***Superstore Sales Case Study:***

***Overview*** : This case study involves performing an Exploratory Data Analysis (EDA) on the Superstore Sales dataset, which is widely used for educational purposes. The dataset provides a diverse set of features, including categorical, numerical, and date-time data, making it ideal for practicing data cleaning, feature engineering, and visualization techniques.

The goal of this case study is to apply systematic EDA techniques to uncover insights, identify trends, and prepare the data for further analysis or modeling. The dataset has been modified to introduce various inconsistencies, missing values, and data quality issues, offering an opportunity to apply advanced data cleaning and transformation skills.

Column Names and Descriptions:

**Row** **ID**: Unique identifier for each row.

**Order** **ID**: Identifier for each order, includes the year of the order.

**Order** **Date**: Date when the order was placed.

**Ship** **Date**: Date when the order was shipped.

**Ship** **Mode**: Mode of shipping (e.g., Standard Class, First Class).

**Customer** **ID**: Unique identifier for each customer.

**Customer** **Name**: Full name of the customer (to be masked).

**Segment**: The market segment of the customer (e.g., Consumer. Corporate)

**Country**: Country where the order was placed.

**City**: City where the order was placed.

**State**: State where the order was placed..

**Postal** **Code**: Postal code of the shipping address.

**Region**: Geographic region of the shipping address.

**Product** **ID:** Unique identifier for each product.

**Category**: Product category (e.g. Furniture, Office Supplies).

**Sub-Category**: Product sub-category (e.g., Chairs, Storage).

**Product** **Name**: Name of the product

**Sales** **Price**: The final price at which the product was sold after applying any discounts

**Quantity**: Number of units sold.

**Discount**. The discount applied to the original price of the product.

**Profit**: Profit earned per quantity sola

EDA Steps and Requirements:

**1. Data Loading and Initial Exploration:**

1. Load the dataset into a pandas DataFrame.

2. Explore the first few rows to understand the structure of the data.

3. Check the data types, summary statistics, and unique values of each column.

4. Identify any obvious data quality issues or inconsistencies.

**2. Handling Duplicates:**

1. Identify and remove duplicate rows in the dataset.

2. Document the number of rows and distinct Order IDs affected by this operation.

**3. Date Handling:**

1. Normalize the Order Date and Ship Date columns to ensure consistent date formats. Ensure that the format of date is consistent across all rows.

2. Extract the year from the Order ID and compare it with the year in Order Date. Correct any inconsistencies.

3. Document the number of rows and distinct Order IDs affected by these operations.

**4. Imputation of Missing Values:**

1. Impute missing values in the Ship Mode column using the calculated Days to Ship column.

2. Calculate Days to Ship as the difference between Ship Date and Order Date. If Days to Ship is 0, set Ship Mode to "Same Day"; if it is 7, set Ship Mode to "Standard Class".

3. Impute missing values in the Quantity column using a method of your choice. Print the rationale for selecting the method for imputation.

4. Document the number of rows and distinct Order IDs affected by these operations.

**5. Data Masking and String Handling:**

1. Drop the Customer Name column to protect Personal Identifiable Information (PII).

2. Create a new column called Customer Name Masked, containing only the initials of the customer name.

(Note : It's important to protect Pil in datasets to maintain customer privacy and comply with data protection regulations. Masking or dropping sensitive data like customer names is a crucial step in this process.)

3. Convert the Postal Code column from numeric to text format, ensuring all codes are 5 characters long. Add a leading '0' where necessary.

**6. Data Type Conversion:**

1. Convert the Quantity and Sales Price columns from strings to their appropriate numeric types (int and float, respectively).

**7. Handling Inconsistent Categorical Data:**

1. Clean the State column by replacing abbreviations with full state names (e.g.. "CA" should be changed to "California"). You may need to research state abbreviations online to ensure all entries are corrected consistently.

**8. Feature Engineering:**

1. Create new columns.

1. Original Price: The price before any discount is applied.

2. Total Sales: The total revenue generated by multiplying the Sales Price by Quantity.

3. Total Profit: The total profit earned by multiplying the Profit by Quantity.

4. Discount Price: The amount of discount applied, calculated based on the Original Price and Discount.

5. Total Discount: The total discount value for the quantity sold.

2. Create a new column Shipping Urgency based on Days to Ship:

1. If Days to Ship is 0, set to "Immediate".

2. If Days to Ship is between 1 and 3, set to "Urgent".

3. If Days to Ship is more than 3, set to "Standard".

3. Create a column that calculates days since last order.

4. Create a new dataset which stores the total sales, quantity and discount per customer and then merge these back to the original dataset

**9. Outlier Detection and Handling:**

1. Identify and handle outliers in the Sales Price column

2. Create a function called remove\_outliers which takes as parameter the dataframe and the column which needs to be searched for outliers. Using the 3 IQR rule, the function should detect and remove the outliers to return the cleaned dataframe

**Why 3\*IQR?**

The 3 IQR method is applied in situations where the dataset has a high variance, and the standard 1.5 IQR might flag too many points as outliers. This method ensures that only the most extreme values are removed, preserving the integrity of the dataset while still mitigating the influence of true outliers

**10. Customer Segmentation and Analysis:**

1. Calculate Customer Sales Quintile and Customer Profit Quintile based on total sales and total profit per Customer ID.

2. What is a Quintile? Quintiles are a statistical way of dividing data into five equal parts, each representing 20% of the data. For example, customers in the top quintile (Q5) represent the top 20% of sales or profit.

3. Create a cross-grid (cross-tabulation) based on these two quintiles to analyze the relationship between customer sales and profitability.

**11. Final Analysis and Dashboard Creation:**

1. Sales and Profit Analysis:

1. Top 10 Most Profitable Products: Use a bar chart to display the products with the highest total profit.

2. Top 10 Most Loss-Making Products: Use a bar chart to display the products with the highest total losses (negative profit).

3. Sales vs. Profit Correlation. Use a scatter plot to visualize the correlation between Total Sales and Tatal Profit. Add a regression line to show the trend.

4. Joint Distribution of Sales and Profit. Create a joint plot to analyze the relationship between Total Sales and Total Profit across different products.

2. Customer Segmentation and Analysis:

1. Customer Sales Quintile vs. Customer Profit Quintile: Create a heatmap or cross-tabulation to explore the relationship between customers' sales and profit quintiles. This will help identify which segments of customers are most valuable.

2. Understand how different product categories perform across customer segments: Create a pivot table to analyze the total Sales and total Profit by Category and Segment. Sort the pivot table to highlight the most profitable and least profitable combinations of Category and Segment.

**3. Shipping and Delivery Analysis:**

1. Distribution of Shipping Urgency: Visualize the distribution of orders by Shipping Urgency using a pie chart or bar chart.

2. Days to Ship vs. Profit. Use a violin plot to explore the distribution of Profit across different Days to Ship categories. This will help analyze whether faster shipping correlates with higher or lower profitability.

3. Shipping Mode and Profitability: Create a grouped bar chart to compare the profitability of different shipping modes (e.g., Standard Class, First Class).

4. Using pivot table, determine which shipping modes are most preferred across different regions and analyze the impact on total sales and profit. Create a pivot table that shows the count of Order IDs, total Sales, and total Profit for each Region and Ship Mode. Identify and print your insights.

**4. Regional Sales and Profitability:**

1. Sales and Profit by Region: Use a map or bar chart to visualize total sales and profit by region or state. This will highlight which regions are the most profitable.

2. State-wise Profitability: Create a pivot table to summarize the profitability of each state. Highlight the top and bottom states based on profitability.

3. Correlation between State and Profit: Use a correlation plot to identify any patterns or relationships between the states and the profitability of orders.

(Hint: Convert the categorical 'State' column into numerical values using label encoding using "from sklearn.preprocessing import LabelEncoder")

**5. Discount and Pricing Analysis:**

1. Impact of Discounts on Profitability. Use a scatter plot with a trend line to analyze how different levels of discount affect profitability.

2. Original Price vs. Discounted Price: Create a line plot to compare the original price and the discounted price across various product categories or sub-categories.

**6. Temporal Analysis:**

1. Sales and Profit Trends Over Time: Use a time series plot to analyze how sales and profit have trended over the years or months. This will help in identifying any seasonal patterns.

2. Order Frequency by Month: Use a bar chart or line plot to show the number of orders placed each month. Highlight any months with unusually high or low order frequencies

3. Yearly Growth in Sales and Profit: Use a year-over-year growth chart to compare the sales and profit growth over different years.