Analysis on diwali Sales Importing the necessary libraries In [68]: **import** pandas **as** pd import numpy as np import matplotlib.pyplot as plt %matplotlib inline import seaborn as sns import warnings warnings.filterwarnings("ignore") Importing the data to be analyse In [3]: df=pd.read_csv("Diwali_Sales_Data.csv", encoding="unicode_escape") In [4]: df.head() Out[4]: Age Age Marital_Status User_ID Cust_name Product_ID Gender State Zone Occupation Product_Category Orders Amount Status unnamed1 Group 0 1002903 Sanskriti P00125942 26-35 28 Maharashtra Western Healthcare Auto 1 23952.0 NaN NaN 1 1000732 Kartik P00110942 26-35 35 Andhra Pradesh Southern Govt Auto 3 23934.0 NaN NaN 2 1001990 P00118542 26-35 35 Uttar Pradesh Central Automobile 3 23924.0 NaN NaN Bindu Auto **3** 1001425 0 Sudevi P00237842 0-17 16 Southern Construction 2 23912.0 NaN NaN Karnataka Auto Food 4 1000588 Joni P00057942 26-35 28 Gujarat Western Auto 2 23877.0 NaN NaN Processing In [5]: df.shape Out[5]: (11251, 15) Performing Data Cleaning 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 11251 non-null object 1 Cust_name 11251 non-null object Product_ID Gender 11251 non-null object 11251 non-null object 4 Age Group 11251 non-null int64 5 Age 11251 non-null int64 Marital_Status 6 7 State 11251 non-null object 11251 non-null object 8 Zone 11251 non-null object 9 Occupation Product_Category 11251 non-null object 10 Orders 11 11251 non-null int64 11239 non-null float64 12 Amount 13 Status 0 non-null float64 float64 14 unnamed1 0 non-null dtypes: float64(3), int64(4), object(8)memory usage: 1.3+ MB In [7]: df.drop(['Status', 'unnamed1'], axis=1, inplace=True) In [8]: pd.isnull(df).sum() Out[8]: User_ID 0 Cust_name 0 Product_ID 0 Gender 0 Age Group Age Marital_Status State Zone 0 Occupation Product_Category 0 0 Orders Amount 12 dtype: int64 In [9]: df['Amount'] = df['Amount'].fillna(df['Amount'].mean()) df.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 11251 entries, 0 to 11250 Data columns (total 13 columns): # Column Non-Null Count Dtype ------ - -11251 non-null int64 User_ID 0 Cust_name 11251 non-null object Product_ID 11251 non-null object 2 11251 non-null object 3 Gender Age Group 11251 non-null object Age 11251 non-null int64 Marital_Status 11251 non-null int64 11251 non-null object 7 State 11251 non-null object 8 Zone 9 Occupation 11251 non-null object 10 Product_Category 11251 non-null object 11 Orders 11251 non-null int64 12 Amount 11251 non-null float64 dtypes: float64(1), int64(4), object(8) memory usage: 1.1+ MB In [10]: df.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 11251 entries, 0 to 11250 Data columns (total 13 columns): # Column Non-Null Count Dtype ------ - -User_ID 0 11251 non-null int64 Cust_name 11251 non-null object Product_ID 11251 non-null object 2 11251 non-null object 3 Gender Age Group 11251 non-null object Age 11251 non-null int64 Marital_Status 11251 non-null int64 11251 non-null object 7 State 11251 non-null object 8 Zone Occupation 11251 non-null object 10 Product_Category 11251 non-null object 11 Orders 11251 non-null int64 12 Amount 11251 non-null float64 dtypes: float64(1), int64(4), object(8) memory usage: 1.1+ MB In [11]: df["Amount"]=df["Amount"].astype("int") In [12]: df.columns Out[12]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age', 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category', 'Orders', 'Amount'], dtype='object') In [13]: df.rename(columns={"Marital_Status":"Marriage"}) Out[13]: User_ID Cust_name Product_ID Gender Age Group Age Marriage Occupation Product_Category Orders Amount State Zone 0 1002903 Sanskriti P00125942 28 23952 Maharashtra Western Healthcare 1 1000732 Kartik P00110942 26-35 35 23934 Andhra Pradesh Southern Govt Auto 3 35 Automobile 23924 2 1001990 Bindu P00118542 26-35 Uttar Pradesh Central Auto **3** 1001425 Sudevi P00237842 0-17 16 Karnataka Southern Construction Auto 23912 **4** 1000588 Joni P00057942 26-35 28 Gujarat Western Food Processing Auto 23877 **11246** 1000695 Manning P00296942 18-25 Maharashtra Western Chemical Office 4 370 11247 1004089 Reichenbach P00171342 26-35 33 Haryana Northern Healthcare 367 Veterinary **11248** 1001209 Oshin P00201342 36-45 40 0 Madhya Pradesh Central Textile Office 213 **11249** 1004023 Noonan P00059442 37 Agriculture Office 206 36-45 Karnataka Southern **11250** 1002744 Brumley P00281742 Maharashtra Western 18-25 19 Healthcare Office 3 188 11251 rows × 13 columns In [14]: df[["Age", "Orders", "Amount"]].describe() Out[14]: Orders Amount count 11251.000000 11251.000000 11251.000000 mean 35.421207 2.489290 9453.609901 12.754122 1.115047 5219.569169 std min 12.000000 1.000000 188.000000 25% 27.000000 1.500000 5443.500000 50% 33.000000 2.000000 8110.000000 75% 43.000000 3.000000 12671.000000 max 92.000000 4.000000 23952.000000 Exploratory Data Analysis: EDA In [15]: df.columns Out[15]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age', 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category', 'Orders', 'Amount'], dtype='object') In [16]: br=sns.countplot(data=df, x="Age Group", hue="Gender") for bars in br.containers: br.bar_label(bars) 3271 Gender 3000 2500 2000 count 1581 1500 1305 1272 1000 705 696 574 554 500 291 273 278 162₁₃₄ 155 0 26-35 0-17 18-25 55 +46-50 51-55 36-45 Age Group In [17]: sales_age = df.groupby(["Age Group"], as_index=False).sum().sort_values(by="Amount", ascending=False) zx=sns.barplot(x="Age Group", y="Amount", data=sales_age) **for** bar **in** zx.patches: height = bar.get_height() zx.text(bar.get_x() + bar.get_width() / 2, height, f"{height:.0f}", ha="center", va="bottom") 1e7 42632348 4.0 3.5 3.0 2.5 -2.0 -22173353 17240732 1.5 9245656 1.0 8280383 4090440 0.5 2699653 0.0 26-35 36-45 18-25 46-50 51-55 55 +0-17 Age Group From above graph we can see that the most of the buyer are of age group between 26-35 years In [24]: sales_state = df.groupby(["State"], as_index=False).sum().sort_values(by="Orders", ascending=False).head(10) sns.set(rc={"figure.figsize":(15,5)}) xy=sns.barplot(x="State", y="Orders", data=sales_state, hue="State") 5000 4000 3000 Orders 2000 1000 0 Delhi Madhya Pradesh Andhra PradeshHimachal Pradesh Uttar Pradesh Maharashtra Karnataka Kerala Gujarat Haryana State So from above graph we see that the Uttarprades have maximum number of orders and Gujrat have minimum of orders In [38]: sales_state_Amt = df.groupby(["State"], as_index=False).sum().sort_values(by="Amount", ascending=False).head(10) sns.set(rc={"figure.figsize":(15,5)}) xy=sns.barplot(x="State", y="Amount", data=sales_state_Amt, hue="State") 1e7 2.00 1.75 1.50 1.25 Amount 1.00 0.75 0.50 0.25 0.00 Uttar Pradesh Delhi Madhya Pradesh Andhra PradeshHimachal Pradesh Bihar Maharashtra Karnataka Gujarat State Now we can see that the Haryana have less orders then Bihar but Haryana spends more finacialy In [70]: sales_Marital_Status = df.groupby(["Marital_Status", "Gender"], as_index=False).sum().sort_values(by="Amount", ascending=False) sns.set(rc={"figure.figsize":(4,3)}) # Replace numerical values with categorical labels sales_Marital_Status["Marital_Status"].replace({0: "Married", 1: "Unmarried"}, inplace=True) cx=sns.barplot(x="Marital_Status", y="Amount", hue="Gender", data=sales_Marital_Status) for bars in cx.containers: **for** bar **in** bars: height = bar.get_height() / 1000000 # Dividing by 1,000,000 to convert to millions cx.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.1f}M', ha='center', va='bottom') 1e7 4 Amount 1 43.8M 18.3M 13.6M 0 Married Unmarried Marital_Status From above graph we can see that most of the buyers are married womens In [74]: sns.set(rc={"figure.figsize":(20,5)}) ax=sns.countplot(data=df, x="Occupation", hue="Occupation") **for** bar **in** ax.patches: height = bar.get_height() ax.text(bar.get_x() + bar.get_width() / 2, height, f'{height}', ha='center', va='bottom') 1588.0 1600 1408.0 1400 1310.0 1200 1139.0 1000 854.0 800 705.0 637.0 566.0 600 542.0 531.0 501.0 423.0 414.0 400 350.0 283.0 200 0 Banking Hospitality Healthcare Govt Automobile Construction Food Processing Lawyer IT Sector Retail Agriculture Textile Chemical Media Aviation Occupation In [75]: sales_Occupation = df.groupby(["Occupation"], as_index=False).sum().sort_values(by="Amount", ascending=False).head(10) sns.set(rc={"figure.figsize":(15,5)}) xy=sns.barplot(x="Occupation", y="Amount", data=sales_Occupation , hue="Occupation") 1e7 1.4 1.2 1.0 Amount 0.8 0.6 0.4 0.2 0.0 IT Sector Healthcare Govt Media Aviation Banking Hospitality Automobile Chemical Lawyer Occupation **From above graph we can see that most of the buyers are working in IT Sector, Healthcare, Aviation. *** In [78]: sales_Product_Category = df.groupby(["Product_Category"], as_index=False).sum().sort_values(by="Amount", ascending=False).head(10) sns.set(rc={"figure.figsize":(20,5)}) xy=sns.barplot(x="Product_Category", y="Amount", data=sales_Product_Category , hue="Product_Category") 3.5 3.0 2.5 1.0 0.5 0.0 Food Clothing & Apparel Electronics & Gadgets Footwear & Shoes Games & Toys Sports Products Beauty Stationery Product_Category In [80]: sales_Product_Category = df.groupby(["Product_Category"], as_index=False).sum().sort_values(by="Orders", ascending=False).head(10) sns.set(rc={"figure.figsize":(20,5)}) xy=sns.barplot(x="Product_Category", y="Orders", data=sales_Product_Category , hue="Product_Category") 6000 5000 4000 3000 3000 2000 1000 0 Household items Clothing & Apparel Food Electronics & Gadgets Footwear & Shoes Beauty Games & Toys Furniture Sports Products Pet Care Product_Category From above two grapg we can see that the clothing&apparel have highest number of order but amount spen on food is higher then clothing&apparel Conclusion The Diwali sales data analysis reveals key insights into consumer behavior and preferences during the festive season. To maximize sales during Diwali: -Target marketing efforts towards the age group between 26-35 years, especially focusing on married women. -Tailor promotional campaigns to resonate with buyers from states like Uttar Pradesh and Haryana, considering their high contribution to sales. -Focus on product categories like clothing & apparel and food, as they attract significant orders and spending.