Comprehensive Analysis of Diwali Sales Trends

Introduction:

Diwali, India's festival of lights, is a peak shopping season marked by significant consumer spending. This project analyses Diwali sales data to uncover trends in consumer behaviour, highlighting key demographics, popular products, and spending patterns. By exploring data across factors like age, gender, and occupation, the study provides actionable insights for businesses to optimize their festive strategies.

 # 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('/content/Diwali Sales Data.csv', encoding= 'unicode_escape')
```

df.shape

```
(11251, 15)
```

df.head()

	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

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()
```

	0
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'})

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns

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

df.describe()

	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

 # use describe() for specific columns

df[['Age', 'Orders', 'Amount']].describe()

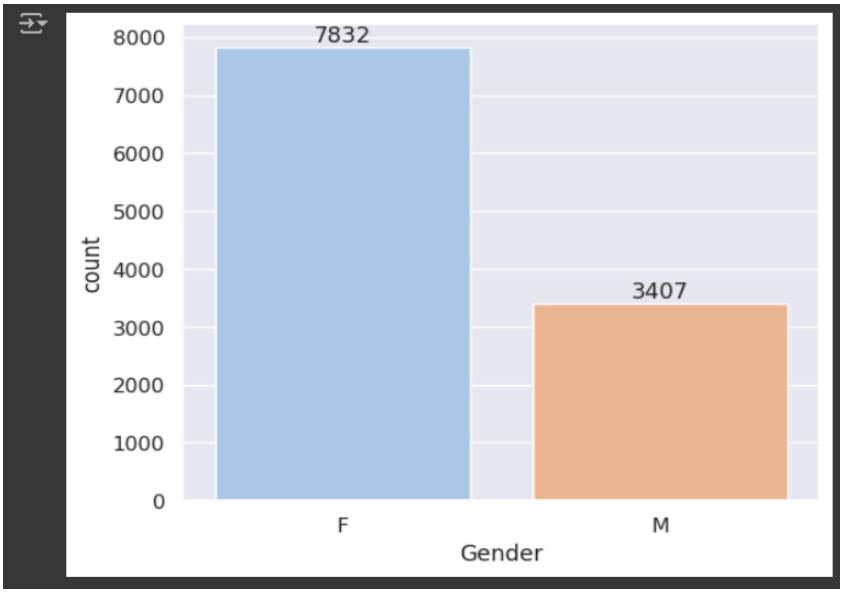
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis

GENDER

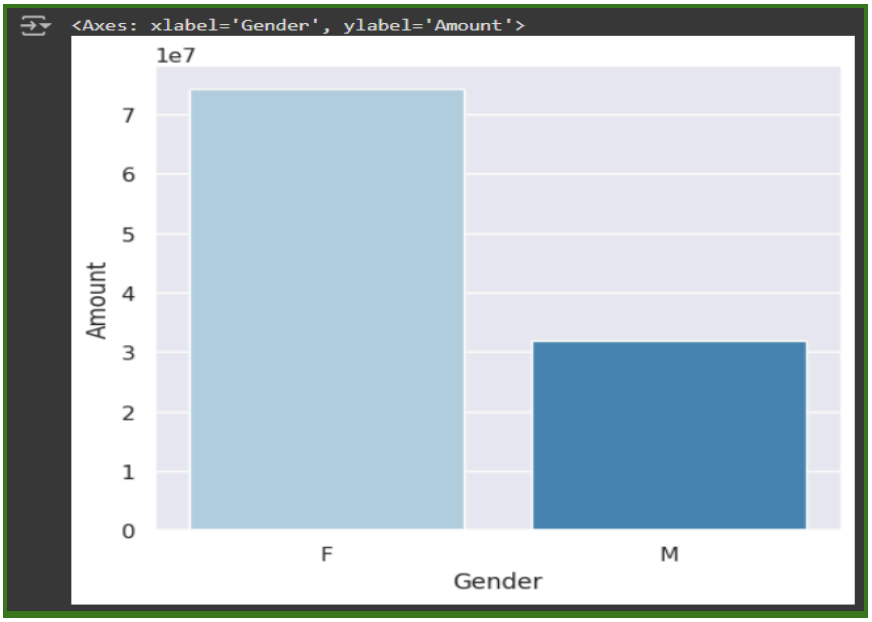
 # plotting a bar chart for Gender and it's count

```
ax = sns.countplot(x = 'Gender',data = df, palette='pastel', hue='Gender')
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, palette='Blues', hue='Gender')
```

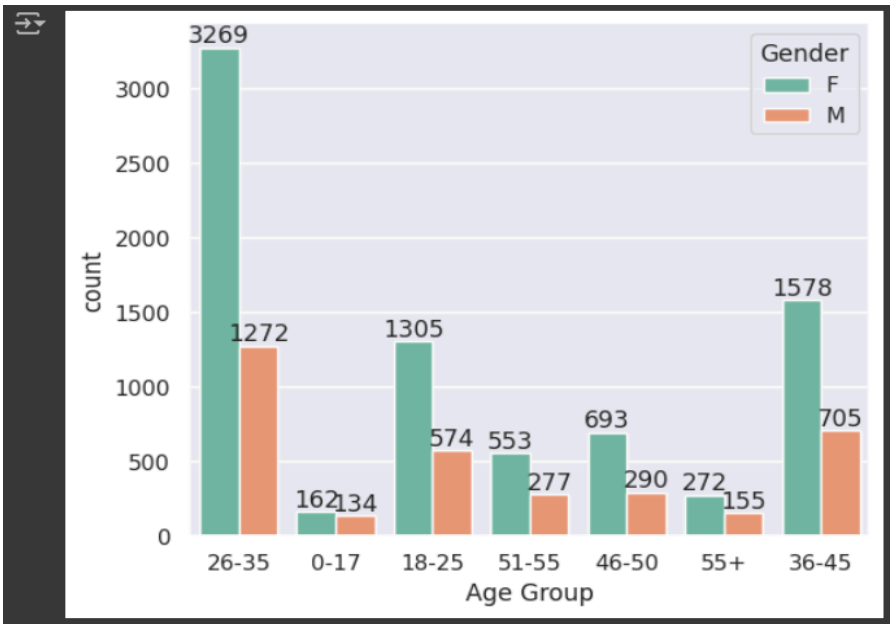


From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

AGE

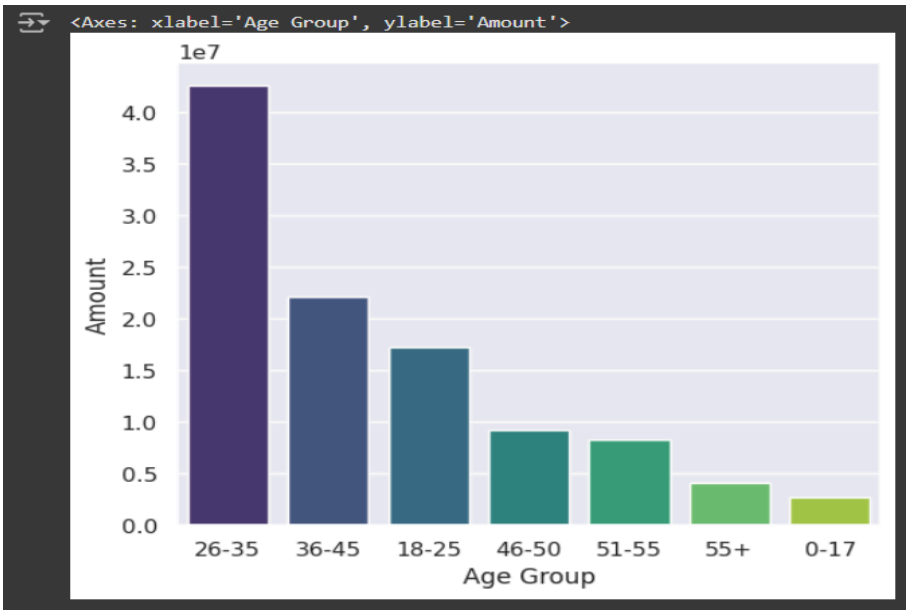
 # plotting a bar chart for Age Group and Gender

```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender', palette='Set2')
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, palette='viridis', hue='Age
Group')
```

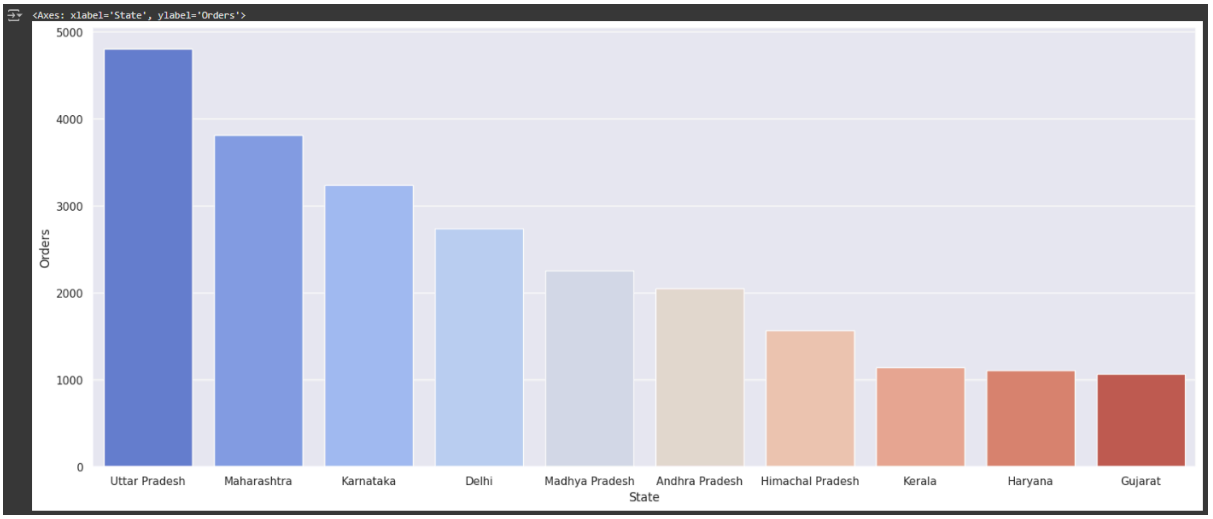


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

STATE

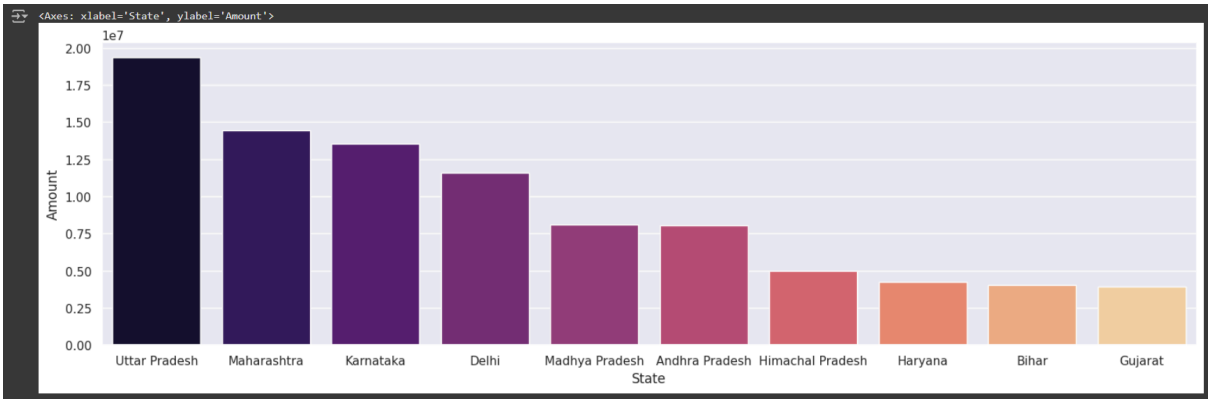
 # total number of orders from top 10 states

```
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,8)})
sns.barplot(data = sales_state, x = 'State', y = 'Orders', palette='coolwarm', hue='State')
```



 # 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':(18,5)})
sns.barplot(data = sales_state, x = 'State', y = 'Amount', palette='magma', hue='State')
```

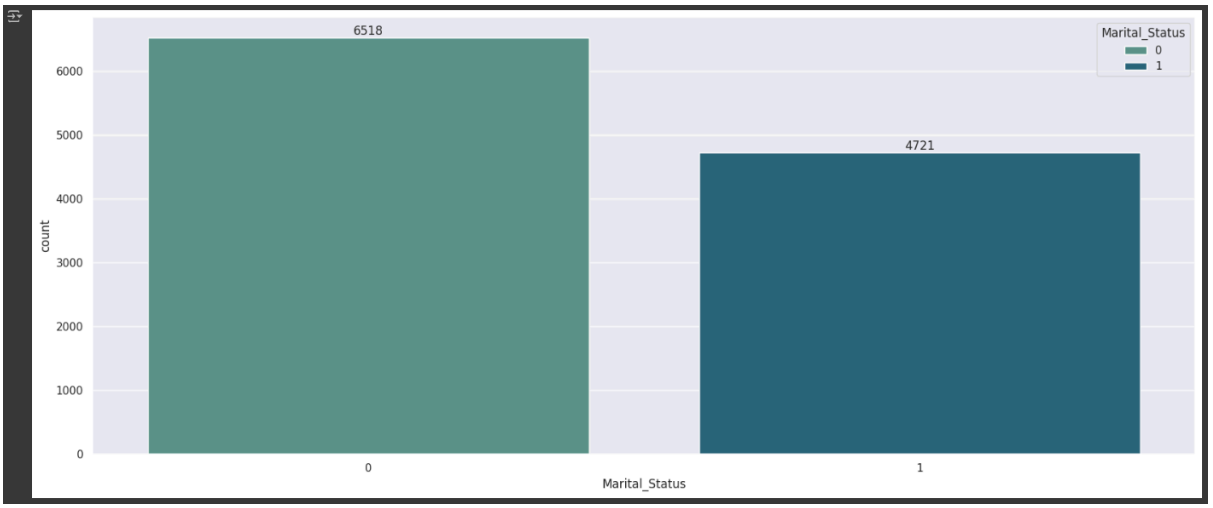


From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

MARITAL STATUS

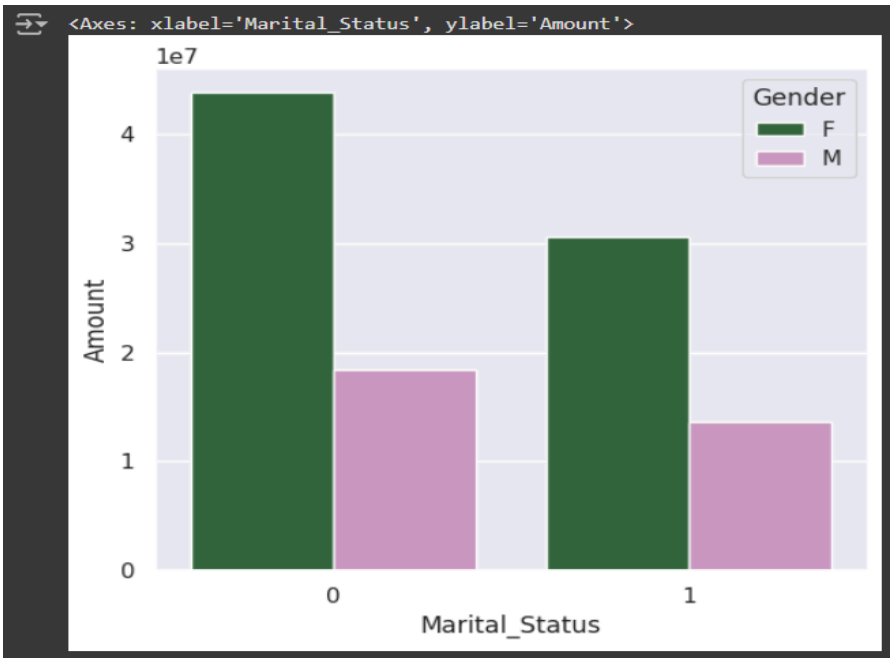
 # Marital Status count plot

```
ax = sns.countplot(data = df, x = 'Marital_Status', palette='crest', hue='Marital_Status')
for bars in ax.containers:
    ax.bar_label(bars)
```



 # Marital Status vs Amount by Gender

```
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',
palette='cubehelix')
```

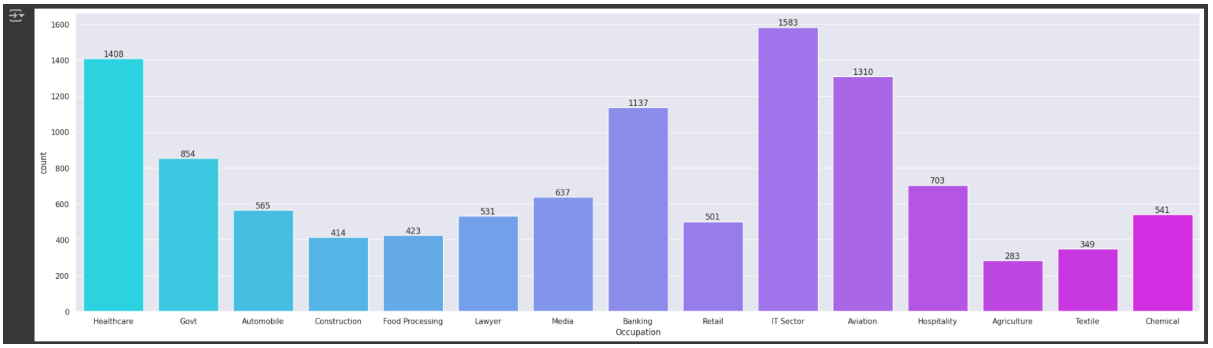


From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

OCCUPATION

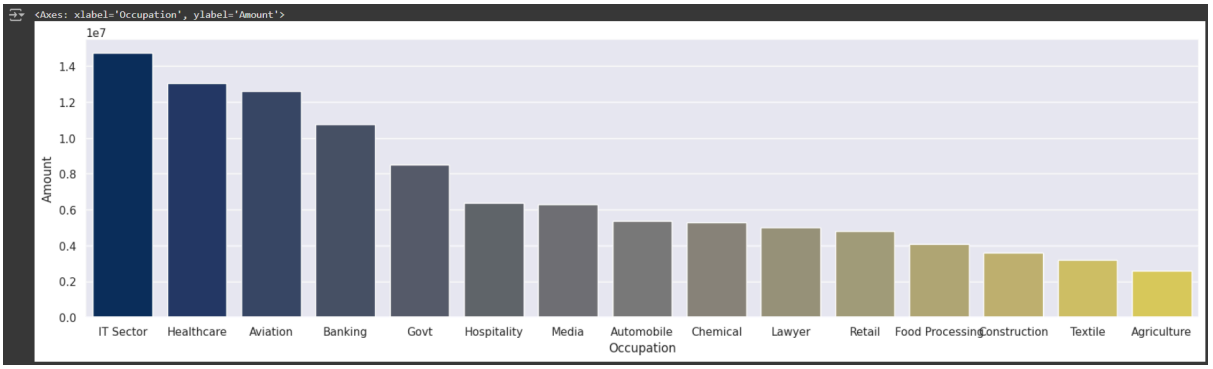
 # Occupation count plot

```
ax = sns.countplot(data = df, x = 'Occupation', palette='cool', hue='Occupation')
sns.set(rc={'figure.figsize':(22,8)})
for bars in ax.containers:
    ax.bar_label(bars)
```



 # Occupation vs Amount

```
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', palette='cividis',
hue='Occupation')
```

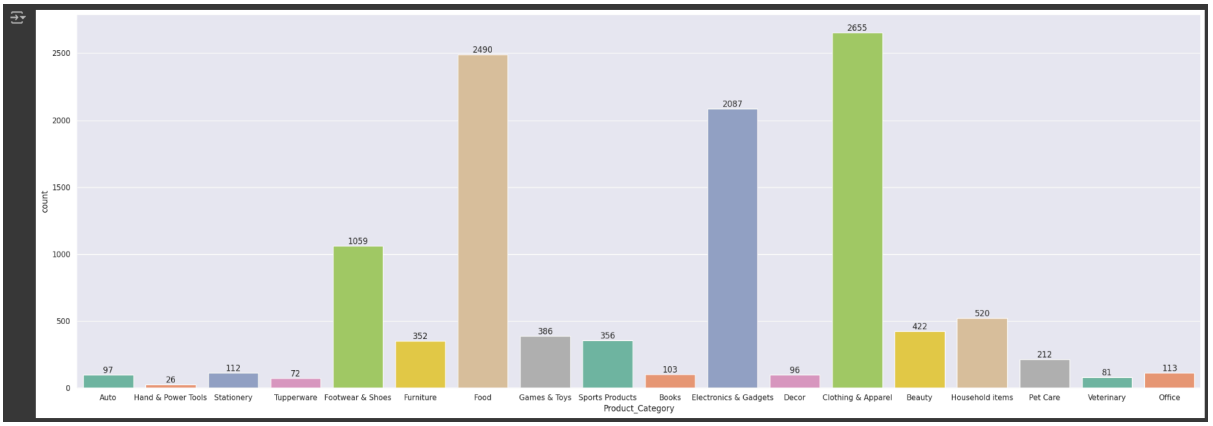


From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

PRODUCT CATEGORY

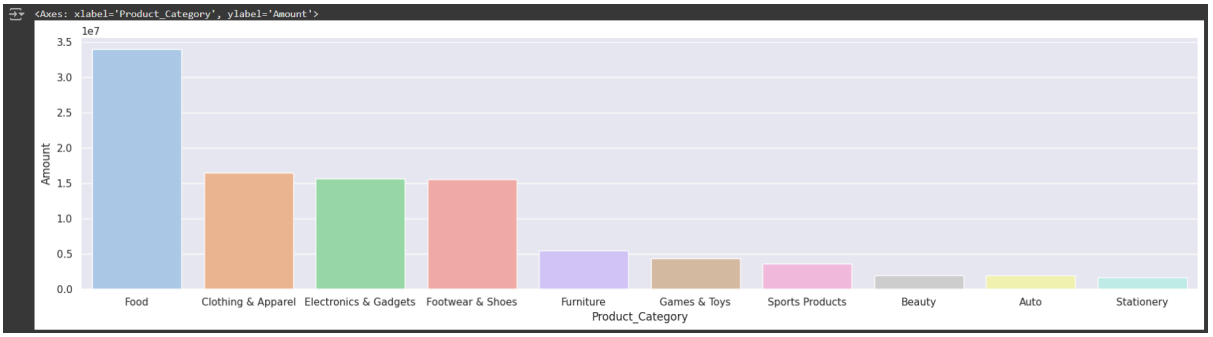
 # Product Category count plot

```
sns.set(rc={'figure.figsize':(30,10)})
ax = sns.countplot(data = df, x = 'Product_Category', palette='Set2', hue='Product_Category')
for bars in ax.containers:
    ax.bar_label(bars)
```



 # Product Category vs Amount

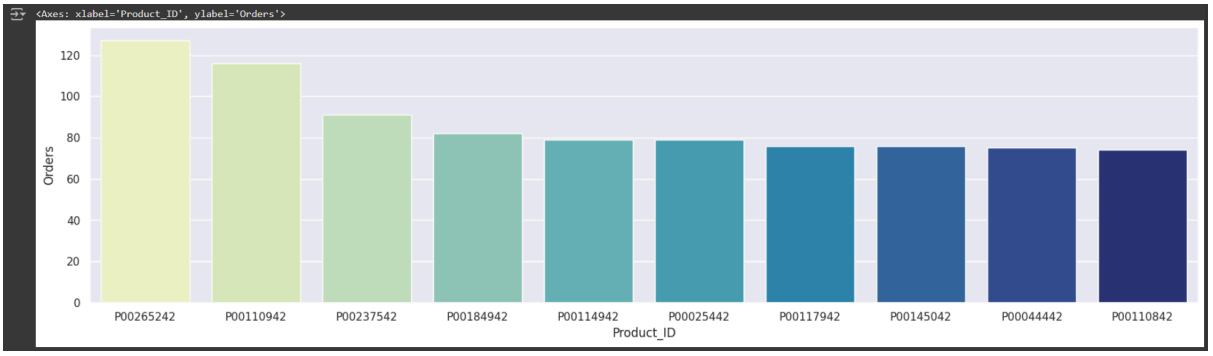
```
sales_state = df.groupby(['Product_Category'],
as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize':(22,5)})
sns.barplot(data = sales_state, x = 'Product_Category', y = 'Amount', palette='pastel',
hue='Product_Category')
```



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

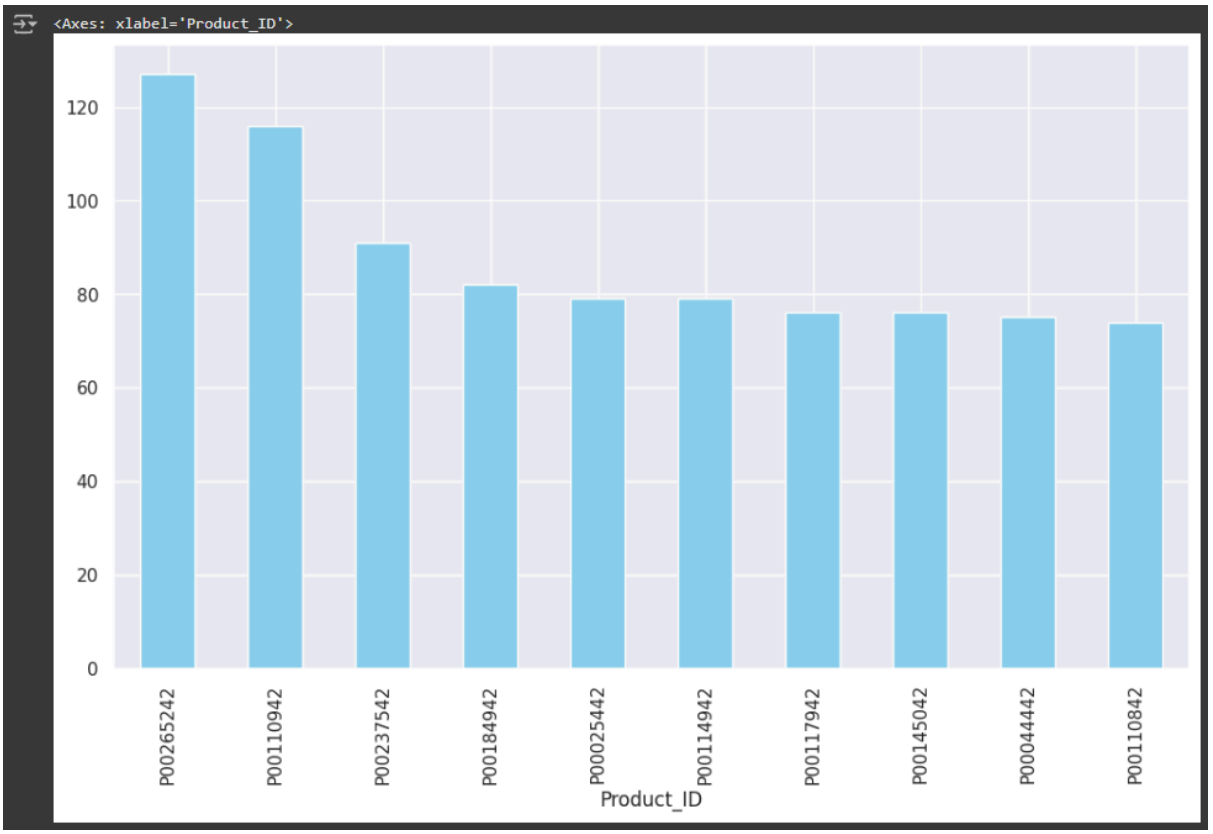
 # Top 10 most sold products

```
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', palette='YlGnBu',
hue='Product_ID')
```



 # 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', color='skyblue')
```



Conclusion:

The analysis reveals that married women aged 26-35 years, especially from Uttar Pradesh, Maharashtra, and Karnataka, are prominent buyers, favoring food, clothing, and electronics. Professionals in IT, healthcare, and aviation contribute significantly to sales. These insights can help businesses refine their marketing strategies and better target customers during festive seasons.