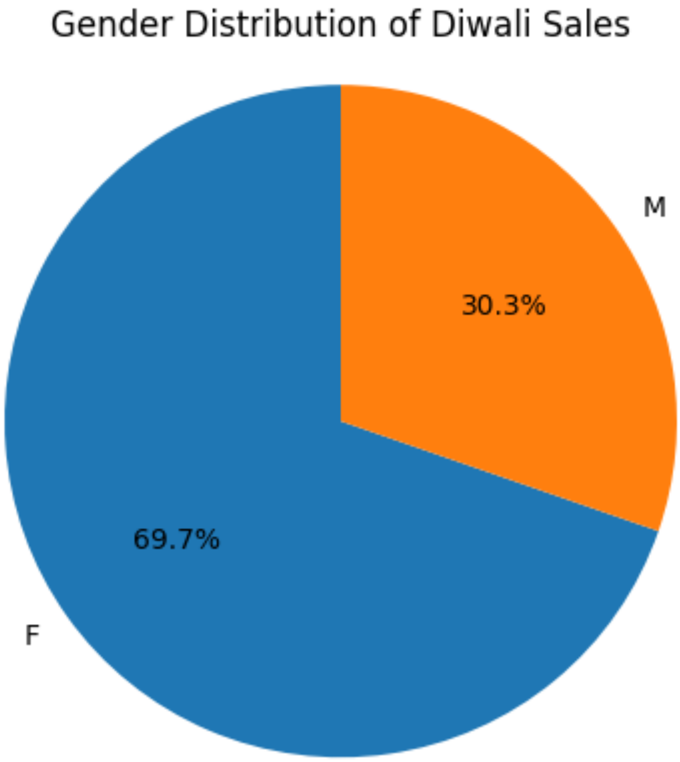
```
dtype('int64')
```

Exploratory Data Analysis

Gender

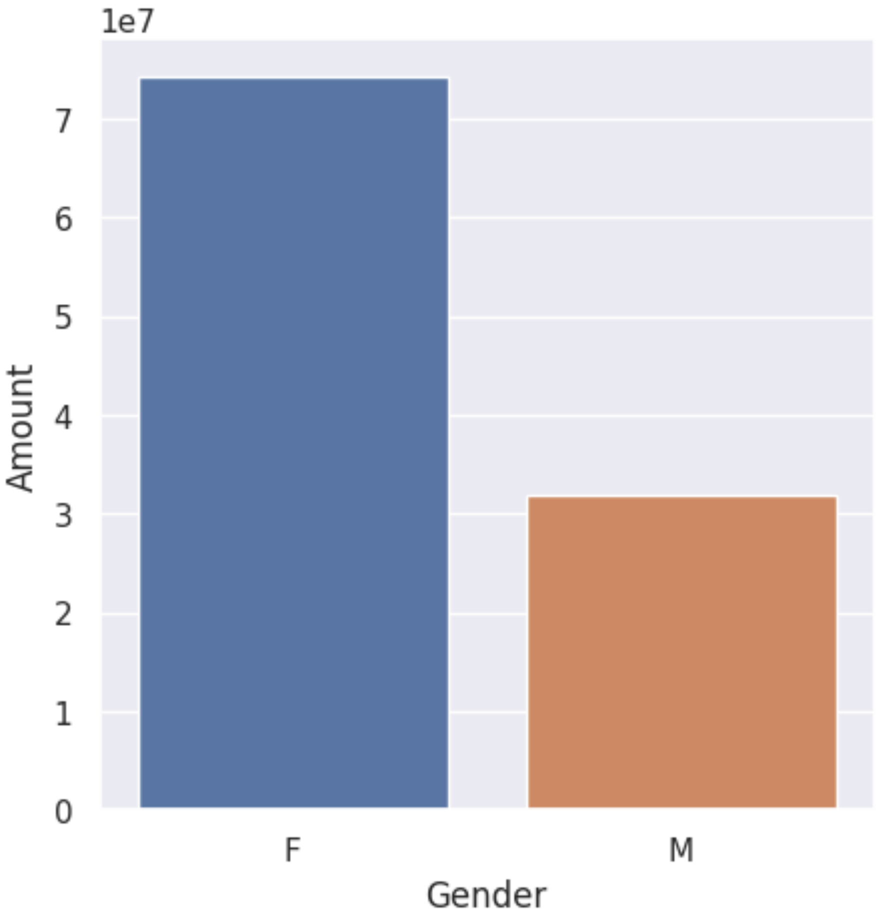
In [15]:

```
gender_counts = df['Gender'].value_counts()

# Create a pie chart
fig, ax = plt.subplots()
ax.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', star
ax.axis('equal')  # Equal aspect ratio ensures the pie chart is circular.

# Add a title
plt.title("Gender Distribution of Diwali Sales")

# Show the pie chart
plt.show()
```



```
In [16]: # gender vs total amount

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

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

```
Out[16]: <Axes: xlabel='Gender', ylabel='Amount'>
```



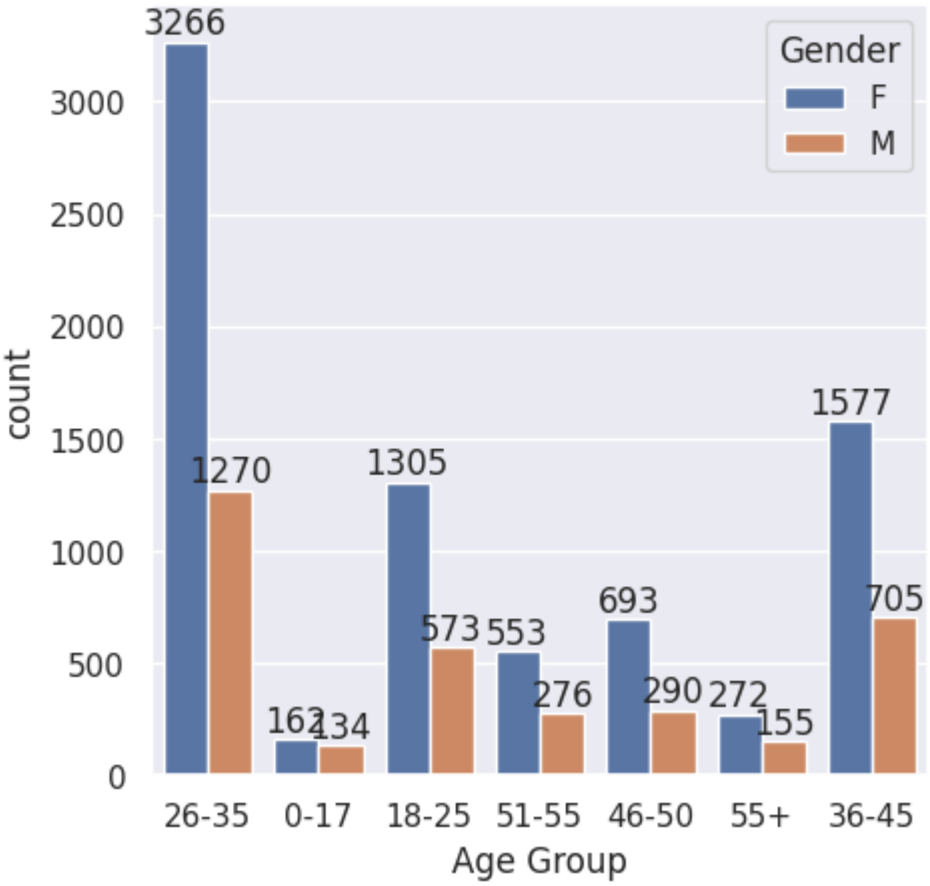
We can conclude that female buyers are more than double of male, not just count even the purchasing power of females are greater than men

---

Age

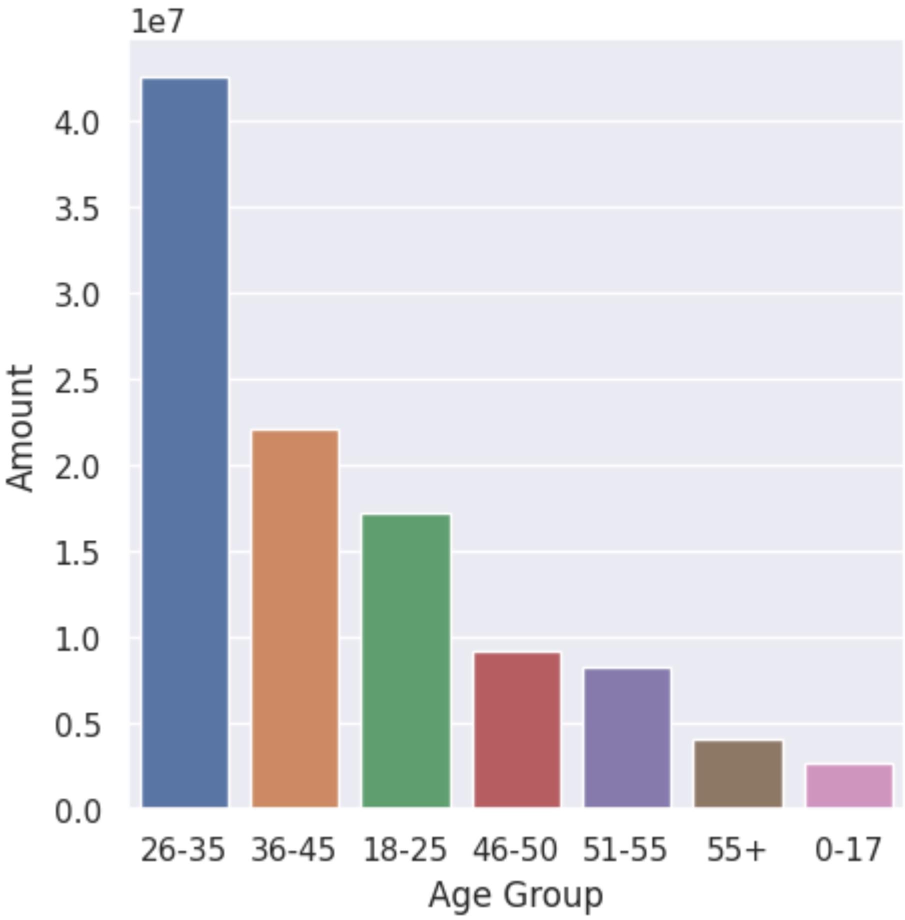
In [17]:

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

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



In [18]:

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

ax = sns.barplot(x = 'Age Group',y= 'Amount' ,data = sales_age)

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



From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

---

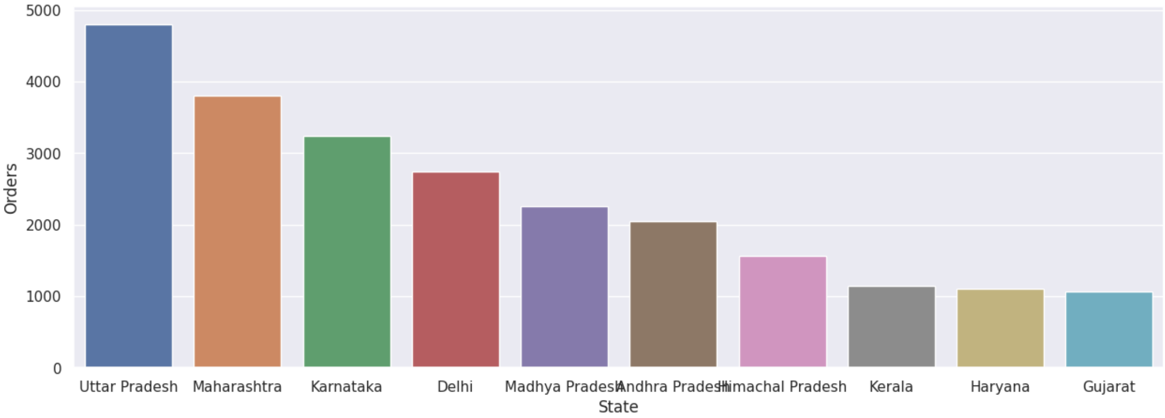
total number of orders from top 10 states

In [19]:

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

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

Out[19]:

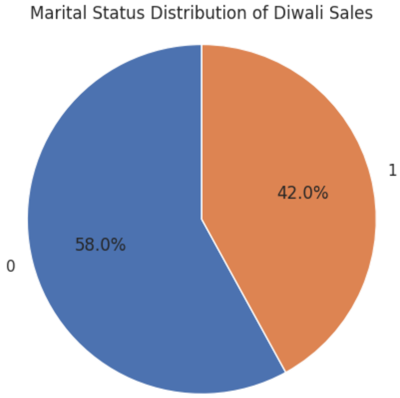
```
<Axes: xlabel='State', ylabel='Orders'>
```



Marital status

In [20]:

```
mar_counts = df['Marital_Status'].value_counts()

# Create a pie chart
fig, ax = plt.subplots()
ax.pie(mar_counts, labels=mar_counts.index, autopct='%1.1f%%', startangle
ax.axis('equal')  # Equal aspect ratio ensures the pie chart is circular.

# Add a title
plt.title("Marital Status Distribution of Diwali Sales")

# Show the pie chart
plt.show()
```

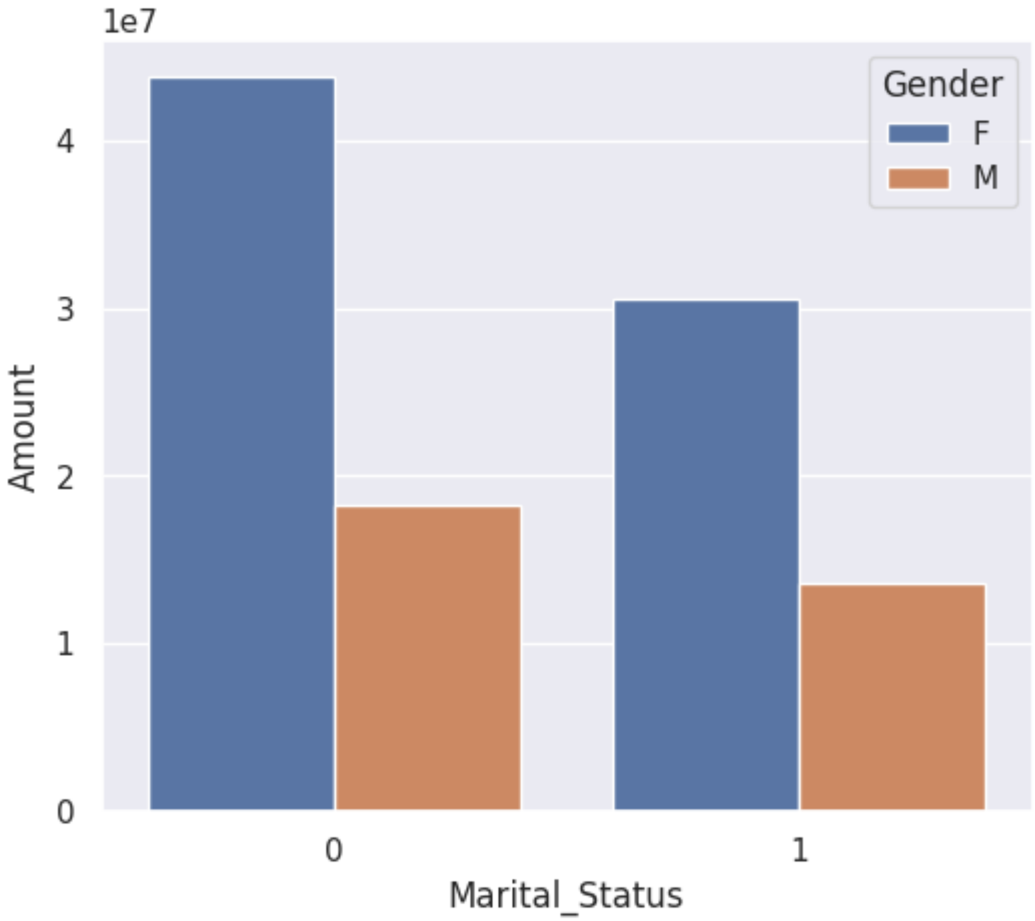


In [21]:

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

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

Out[21]:

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

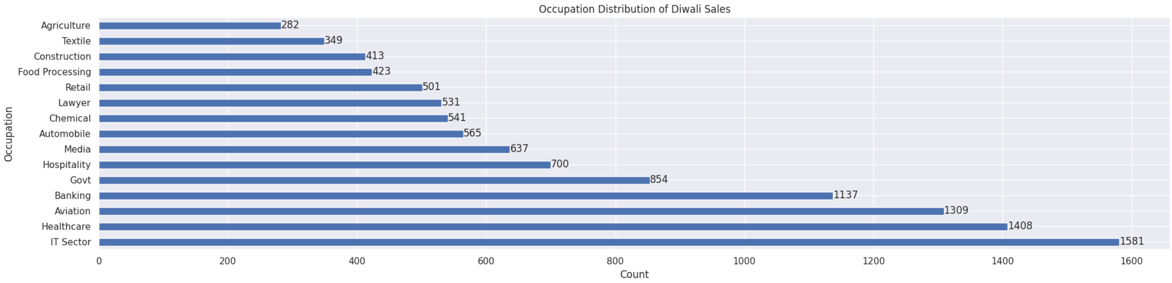


From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

---

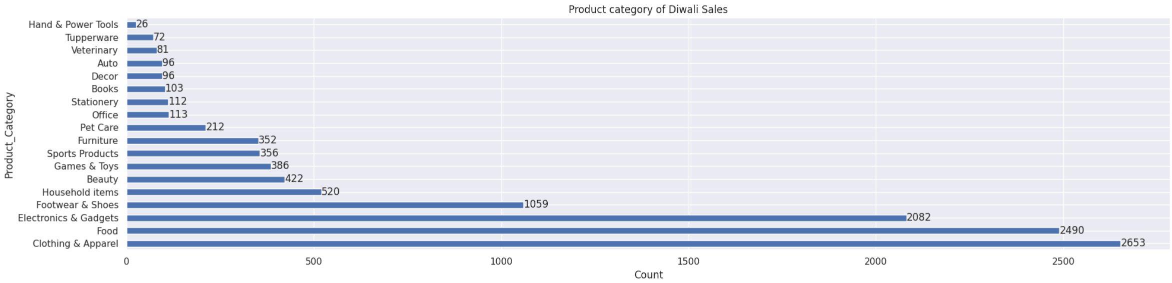
Occupation

In [30]:
```
occupation_counts = df['Occupation'].value_counts()

fig, ax = plt.subplots()
occupation_counts.plot(kind='barh', ax=ax)

ax.set_xlabel('Count')
ax.set_ylabel('Occupation')
plt.title("Occupation Distribution of Diwali Sales")

sns.set(rc={'figure.figsize':(20,5)})

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

plt.tight_layout()

plt.show()
```



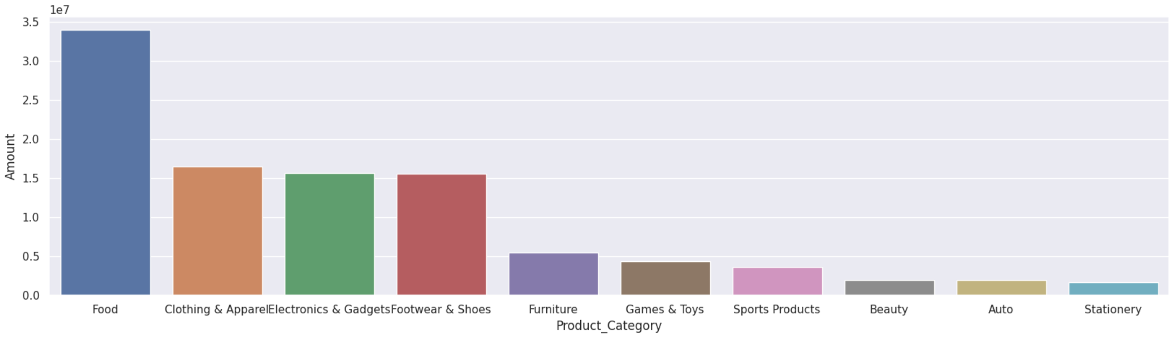
Product Category

In [23]:
```python
# sns.set(rc={'figure.figsize':(20,5)})
# ax = sns.countplot(data = df, x = 'Product_Category')

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

prod = df['Product_Category'].value_counts()

fig, ax = plt.subplots()
prod.plot(kind='barh', ax=ax)

ax.set_xlabel('Count')
ax.set_ylabel('Product_Category')
plt.title("Product category of Diwali Sales")

sns.set(rc={'figure.figsize':(20,5)})

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

plt.tight_layout()

plt.show()
```



In [24]:
```python
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')
```

Out[24]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



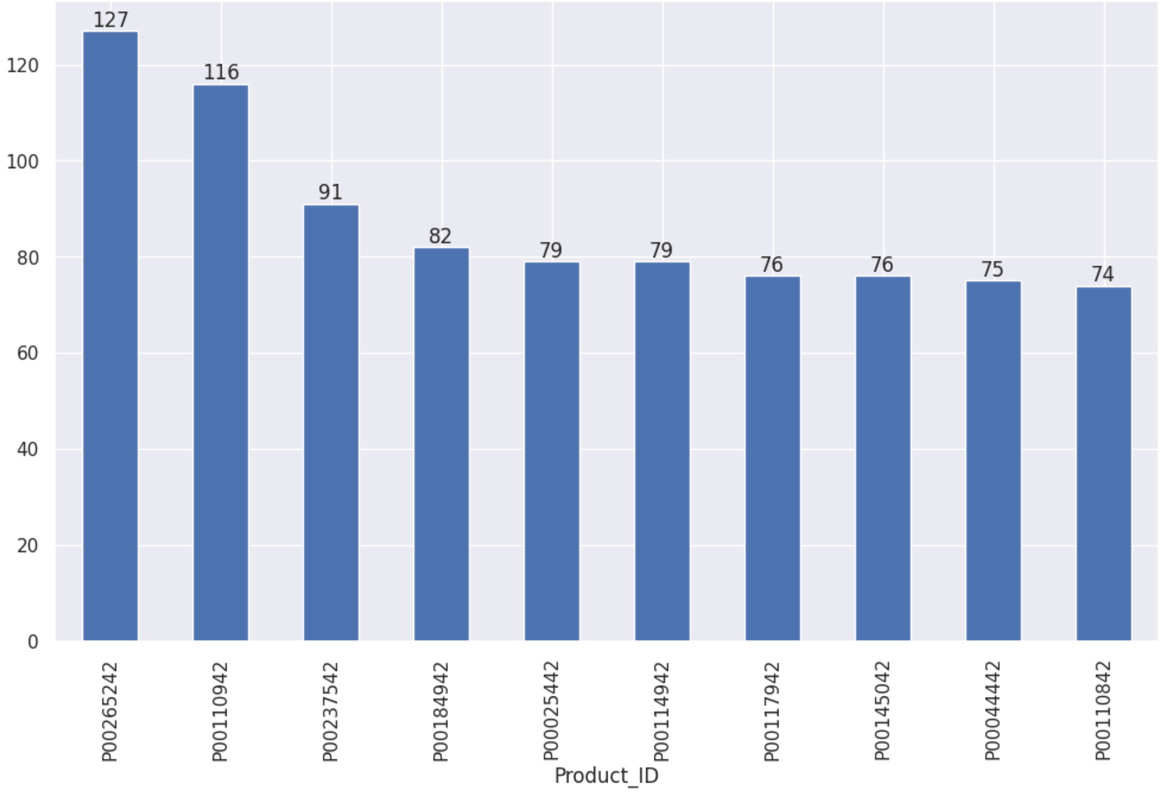
From above graphs we can coclude

Food, Clothing and Electronics category are the most sold products

---

Top 10 most sold products

```
In [25]: fig1, ax1 = plt.subplots(figsize=(12,7))
         df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascendi
         for bars in ax1.containers:
             ax1.bar_label(bars)
```



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 specially from Food, Clothing and Electronics category during Diwali!