Diwali Sales Analysis

(Python+Sql+Power BI Project)

1. Introduction

This project focuses on analyzing sales data using Python to understand customer behavior and purchasing patterns. The objective of the project was to clean and analyze raw sales data, extract meaningful insights, and support business decision-making related to customer targeting, sales improvement, and inventory planning.

2. Tools and Technologies Used

Programming Language: Python

Libraries:

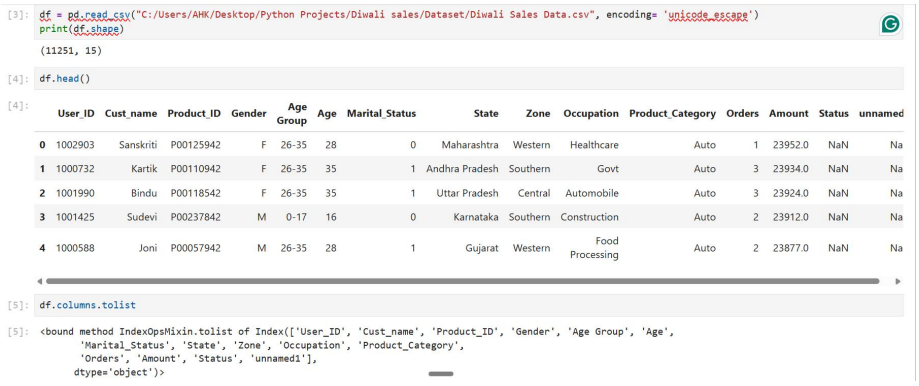
- Pandas – data cleaning and manipulation
- Matplotlib & Seaborn – data visualization

Environment: Jupyter Notebook

3. Data Cleaning and Preparation

The dataset was first cleaned to ensure accuracy and consistency. This included handling missing values, correcting data types, removing unnecessary columns, and standardizing categorical variables. Pandas was used extensively to manipulate and prepare the data for analysis.

Figure 1: Python code for data cleaning and preprocessing.



```
[3]: df = pd.read_csv("C:/Users/AHK/Desktop/Python Projects/Diwali sales/Dataset/Diwali Sales Data.csv", encoding= 'unicode_escape')
     print(df.shape)

     (11251, 15)

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

[4]:

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

```
[5]: df.columns.tolist

[5]: <bound method IndexOpsMixin.tolist of Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
           'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
           'Orders', 'Amount', 'Status', 'unnamed1'],
          dtype='object')>
```

[8]:
```
#droping the blank columns/unrelated columns
df.drop(['Status','unnamed1'], axis=1, inplace = True)
```

[9]:
```
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
```

[11]:
```
#to check the null values
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             12
dtype: int64
```

[12]:
```
df.shape
```

[12]:
```
(11251, 13)
```

[13]:
```
#to delete the null values
df.dropna(inplace=True)
```

[17]:
```
pd.isnull(df).sum()
```

[17]:
```
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
```

[14]:
```
df.shape
```

[14]:
```
(11239, 13)
```

[15]:
```
df['Amount'] = df['Amount'].astype('int')
```

[16]:
```
df['Amount'].dtypes
```

[16]:
```
dtype('int64')
```

[22]:
```
df.rename(columns={'Marital_Status':'Shaadi'})
#it wont be saved because not used inplace= True. now it created new table/dataframe, wont effect the orginal one
```

[22]:

	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

4. Exploratory Data Analysis (EDA)

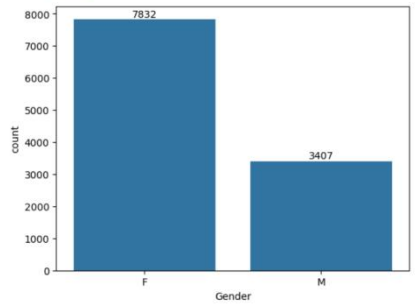
Exploratory Data Analysis was performed to understand data distribution and identify trends. Visualizations were created using Matplotlib and Seaborn to analyze sales performance across different dimensions such as:

- Gender
- Age groups
- States
- Occupation
- Product categories

Gender-wise Sales Analysis

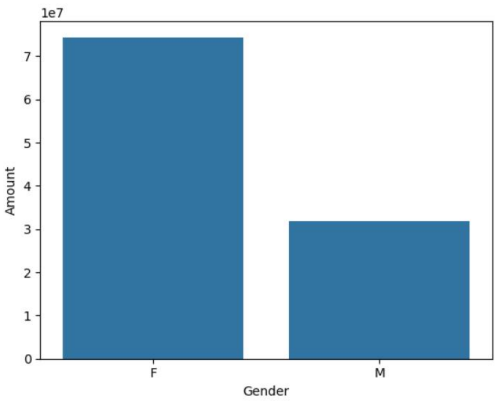
This analysis highlights differences in purchasing behavior between male and female customers.

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

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




```
*[52]: sales_gen = df.groupby(['Gender'],as_index=False).agg({'Amount':'sum'}).sort_values(by='Amount', ascending=False)

       sns.barplot(data= sales_gen, x = 'Gender', y= 'Amount')
```

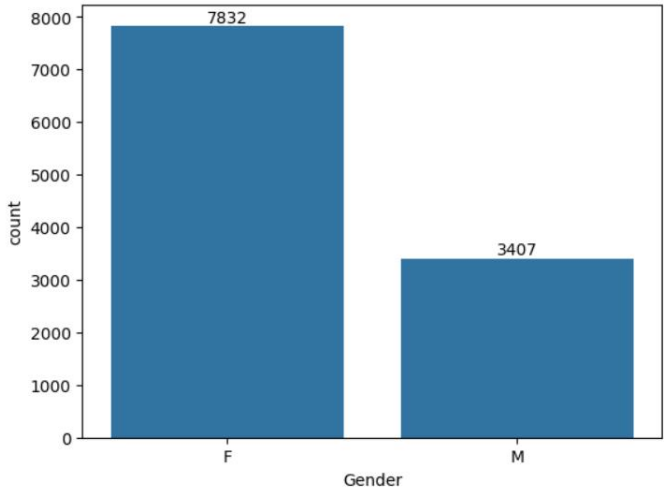
```
[52]: <Axes: xlabel='Gender', ylabel='Amount'>
```



Age Group Analysis

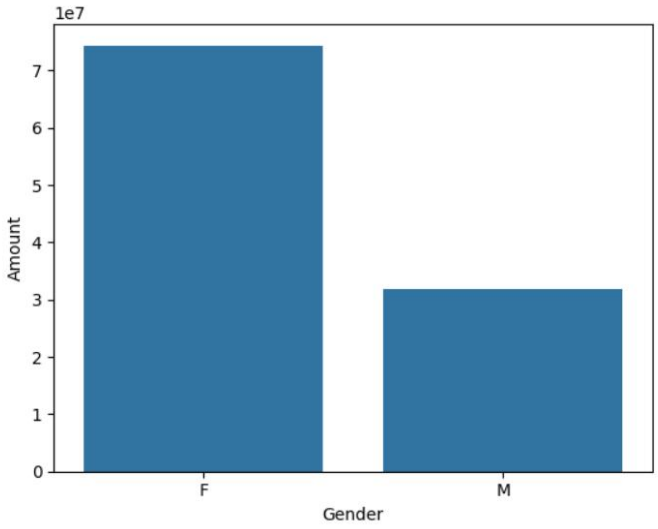
Sales performance was analyzed across different age groups to understand which segment contributes the most to revenue.

[37]:
```
ax=sns.countplot(data = df,x='Gender')

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

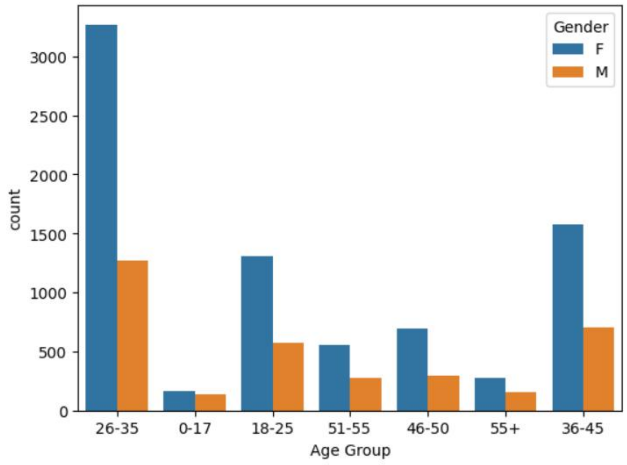


[52]:
```
sales_gen = df.groupby(['Gender'],as_index=False).agg({'Amount':'sum'}).sort_values(by='Amount', ascending=False)

sns.barplot(data= sales_gen, x = 'Gender', y= 'Amount')
```

[52]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```

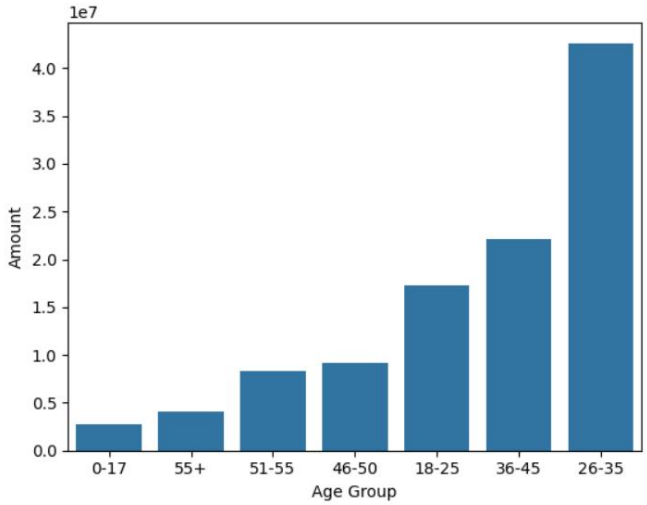


[61]:
```
sns.countplot(data=df,x='Age Group',hue='Gender')
plt.show()
```



[73]:
```
sales_age= df.groupby(['Age Group'],as_index=False).agg({'Amount':'sum'}).sort_values(by='Amount')

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

[73]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

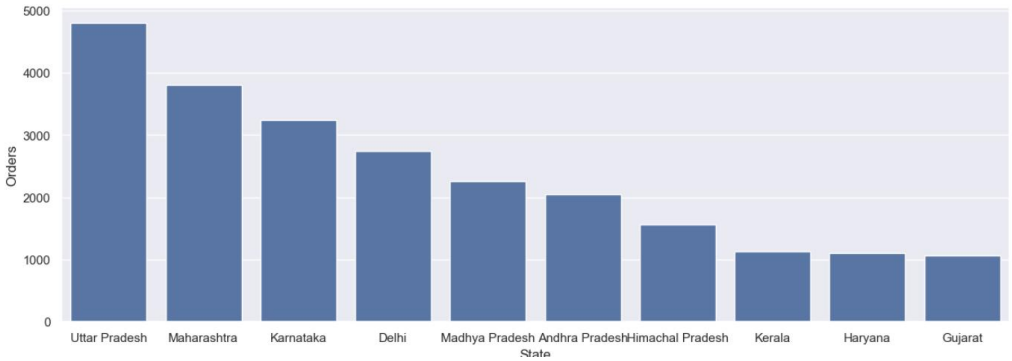


State-wise and Occupation-wise Analysis

State-wise and occupation-wise analysis helped identify potential customer regions and professions contributing significantly to sales.

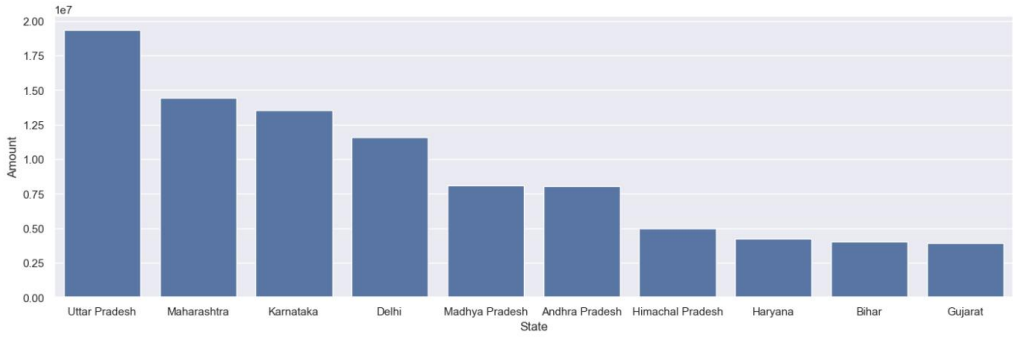
```
[89]: #total no.of orders from top 10 states

Orders_state= df.groupby('State').agg({'Orders':'sum'}).sort_values(by='Orders',ascending=False).head(10)
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = Orders_state, x='State',y='Orders')
```

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




```
Total sales from to 10 states

[95]: Amount_states= df.groupby('State').agg({'Amount':'sum'}).sort_values(by='Amount',ascending=False).head(10)
sns.set(rc={'figure.figsize':(17,5)})
sns.barplot(data=Amount_states, x='State', y='Amount')
```

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



From above graphs we can see that most of the orders and total sales/amount are from uttarpradesh,maharastra and karnataka respectively

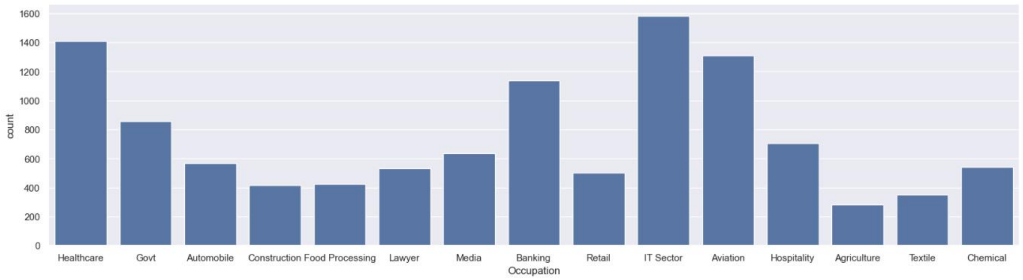
Occupation

[115]:
```
df.columns
```

[115]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

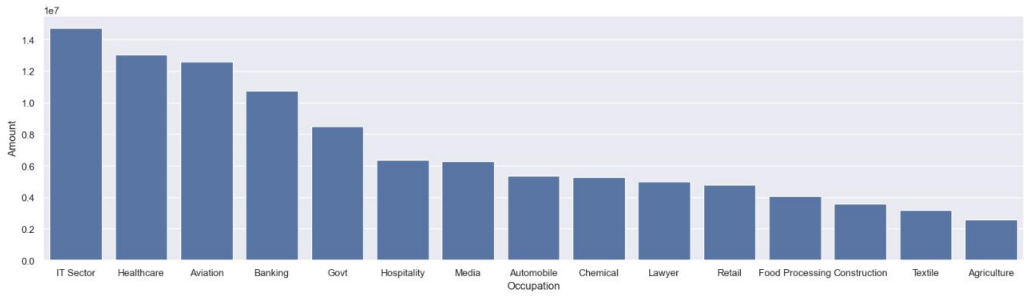
[119]:
```
sns.set(rc={'figure.figsize':(20,5)})
sns.countplot(data=df,x='Occupation')
```

[119]:
```
<Axes: xlabel='Occupation', ylabel='count'>
```



[122]:
```
#which occupation people purchased more

occ_sales=df.groupby('Occupation').agg({'Amount':'sum'}).sort_values(by='Amount',ascending=False)

sns.barplot(data=occ_sales,x='Occupation',y='Amount')
```

[122]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

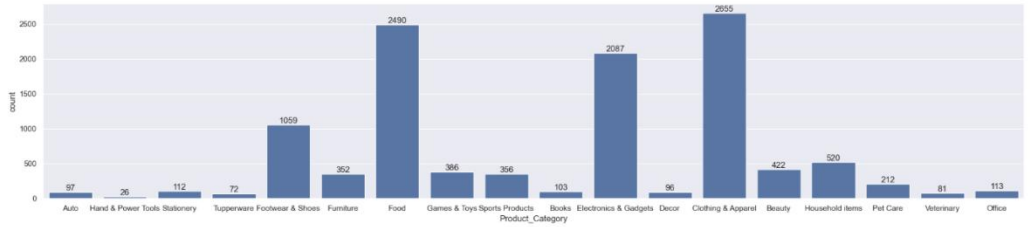


From the above graph, we can see that most of the buyers from IT sector, Healthcare and Aviation sector.

Product Category Analysis

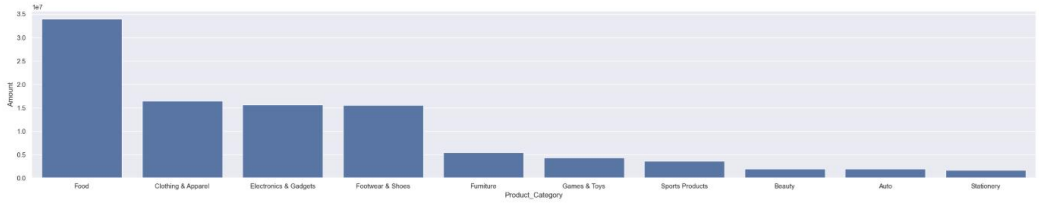
This analysis identified the most selling product categories and products, which can help in inventory planning and demand forecasting.

```
[129]: sns.set(rc={'figure.figsize':(25,5)})
       ax=sns.countplot(data=df,x='Product_Category')

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




```
[130]: #which product category got more sales/amount

       procat_sales= df.groupby('Product_Category').agg({'Amount':'sum'}).sort_values(by='Amount',ascending=False).head(10)
       sns.set(rc={'figure.figsize':(30,5)})
       sns.barplot(data=procat_sales,x='Product_Category',y='Amount')
```

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



From above graph, we can see that most of the sold products are from Food,clothing& apparel and Electronics&Gadgets

EDA helped in identifying patterns, outliers, and relationships between customer demographics and purchasing behavior.

5. Key Insights and Findings

- Identified potential customers across different **states, occupations, genders, and age groups**, helping improve customer targeting strategies.
- Analyzed **most selling product categories and products**, which can support better inventory planning.
- Observed purchasing trends that can help businesses align supply with customer demand.
- Insights from the analysis can be used to enhance **customer experience and sales performance**.

6. Business Impact

The insights derived from this project can help businesses:

- Improve customer experience through targeted marketing
- Increase sales by focusing on high-performing products
- Optimize inventory planning to meet demand efficiently
- Make data-driven decisions using customer behavior analysis

SQL QUERIES

1. **What is the total sales amount by gender?**

```
SELECT gender,SUM(Amount) AS total_sales
FROM diwali_sales
GROUP BY gender;
```

Data Output | Messages | Notifications

	gender text	total_sales numeric
1	M	31913276
2	F	74335853

2. **Which age group has generated the highest total sales?**

```
SELECT age_group,SUM(amount) AS  highest_total_sales
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Data Output | Messages | Notifications

	age_group text	highest_total_sales numeric
1	26-35	42613442

3. **What are the top 5 states by total sales amount?**

```
SELECT state,SUM(amount) AS total_sales
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

Data Output | Messages | Notifications

	state text	total_sales numeric
1	Uttar Pradesh	19374968
2	Maharashtra	14427543
3	Karnataka	13523540
4	Delhi	11603818
5	Madhya Prade...	8101142

4. **Which product categories contribute the most to overall revenue?**

```
SELECT product_category,SUM(amount) AS  overall_revenue
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC;
```

Data Output | Messages | Notifications

	product_category text	overall_revenue numeric
1	Food	33933883
2	Clothing & Apparel	16495019
3	Electronics & Gadgets	15643846
4	Footwear & Shoes	15575209
5	Furniture	5440051
6	Games & Toys	4331694
7	Sports Products	3635933
8	Beauty	1959484
9	Auto	1958609
10	Stationery	1676051
11	Household items	1569337
12	Tupperware	1155642
13	Books	1061478
14	Decor	730360
15	Pet Care	482277
16	Hand & Power Tools	405618
17	Veterinary	112702
18	Office	81936

5. **What are the top 10 most purchased products based on transaction count?**

```
SELECT product_id,COUNT(*) AS transaction_count
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
```

Data Output | Messages | Notifications

	product_id text	transaction_count bigint
1	P00265242	53
2	P00110942	44
3	P00184942	37
4	P00237542	35
5	P00112142	34
6	P00114942	33
7	P00110742	32
8	P00112542	30
9	P00110842	30
10	P00145042	30

6. **How does marital status (Married vs Single) impact total sales?**

```
SELECT marital_status, SUM(orders) AS total_orders, SUM(amount) AS total_sales
FROM diwali_sales
GROUP BY 1;
```

Data Output | Messages | Notifications

	marital_status bigint	total_orders numeric	total_sales numeric
1	0	16249	62125384
2	1	11732	44123745

7. **What is the average purchase amount for each age group?**

```
SELECT age_group, ROUND(AVG(amount),2) AS average_purchase_amount
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC;
```

Data Output | Messages | Notifications

	age_group text	average_purchase_amount numeric
1	51-55	9953.59
2	36-45	9699.95
3	55+	9557.35
4	26-35	9384.15
5	46-50	9367.08
6	18-25	9175.48
7	0-17	9120.45

8. **Which occupations contribute the highest sales revenue?**

```sql
SELECT occupation, SUM(amount) AS sales_revenue
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3; -- TOP 3 OCCUPATIONS CONTRIBUTED HIGHEST SALES REVENUE
```

Data Output | Messages | Notifications

	occupation text	sales_revenue numeric
1	IT Sector	14755079
2	Healthcare	13034586
3	Aviation	12602298

9. **What percentage of total sales does each product category represent?**

```sql
SELECT product_category, ROUND(100 * SUM(amount) / (SELECT SUM(amount) FROM diwali_sales),2) AS sales_percentage
FROM diwali_sales
GROUP BY 1
ORDER BY 2 DESC;
```

Data Output | Messages | Notifications

	product_category text	sales_percentage numeric
1	Food	31.94
2	Clothing & Apparel	15.52
3	Electronics & Gadgets	14.72
4	Footwear & Shoes	14.66
5	Furniture	5.12
6	Games & Toys	4.08
7	Sports Products	3.42
8	Beauty	1.84
9	Auto	1.84
10	Stationery	1.58
11	Household items	1.48
12	Tupperware	1.09
13	Books	1.00
14	Decor	0.69
15	Pet Care	0.45
16	Hand & Power Tools	0.38
17	Veterinary	0.11
18	Office	0.08

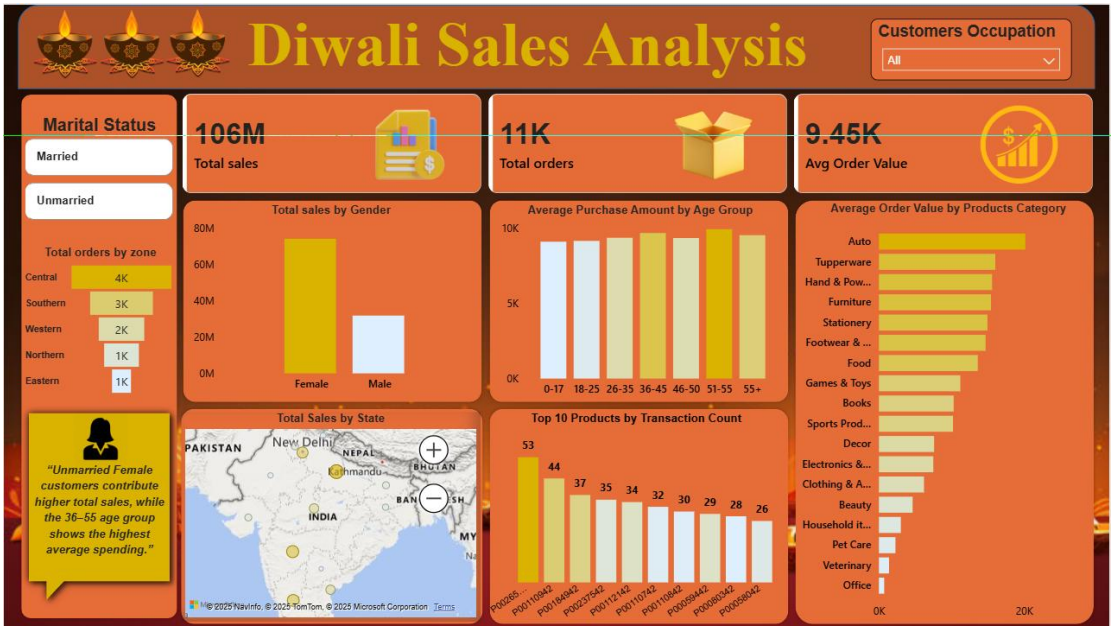
10. **Which age group and gender combination generates the highest average sales?**

```
SELECT age_group,gender, ROUND(AVG(amount),2) as highest_average_sales
FROM diwali_sales
GROUP BY 1,2
ORDER BY 3 DESC
LIMIT 1;
```

Data Output | Messages | Notifications

	age_group text	gender text	highest_average_sales numeric
1	55+	M	10813.26

Power BI



Dashboard Link:
https://app.powerbi.com/view?r=eyJrIjoiZDA0NzEwY2QtMjAxMC00M2M1LWJhMDMtZGM1NWJjNDM0YTkzIiwidCI6ImUxNGU3M2ViLTUyNTEtNDM4OC04ZDY3LThmOWYyZTJkNWE0NiIsImMiOjEwfQ%3D%3D

7. Conclusion

This project demonstrates an end-to-end data analytics workflow using **Python, SQL, and Power BI** to analyze Diwali sales data and uncover meaningful business insights. Python was used for data cleaning, pre-prossessing, and exploratory data analysis to ensure data quality and understand customer behavior patterns. SQL was applied to answer key business questions related to customer segmentation, product performance, and sales distribution through structured queries. Power BI was then used to transform these insights into an interactive and visually intuitive dashboard.

The analysis revealed important trends such as higher sales contribution from female and unmarried customers, strong purchasing power among the 36–55 age group, and clear identification of top-performing product categories and products. Geographic and zone-wise analysis further supported regional sales understanding and operational planning.

Overall, this project highlights the ability to combine **programming, querying, and business intelligence tools** to convert raw data into actionable insights. It reflects practical skills in data analysis, data visualization, and data-driven decision-making, making it relevant for real-world business and analytics roles.