Case Study

Title of Case Study

Analyzing Consumer Trends During Diwali Sales

Subject

DS8001 - Data Analytics and Visualization

Submitted by

Mitul Rathod (202304104610017)

MCA 3rd Semester

Index

1. Introduction

1.1 Brief description of the problem statement and objectives

During the Diwali season, one of India's most significant shopping events, consumers engage in extensive purchasing across various product categories, including electronics, clothing, and home goods. Retailers experience a remarkable increase in sales, making it essential for businesses to comprehend consumer behaviors, buying patterns, and regional preferences during this peak period. This understanding is vital for tailoring marketing strategies, managing inventory effectively, and enhancing customer experiences.

1.2 Overview of the dataset used

The dataset contains 11,251 records of Diwali sales, with information on customer demographics like age, gender, and occupation, along with details on product categories, orders, and purchase amounts. It also includes data on regional sales patterns across different states and zones in India.

2. Description of data collection methods

2.1 Data cleaning and preprocessing steps

- ✓ Remove Unnecessary Columns
- ✓ Handle Missing Values
- ✓ Check for Duplicates
- ✓ Verify Data Types
- ✓ Normalize Categorical Data

```
[40]  # import python libraries

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

```
# Use the raw link and specify the encoding
url = 'https://raw.githubusercontent.com/mitul-rathod02/Analyzing-Consumer-Trends-During-Diwali-Sales/main/DiwaliSalesData.csv'
df = pd.read_csv(url, encoding= 'unicode_escape')

# Display the first few rows of the dataset
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

```
[15] df.shape
```

```
(11251, 15)
```

```
[16] 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
```

```
[17] #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

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

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

[21] 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')
```

3. Descriptive statistics and summary of key findings

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

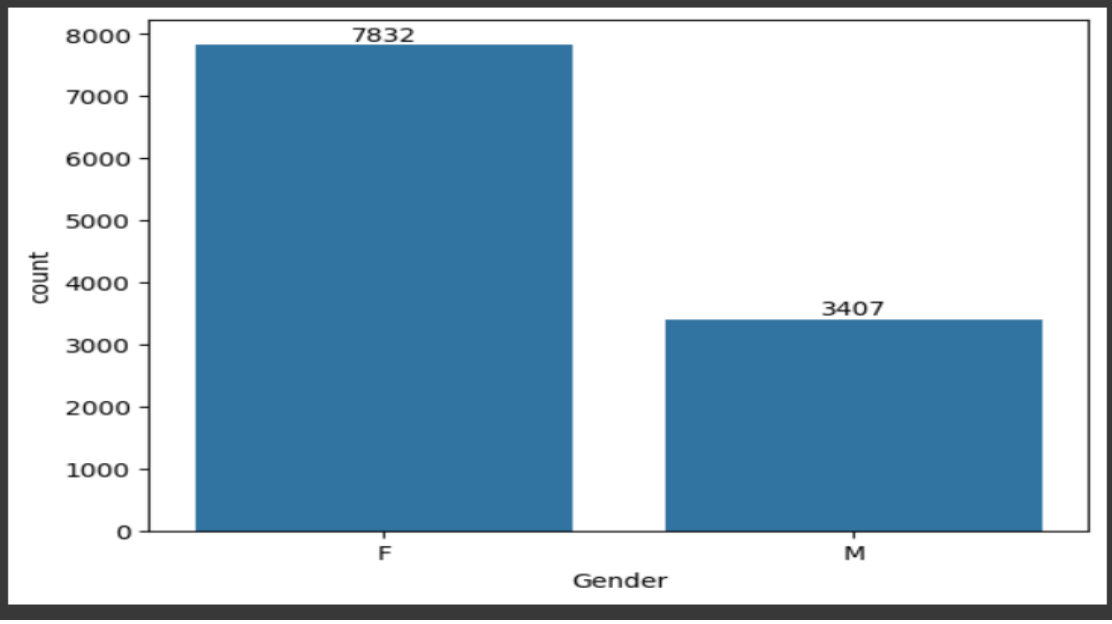
	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000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

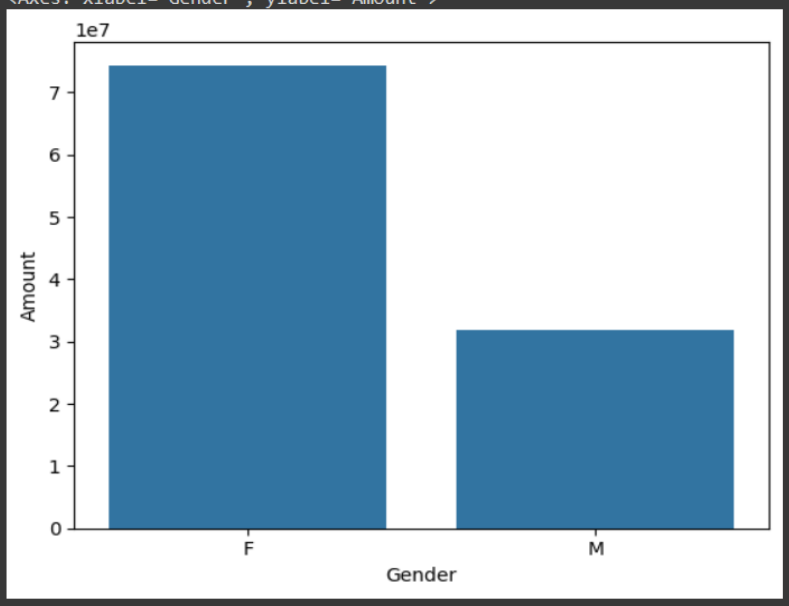
Gender

```
[26] # 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)
```

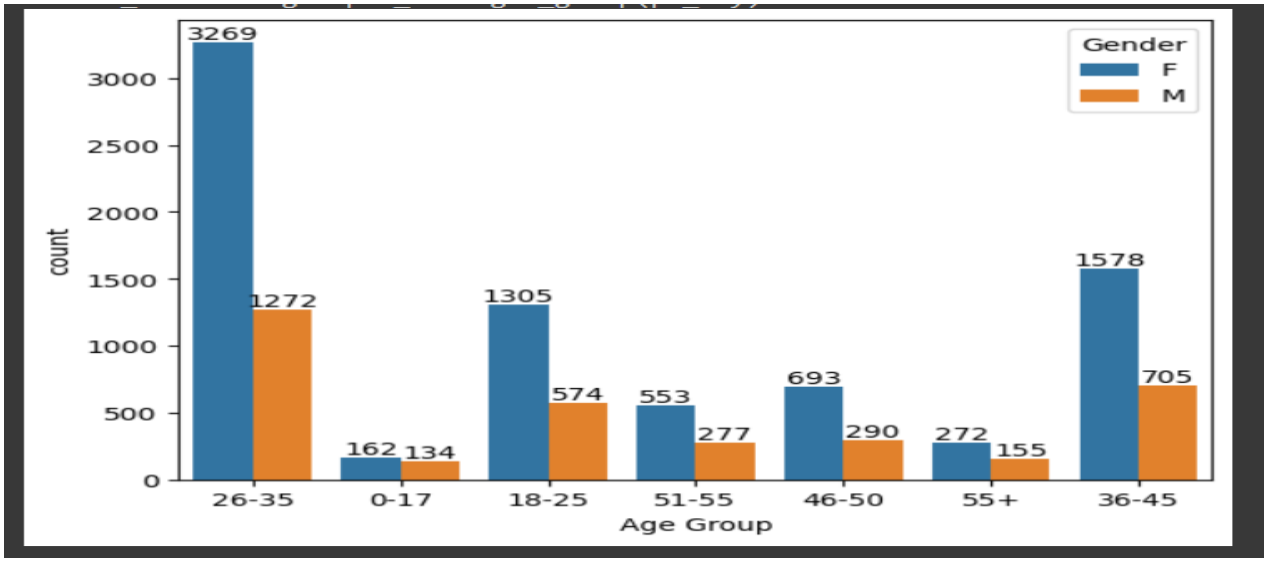
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



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

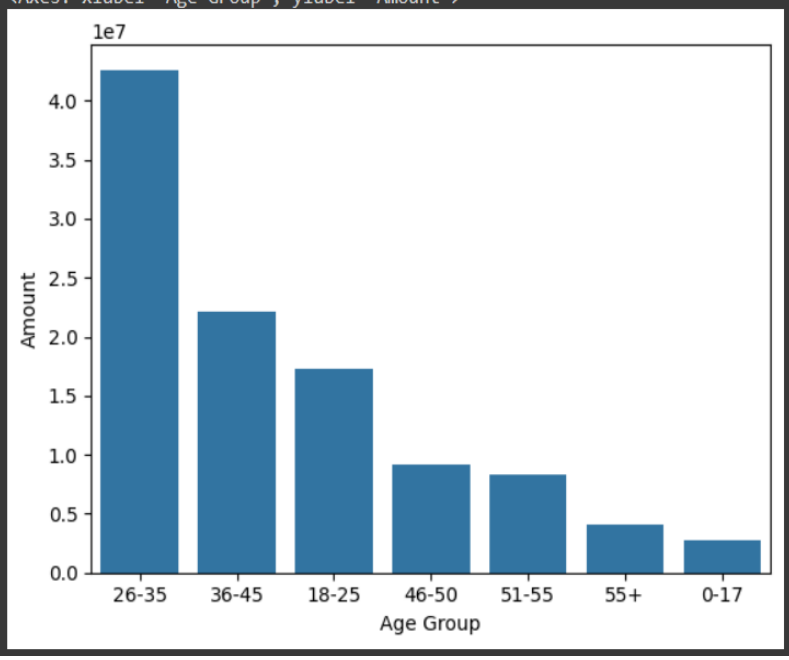
Age

```
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)
```

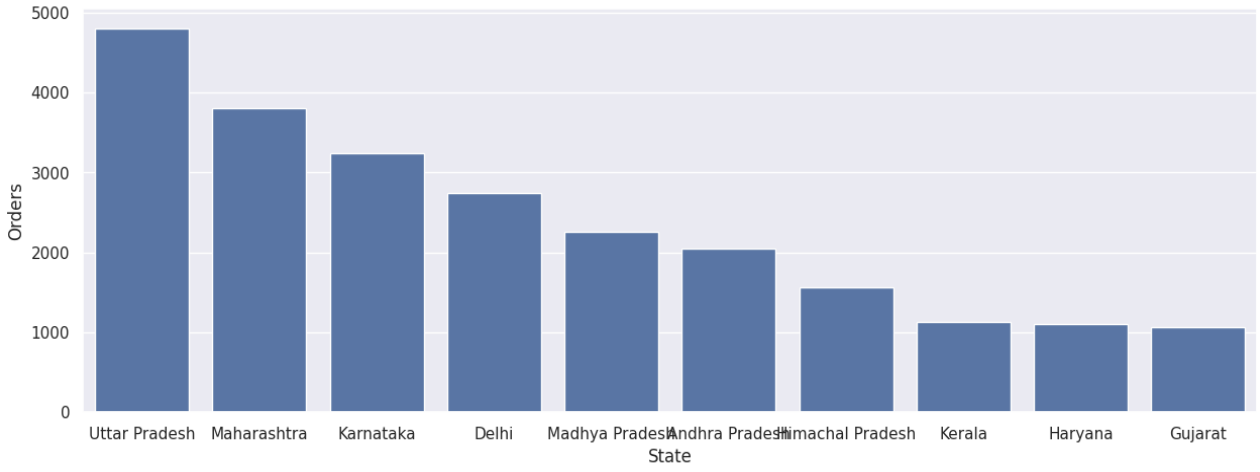
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



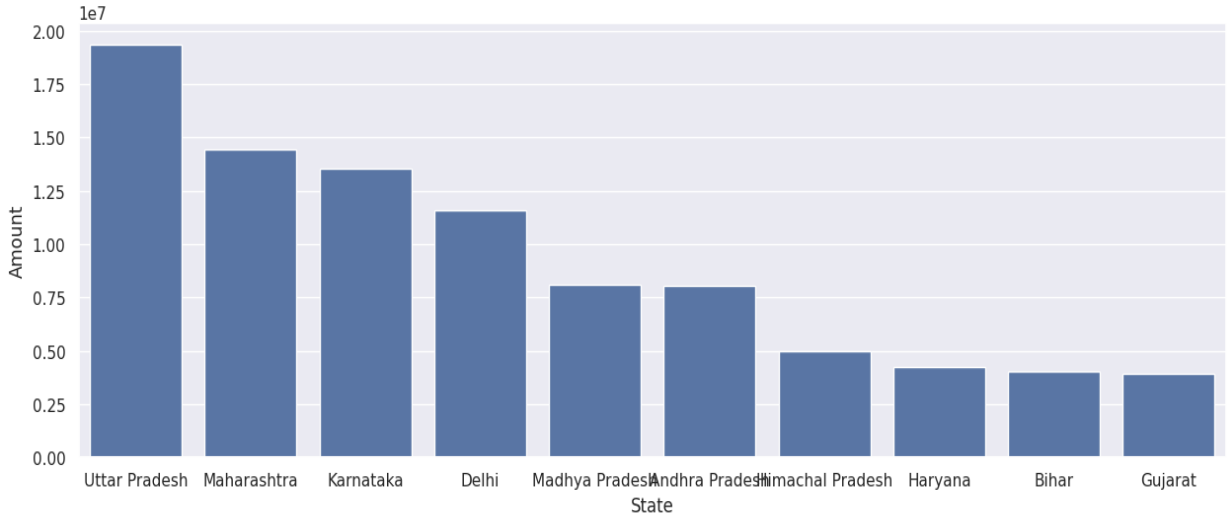
- *From the above graphs we can see that most of the buyers are of age group between 26-35 yrs female.*

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')
```



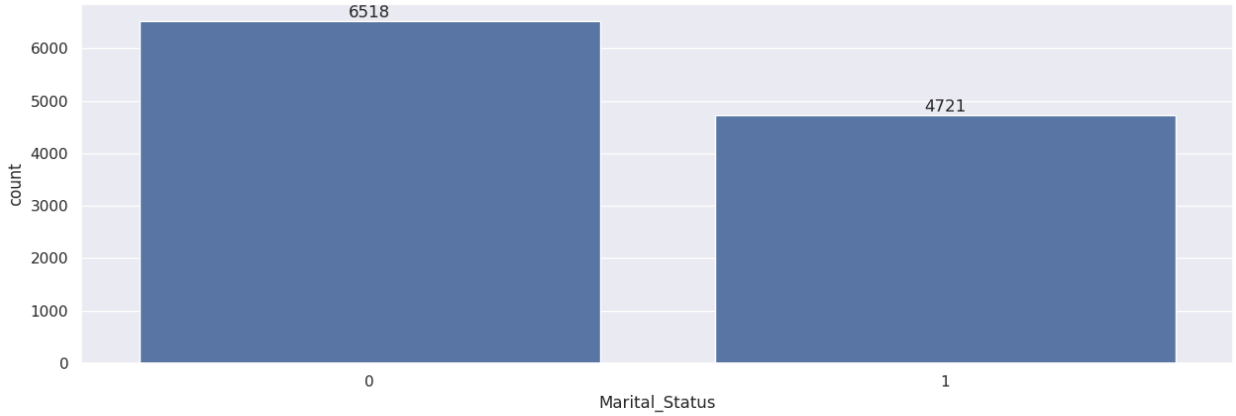

```
# total amount/sales from top 10 states

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')
```



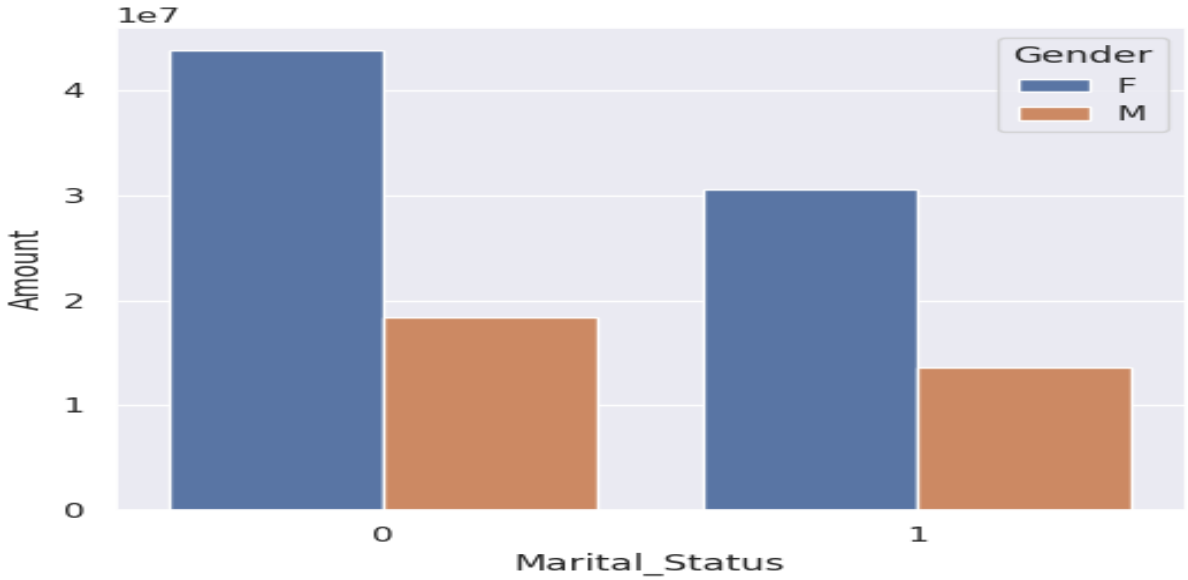
- *From the above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively.*

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)
```



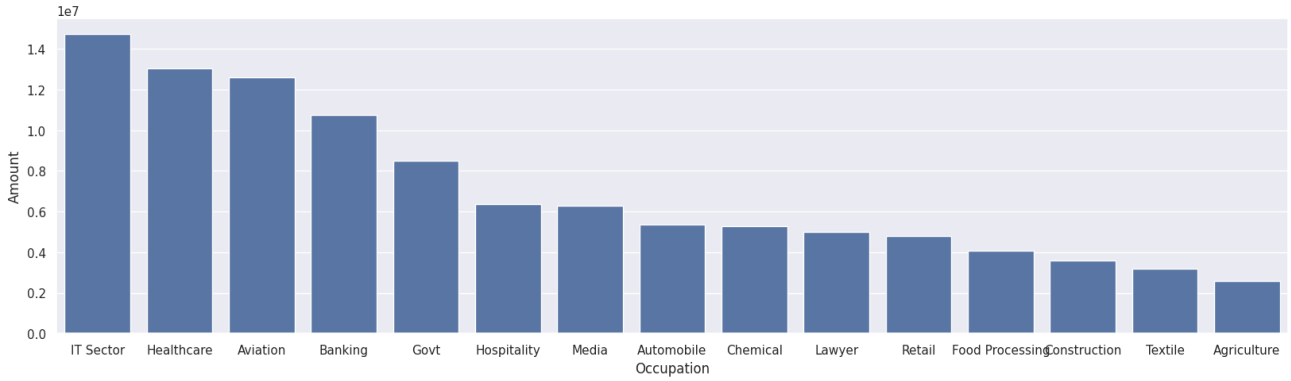

```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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



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

Occupation

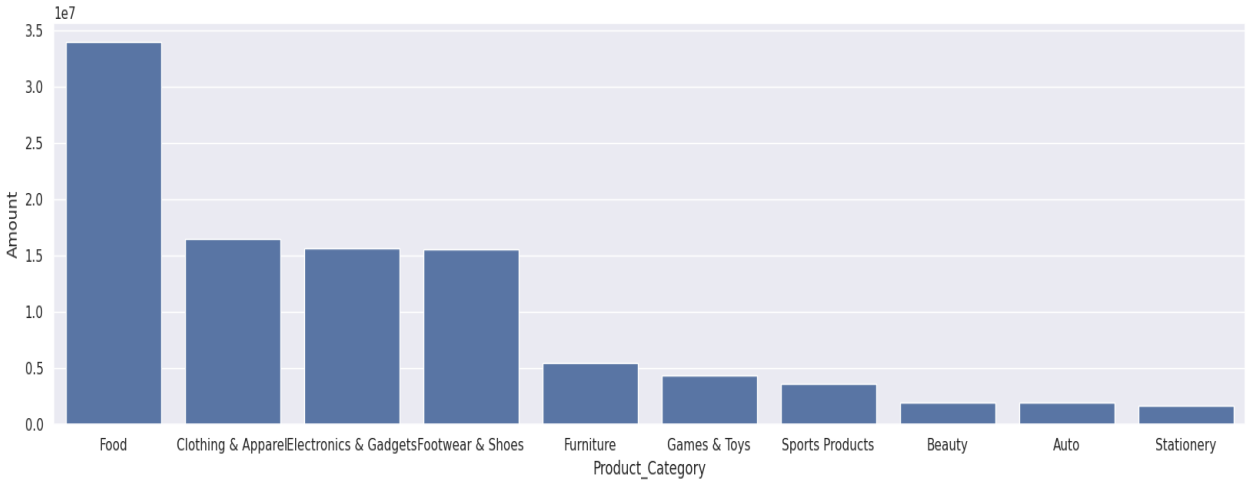
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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



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

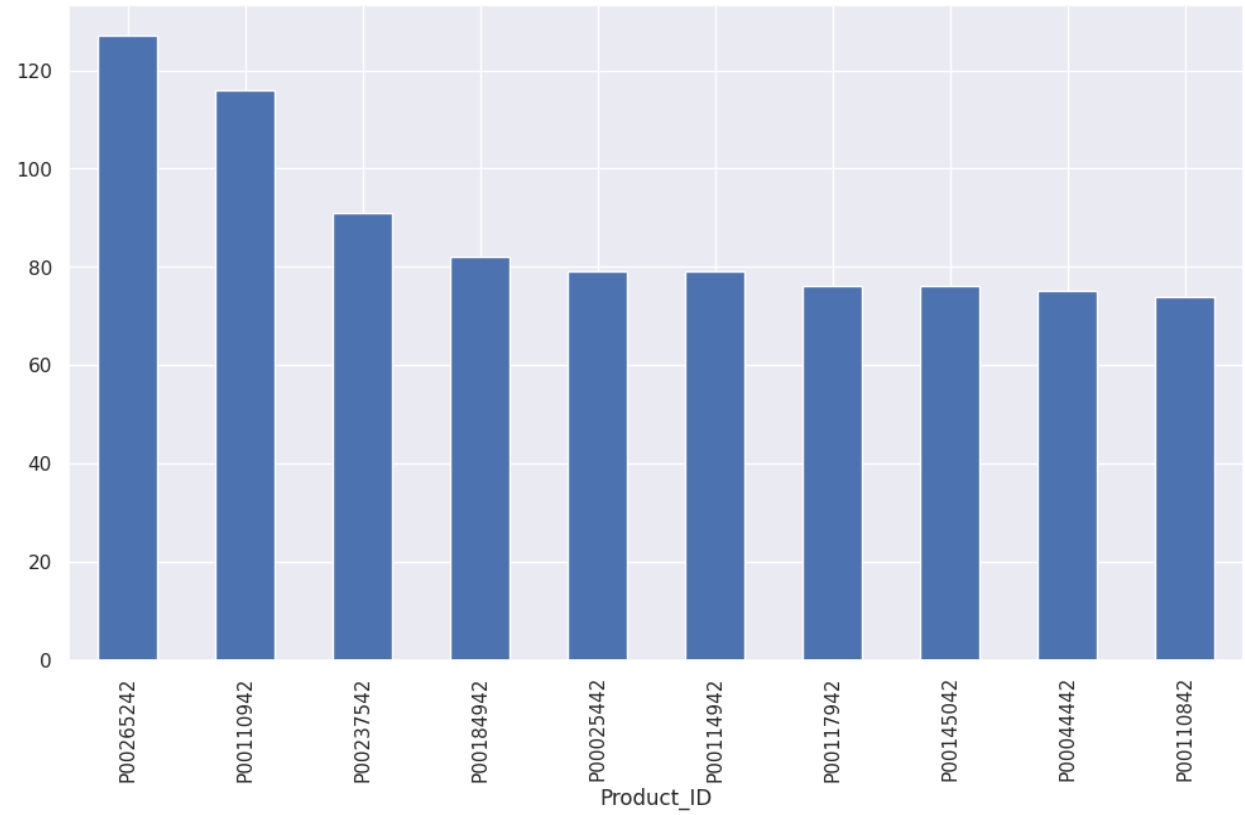
Product Category

```
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')
```



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

```
# 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')
```



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

4. Results and Discussion

4.1 Summary of the findings and insights

The analysis highlights that most Diwali shoppers are females, particularly those aged 26-35. States like Uttar Pradesh, Maharashtra, and Karnataka account for the highest sales, with married women in the IT, healthcare, and aviation sectors showing significant purchasing power, especially for food, clothing, and electronics.

4.2 Interpretation of visualizations

The graphs show clear patterns in shopping habits. They indicate that women buy more than men, and certain age groups and states have higher sales. This helps identify who the main customers are.

4.3 Discussion on the implications of the results

These findings suggest that businesses should target female customers, especially those in their late twenties and early thirties. Advertising should focus on what these women like to buy, especially in the key states mentioned.

4.4 Conclusion and Recommendations

In conclusion, the analysis of Diwali sales data shows that women, especially those aged 26-35, are the main shoppers. They spend the most money on food, clothing, and electronics, particularly in states like Uttar Pradesh, Maharashtra, and Karnataka. Businesses should focus on this group when planning their sales strategies.

It is recommended that companies create advertisements that appeal to women and highlight products they want to buy. Additionally, targeting marketing efforts in the top-performing states can help boost sales during the Diwali season. By understanding customer preferences, businesses can better meet their needs and increase their sales.

5. References

- *https://www.kaggle.com/datasets/saadharoon27/diwali-sales-dataset*
- *https://www.geeksforgeeks.org/data-analyst-projects/*
- *https://www.grammarly.com/plagiarism-checker*