Data Analysis Project Report on ‘Amazon Sales’ Dataset

(By Gitanjali Nikam)

Contents:

Executive Summary:

This report focuses on analysing the sales and revenue data from April to June 2022, with the objective

of developing strategies to increase overall revenue by the end of Q3 2022. The analysis highlights the

performance of various product categories, customer spending patterns, and the impact of

promotions. Our primary goal is to increase revenue by at least 5% for the lowest-performing product

among the top 3 revenue-generating categories, and to target specific states for promotions, with Western Dress identified as a key category for driving additional sales.

Key Findings:

1. Top 3 Products: The product categories "Set," "Kurta," and "Western Dress" generate the highest revenue. However, Western Dress contributes the least among these, making it a primary target for revenue growth through targeted promotions.

2. State-wise Performance: Sales are concentrated in a few key states. The top 50% of states contribute significantly to overall revenue, particularly for Western Dress, making them ideal targets for promotions.

3. Promotion Strategy: To achieve the target of increasing revenue by 5% for the third-highest revenue-generating product, we propose implementing discounts, Buy 1 Get 1 Free offers, and free shipping in the top-performing states.

4. Customer Spending: Business customers tend to have a higher average order value compared to regular customers, providing an opportunity to target B2B customers with specialized promotions.

Recommendations:

• Promote Western Dress and other underperforming categories in the top 50% of states based on revenue.

• Implement targeted promotions such as discounts and Buy 1 Get 1 Free offers to boost sales.

• Focus on Q3 2022 promotions and track performance to ensure the revenue target of a 5% increase is met by the end of Q3 2022.

This report provides a detailed analysis of current sales trends and a well-defined promotion strategy

aimed at driving revenue growth across multiple categories by the end of Q3 2022.

Introduction:

The purpose of this report is to analyse the sales and revenue data from April to June 2022 and develop

strategies to drive revenue growth by the end of Q3 2022. The data includes a comprehensive review of

product category performance, customer spending patterns, and state-wise sales, which are used to

create targeted promotional strategies aimed at increasing overall revenue.

The analysis identifies key product categories, such as Set, Kurta, and Western Dress, which are the

top revenue-generating categories during the analysis period. Among these, Western Dress presents a

significant opportunity for growth as it currently ranks third in revenue generation, making it a prime

target for promotional efforts.

8788385365

Additionally, the report examines customer spending behaviour, highlighting the differences between

business customers and regular customers, and how these segments can be effectively targeted to

maximize revenue. The state-wise sales analysis further emphasizes the importance of focusing

promotions in the top-performing states that contribute significantly to overall sales.

The primary objective is to achieve a 5% increase in revenue for the third-highest revenue-generating

product category by the end of Q3 2022 through well-planned promotions and customer engagement

strategies. This report will outline the steps necessary to reach this goal, ensuring that both product

category performance and revenue growth are optimized.

Data Cleaning:

1. Dropping irrelevant columns: There are some columns present in the dataset which is irrelevant and have no use to be in the analysis. Example: Unnamed 22, fulfilled-by, Ship Country, Currency, Sales Channel.

```
columns_to_drop = ['Unnamed: 22', 'fulfilled-by', 'ship-country','currency','Sales Channel ']
df= df.drop(columns=columns_to_drop,axis=1)
df
```

2. Dropping duplicates: Dataset contains few duplicate values which is being removed during analysis.

```
df = df.drop_duplicates(subset=['Order ID','ASIN'])
df
```

3. Filling missing values: Dataset contains some missing values which can be fatal to the analysis so, it has been removed.

```
fill_na_values = {'Courier Status': 'Unknown', 'promotion-ids': 'No Promotion', 'Amount': 0}
df= df.fillna(value=fill_na_values)
df
```

4. Renaming columns: In the provided dataset a column is present which is playing an important role but its name is not as accurate as it should be. So, it has been changed.

```
[37] column_rename_dict={'B2B':'customer_type','Amount':'order_amount_($)'}
     df = df.rename(columns=column_rename_dict)
     df
```

5. Adding Datetime & month column: A new column named month is being added to the dataset in order to perform some useful analysis.

```
[17] #create datetime and month columns
     #column creation
     #month - to use in analysis and groupbys
     #column value ordering
     df['Month'] =df['Date'].dt.month_name()
     df
```

```
df['Month'] =df['Date'].dt.month_name()
```

6. Creating a product category on the basis of product sizes.

```
[ ]  # CREATE AN ORDERED  CATEGORY FOR THE  'SIZE' COLUMN
     size_order =['S','M','L','XL','XS','XXL','3XL','4XL','5XL','6XL','Free']
     df['size']=pd.Categorical(df['Size'],categories=size_order, ordered=True)
     df
```

7. Renaming the Customer type column values: This column contains True and False which is not conveying a correct message in the analysis so it is being changed to Business Customer and Regular Customer.

```
[14] df['customer_type']=df['customer_type'].astype(str)
     df['customer_type'] = df['customer_type'].str.strip().replace({'True':'Business','False':'Customer'})
     df
```

8. Converting INR to USD:

```
[15] #AMOUNT - changing  to order_amount _($) and  converting from to $
     exchange_rate = 0.0120988
     df['order_amount_($)'] =(df['order_amount_($)'].apply(lambda x:x * exchange_rate)).round(2)
     df
```

Data Analysis:

The data from April to June 2022 was analysed to identify trends in product category performance,

customer spending behaviour, cancellation/return patterns, and state-wise sales. The goal is to

develop strategies to boost overall revenue and improve product category performance. The analysis

highlights the following key findings:

1. Top Products Analysis:

The top 3 revenue-generating product categories are Set, Kurta, and Western Dress:

• Set: Generates the highest revenue, accounting for 50% of the total sales.

8788385365

• Kurta: Contributes 27% of the total revenue.

• Western Dress: Contributes 14%, and despite being third, it offers significant potential for

growth through targeted promotions.

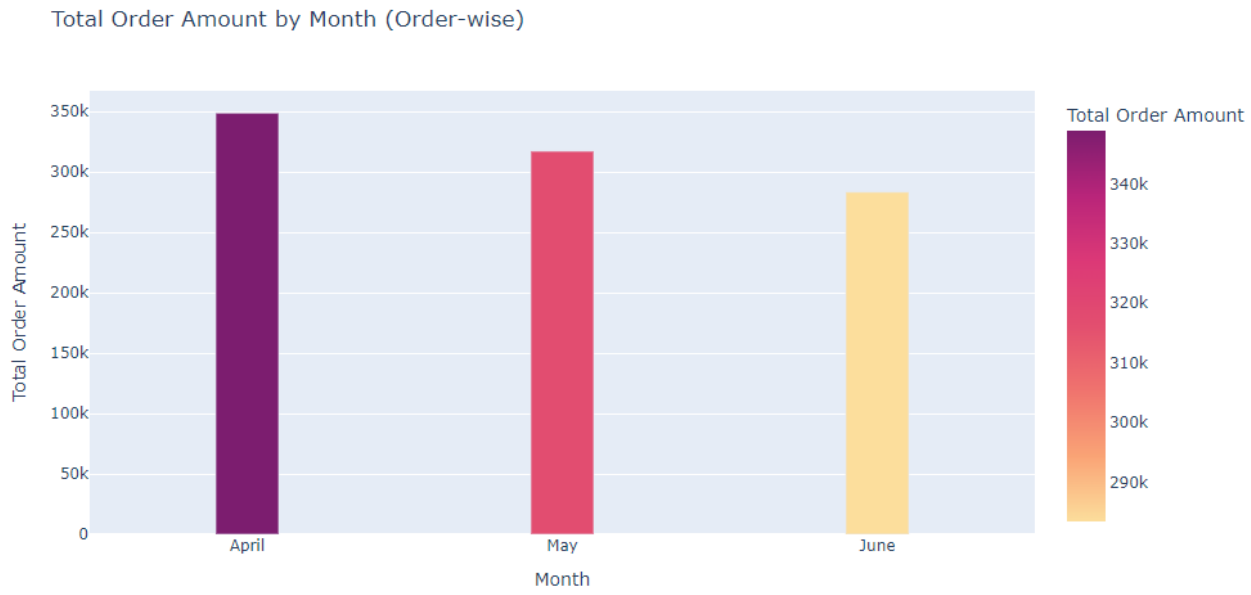
```
[19] import pandas as pd
     import plotly.express as px

     #ASSUMING YOUR DATAFRAME IS NAMED 'df'
     #REPLACAE 'Amount' and 'Category' with the actual column names in your  dataframes

     #Step 1 : Group by Category  and Calculate Sum  of Amount
     grouped_df =df.groupby('Category')['order_amount_($)'].sum().reset_index()

     #Step 2 :Select  Top 3 categories
     top3_categories=grouped_df.nlargest(3,'order_amount_($)')

     #Step 3: Create  a Bar Chart for the top 3 categories
     fig=px.bar(top3_categories,x='Category',y='order_amount_($)',
                title ='Top 3 Categories by Total Amount ',
                labels={'order_amount_($)':'Total Amount','Category':'Product Category'},
                color='order_amount_($)',color_continuous_scale='greens')

     #cusomize the layout  and appearance  of the  chart
     fig.update_layout(xaxis_title='Product Category',yaxis_title='Total Amount')

     # show the chart
     fig.show()
```

Preliminary Analysis

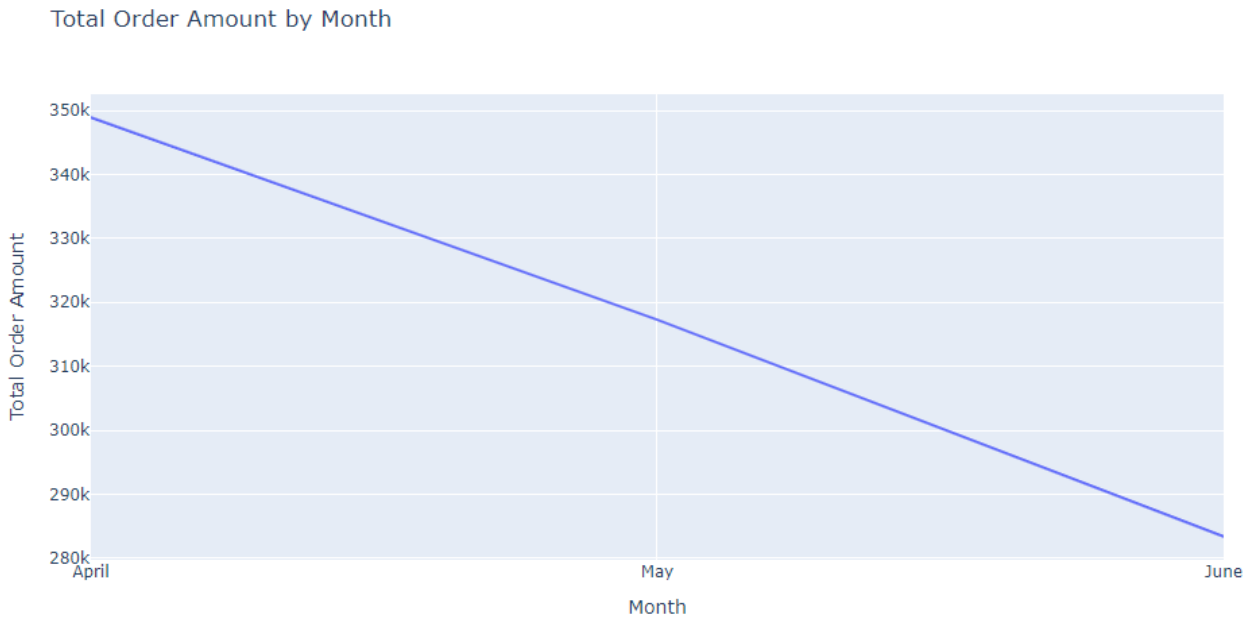
--Based on the analysis, we can draw some preliminary insights about the Q2 2022 performance of Amazon India:

The total revenue for Q2 2022 decreased by -18.77% from April to June, with May revenue experiencing a -9.06% decrease from April, and June revenue seeing a -10.68% decrease from May, which is a cause of concern.

```
[20] import pandas as pd
     import plotly.express as px

     # Step 1: Group by Month and Calculate Total Order Amount
     group_df = df.groupby('Month')['order_amount_($)'].sum().reset_index()

     # Step 2: Sort the DataFrame by Month in the order of appearance in the original DataFrame
     months_order = df['Month'].unique()
     group_df_sorted = group_df.set_index('Month').reindex(months_order).reset_index()

     # Step 3: Create a Bar Chart for Total Order Amount by Month
     fig = px.bar(group_df_sorted, x='Month', y='order_amount_($)',
                  color='order_amount_($)',
                  color_continuous_scale='sunsetdark',  # You can choose a different color scale
                  labels={'order_amount_($)': 'Total Order Amount', 'Month': 'Month'},
                  title='Total Order Amount by Month (Order-wise)')

     # Customize the layout and appearance of the chart
     fig.update_layout(bargap=0.8, xaxis_title='Month', yaxis_title='Total Order Amount', width=1000)

     # Show the chart
     fig.show()
```

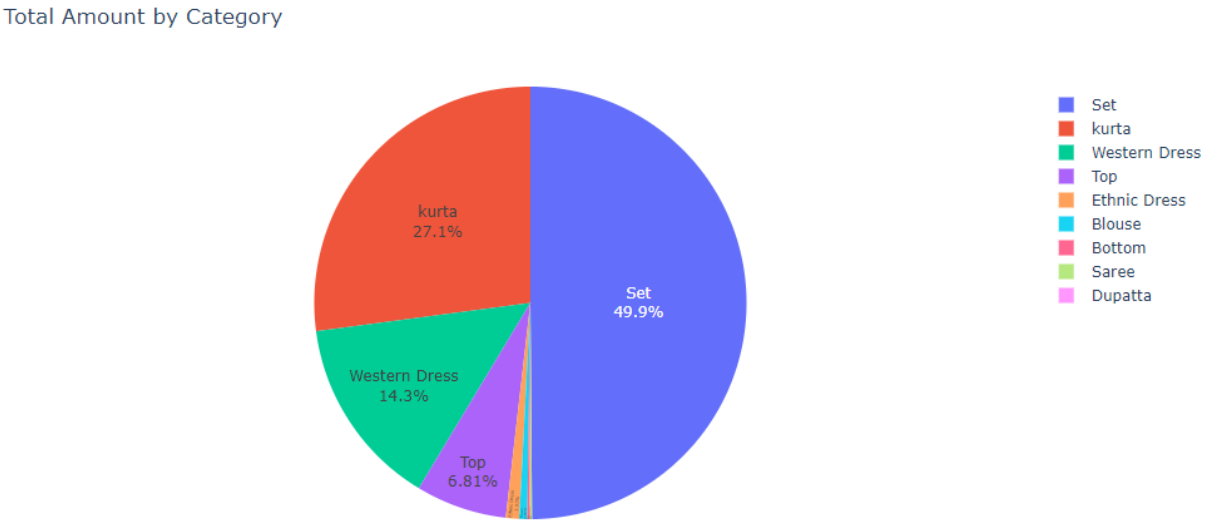



```
[20] import pandas as pd
     import plotly.express as px

     # Step 1: Group by Month and Calculate Total Order Amount
     group_df = df.groupby('Month')['order_amount_($)'].sum().reset_index()

     # Step 2: Sort the DataFrame by Month in the order of appearance in the original DataFrame
     months_order = df['Month'].unique()
     group_df_sorted = group_df.set_index('Month').reindex(months_order).reset_index()

     # Step 3: Create a Bar Chart for Total Order Amount by Month
     fig = px.bar(group_df_sorted, x='Month', y='order_amount_($)',
                  color='order_amount_($)',
                  color_continuous_scale='sunsetdark',  # You can choose a different color scale
                  labels={'order_amount_($)': 'Total Order Amount', 'Month': 'Month'},
                  title='Total Order Amount by Month (Order-wise)')

     # Customize the layout and appearance of the chart
     fig.update_layout(bargap=0.8, xaxis_title='Month', yaxis_title='Total Order Amount', width=1000)

     # Show the chart
     fig.show()
```



The revenue is dominated by the product category "Set," which accounts for 49.88% of total revenue,followed by kurta with 27.09% and Western Dress with 14.28%.

```
[22] import pandas as pd
     import plotly.express as px

     # Assuming your DataFrame is named 'df'
     # Replace 'Amount' and 'Category' with the actual column names in your DataFrame

     # Step 1: Group by Category and Calculate Sum of Amount
     grouped_df = df.groupby('Category')['order_amount_($)'].sum().reset_index()

     # Step 2: Create a Dashboard with a Pie Chart
     fig = px.pie(grouped_df, names='Category', values='order_amount_($)',
                  title='Total Amount by Category',
                  labels={'order_amount_($)': 'Total Amount', 'Category': 'Product Category'})

     # Customize the layout and appearance of the dashboard
     fig.update_traces(textposition='inside', textinfo='percent+label')

     # Show the dashboard
     fig.show()
```



The top 5 product categories by average price in dollars are Set (9.43),$Saree$($9.14),$WesternDress$(8.75), Ethnic Dress (8.26),$andTop$(6.09), indicating that these products are high-value orders

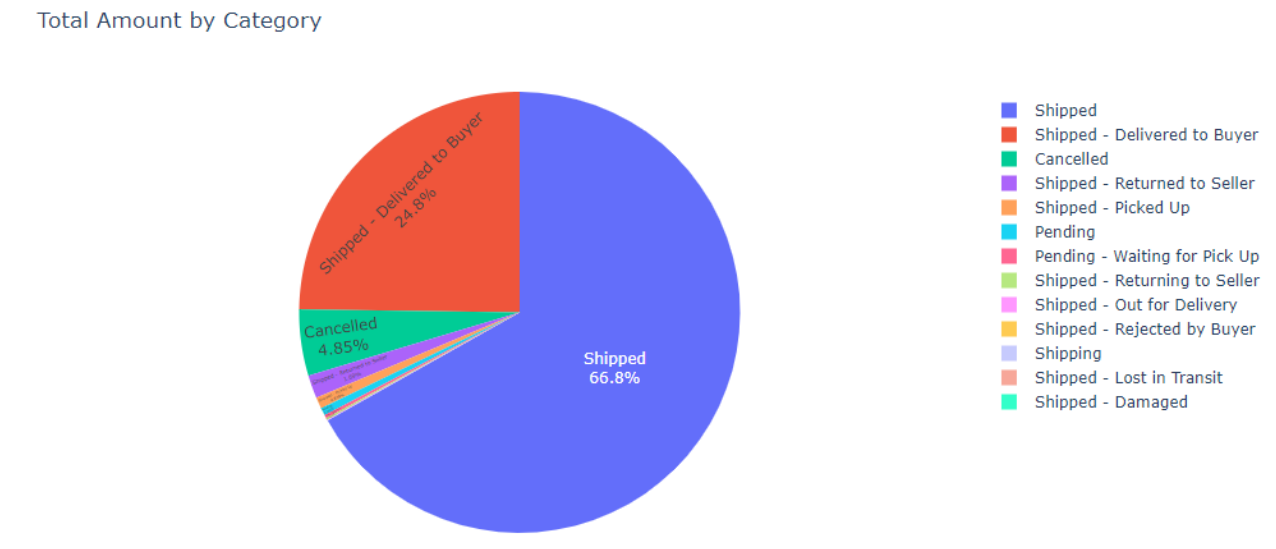
```
[23] import pandas as pd
     import plotly.express as px

     avg_price_df = df.groupby('Category')['order_amount_($)'].mean().reset_index()
     avg_price_df
```

	Category	order_amount_($)
0	Blouse	5.991805
1	Bottom	4.143409
2	Dupatta	3.690000
3	Ethnic Dress	8.254836
4	Saree	9.143537
5	Set	9.433817
6	Top	6.090487
7	Western Dress	8.753854
8	kurta	5.166311

```
[ ] import pandas as pd
    import plotly.express as px

    # Assuming your DataFrame is named 'df'
    # Replace 'order_amount_($)' and 'Category' with the actual column names in your DataFrame

    # Step 1: Group by Category and Calculate the Average Selling Price
    avg_price_by_category = df.groupby('Category')['order_amount_($)'].mean().reset_index()

    # Step 2: Select Top 5 Categories
    top5_categories = avg_price_by_category.nlargest(5, 'order_amount_($)')

    # Step 3: Create a Bar Chart for Top 5 Categories by Average Selling Price
    fig = px.bar(top5_categories, x='Category', y='order_amount_($)',
                 title='Top 5 Categories by Average Selling Price',
                 labels={'order_amount_($)': 'Average Selling Price', 'Category': 'Product Category'},
                 color='order_amount_($)', color_continuous_scale='blues')

    # Customize the layout and appearance of the chart
    fig.update_layout(xaxis_title='Product Category', yaxis_title='Average Selling Price')

    # Show the chart
    fig.show()
```

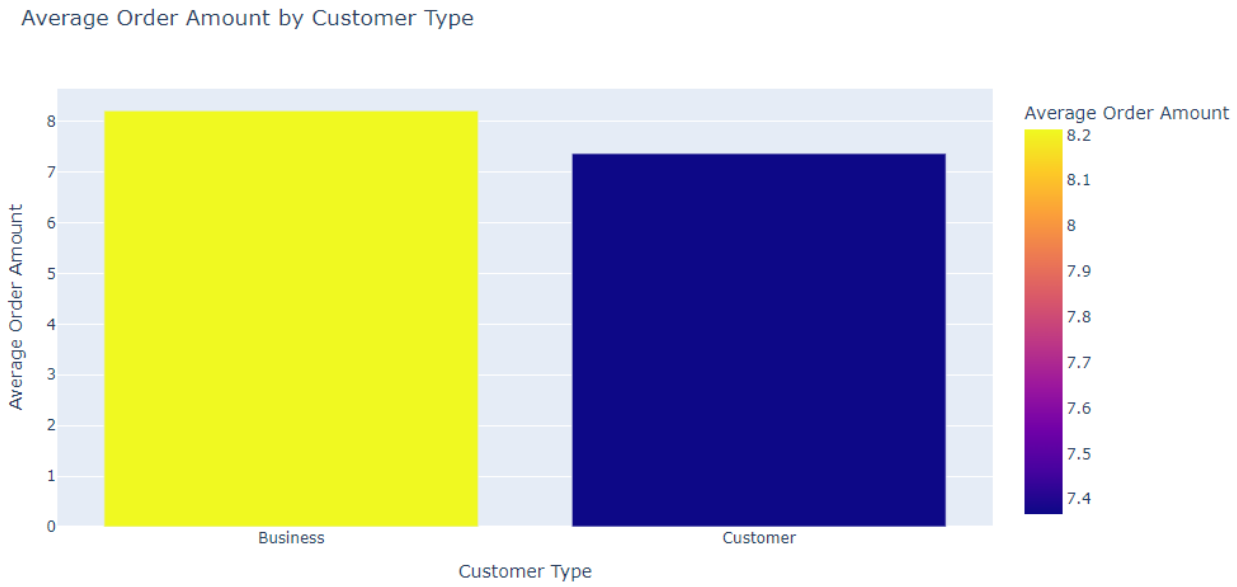
The total number of cancelled and returned orders was 49,178, which represents 17.53% of all orders. Of these, 14.22% were cancelled and 1.64% were returned.

```
[26] import pandas as pd
     import plotly.express as px

     grouped_df = df.groupby('Status',as_index=False)['Qty'].sum()

     fig = px.pie(grouped_df, names='Status', values='Qty',
                  title='Total Amount by Category',
                  labels={'Qty': 'Total Amount', 'Status': 'Order Status'})

     fig.update_traces(textposition='inside', textinfo='percent+label')

     fig.show()
```



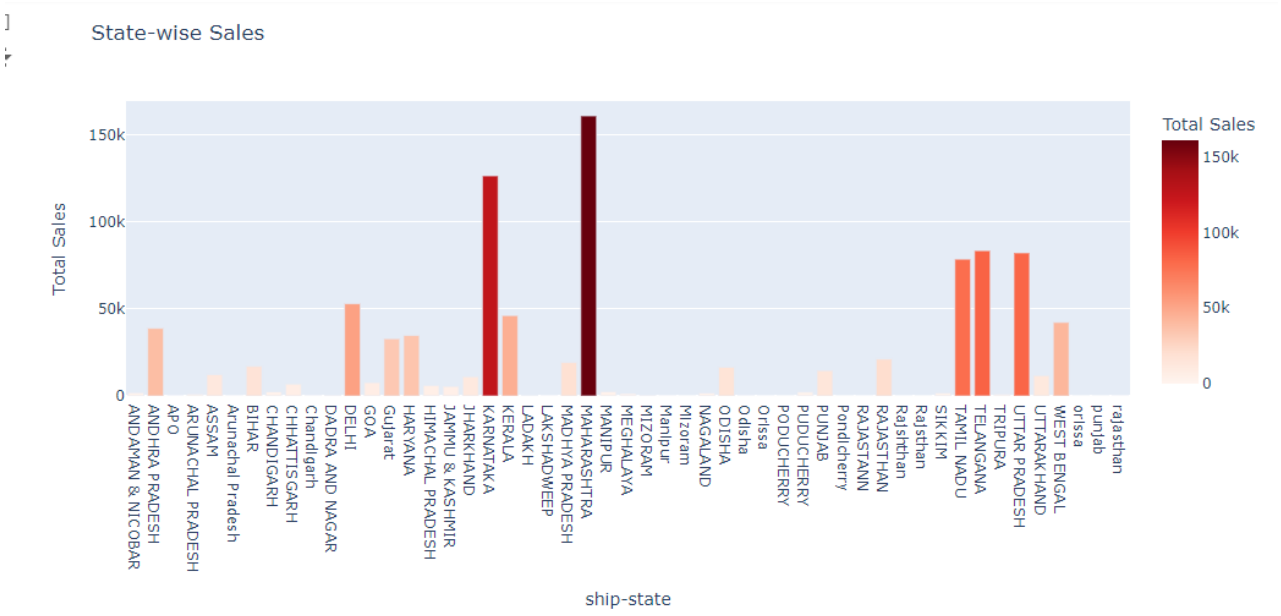
The average order amount by customer type is 8.21$for businesscustomersand$ 7.37 for regular customers.

```
[27] import pandas as pd
     import plotly.express as px

     # Group by 'customer_type' and calculate the average order amount
     average_order_amount = df.groupby('customer_type')['order_amount_($)'].mean().reset_index()

     # Create a bar chart using Plotly Express
     fig = px.bar(average_order_amount, x='customer_type', y='order_amount_($)',
                  title='Average Order Amount by Customer Type',
                  labels={'order_amount_($)': 'Average Order Amount', 'customer_type': 'Customer Type'},
                 color ='order_amount_($)')

     # Show the chart
     fig.show()
```



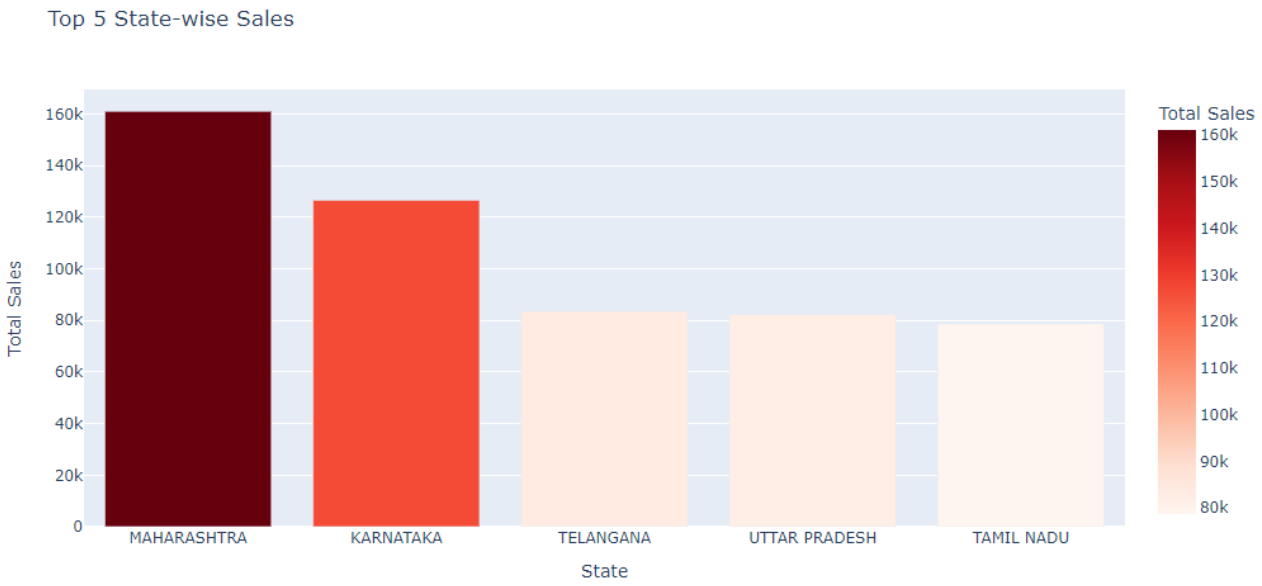
Statewise sale

```
[28] #statewise Sale


     import pandas as pd
     import plotly.express as px

     # Group by 'State' and calculate the total sales for each state
     statewise_sales = df.groupby('ship-state')['order_amount_($)'].sum().reset_index()

     # Create a bar graph using Plotly Express
     fig = px.bar(statewise_sales, x='ship-state', y='order_amount_($)',
                  title='State-wise Sales',
                  labels={'order_amount_($)': 'Total Sales', 'ship-state': 'State'},
                  color='order_amount_($)', color_continuous_scale='reds')

     # Customize the layout and appearance of the chart
     fig.update_layout(xaxis_title='ship-state', yaxis_title='Total Sales')

     # Show the chart
     fig.show()
```



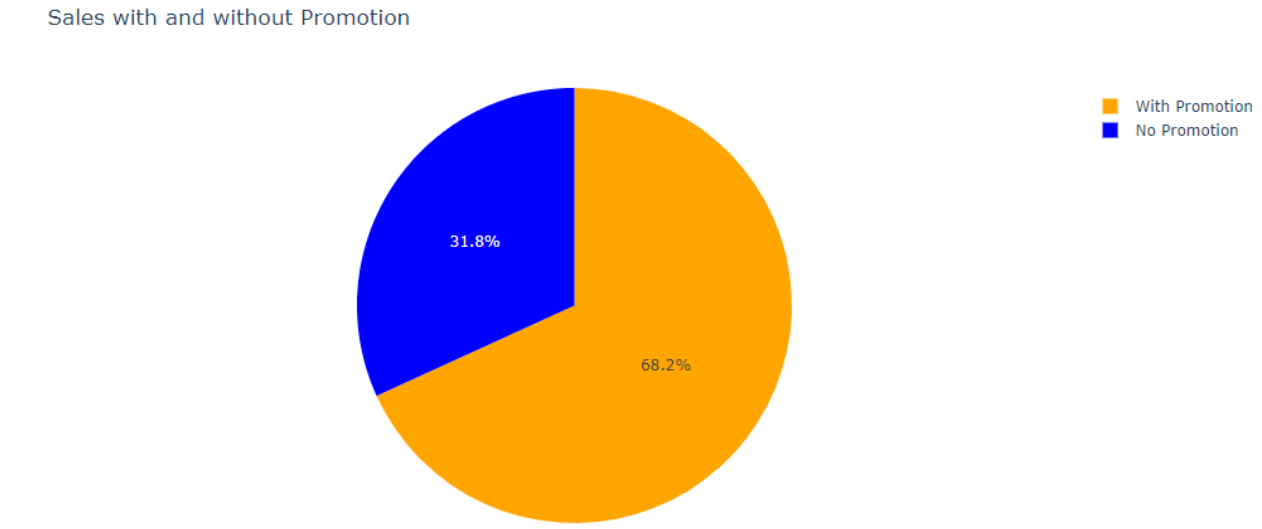
Top 5 state salewise

```
[ ]  #Top 5 State salewise
     import pandas as pd
     import plotly.express as px

     # Assuming statewise_sales DataFrame is already defined

     # Sort the DataFrame by 'order_amount_($)' in descending order
     statewise_sales_sorted = statewise_sales.sort_values(by='order_amount_($)', ascending=False)

     # Select the top 5 states
     top5_states = statewise_sales_sorted.head(5)

     # Create a bar graph using Plotly Express for the top 5 states
     fig = px.bar(top5_states, x='ship-state', y='order_amount_($)',
                  title='Top 5 State-wise Sales',
                  labels={'order_amount_($)': 'Total Sales', 'ship-state': 'State'},
                  color='order_amount_($)', color_continuous_scale='reds')

     # Customize the layout and appearance of the chart
     fig.update_layout(xaxis_title='State', yaxis_title='Total Sales')

     # Show the chart
     fig.show()
```

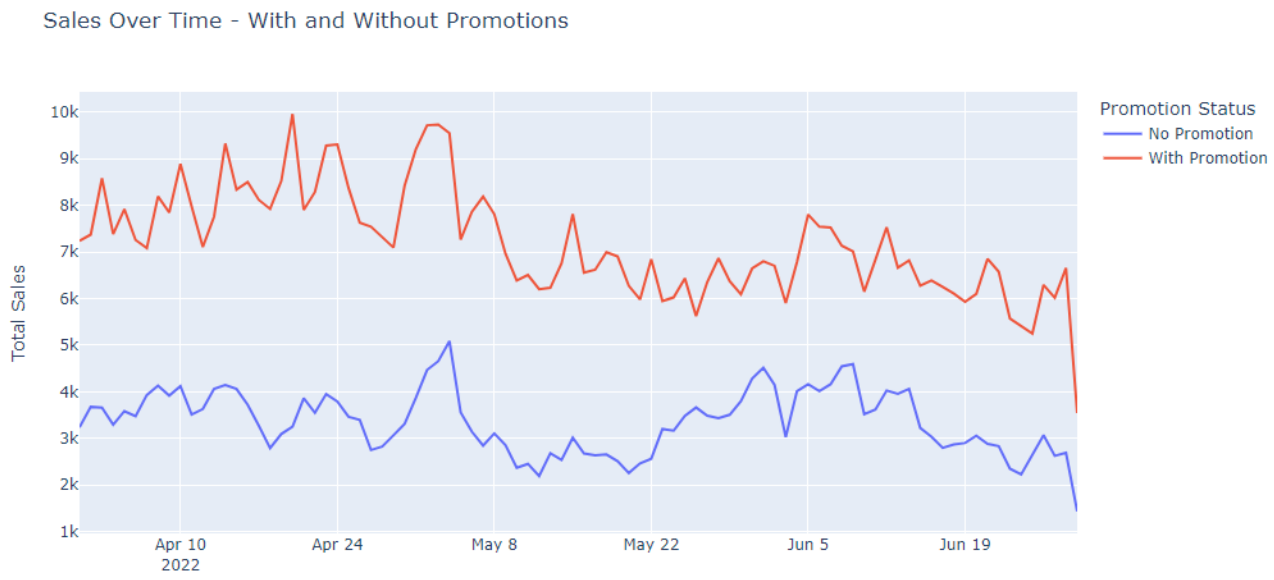


Sell with promotion and without promotion

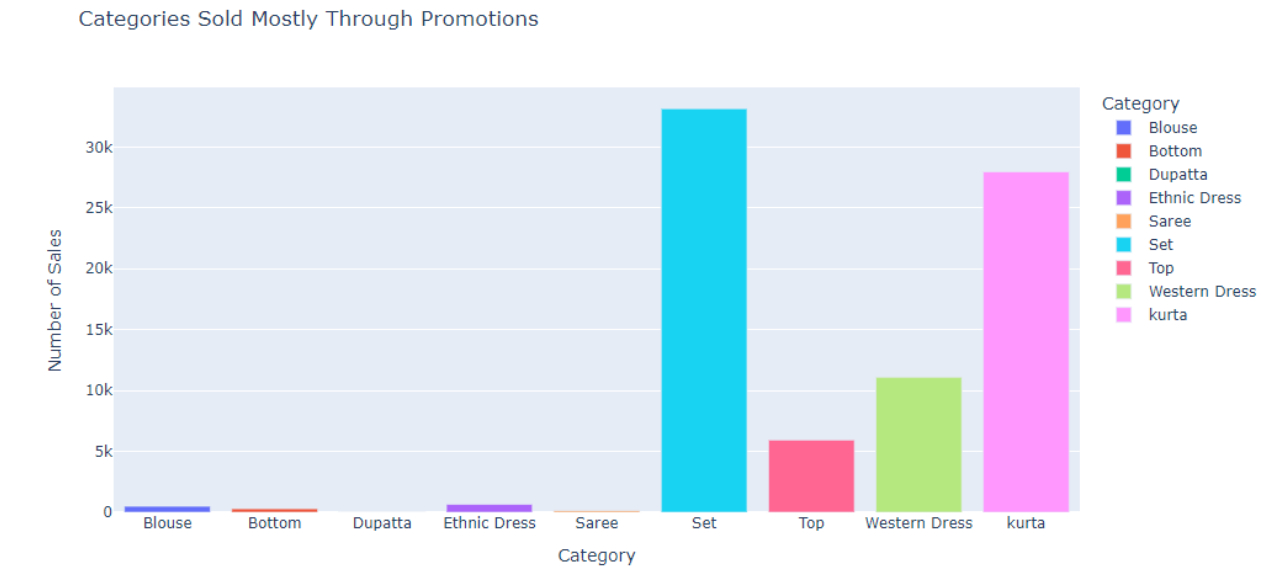
```
[ ] #promotion wise sell
    import pandas as pd
    import plotly.express as px

    # Create a new column indicating whether there is a promotion
    df['Promotion Status'] = df['promotion-ids'].apply(lambda x: 'No Promotion' if x == 'No Promotion' else 'With Promotion')

    # Aggregate data for plotting
    agg_data = df.groupby('Promotion Status')['order_amount_($)'].sum().reset_index()

    # Create a pie chart using Plotly Express
    fig = px.pie(agg_data, values='order_amount_($)', names='Promotion Status',
                 title='Sales with and without Promotion',
                 color='Promotion Status', color_discrete_map={'No Promotion': 'blue', 'With Promotion': 'orange'})

    # Show the chart
    fig.show()
```



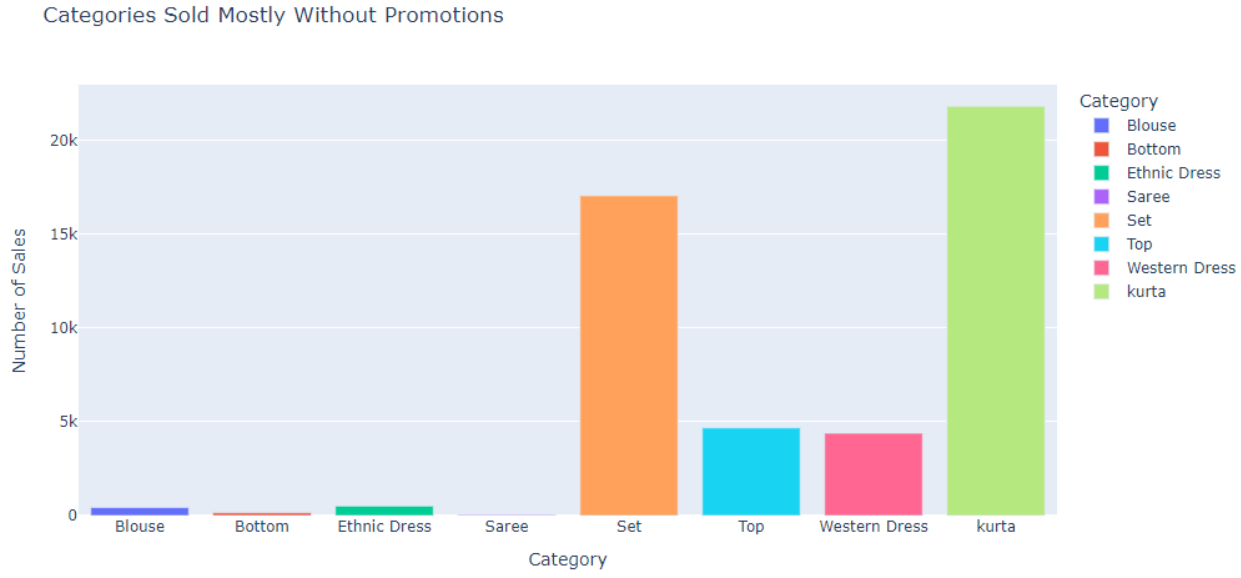
```
[ ] import pandas as pd
    import plotly.express as px


    # Create a new column indicating whether there is a promotion
    df['Promotion Status'] = df['promotion-ids'].apply(lambda x: 'No Promotion' if x=='No Promotion' else 'With Promotion')

    # Group by Date and Promotion Status, and sum the order amounts
    sales_by_date = df.groupby(['Date', 'Promotion Status'])['order_amount_($)'].sum().reset_index()

    # Create a line chart using Plotly Express
    fig = px.line(sales_by_date, x='Date', y='order_amount_($)',
                  color='Promotion Status',
                  title='Sales Over Time - With and Without Promotions',
                  labels={'order_amount_($)': 'Total Sales', 'Date': 'Date'})

    # Show the chart
    fig.show()
```



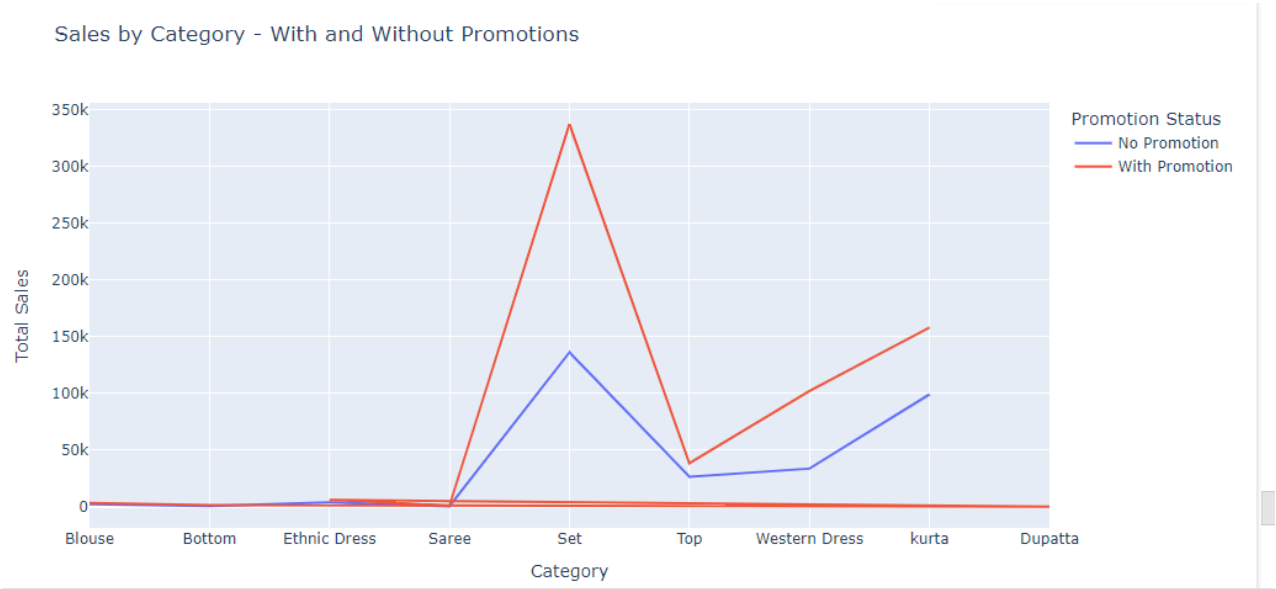

```
[ ] #promotionwise or without promotion sold category
    import pandas as pd
    import plotly.express as px

    # Create a new column indicating whether there is a promotion
    df['Promotion Status'] = df['promotion-ids'].apply(lambda x: 'No Promotion' if x=='No Promotion' else 'With Promotion')

    # Group by Category and Promotion Status, and count the occurrences
    category_promotion_count = df.groupby(['Category', 'Promotion Status']).size().reset_index(name='Count')

    # Filter rows with promotions
    category_promotion_count_with_promotion = category_promotion_count[category_promotion_count['Promotion Status'] == 'With Promotion']

    # Create a bar chart using Plotly Express
    fig = px.bar(category_promotion_count_with_promotion, x='Category', y='Count',
                 title='Categories Sold Mostly Through Promotions',
                 labels={'Count': 'Number of Sales'},
                 color='Category')

    # Show the chart
    fig.show()
```



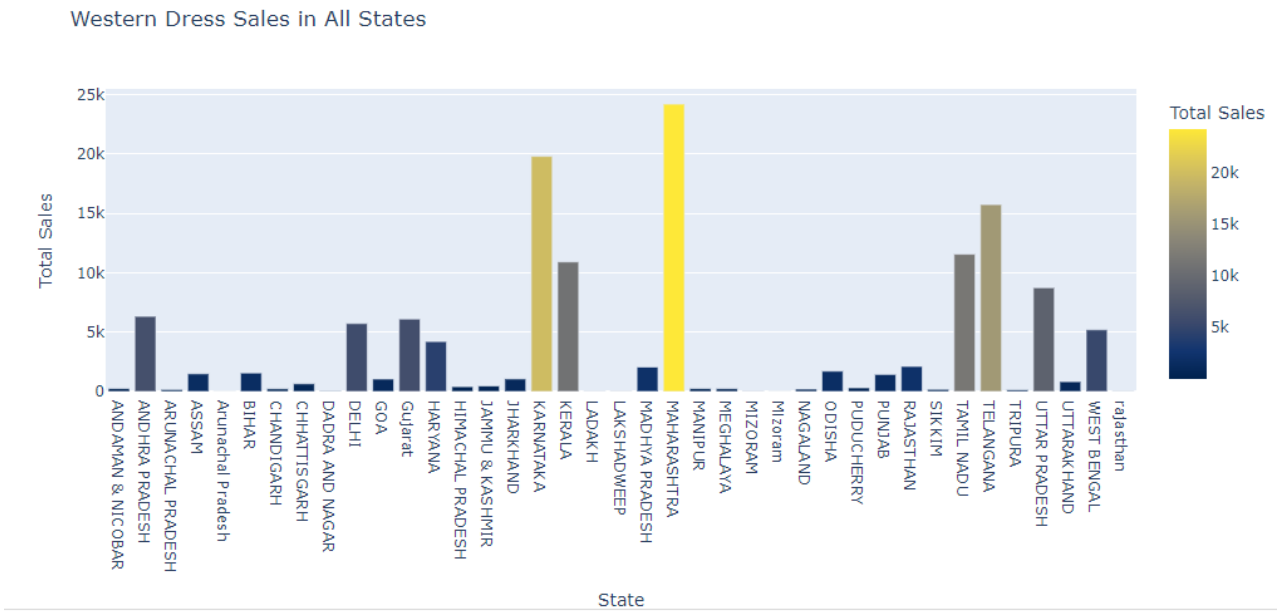

```
[ ]  import pandas as pd
     import plotly.express as px


     # Create a new column indicating whether there is a promotion
     df['Promotion Status'] = df['promotion-ids'].apply(lambda x: 'No Promotion' if x=='No Promotion' else 'With Promotion')

     # Group by Category and Promotion Status, and count the occurrences
     category_promotion_count = df.groupby(['Category', 'Promotion Status']).size().reset_index(name='Count')

     # Filter rows without promotions
     category_promotion_count_no_promotion = category_promotion_count[category_promotion_count['Promotion Status'] == 'No Promotion']

     # Create a bar chart using Plotly Express
     fig = px.bar(category_promotion_count_no_promotion, x='Category', y='Count',
                  title='Categories Sold Mostly Without Promotions',
                  labels={'Count': 'Number of Sales'},
                  color='Category')

     # Show the chart
     fig.show()
```



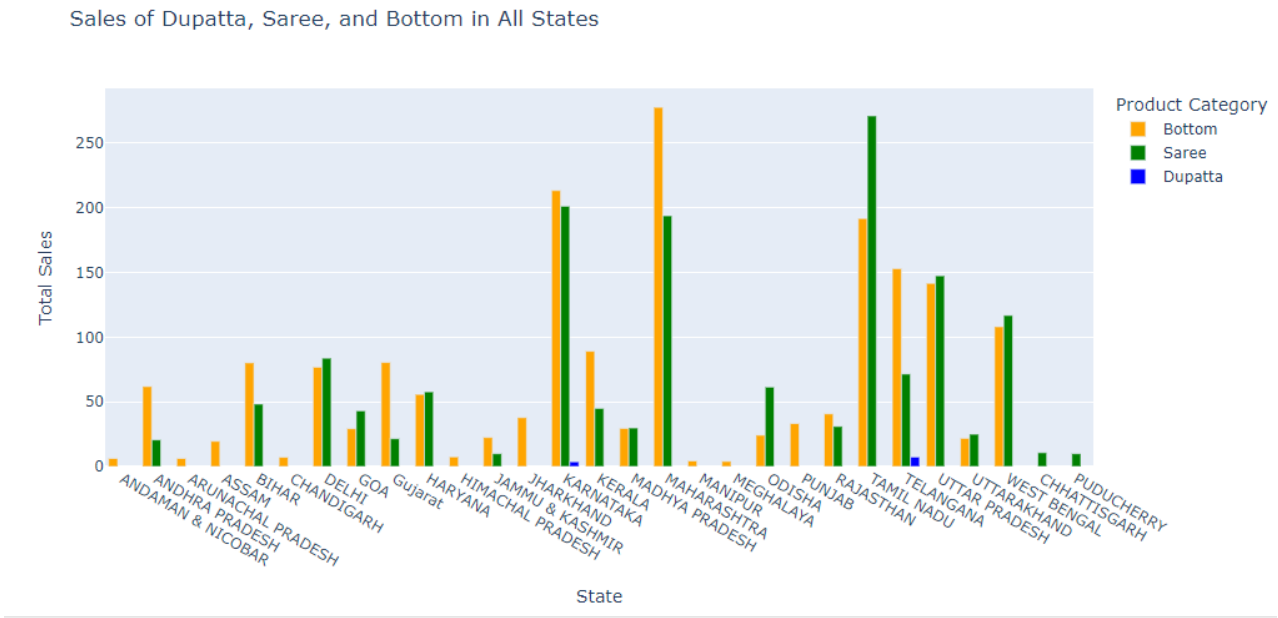
```
[ ]  import pandas as pd
     import plotly.express as px


     # Create a new column indicating whether there is a promotion
     df['Promotion Status'] = df['promotion-ids'].apply(lambda x: 'No Promotion' if x=='No Promotion' else 'With Promotion')

     # Group by Category and Promotion Status, and sum the order amounts
     sales_by_category = df.groupby(['Category', 'Promotion Status'])['order_amount_($)'].sum().reset_index()

     # Create a line chart using Plotly Express
     fig = px.line(sales_by_category, x='Category', y='order_amount_($)',
                   color='Promotion Status',
                   title='Sales by Category - With and Without Promotions',
                   labels={'order_amount_($)': 'Total Sales', 'Category': 'Category'})

     # Show the chart
     fig.show()
```




```
[ ]  #Western Dress sale in all states
     import pandas as pd
     import plotly.express as px

     # Assuming your DataFrame is named 'df'
     # Replace 'Category', 'ship-state', and 'order_amount_($)' with the actual column names in your DataFrame

     # Step 1: Filter data for 'Western Dress' category
     western_dress_sales = df[df['Category'] == 'Western Dress']

     # Step 2: Group by State and Calculate Total Sales
     statewise_western_dress_sales = western_dress_sales.groupby('ship-state')['order_amount_($)'].sum().reset_index()

     # Step 3: Create a Bar Chart for Western Dress Sales in All States
     fig = px.bar(statewise_western_dress_sales, x='ship-state', y='order_amount_($)',
                  title='Western Dress Sales in All States',
                  labels={'order_amount_($)': 'Total Sales', 'ship-state': 'State'},
                  color='order_amount_($)', color_continuous_scale='cividis')

     # Customize the layout and appearance of the chart
     fig.update_layout(xaxis_title='State', yaxis_title='Total Sales')

     # Show the chart
     fig.show()
```




```
[ ]  #Calculating bottom three product sale in all states
     import pandas as pd
     import plotly.express as px

     # Assuming your DataFrame is named 'df'
     # Replace 'Category', 'ship-state', and 'order_amount_($)' with the actual column names in your DataFrame

     # Step 1: Filter data for 'Dupatta,' 'Saree,' and 'Bottom' categories
     selected_categories = ['Dupatta', 'Saree', 'Bottom']
     selected_category_sales = df[df['Category'].isin(selected_categories)]

     # Step 2: Group by State and Calculate Total Sales for selected categories
     statewise_selected_category_sales = selected_category_sales.groupby(['ship-state', 'Category'])['order_amount_($)'].sum().reset_inde

     # Step 3: Create a Bar Chart for Sales of selected categories in All States
     fig = px.bar(statewise_selected_category_sales, x='ship-state', y='order_amount_($)',
                  color='Category', barmode='group',
                  title='Sales of Dupatta, Saree, and Bottom in All States',
                  labels={'order_amount_($)': 'Total Sales', 'ship-state': 'State', 'Category': 'Product Category'},
                  color_discrete_map={'Dupatta': 'blue', 'Saree': 'green', 'Bottom': 'orange'})

     # Customize the layout and appearance of the chart
     fig.update_layout(xaxis_title='State', yaxis_title='Total Sales')

     # Show the chart
     fig.show()
```



Insight:

1.Top 3 Categories of products by sales are Set=473634,$which\ is$ 49.88 ,which is 27.09% and Western Dresses=135606$ contribute 14.28%

2..Bottom 3 Categories of products by sales are Dupatta=11.07,$Saree$=1500 ,Bottom=1823$

3.Monthwise Revenue April=348904,May=317284 ,June=283491$, it shows that sales has been falling monthwise

4.The top 5 product categories by average price in dollars are Set (9.43),$Saree$($9.14),$WesternDress$(8.75), Ethnic Dress (8.26),$andTop$(6.09), indicating that these products are high-value orders

5.The average order amount by customer type is 8.21$forbusinesscustomersand$ 7.37 for regular customers

6.Top 5 states salewise are Maharashtra=161128,$Karnataka$=126640 ,Telangana=83631,$Utterpradesh$82331 , Tamilnadu=78719

promotion wise sales contribute 68.2% while without promotion contribute31.8%
8.Western Dress has highest sale in maharashtra=24209,$Karnataka$=19802 ,Telangana=15743,$Tamilnadu$=11562 ,Kerala=10914$

9.Bottom three categories have great responce in maharashtra, Tamilnadu, Karnataka, Telangana and Uttarpradesh

Conclusion From the above insights we can conclude to increase the sales of western dresses we need to do more promotion in Mahrashtra, Karnataka, Telangana, Tamilnadu and Kerala and for bottom 3 product we need to do promotion in maharashtra, Tamilnadu, Karnataka, Telangana and Uttarpradesh

Summary of Analysis:

• The top 3 product categories are key drivers of revenue, with Western Dress identified as a focus for future growth.

• Business customers show higher spending potential, making them a critical segment for revenue growth through specialized promotions.

• State-wise sales analysis indicates that focusing on the top-performing states will help maximize the effectiveness of promotional efforts.

• Reducing the cancellation and returning rates can obviously lead to increase the revenue.

• During the analysis of the dataset it was found that western dress is playing an important role here because we have been given that we have to increase the revenue of the bottom first product category from the top three revenue generating product category by 5% by the end of Q3 2022. Also, it is given to us that we have to promote western dress which is a plus point here.

• A future-based analysis shows that applying promotions on product categories and states can lead to the increment of revenue. As this analysis is hypothetical so the real analysis depends on the sales data of Q3 2022.

Promotional Strategies for western dress:

Some promotional can be applied in order to increase the sales revenue are listed below:

1. Targeted Discounts and Offers

• Flash Sales: Offer limited-time discounts on Western Dresses to create urgency and encourage quick purchases.

• Bundle Deals: Pair Western Dresses with complementary products (e.g., accessories) at a discounted rate to increase average order value.

• Buy and Get: Offers like Buy 1 Get 1 free can be applied to the products in order to increase the order values and sales.

2. Seasonal Promotions

• Festive Sales: Align promotions with upcoming festivals or holidays, showcasing Western Dresses as ideal choices for celebrations.

8788385365

• End-of-Season Sales: Clear out inventory with significant discounts on seasonal Western Dresses to make room for new styles.

3. Free Shipping Promotions

• Threshold for Free Shipping: Offer free shipping on orders above a certain amount, encouraging customers to purchase more items, including Western Dresses.

4. In-Store Events

• Try-On Events: Host events where customers can try on Western Dresses, receive styling advice, and enjoy refreshments, creating a fun shopping experience.

5. Feedback and Improvement

• Customer Surveys: Gather feedback from customers who purchased Western Dresses to understand their preferences and improve future offerings.

Implementing a combination of these strategies will help increase awareness, engagement, and sales

for Western Dresses. It’s essential to monitor the performance of each strategy and adapt based on

customer responses and sales data.

KPI and Revenue Growth:

Key Performance Indicators (KPIs)

1. Revenue Growth Rate

• Measures the percentage increase in revenue over a specific period.

• Target: Aim for a 15% increase in revenue for the Western Dress category over the next quarter. Not only western dresses but other products can also be included for promotional purpose in order to gain revenue increase.

2. Customer Acquisition Cost (CAC)

• The total cost of acquiring a new customer, including marketing and advertising expenses.

• Target: Keep CAC below $30 per new customer acquired.

3. Customer Retention Rate

• The percentage of customers who continue to purchase over a specific period.

• Target: Achieve a retention rate of 70% for customers purchasing Western Dresses.

4. Average Order Value (AOV)

• The average amount spent by customers in a single transaction.

• Target: Increase the AOV by 10% through upselling and cross-selling strategies.

5. Conversion Rate

8788385365

• The percentage of website visitors who make a purchase.

• Target: Achieve a conversion rate of 3% on the product pages for Western Dresses.

6. Sales Volume

• The total number of units sold within a specific timeframe.

• Target: Increase sales volume of Western Dresses by 20% in the upcoming promotional campaign.

7. Customer Feedback and Satisfaction

• Measure customer satisfaction through surveys and reviews.

• Target: Maintain an average customer satisfaction score of 4.5/5 for Western Dresses.

Revenue Growth Strategies:

1. Increase Sales from Promotions: Implement targeted promotions aimed at the Western Dress category to stimulate purchases and drive revenue growth.

2. Enhance Online Presence: Optimize the website for better visibility, ensuring a smooth shopping experience to increase conversions and boost revenue.

3. Leverage Social Media Advertising: Use targeted ads to reach potential customers who may be interested in Western Dresses, driving more traffic and sales.

4. Expand Product Range: Introduce new styles or variations of Western Dresses based on market trends and customer feedback to attract more buyers.

5. Improve Customer Engagement: Foster relationships with customers through personalized communication and engagement, encouraging repeat purchases.

6. Utilize Data Analytics: Analyse sales data to identify high-performing products and customer preferences, allowing for informed decisions on inventory and promotions.

7. Implement Referral Programs: Encourage satisfied customers to refer friends and family, offering discounts or incentives for successful referrals to drive new customer acquisition.

By monitoring these KPIs and implementing effective revenue growth strategies, you can enhance the performance of the Western Dress category and drive overall business growth.

Conclusions and Recommendations:

Conclusions

1. Revenue Performance: The Western Dress category has shown potential for growth, contributing 14% to the overall revenue. Targeted promotions can significantly enhance its performance.

2. Top Categories: The top three revenue-generating categories—Set, Kurta, and Western Dress—highlight the importance of leveraging popular products to drive sales.

8788385365

3. Promotional Impact: Initial analyses indicate that promotions can lead to a considerable increase in sales volume and revenue, as seen in the post-promotion revenue strategies.

4. Customer Engagement: High customer retention rates and average order values suggest that customers are willing to spend more on appealing products, indicating effective marketing strategies.

Recommendations:

1. Implement Targeted Promotions: Utilize discounts, bundle offers, and seasonal promotions focused on the Western Dress category to stimulate sales and increase market share.

2. Enhance Marketing Strategies: Invest in social media marketing and influencer collaborations to reach a wider audience, showcasing the appeal and versatility of Western Dresses. Other underperforming categories can also benefit from similar promotional efforts.

3. Expand Product Range: Introduce new styles and variations of Western Dresses based on customer preferences and feedback, ensuring the product line remains fresh and appealing.

4. Leverage Data Analytics: Continuously analyse sales data and customer feedback to adjust marketing strategies and inventory management, ensuring alignment with market trends.

5. Improve Customer Experience: Enhance the online shopping experience with user-friendly navigation, detailed product descriptions, and high-quality images to boost conversion rates.

6. Foster Customer Loyalty: Develop loyalty programs and referral incentives to encourage repeat purchases and attract new customers through existing satisfied customers.

7. Monitor and Adjust Strategies: Regularly assess the effectiveness of promotional strategies through KPIs and adjust approaches based on data insights to optimize performance.

8. Utilize Feedback for Improvement: Collect customer feedback on Western Dresses to identify areas for improvement in product quality, pricing, and overall shopping experience.

Hence, by implementing these recommendations, the business can capitalize on growth opportunities

in the Western Dress category, enhance overall revenue performance, and build stronger customer

relationships. Continuous monitoring and adaptation will be key to sustaining success in a competitive

market.