```
# import python libraries

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

# import csv file
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')

df.shape
```

```
(11251, 15)
```

```
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11251,\n  \"fields\":
[\n    {\n      \"column\": \"User_ID\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1716,\n        \"min\":
1000001,\n        \"max\": 1006040,\n        \"num_unique_values\":
3755,\n        \"samples\": [\n          1005905,\n          1003730,\
n          1005326\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Cust_name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1250,\n
\"samples\": [\n          \"Nida\",\n          \"Lacy\",\n
\"Caudle\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Product_ID\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2351,\n
\"samples\": [\n          \"P00224442\",\n          \"P00205242\",\n
\"P00347442\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"M\",\n          \"F\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age Group\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n          \"26-
35\",\n          \"0-17\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12,\n        \"min\": 12,\n
\"max\": 92,\n        \"num_unique_values\": 81,\n        \"samples\":
[\n          18,\n          28\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Marital_Status\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
```

```
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"State\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 16,\n
\"samples\": [\n          \"Maharashtra\",\n          \"Andhra\\
u00a0Pradesh\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Zone\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Southern\",\n          \"Eastern\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Occupation\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 15,\n        \"samples\": [\n
\"Retail\",\n          \"Aviation\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Product_Category\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 18,\n        \"samples\": [\n
\"Auto\",\n          \"Hand & Power Tools\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Orders\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 1,\n
\"min\": 1,\n        \"max\": 4,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          3,\n          4\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Amount\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
5222.355869186444,\n        \"min\": 188.0,\n        \"max\":
23952.0,\n        \"num_unique_values\": 6584,\n        \"samples\":
[\n          19249.0,\n          13184.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Status\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": null,\n
\"min\": null,\n        \"max\": null,\n        \"num_unique_values\":
0,\n        \"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"unnamed1\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": null,\n        \"min\": null,\n
\"max\": null,\n        \"num_unique_values\": 0,\n
\"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
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
```

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

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

```
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
```

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

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

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

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

```
df.columns
```

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

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


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11239,\n  \"fields\":
[\n    {\n      \"column\": \"User_ID\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1716,\n        \"min\":
1000001,\n        \"max\": 1006040,\n        \"num_unique_values\":
3752,\n        \"samples\": [\n          1002014,\n          1003491,\
n          1001842\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Cust_name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1250,\n
\"samples\": [\n          \"Hallsten\",\n          \"Shubham\",\n
\"Riya\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Product_ID\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2350,\n
\"samples\": [\n          \"P00133342\",\n          \"P00302142\",\n
\"P00227542\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"M\",\n          \"F\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age Group\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n          \"26-
35\",\n          \"0-17\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12,\n        \"min\": 12,\n
\"max\": 92,\n        \"num_unique_values\": 81,\n        \"samples\":
[\n          38,\n          28\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Shaadi\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"State\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 16,\n
```

\"samples\": [\n   \"Maharashtra\",\n   \"Andhra\\
u00a0Pradesh\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Zone\",\n   \"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n   \"samples\": [\n
\"Southern\",\n   \"Eastern\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Occupation\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 15,\n   \"samples\": [\n   \"IT
Sector\",\n   \"Hospitality\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Product_Category\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 18,\n   \"samples\": [\n
\"Auto\",\n   \"Hand & Power Tools\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Orders\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 1,\n
\"min\": 1,\n   \"max\": 4,\n   \"num_unique_values\": 4,\n
\"samples\": [\n   3,\n   4\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Amount\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 5222,\n
\"min\": 188,\n   \"max\": 23952,\n   \"num_unique_values\":
6583,\n   \"samples\": [\n   19247,\n   5293\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe"}

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

{"summary":"{\n \"name\": \"df\",\n \"rows\": 8,\n \"fields\": [\n
{\n   \"column\": \"User_ID\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 461312.8299795869,\n
\"min\": 1716.0388257054726,\n   \"max\": 1006040.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
1003003.5217546045,\n   1003064.0,\n   11239.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 3960.7779927819724,\n   \"min\": 12.0,\n   \"max\":
11239.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n
35.41035679330901,\n   33.0,\n   11239.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Marital_Status\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
3973.439417307323,\n   \"min\": 0.0,\n   \"max\": 11239.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n

```
0.42005516505027135,\n          1.0,\n          0.4935894048750214\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Orders\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
3972.7985251346995,\n        \"min\": 1.0,\n        \"max\": 11239.0,\
n        \"num_unique_values\": 7,\n        \"samples\": [\n
11239.0,\n          2.4896343091022333,\n          3.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Amount\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
7024.070687950828,\n        \"min\": 188.0,\n        \"max\":
23952.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
9453.610552540262,\n          8109.0,\n          11239.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


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

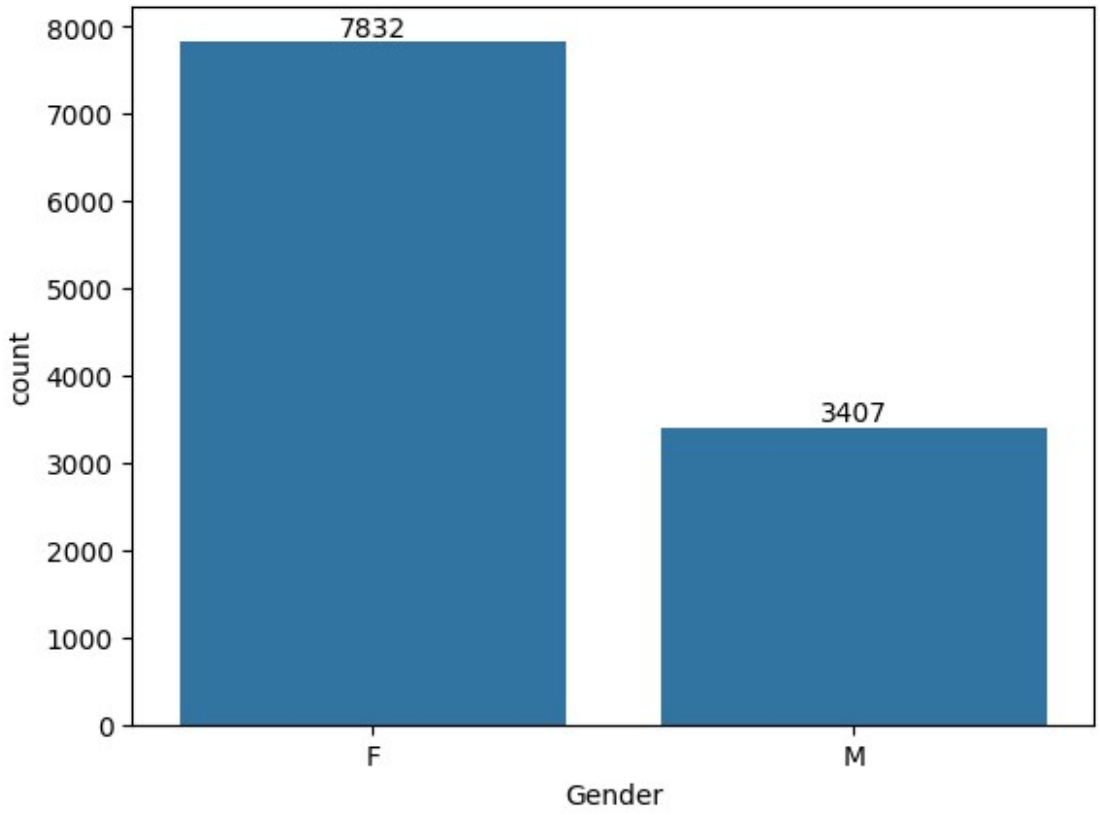

```
{"summary":"{\n  \"name\": \"df[['Age', 'Orders', 'Amount']]\",\n
\"rows\": 8,\n  \"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3960.7779927819724,\n        \"min\": 12.0,\n        \"max\":
11239.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
35.41035679330901,\n          33.0,\n          11239.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Orders\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
3972.7985251346995,\n        \"min\": 1.0,\n        \"max\": 11239.0,\
n        \"num_unique_values\": 7,\n        \"samples\": [\n
11239.0,\n          2.4896343091022333,\n          3.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Amount\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
7024.070687950828,\n        \"min\": 188.0,\n        \"max\":
23952.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
9453.610552540262,\n          8109.0,\n          11239.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


Exploratory Data Analysis

Gender

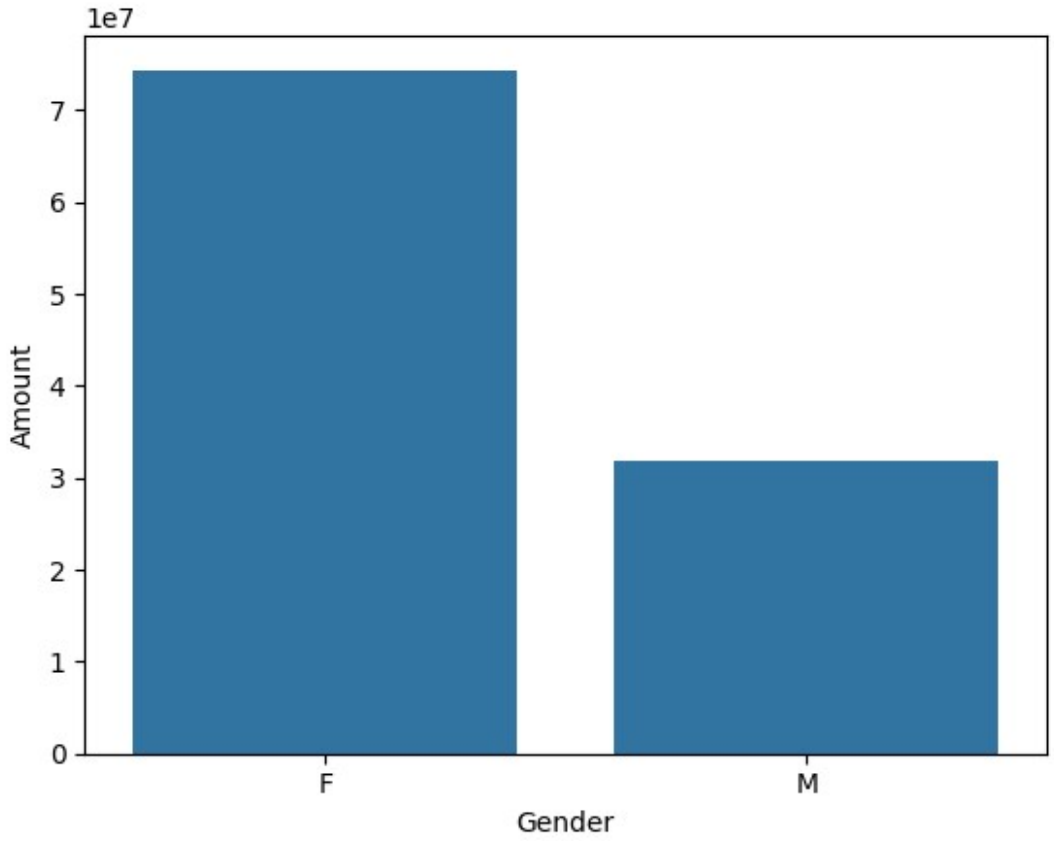
```
# plotting a bar chart for Gender and it's count

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

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




```
# plotting a bar chart for gender vs total 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)
```

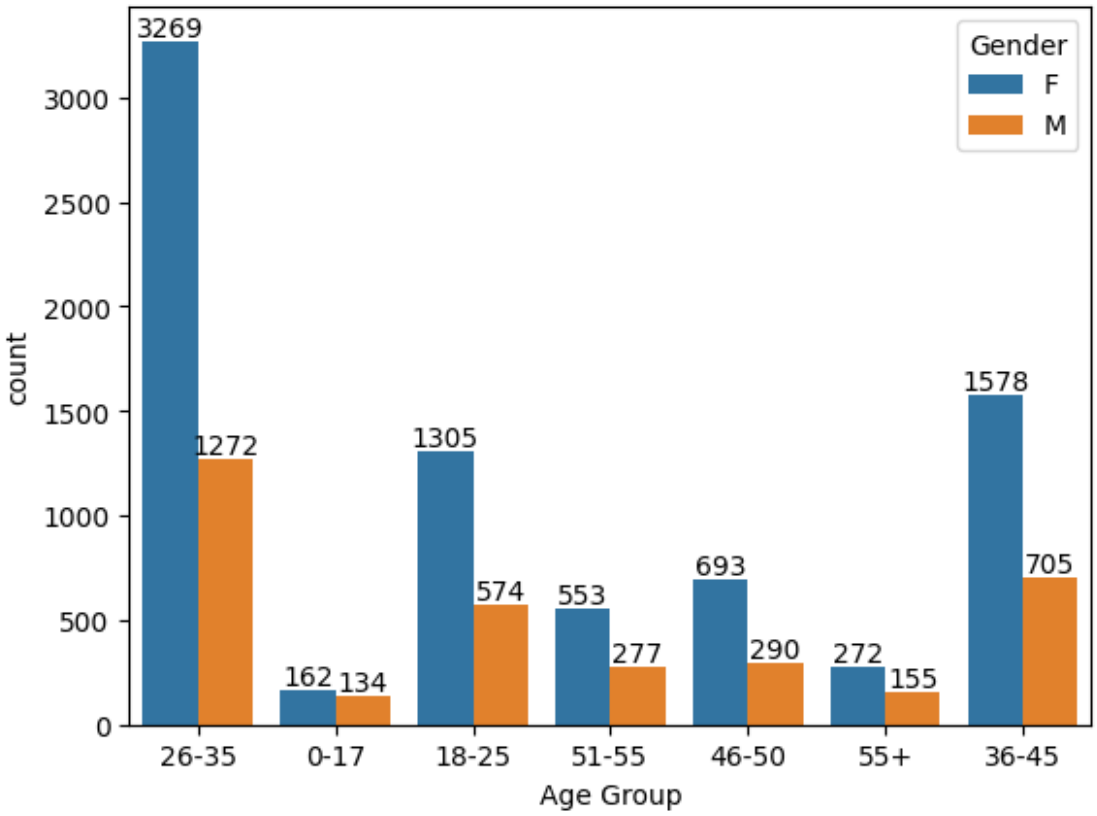
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



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

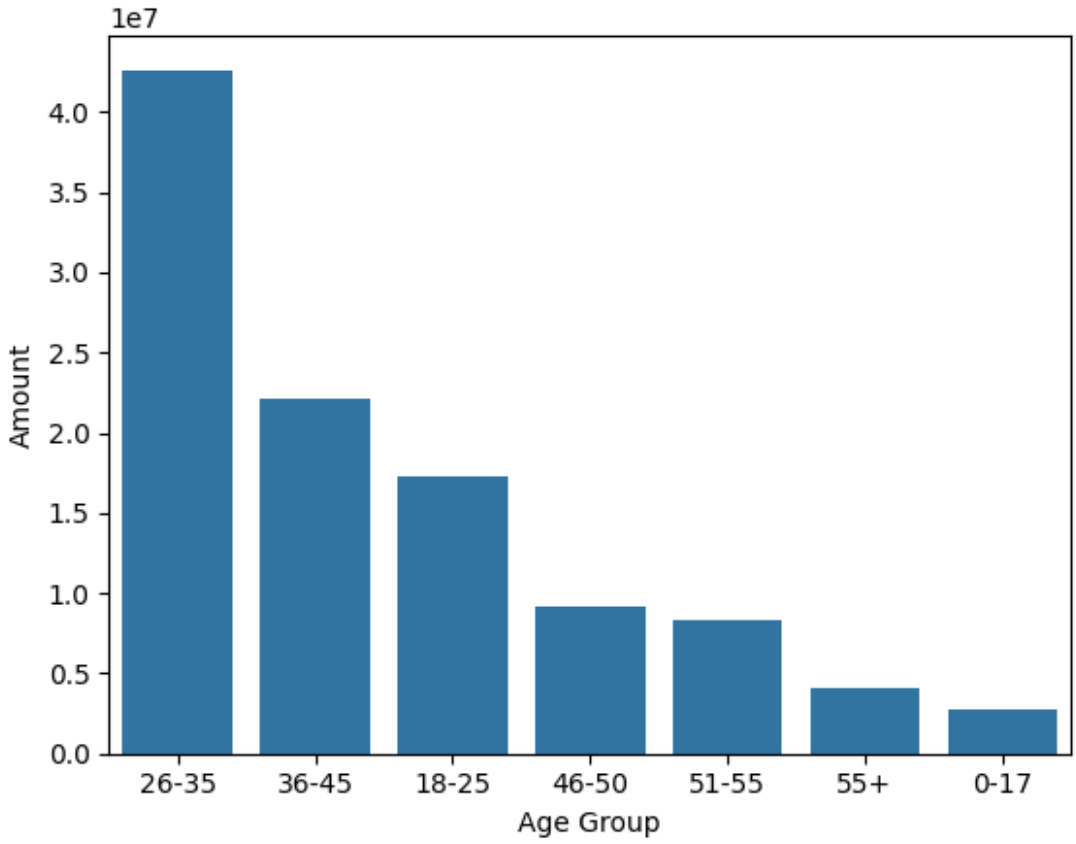
Age

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

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




```
# 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)
```

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

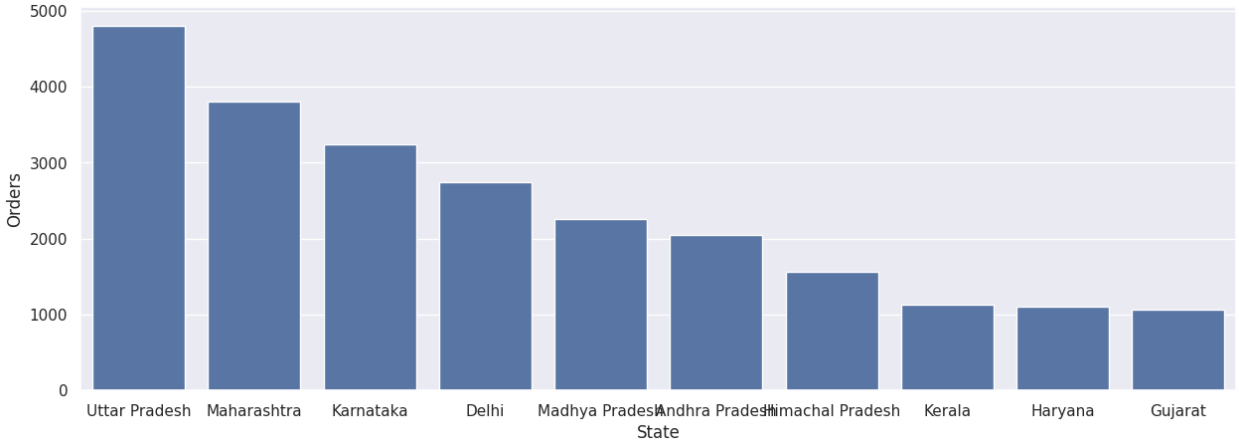


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

State

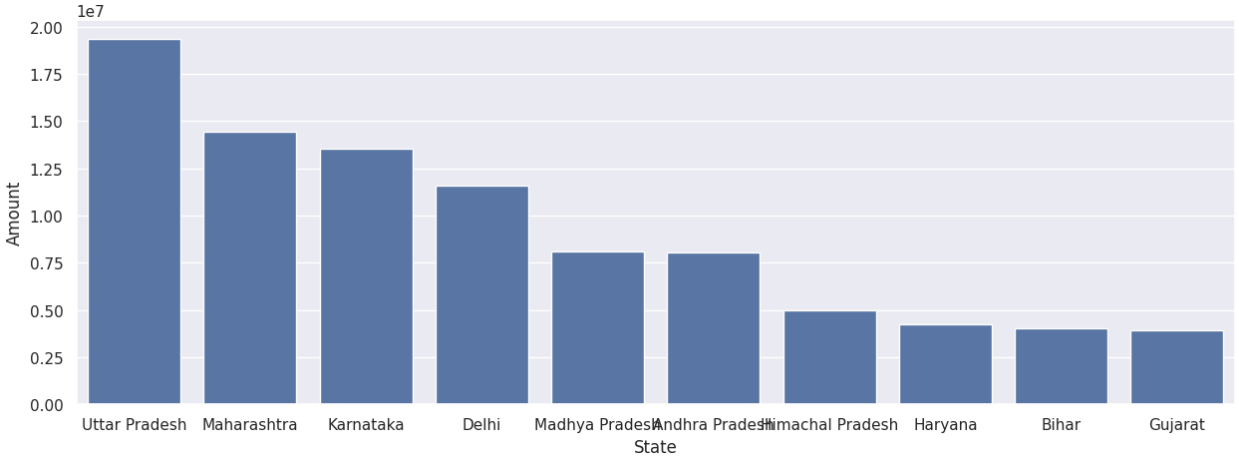
```
# total number of orders from top 10 states

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')
```

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




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

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')
```

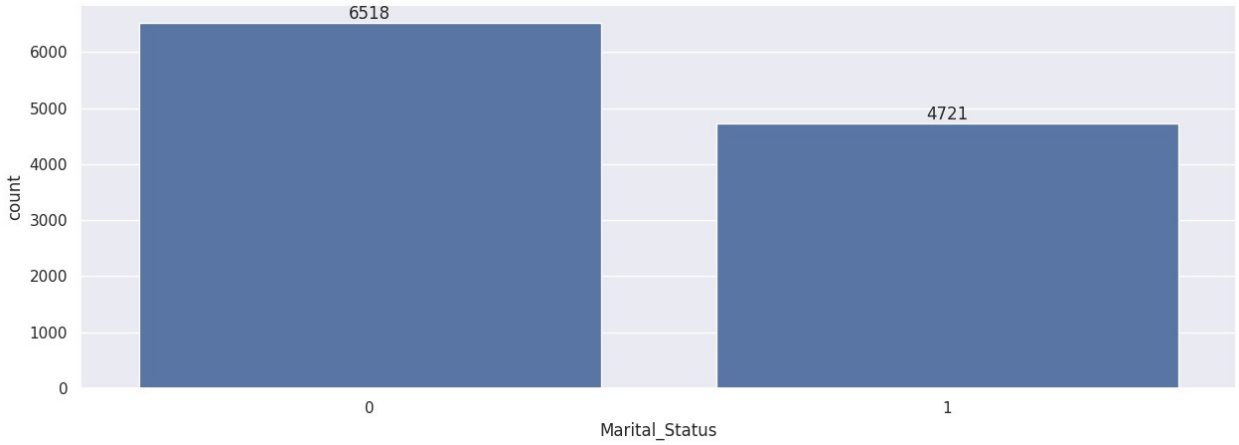
```
<Axes: xlabel='State', ylabel='Amount'>
```



From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

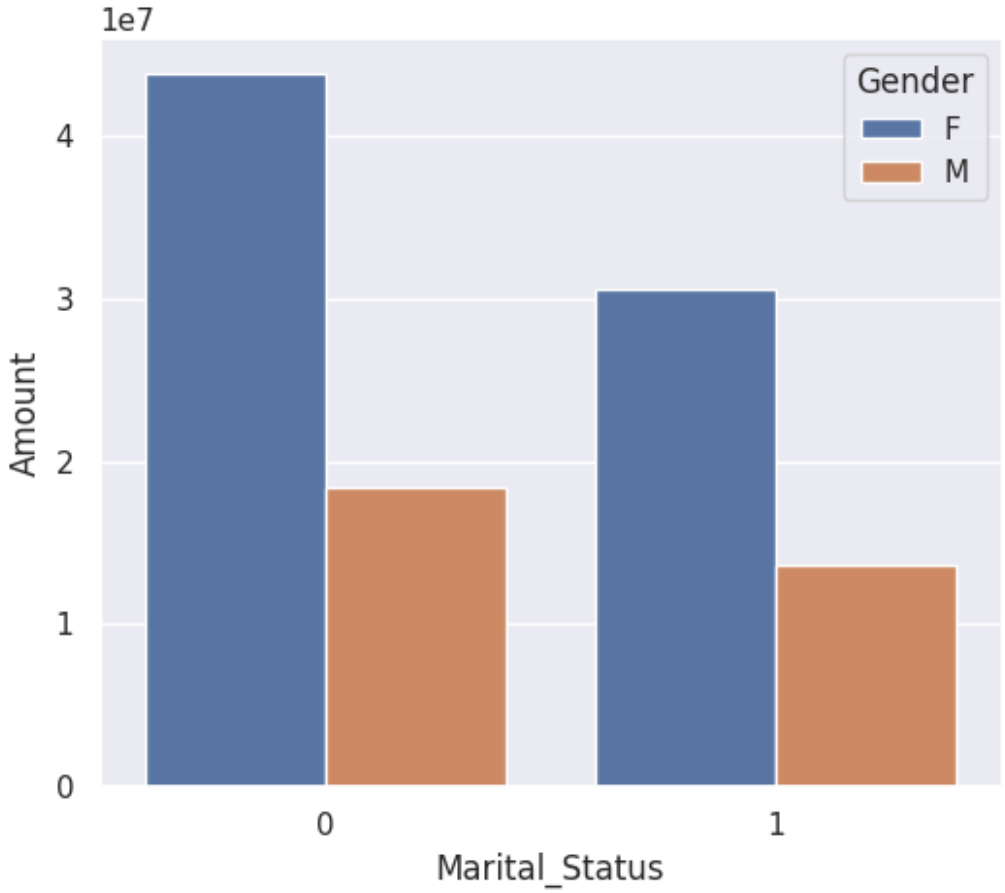
Marital Status

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

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




```
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')
```

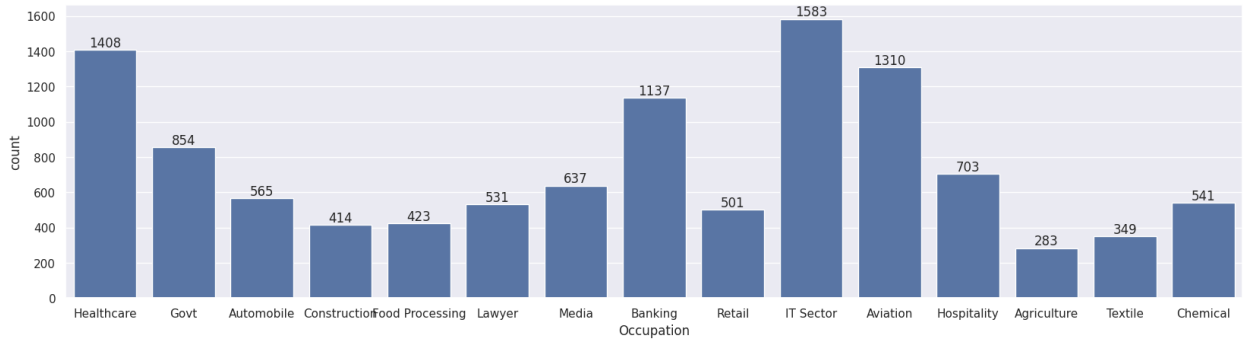
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



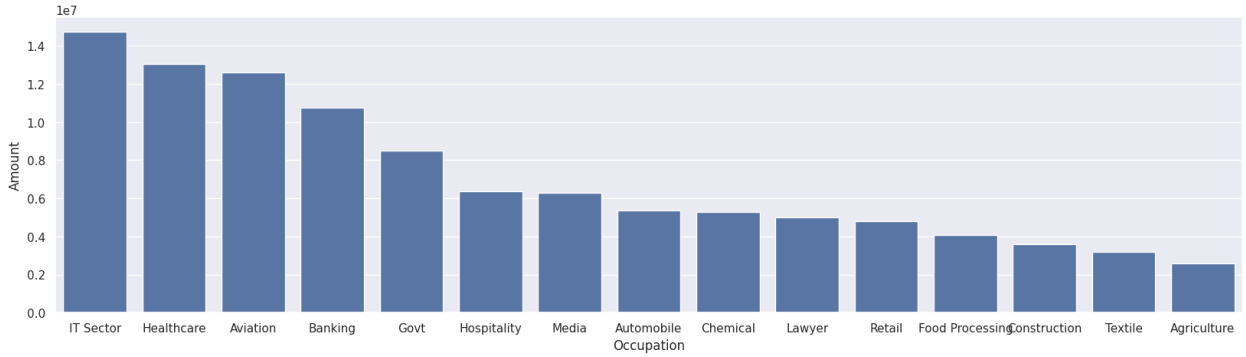
From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

Occupation

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

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




```
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')

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

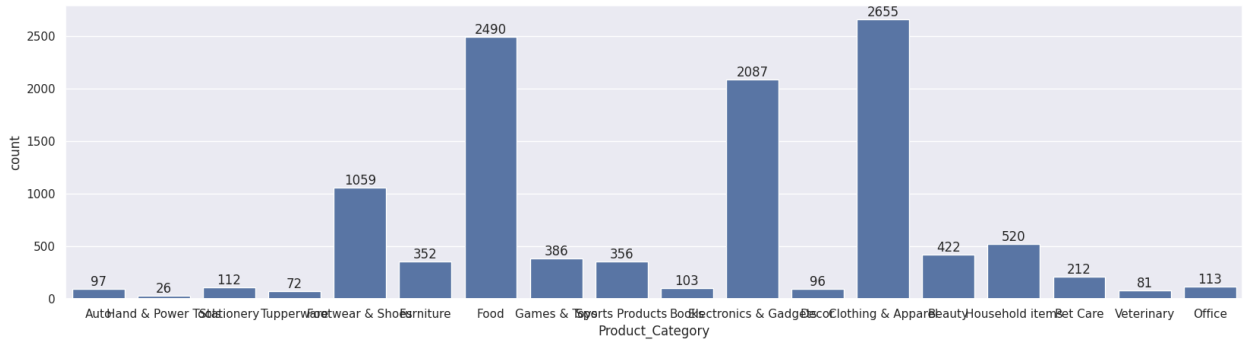


From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

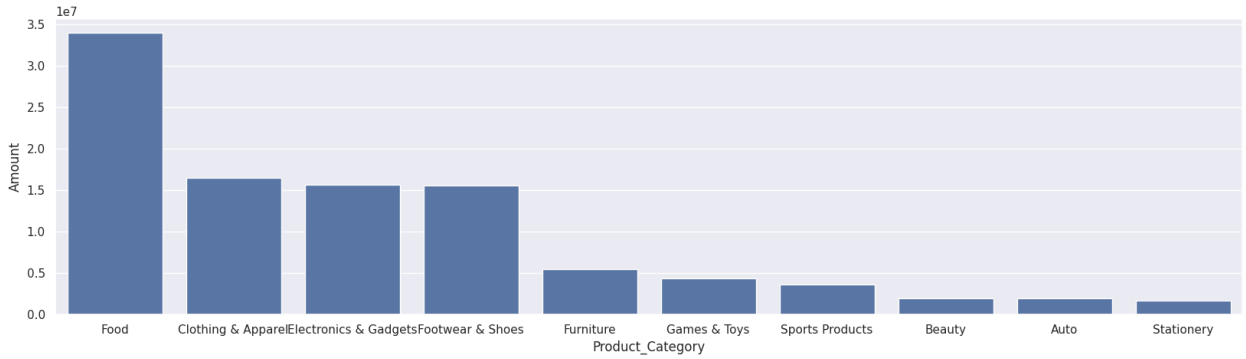
Product Category

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

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

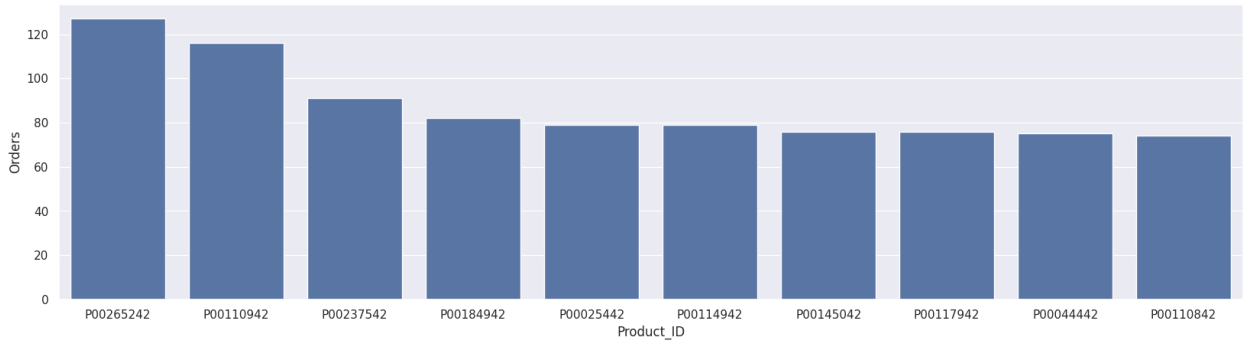



```
sales_state = df.groupby(['Product_Category'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

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

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



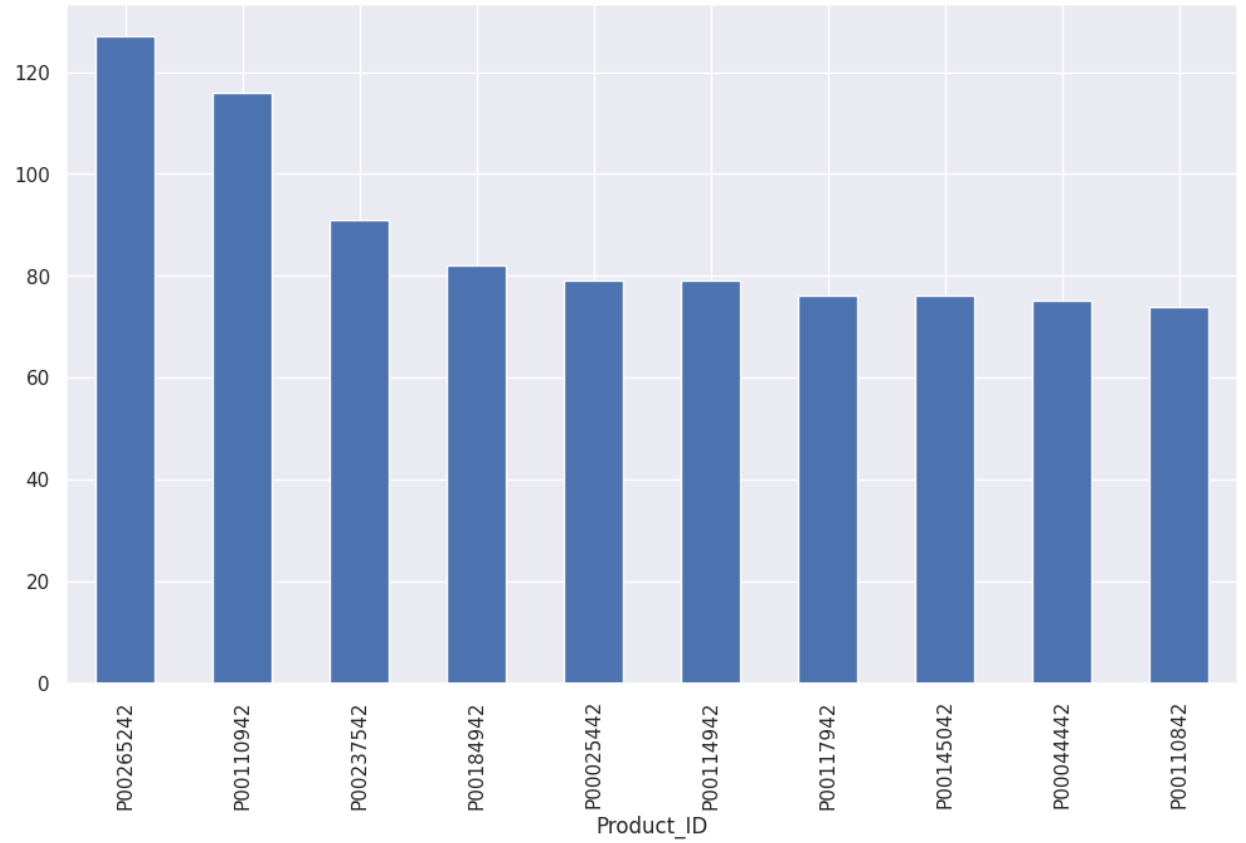
From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

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

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

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




```
# 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(ascending=False).plot(kind='
bar')
```

```
<Axes: 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

complete project on YouTube: https://www.youtube.com/@RishabhMishraOfficial

complete project on GitHub: https://github.com/rishabhnmishra/Python_Diwali_Sales_Analysis

Thank you!