## **CRM Data Engineering and Analysis Pipeline**

## **Table of Contents**

|  |
| --- |
| [**Chapter - 1 Introduction 2**](#_kcsvq4u9466r)  [CRM Data Engineering and Analysis Pipeline 2](#_x4dhnybb6bl)  [1.1 Purpose of the Project 3](#_wfxlqpg8inj6)  [1.2 Main Objectives 3](#_k8qibopvixlh)  [**Chapter - 2 Methodology and Architecture Diagram 4**](#_piiazzsoooh8)  [2.1 Entity Relationship Diagram 4](#_qgg13fs2nxvy)  [2.2 Architecture Diagram 5](#_weu2rrpyneth)  [**Chapter - 3 Objectives and Problem Statements 9**](#_8cwcm3s94tr1)  [Problem Statement 3.1: Ensuring Data Accuracy and Data Cleaning 9](#_azndy2p1503s)  [Approach: 9](#_4jq9hmpmu8c9)  [3.1.1 Approach for Cleaning Customers Table 10](#_h54rmwq3ewgp)  [3.1.2 Approach for Cleaning Interactions Table 11](#_1egwy28fnz77)  [3.1.3 Approach for Cleaning Products Table 12](#_8koypgrss5ge)  [3.1.4 Approach for Cleaning Sales Team Table 13](#_jni5gxgnlw3j)  [3.1.5 Approach for Cleaning Transactions Table 14](#_qoey7gkizbgr)  [Problem Statement 3.2: Customer Purchase Behaviour Analysis 15](#_16x4nwwf5p9p)  [3.2.1 Approach and Outcome 15](#_75lta48jn5sd)  [Problem Statement 3.3: Product Sales Performance Evaluation 21](#_bs6i6gdjiz6b)  [3.3.1 Approach and Outcome 21](#_9cw9asxk508i)  [Problem Statement 3.4: Customer Interaction Effectiveness 27](#_uj9q8x5vc6r9)  [3.4.1 Approach and Outcome 27](#_jeuqboaxp60l)  [Problem Statement 3.5: Sales Team Performance Assessment 31](#_18eiriuyiblj)  [3.5.1 Approach and Outcome 31](#_5jmmtxg6yi0h)  [**Chapter - 4 Conclusion 35**](#_r6mzcosy0fhi) |

# 

# **Chapter - 1 Introduction**

## **CRM Data Engineering and Analysis Pipeline**

This project focuses on implementing a comprehensive data cleaning and preprocessing workflow for multiple datasets related to sales and customer interactions. The datasets include customers, interactions, products, sales teams, and transactions. The goal of the project is to ensure that the data used for further analysis is accurate, consistent, and free from anomalies that could skew insights or decision-making.

The workflow begins with identifying and handling missing values across the datasets. For numeric fields, such as sales or transaction amounts, missing values are typically filled with calculated averages to maintain the integrity of the data. Text fields, such as product names or customer details, are standardized to ensure consistency in capitalization and formatting. This step is crucial for maintaining uniformity, especially when the data comes from multiple sources or has been manually entered.

Duplicates are another common issue in data processing. For each dataset, duplicates are identified and removed based on unique identifiers like Customer\_ID, Transaction\_ID, or Sales\_Rep\_ID. This step prevents any inflated figures that could arise from multiple records representing the same data point.

Special attention is given to date and boolean fields, ensuring that date formats are validated and corrected to follow a standardized format, such as YYYY-MM-DD. This guarantees compatibility with analytical tools and facilitates time-based analyses like trends or customer behavior patterns. Similarly, boolean fields such as issue resolution statuses are validated to ensure they only contain appropriate true or false values.

Anomalies, such as negative values in sales amounts or zero prices in product data, are addressed by replacing these incorrect values with appropriate measures like average values or predefined defaults. This step helps maintain the accuracy of financial and performance metrics.

Once the data cleaning process is complete, the cleaned datasets are saved for further analysis. This ensures that the data is not only error-free but also aligned with best practices for data management. By following this structured approach, the project lays a strong foundation for reliable and insightful data analysis, which is critical for supporting strategic decision-making in sales performance evaluation and customer relationship management.

## **1.1 Purpose of the Project**

The primary goal of this project is to leverage CRM data to gain deep insights into **customer behaviors**, **product performance**, **sales trends**, and **team effectiveness**. By systematically analyzing data from five key domains—customers, products, transactions, interactions, and the sales team—the project seeks to uncover patterns and trends that will help optimize operations and improve customer relationships.

## **1.2 Main Objectives**

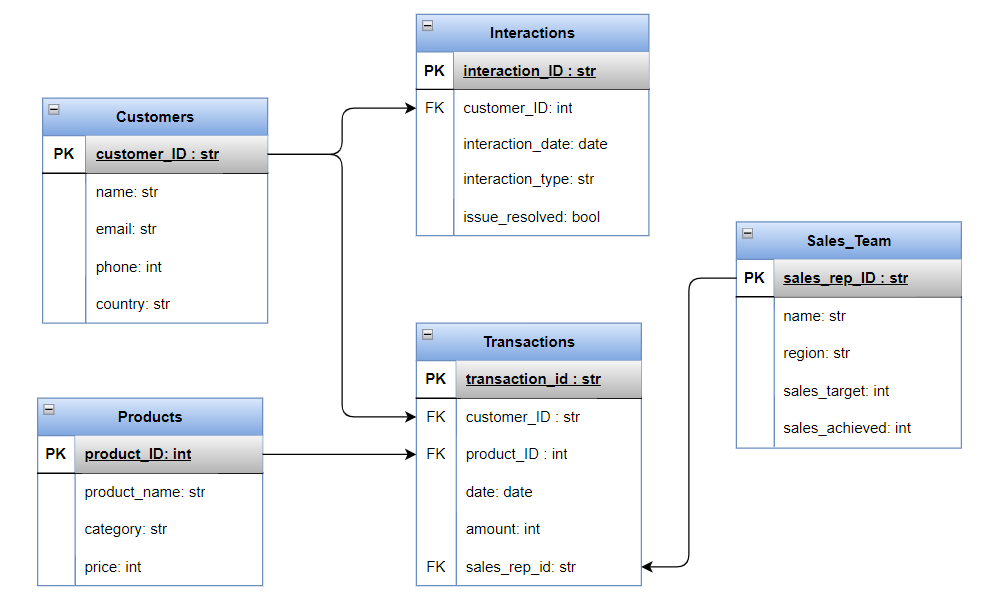
1. **Ensure Data Accuracy and Integrity**: Validate and ensure that CRM data is accurate, consistent, and correctly linked across multiple datasets. This is a foundational step to ensure reliable analysis and informed decision-making.
2. **Customer Purchase Behavior Analysis**: Examine customer interactions with products and services to understand **spending patterns**, **purchase frequency**, and **geographic influences**. This will help in identifying high-value customer segments and tailoring marketing efforts.
3. **Product Sales Performance Evaluation**: Assess the performance of various products and categories, focusing on which products drive revenue and how **pricing strategies** impact sales. This analysis helps in identifying top-performing items and optimizing product offerings.
4. **Customer Interaction Effectiveness**: Evaluate the success of various customer interaction methods (e.g., email, chat) in resolving issues and improving satisfaction. This will guide strategies to enhance customer support through effective communication channels.
5. **Sales Team Performance Assessment**: Analyze sales team performance to identify high performers and those who require additional support. This helps in **target setting**, **recognizing top talent**, and addressing **performance gaps**.

# **Chapter - 2 Methodology and Architecture Diagram**

## **2.1 Entity Relationship Diagram**

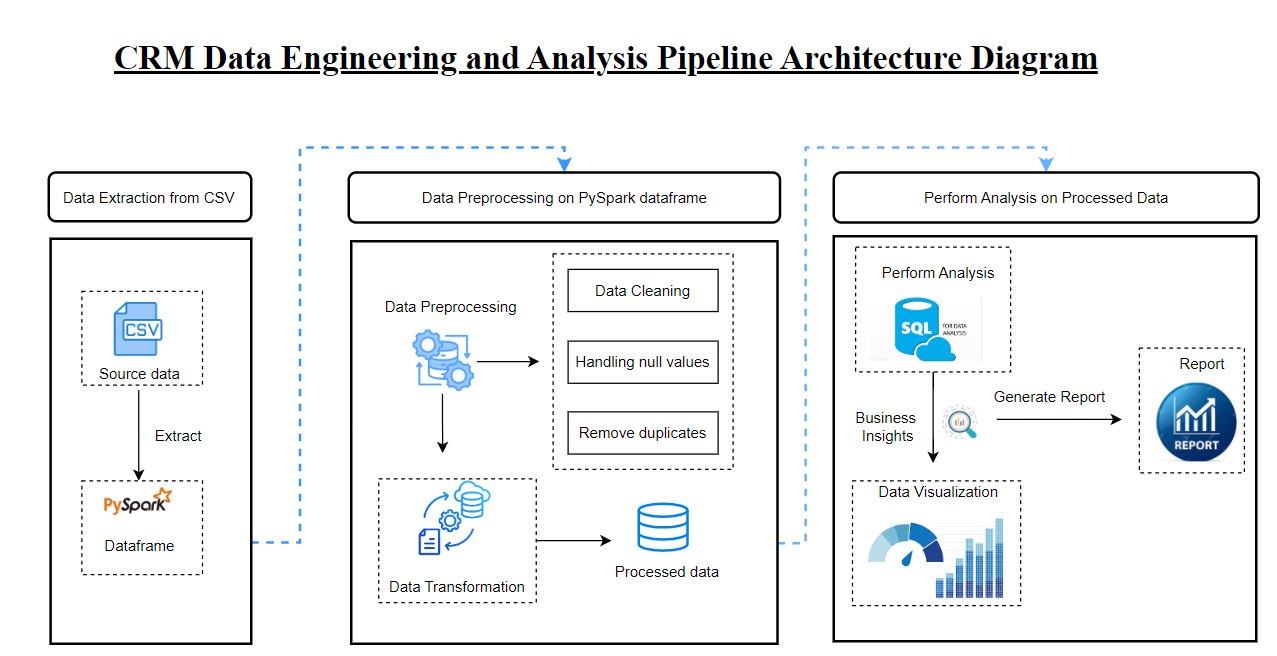
This ER (Entity-Relationship) diagram represents the relationships between five primary entities in a customer and sales management system: Customers, Transactions, Products, Sales Team, and Interactions.

* Customers can have multiple Transactions and Interactions, indicating the one-to-many relationships between Customers and these entities.
* Transactions link Customers, Products, and the Sales Team, meaning that each transaction involves a customer, a product, and a sales representative.
* Interactions are associated with Customers to track any issues or communications, allowing the system to record customer service or support interactions.

****

**Fig 2.1 Entity Relationship Diagram**

## **2.2 Architecture Diagram**



**Fig 2.2** **Architechture Diagram**

**1. Data Extraction from CSV**

This stage involves extracting raw data from CSV files, which is often the standard format for structured datasets in CRM systems.

* **Source Data (CSV)**:
  + The raw CRM data is typically stored in CSV format. This format is widely used due to its simplicity and compatibility with various data-processing systems.
* **Data Extraction**:
  + The process of extracting data from these CSV files is facilitated by **PySpark**, a powerful and efficient distributed computing framework. PySpark is employed here due to its ability to handle large-scale data and parallel processing.
  + Once the data is extracted, it is loaded into a **PySpark DataFrame**, which is optimized for distributed computing and further transformation operations.

**2. Data Preprocessing on PySpark DataFrame (Middle Section)**

After extraction, the raw data undergoes various preprocessing steps to clean and transform it, making it suitable for analysis. This step is crucial to ensure the quality of the data.

* **Data Preprocessing**:
  + This block includes key steps such as **data cleaning**, **handling null values**, and **removing duplicates**.
  + These tasks are performed to ensure the integrity of the data, which may contain missing, inconsistent, or duplicated entries.
* **Data Cleaning**:
  + This involves filtering out or correcting erroneous or irrelevant data that could lead to inaccurate analysis results.
* **Handling Null Values**:
  + Missing data is handled by either removing null entries or imputing them with relevant default values, depending on the business use case.
* **Remove Duplicates**:
  + Duplicates in the dataset are identified and removed to avoid skewing results and ensure the analysis is based on unique data points.
* **Data Transformation**:
  + After preprocessing, the data is transformed into a format that is easier to query and analyze. This step might involve changing data types, aggregating data, or applying business-specific transformations.
  + The processed data is now stored in a transformed and structured format, which will be used in the next stage of analysis.

**3. Perform Analysis on Processed Data (Right Section)**

This section covers the final stage of the pipeline, where the processed data is analyzed to extract business insights, generate reports, and visualize the results.

* **Perform Analysis**:
  + The processed data is analyzed using **SQL**, a widely-used querying language that enables efficient data retrieval and manipulation. SQL is employed here for running queries, performing aggregations, and generating key performance indicators (KPIs).
* **Business Insights**:
  + The results of the SQL analysis help in extracting actionable business insights. These insights can be used to inform decision-making processes, optimize customer strategies, or identify trends within the CRM data.
* **Data Visualization**:
  + Data visualization techniques are applied to represent the analytical outcomes in a graphical format, such as charts or dashboards. This helps in conveying complex data patterns to stakeholders in an easily understandable manner.
* **Generate Report**:
  + Based on the insights and visualizations, comprehensive reports are generated. These reports summarize key findings, offer data-driven recommendations, and may include visual elements like graphs, charts, and tables.
  + The reports provide crucial insights for business stakeholders, allowing them to take informed actions based on the CRM data analysis.

# 

# 

# **Chapter - 3 Objectives and Problem Statements**

## **Problem Statement 3.1: Ensuring Data Accuracy and Data Cleaning**

**Objective:** This problem addresses the challenge of ensuring data consistency across multiple datasets in a CRM environment, specifically focusing on validating key IDs (e.g., Customer\_ID, Product\_ID, Sales\_Rep\_ID) between the datasets.

### **Approach:**

1. **Data Loading**The first step involves loading multiple CSV files (customers, products, transactions, interactions, and sales team) into **PySpark DataFrames** using the load\_data\_files() function. This allows efficient handling of large datasets.
2. **Initial Data Display**The display\_dataframes() function is used to show the first few records from each dataset for an initial inspection and to log the data structure.
3. **Validation Logic**
   * The **validate\_ids()** function checks for the presence of corresponding IDs between two datasets.
   * It uses a **left-anti join** in PySpark to find missing records in one dataset that should match another.
   * The function prints any missing IDs in **green** for emphasis and logs the results.
4. **Validation Process**
   * The validate\_data() function iterates over the predefined validation rules, which compare key IDs between the datasets.
   * The function validates:
     + **Customer\_IDs** between transactions and customers, as well as interactions and customers.
     + **Product\_IDs** between transactions and products.
     + **Sales\_Rep\_IDs** between transactions and the sales team.
5. **Logging and Reporting**The process includes comprehensive logging of the validation checks and their results, ensuring visibility into data integrity and highlighting missing records.

**6. Data Cleaning**

#### **3.1.1 Approach for Cleaning Customers Table**

1. **Count Missing Values Before Filling:**
   * Counted the number of missing (null) values in each column before any data imputation to identify columns with incomplete data.
2. **Check for Duplicates:**
   * Identified duplicate entries in each column to understand the extent of redundancy before any cleaning.
3. **Format and Clean Phone Numbers:**
   * Processed and standardized phone numbers using an external dataset for country-specific formatting.
4. **Remove Duplicate Records:**
   * Dropped duplicate records based on the Customer\_ID column to ensure each customer is uniquely represented.
5. **Format and Clean Email Addresses:**
   * Validated and standardized email addresses to correct any formatting issues.
6. **Fill Missing Values:**
   * Imputed missing values in the Email and Phone columns with default values ('unknown') to handle incomplete data.
7. **Capitalize Names and Country:**
   * Capitalized the first letter of each word in the Name and Country columns to ensure consistent formatting.
8. **Cross-Validation for Missing Values:**
   * Recounted missing values after filling them to verify that all intended imputation steps were successfully applied.
9. **Check for Duplicates Post-Cleaning:**
   * Rechecked for duplicates in each column after cleaning to ensure that the data is free of redundancy.
10. **Export Cleaned Data:**
    * Exported the cleaned DataFrame to a CSV file for further use and analysis.
11. **Record Count After Cleaning:**
    * Logged the number of records in the cleaned DataFrame to confirm that the cleaning process maintained data integrity.

#### **3.1.2 Approach for Cleaning Interactions Table**

1. **Identify Missing Values Before Filling:**
   * Identified missing values in each column of the interactions\_df DataFrame to determine the extent of incomplete data before applying any imputation.
2. **Count and Identify Most Occurring Interaction Type:**
   * Counted occurrences of each Interaction\_Type and identified the most frequently occurring type to use as a replacement for missing values.
3. **Replace Null Values in Interaction\_Type:**
   * Replaced null values in the Interaction\_Type column with the most occurring interaction type to ensure completeness in the dataset.
4. **Check for Duplicate Records:**
   * Checked for and removed duplicate records based on the Interaction\_ID column to maintain data integrity.
5. **Capitalize Text Fields:**
   * Capitalized the first letter of each word in the Interaction\_Type column to standardize text formatting.
6. **Validate Dates:**
   * Validated the dates in the Interaction\_Date column to ensure they follow the correct date format.
7. **Validate Boolean Values:**
   * Validated boolean values in the Issue\_Resolved column to confirm they adhere to expected boolean values (e.g., 'Yes'/'No' or True/False).
8. **Cross-Verification of Missing Values:**
   * Rechecked for missing values in each column after filling them to ensure all intended imputation steps were successfully applied.
9. **Display Cleaned Data:**
   * Displayed the cleaned interactions\_df DataFrame to review the results of the cleaning process.
10. **Save Cleaned Data:**
    * Saved the cleaned DataFrame to a CSV file for further use and analysis, ensuring that the cleaned data is properly stored.
11. **Record Count After Cleaning:**
    * Logged the number of records in the cleaned DataFrame to confirm the effectiveness of the cleaning process and ensure data integrity.

#### 

#### **3.1.3 Approach for Cleaning Products Table**

1. **Count Missing Values Before Filling:**
   * Counted missing values in each column of the products\_df DataFrame to assess the extent of missing data before applying any imputation.
2. **Count Duplicates Before Dropping:**
   * Checked for duplicate entries in each column of the DataFrame to identify and address potential redundancy in the data.
3. **Drop Duplicates Based on Product\_ID:**
   * Removed duplicate records based on the Product\_ID column to ensure each product is uniquely represented.
4. **Fill Missing Values:**
   * Filled missing values in the Category column with "Uncategorized" to ensure completeness in the dataset.
5. **Capitalize Text Fields:**
   * Capitalized the first letter of each word in the Product\_Name and Category columns to standardize text formatting.
6. **Cross-Validation of Missing Values:**
   * Recounted missing values in each column after filling them to confirm that all intended imputation steps were applied effectively.
7. **Count Duplicates After Dropping:**
   * Checked for duplicate entries in each column again after removing duplicates to ensure no redundant data remains.
8. **Handle Negative or Zero Prices:**
   * Replaced negative or zero values in the Price column with the average price of the products to ensure all prices are valid and positive.
9. **Export Cleaned Data:**
   * Exported the cleaned DataFrame to a CSV file to store the cleaned data for further use and analysis.
10. **Record Count After Cleaning:**
    * Logged the number of records in the cleaned DataFrame to verify the effectiveness of the cleaning process and ensure data integrity.

#### **3.1.4 Approach for Cleaning Sales Team Table**

1. **Identify Missing Values:**
   * Identified missing values in each column of the sales\_team\_df DataFrame before applying any imputation or cleaning processes.
2. **Handle Missing Values:**
   * Calculated the average value of the Sales\_Achieved column.
   * Filled missing values in the Sales\_Achieved column with this average to maintain consistency in the data.
3. **Check for Duplicate Values:**
   * Checked for duplicate entries in the sales\_team\_cleaned\_df DataFrame, focusing on the Sales\_Rep\_ID column to identify potential redundancies.
4. **Drop Duplicates:**
   * Removed duplicate records based on the Sales\_Rep\_ID column to ensure each sales representative is uniquely represented.
5. **Standardize Formats:**
   * Standardized the format of the Name and Region columns by capitalizing the first letter of each word to ensure uniformity.
6. **Cross-Verification of Missing Values:**
   * Recounted missing values in each column after filling them to verify the effectiveness of the imputation process.
7. **Display the Cleaned Data:**
   * Displayed a sample of the cleaned sales\_team\_cleaned\_df DataFrame to review the results of the cleaning operations.
8. **Save the Cleaned Data:**
   * Exported the cleaned DataFrame to a CSV file to store the results of the data cleaning process.
9. **Record Count After Cleaning:**
   * Logged the number of records in the cleaned DataFrame to confirm the success of the cleaning operations and ensure data integrity.

#### **3.1.5 Approach for Cleaning Transactions Table**

1. **Count Missing Values:**
   * Counted missing values in each column of the transactions\_df DataFrame to assess the extent of missing data before any imputation.
2. **Calculate Mean Value:**
   * Calculated the mean value of the Amount column to use it for filling missing values.
3. **Fill Missing Values:**
   * Filled missing values in the Amount column with the rounded mean value to ensure completeness of the dataset.
4. **Check for Duplicate Records:**
   * Checked for duplicate entries based on the Transaction\_ID column to identify and address any redundancies.
5. **Ensure Date Format:**
   * Ensured that the Date column is formatted as YYYY-MM-DD to maintain consistency and facilitate accurate date-related operations.
6. **Validate Dates:**
   * Validated dates in the Date column to ensure they conform to the correct format and represent valid dates.
7. **Cross-Verification After Filling Missing Values:**
   * Recounted missing values in each column after filling them to verify the effectiveness of the imputation process.
8. **Correct Inaccurate Data:**
   * Corrected non-positive values in the Amount column by replacing them with the calculated mean value to maintain data integrity.
9. **Display the Cleaned DataFrame:**
   * Displayed a sample of the cleaned corrected\_transactions\_df DataFrame to review the results of the cleaning process.
10. **Save the Cleaned Data:**
    * Exported the cleaned DataFrame to a CSV file to store the results of the data cleaning process and ensure data is ready for further analysis.
11. **Record Count After Cleaning:**
    * Logged the number of records in the cleaned DataFrame to confirm the success of the cleaning operations and ensure data consistency.

## **Problem Statement 3.2: Customer Purchase Behaviour Analysis**

**Objective:** Analyse and report on customer purchase behaviours to understand spending patterns and customer segmentation.

**Description:**

1. Use transactions.csv to analyse customer purchasing behaviour, including total spending, average transaction amount, and frequency of purchases.
2. Cross-reference with customers.csv to segment customers by country and assess how geographic location impacts their purchasing patterns.

### **3.2.1 Approach and Outcome**

1. **Loading the Data:**

* We loaded the customers.csv, transactions.csv, and products.csv datasets into Spark DataFrames for analysis.
* We ensured that the data was cleaned and preprocessed to handle missing values and incorrect formats.

1. **Calculating Total Spending, Average Transaction Amount, and Purchase Frequency:**

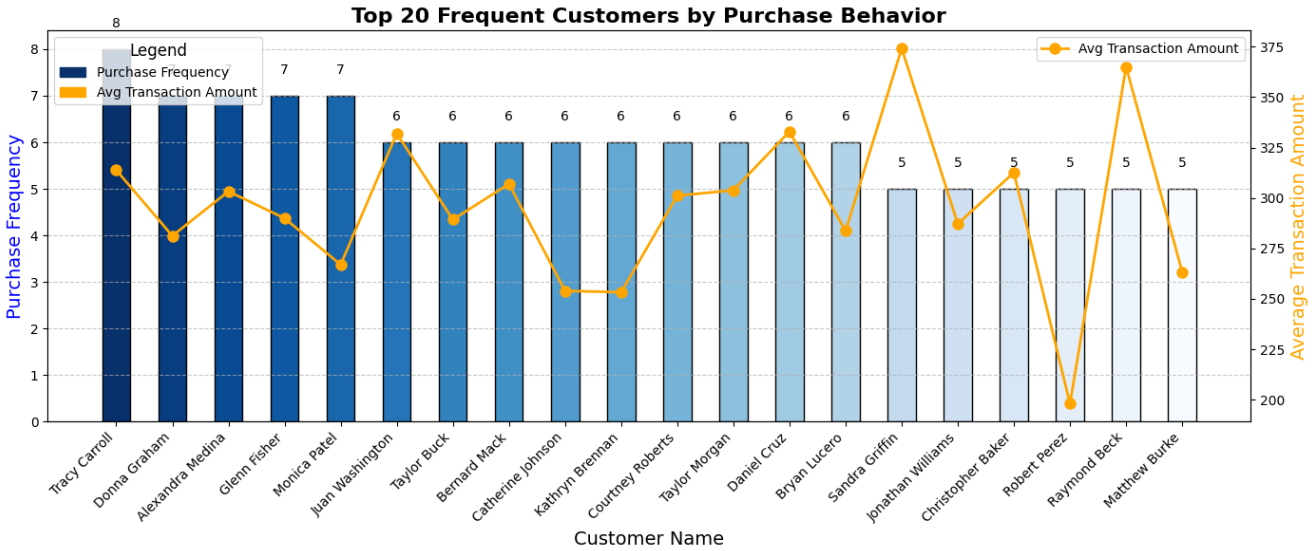
* We used the transactions.csv dataset to group transactions by Customer\_ID and calculated:
  + **Total Spending**: Summing up the Amount column to get the total amount spent by each customer.
  + **Average Transaction Amount**: Averaging the Amount values for each customer to find their typical transaction value.
  + **Purchase Frequency**: Counting the number of transactions (based on Transaction\_ID) to determine how often each customer makes a purchase.

1. **Merging Customer Details with Purchase Behavior:**

* After calculating the above metrics, we joined the result with the customers.csv data to enrich the spending data with customer names and countries.

1. **Identifying Top 20 Most Active Customers by Purchase Frequency:**

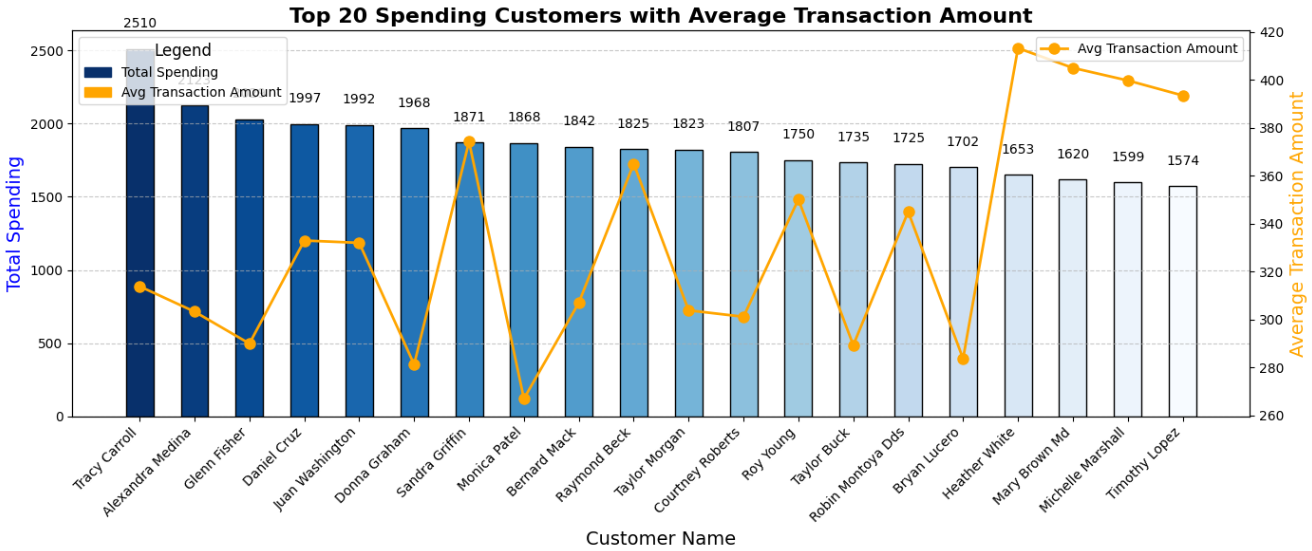
* We sorted the combined dataset by Purchase Frequency in descending order to identify the top 20 customers who made the most purchases.
* This data was converted into a Pandas DataFrame for easier plotting, and a dual-axis bar-line plot was created to visualize the relationship between purchase frequency and average transaction amount for the top customers.

****

**Fig 3.2.1.1 Top 20 Frequent Customers by Purchase Behavior**

1. **Identifying Top 20 Spending Customers:**

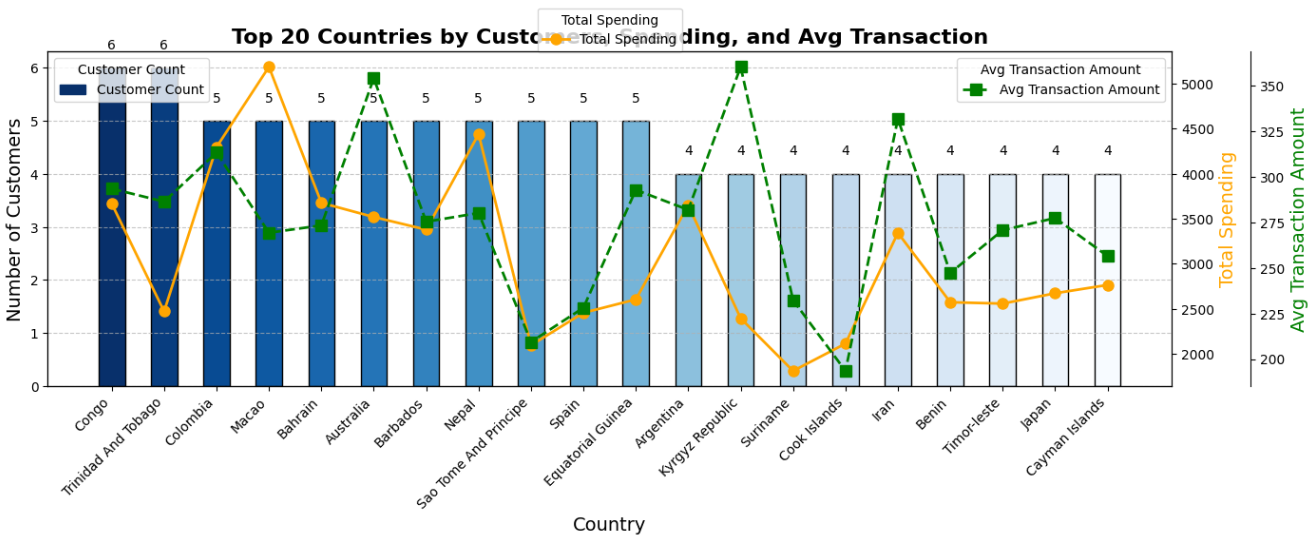
* We repeated a similar process, but this time we sorted customers by Total Spending to identify the top 20 highest-spending customers.
* The data was plotted using another dual-axis bar-line plot, showing the total spending alongside the average transaction amount for each of these top spenders.



**Fig 3.2.1.2 Top 20 Spending Customers with Average Transaction Amount**

1. **Analyzing Spending Distribution by Country:**

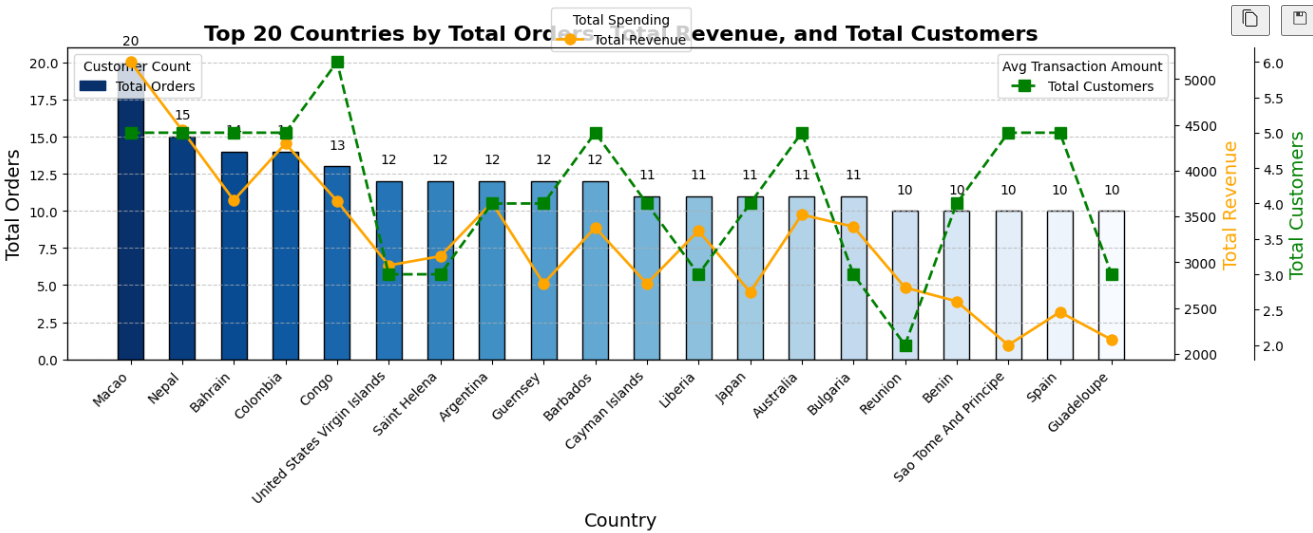
* To understand regional trends, we grouped customers by Country and calculated:
  + **Total Spending By Country**: Sum of Total\_Spending for all customers in that country.
  + **Average Transaction Amount By Country**: Average transaction value per country.
  + **Customer Count By Country**: The number of distinct customers in each country.
* The results were sorted by total spending and customer count, providing insights into which countries have the highest spending and customer base.



**Fig 3.2.1.3 Top 20 Countries By Customer Spending and Avg Transaction**

1. **Plotting Top 20 Countries by Key Metrics:**

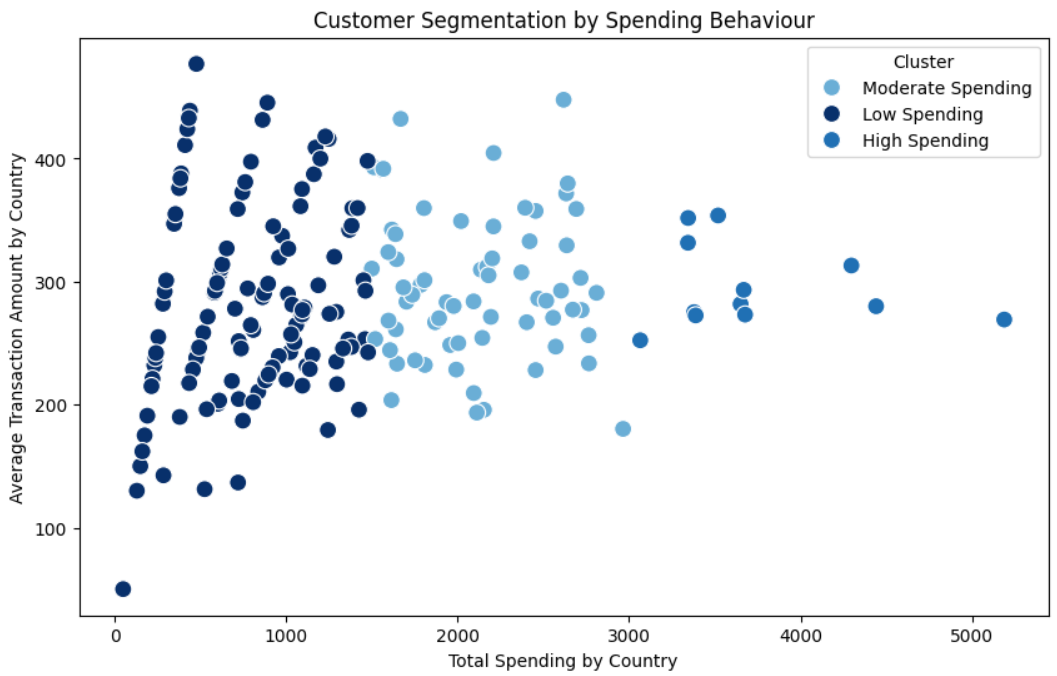
* We plotted the top 20 countries based on customer count, total spending, and average transaction amount using a triple-axis bar-line plot. This visualization helped identify the leading countries in terms of customer base and revenue generation.



**Fig 3.2.1.4** **Top 20 Countries by Total Orders ,Revenue and total customers**

1. **Customer Segmentation by Spending Behavior:**

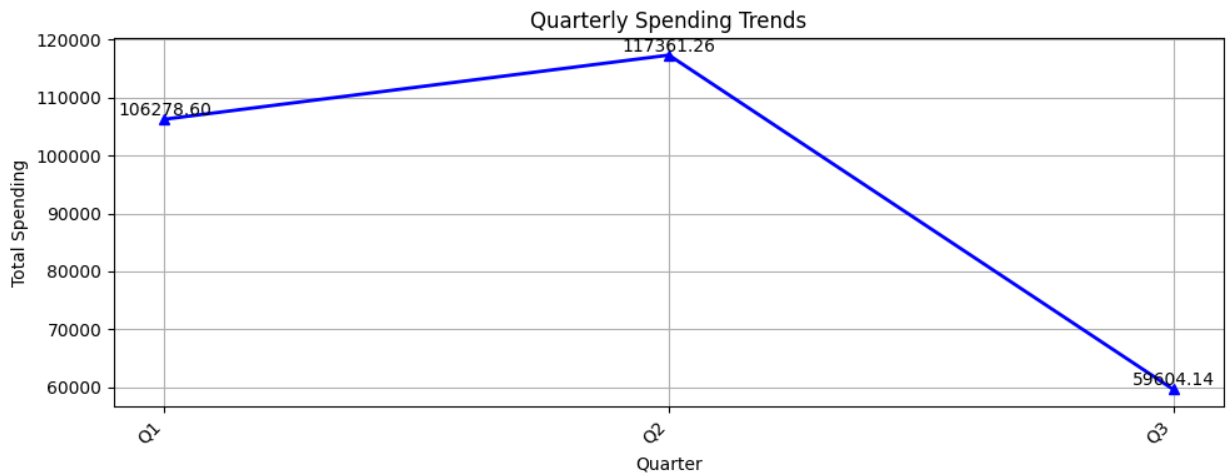
* Customers were manually segmented into three groups based on their total spending:
  + **Low Spending**: Total spending below 1500.
  + **Moderate Spending**: Total spending between 1500 and 3000.
  + **High Spending**: Total spending above 3000.



**Fig 3.2.1.5 Customer Segmentation By Spending Behaviour**

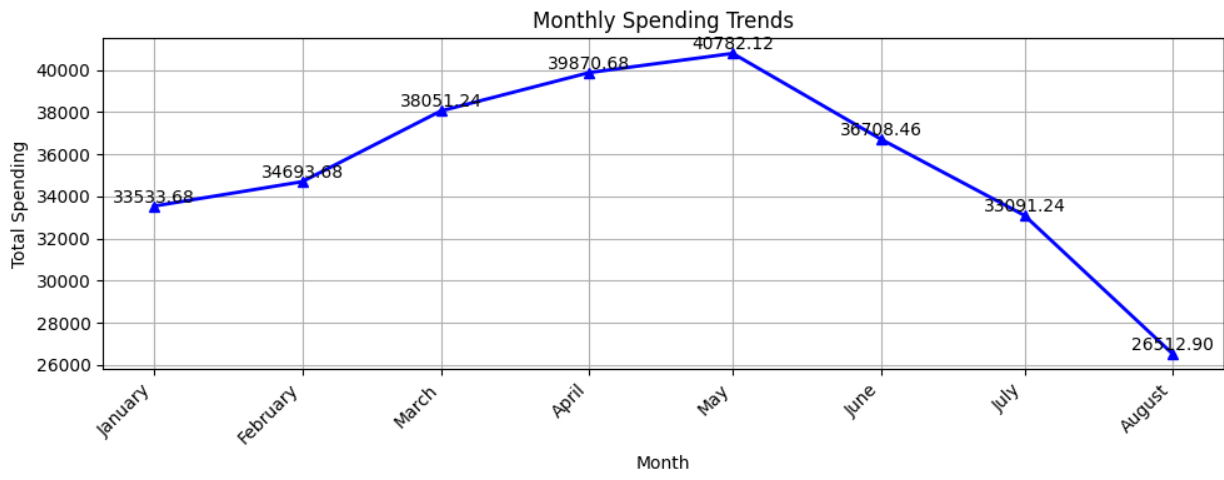
1. **Time-Based Trends (Quarterly, Monthly, and Weekly Analysis):**

* **Quarterly Trends**: We extracted the year and quarter from the Date column in the transactions and calculated total spending per quarter. The results were plotted to visualize spending trends over time.



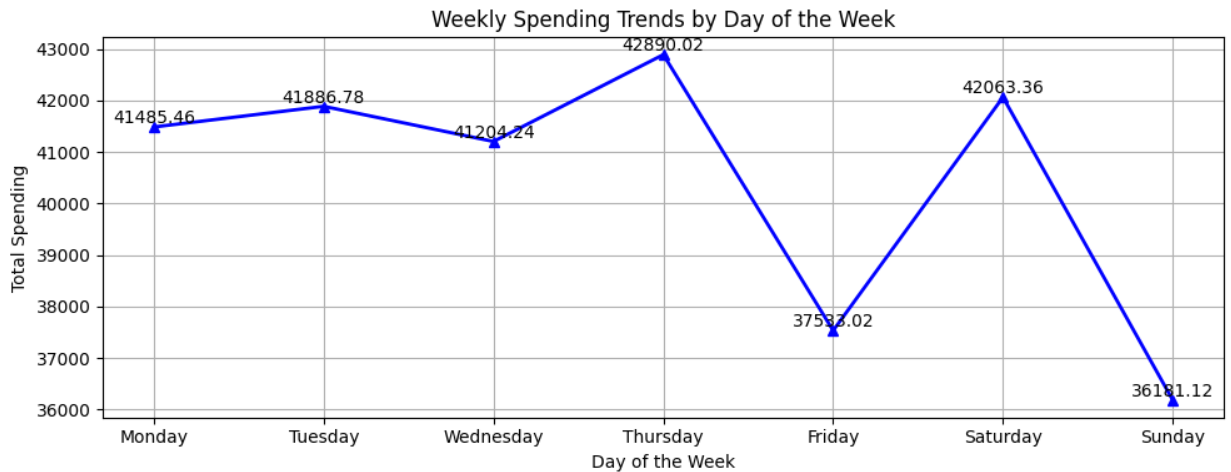
**Fig 3.2.1.6 Quartely Trends**

* **Monthly Trends**: Similarly, monthly spending was calculated by extracting the year and month. We plotted the monthly trends to analyze seasonality in spending patterns.



**Fig 3.2.1.7 Monthly Trends**

* **Weekly Trends**: We calculated total spending by day of the week to observe any weekly purchasing patterns, such as spikes in spending on weekends.



**Fig 3.2.1.8 Weekly Trends**

## **Problem Statement 3.3: Product Sales Performance Evaluation**

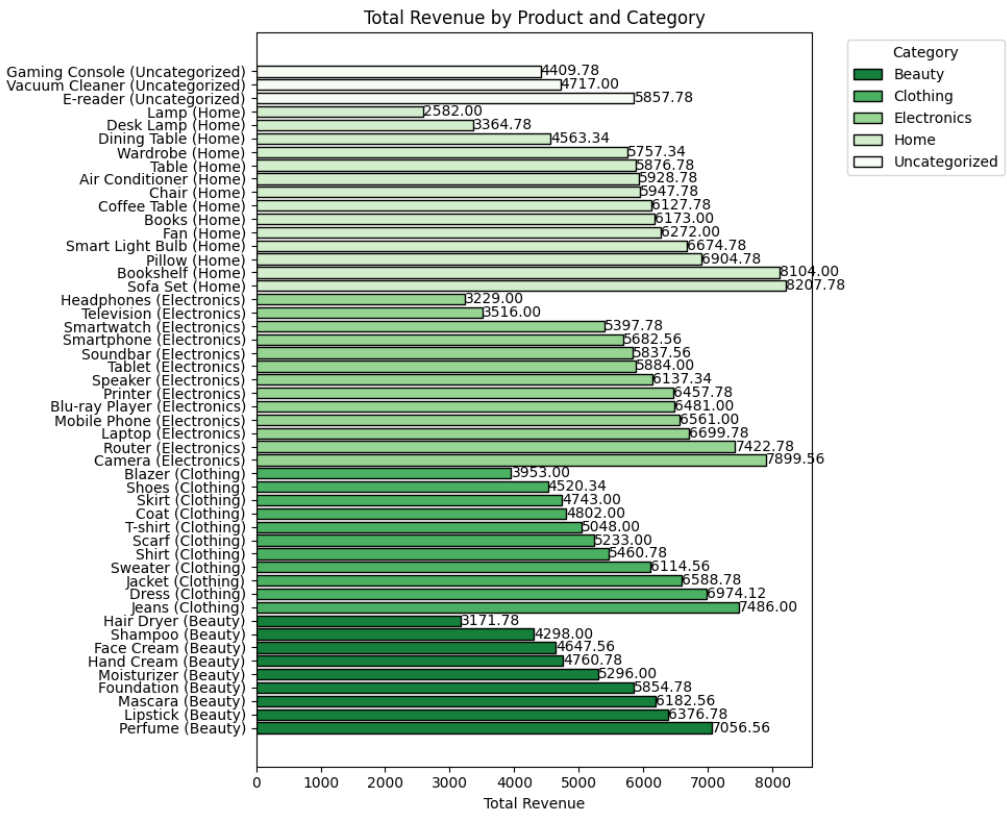
**Objective:** Evaluate the sales performance of products to identify top-selling items and assess category performance.

**Description:**

1. **Analyse transactions.csv** to determine total revenue and sales volume for each product.
2. **Join with products.csv** to categorize products and evaluate performance by product category.

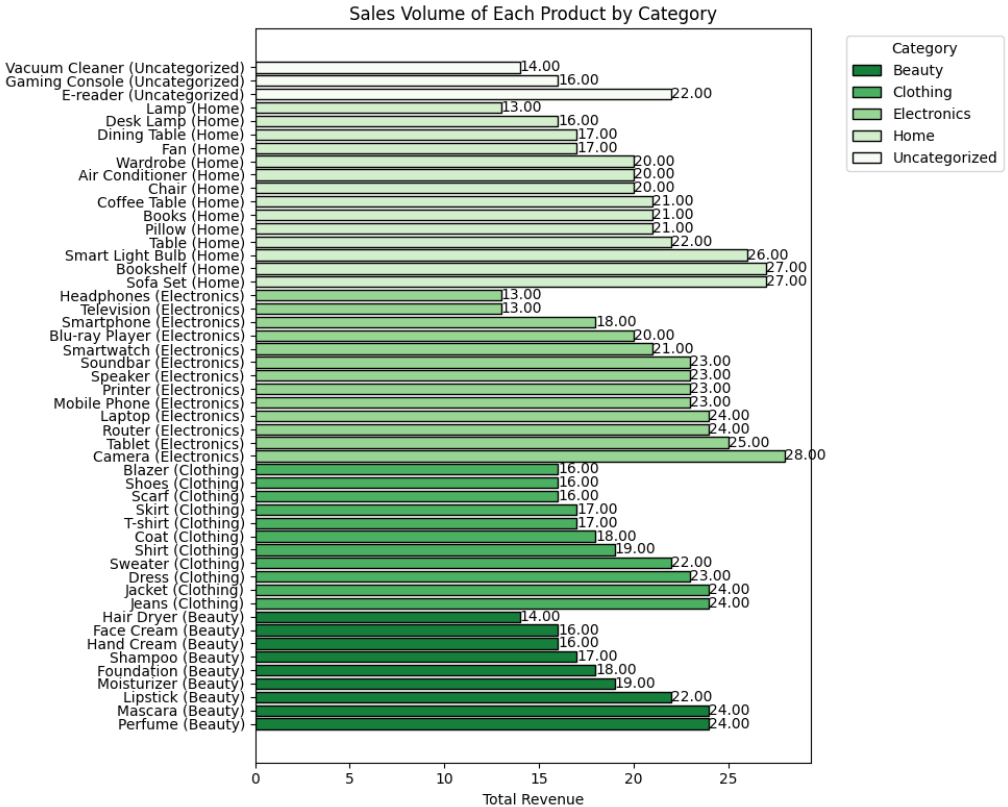
### **3.3.1 Approach and Outcome**

1. **Data Loading:**
   1. Loaded customer, transaction, and product datasets into Spark DataFrames from CSV files.
2. **Join Datasets:**
   1. Joined the transactions\_df with products\_df on Product\_ID to link transaction data with product information.
3. **Calculate Total Revenue and Sales Volume per Product:**
   1. Grouped the joined data by product to calculate total revenue (sum of 'Amount') and sales volume (count of transactions per product).
4. **Visualize Product Revenue by Category:**
   1. Converted Spark DataFrame to Pandas.
   2. Sorted products within categories by total revenue.
   3. Visualized revenue using a horizontal bar plot, with products categorized for color coding.



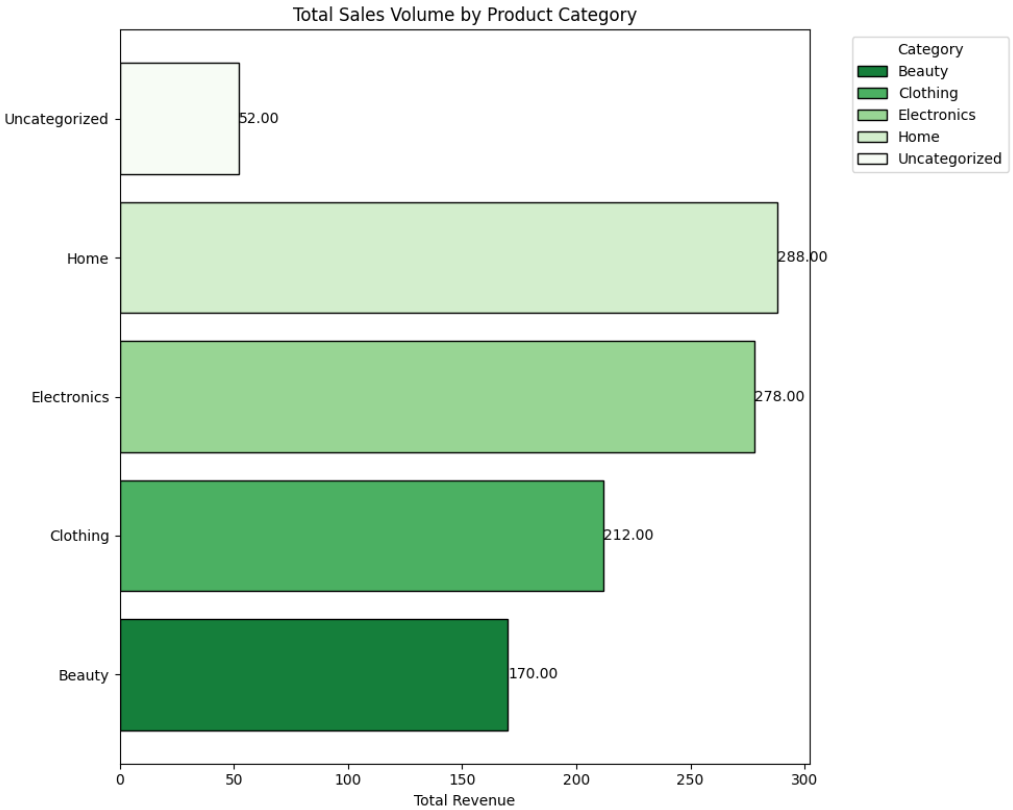
**Fig 3.3.1.1 Total Revenue and Sales Volume per Product**

1. **Visualize Sales Volume by Category:**
   1. Sorted products within categories by sales volume.
   2. Visualized the data using a horizontal bar plot to show product sales volume by category.



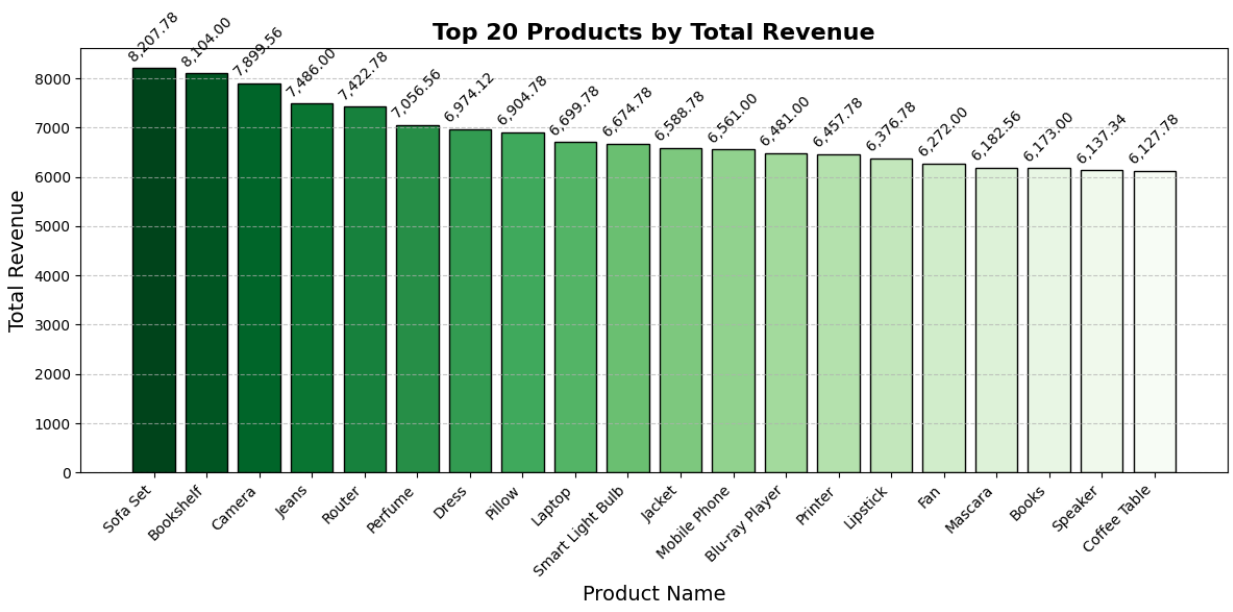
**Fig 3.3.1.2 Sales Volume of each product by Category**

1. **Total Sales Volume by Product Category:**
   1. Aggregated sales volume by category.
   2. Plotted total sales volume per category using horizontal bar plots.



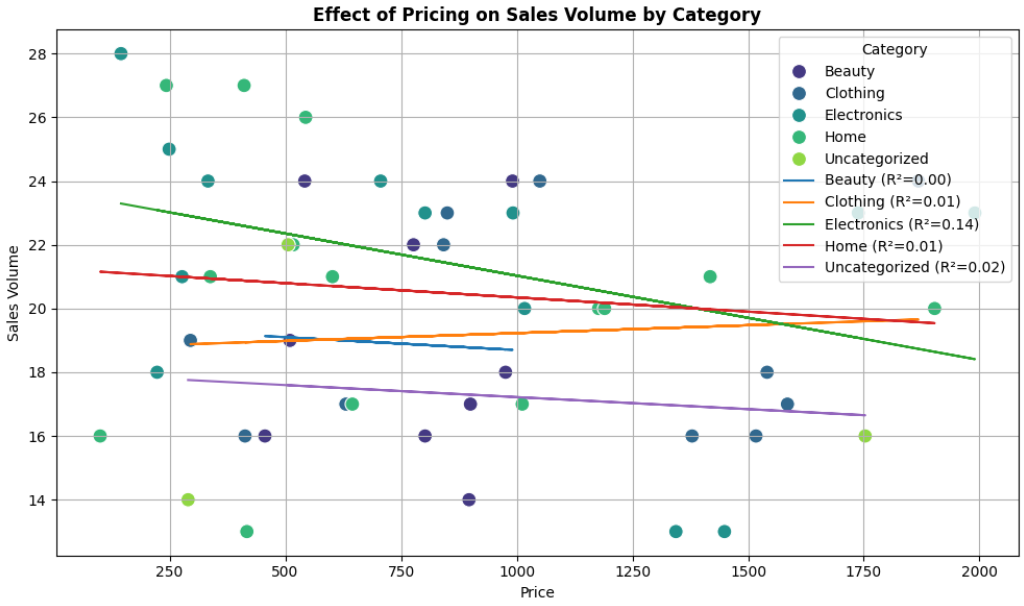
**Fig 3.3.1.3 Total Sales Volume By Product Category**

1. **Top 20 Products by Revenue and Sales Volume:**
   1. Extracted the top 20 products based on revenue and sales volume.
   2. Created vertical bar plots to display the top-performing products.



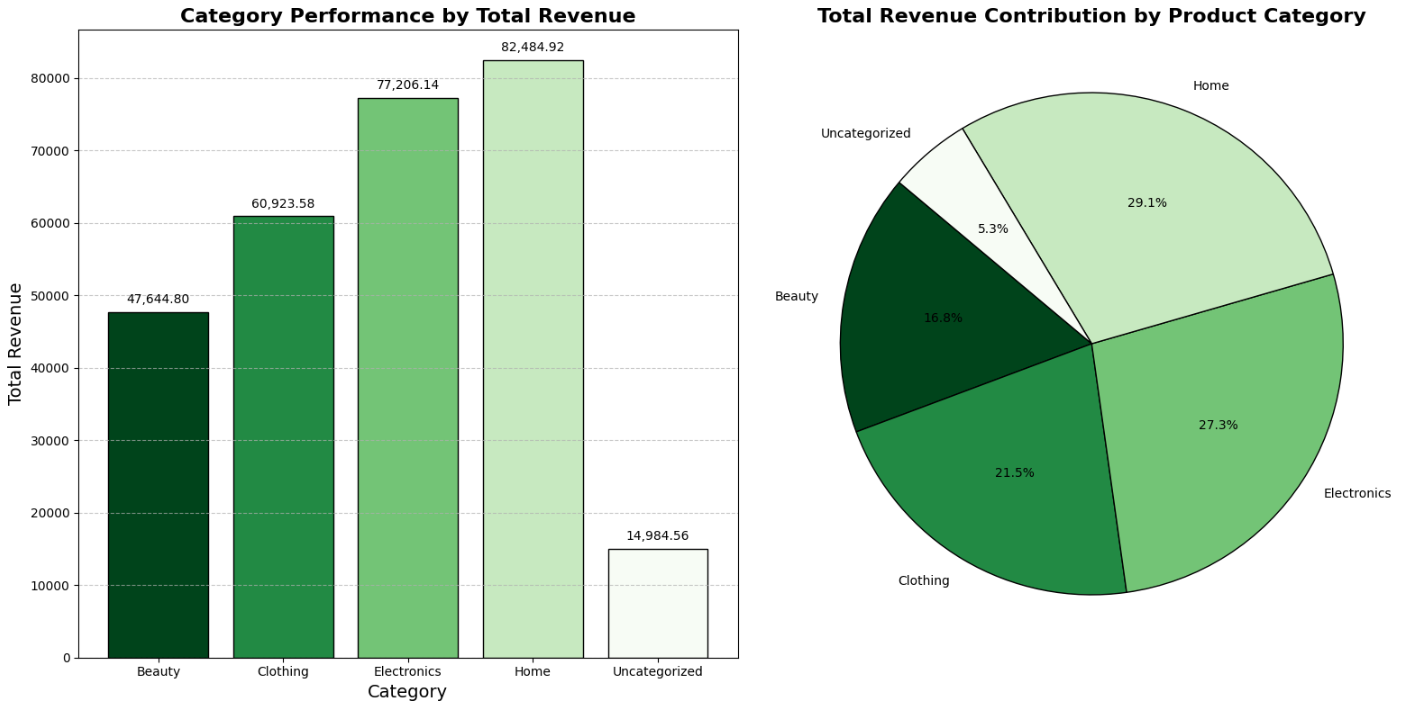
**Fig 3.3.1.4 Top Products By Revenue**

1. **Effect of Pricing on Sales Volume:**
   1. Plotted the relationship between price and sales volume using a scatter plot.
   2. Added regression lines for each category to assess trends.



**Fig 3.3.1.5 Pricing VS Sales Analysis**

1. **Category Performance by Total Revenue:**
2. Aggregated total revenue and sales volume by category.
3. Visualized the data using a bar plot for revenue and a pie chart for revenue contribution across categories.



**Fig 3.3.1.6 Category Performance**

**Problem Statement 3.4: Customer Interaction Effectiveness**

**Objective:** Assess the effectiveness of various customer interaction methods in resolving issues.

**Description:**

1. **Evaluate interactions.csv** to analyse resolution rates of different interaction types (e.g., Email, Chat).
2. **Cross-reference with customers.csv** to identify if specific customer demographics show a preference for certain interaction types.

### **3.4.1 Approach and Outcome**

1. **Data Loading:**

* Loaded cleaned\_interactions.csv and cleaned\_customers.csv into Spark DataFrames.

1. **Merge Interactions with Customer Data:**

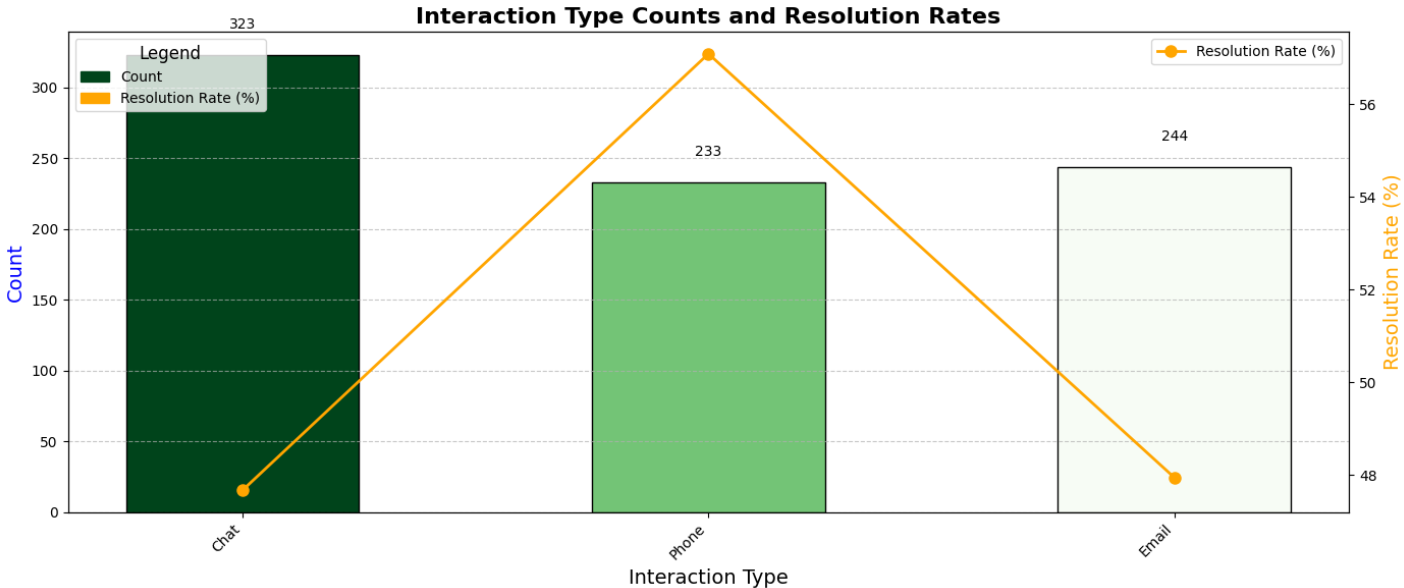
* Merged interactions with customers on Customer\_ID to analyze interaction data in the context of customer demographics (e.g., country, age group).

1. **Interaction Type Analysis:**

* Grouped interactions by Interaction\_Type and calculated the total number of interactions for each type.
* Aggregated resolution rates by calculating the percentage of resolved issues (Issue\_Resolved) for each interaction type.

1. **Visualization of Interaction Counts and Resolution Rates:**

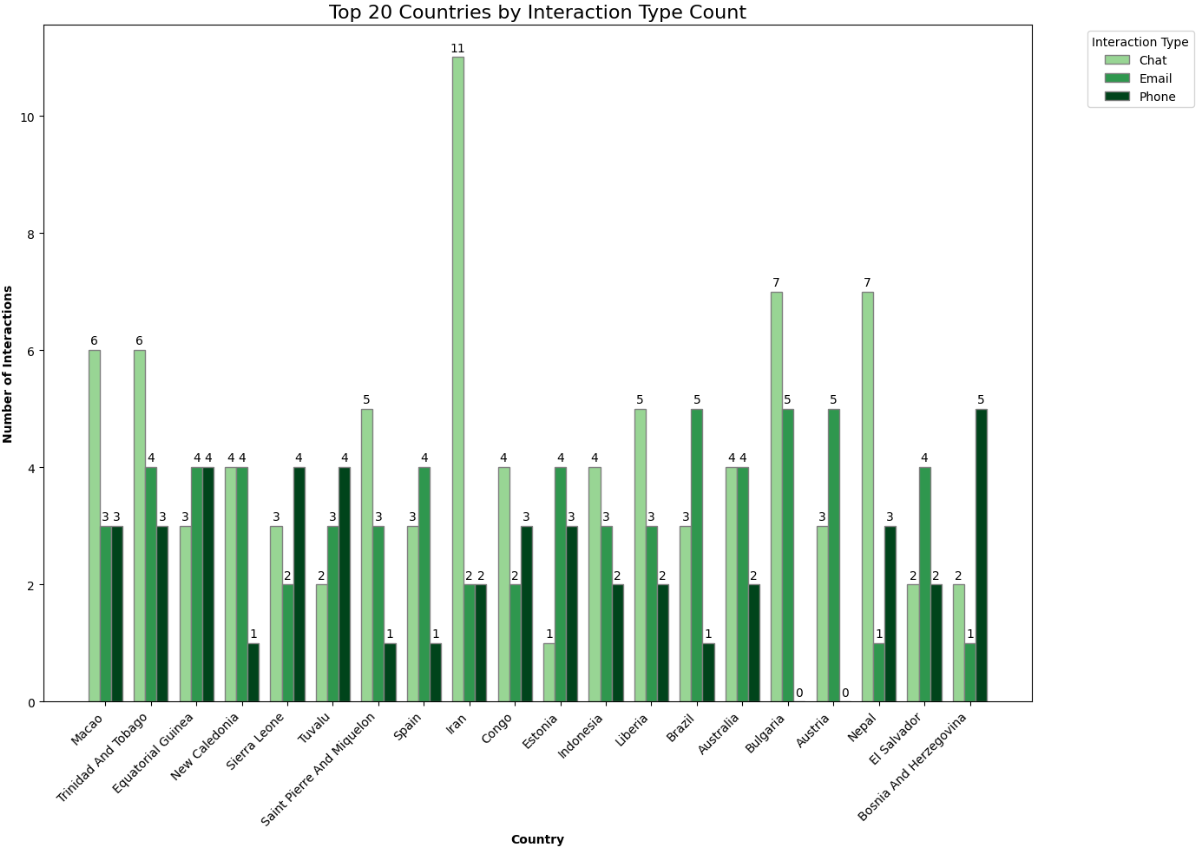
* Joined interaction counts and resolution rates into a single DataFrame.
* Converted the data to Pandas for plotting using a dual-axis bar-line chart to display both interaction counts and resolution rates for each interaction type.



**Fig 3.4.1.1 Interaction Counts and Resolution Rates**

1. **Top 20 Countries by Interaction Type Count:**

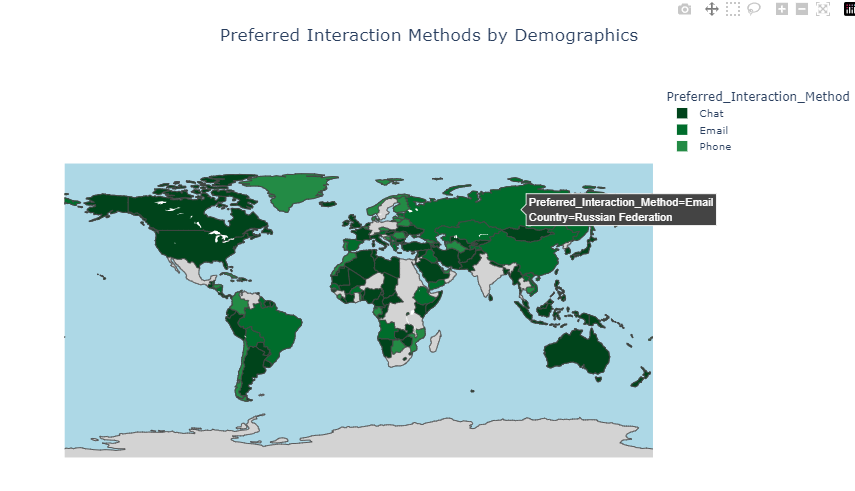
* Counted the total number of interactions for each country and sorted the data to get the top 20 countries based on interaction count.
* Filtered the interaction data for these top 20 countries.
* Grouped by country and interaction type, pivoted the table, and visualized the interaction type distribution across these countries.



**Fig 3.4.1.2 Top Countries by Interaction Type**

1. **Preferred Method of Interaction by Country:**

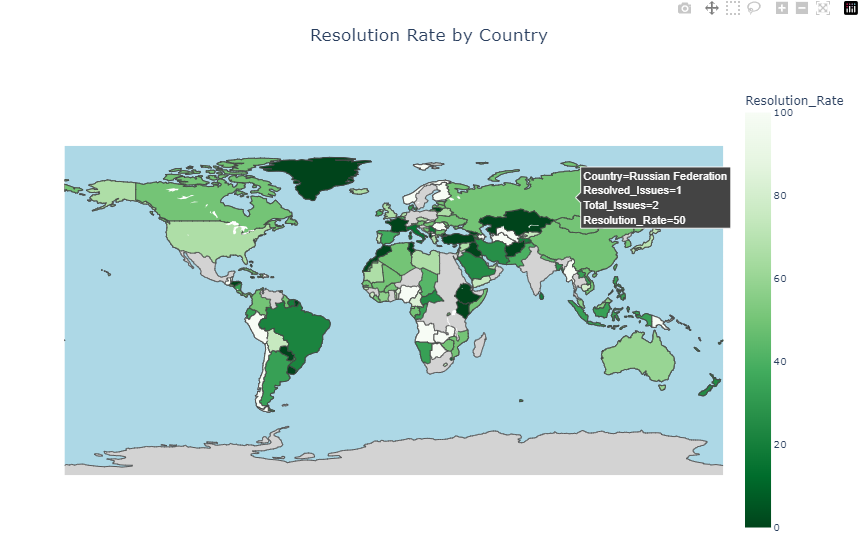
* Calculated the most preferred interaction method for each country by determining which interaction type had the highest count for each country.
* Visualized the preferred interaction method using a world map to identify patterns in customer preferences by demographics.



**Fig 3.4.1.3 Preferred Interaction Methods**

1. **Resolution Rate by Country:**

* Grouped by country and calculated the resolution rate (percentage of resolved issues) for each country.
* Visualized the resolution rate distribution across countries using a world map, showing both total and resolved issues for each country.



**Fig 3.4.1.4 Resolution Rate by country**

## **Problem Statement 3.5: Sales Team Performance Assessment**

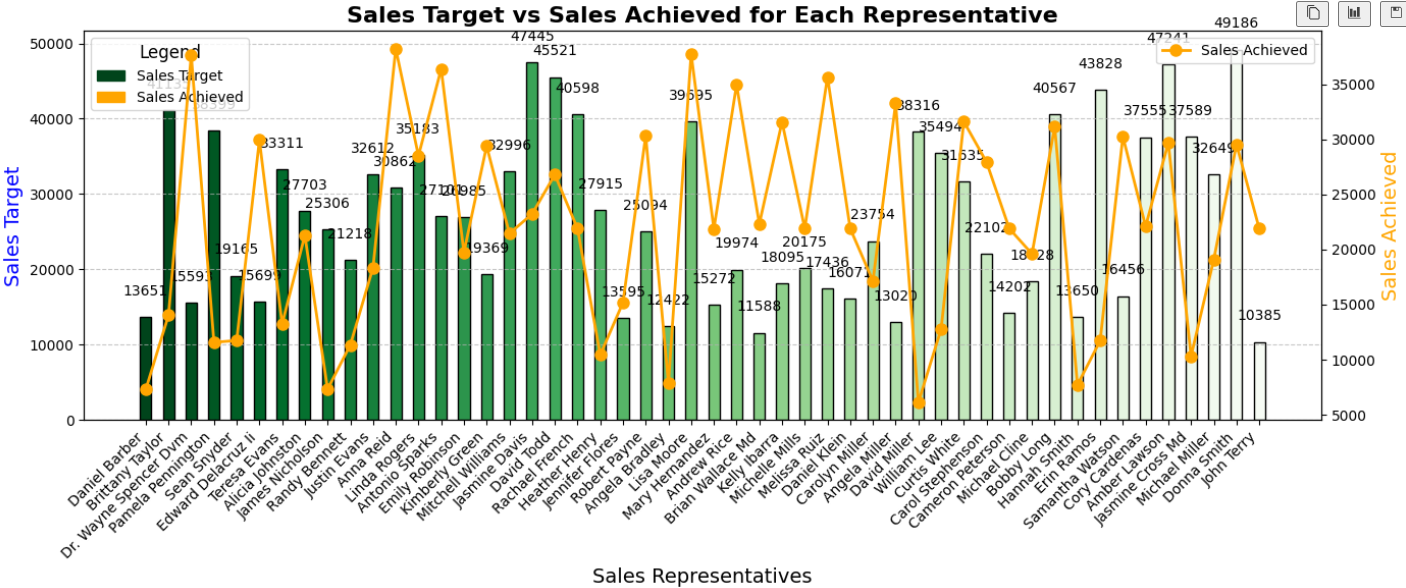
**Objective:** Analyse sales team performance to identify high performers and areas needing improvement.

**Description:**

1. **Assess sales\_team.csv** to evaluate performance metrics such as sales achieved versus sales targets for each sales representative.
2. **Identify top performers** and those who are underperforming relative to their targets.

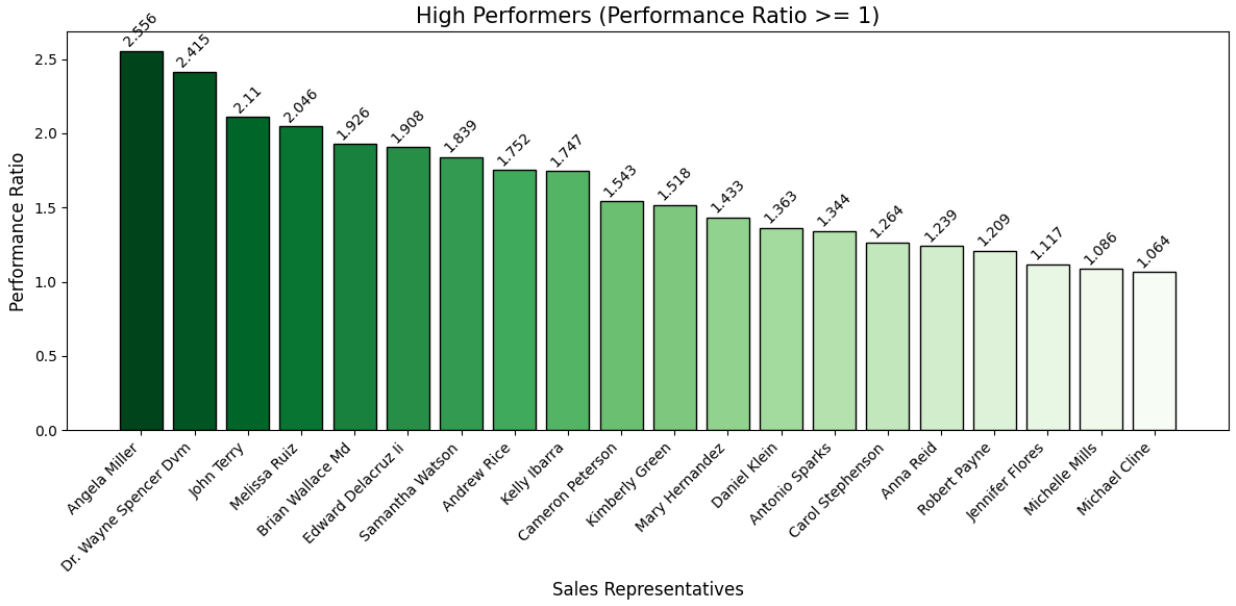
### **3.5.1 Approach and Outcome**

1. **Data Loading:**
   * Loaded cleaned\_sales\_team.csv into a Spark DataFrame.
2. **Performance Metric Calculation:**
   * Calculated the **Performance Ratio** for each representative as the ratio of Sales\_Achieved to Sales\_Target.
   * Classified representatives into **High Performers** (Performance Ratio ≥ 1) and **Underperformers** (Performance Ratio < 1).
3. **Sales Percentage Calculation:**
   * Computed the **Sales Percentage** as the percentage of the target achieved by each representative.
4. **Visualization of Sales Target vs Sales Achieved:**
   * Converted the DataFrame to Pandas.
   * Created a dual-axis bar-line chart to display sales targets and achievements for each representative.

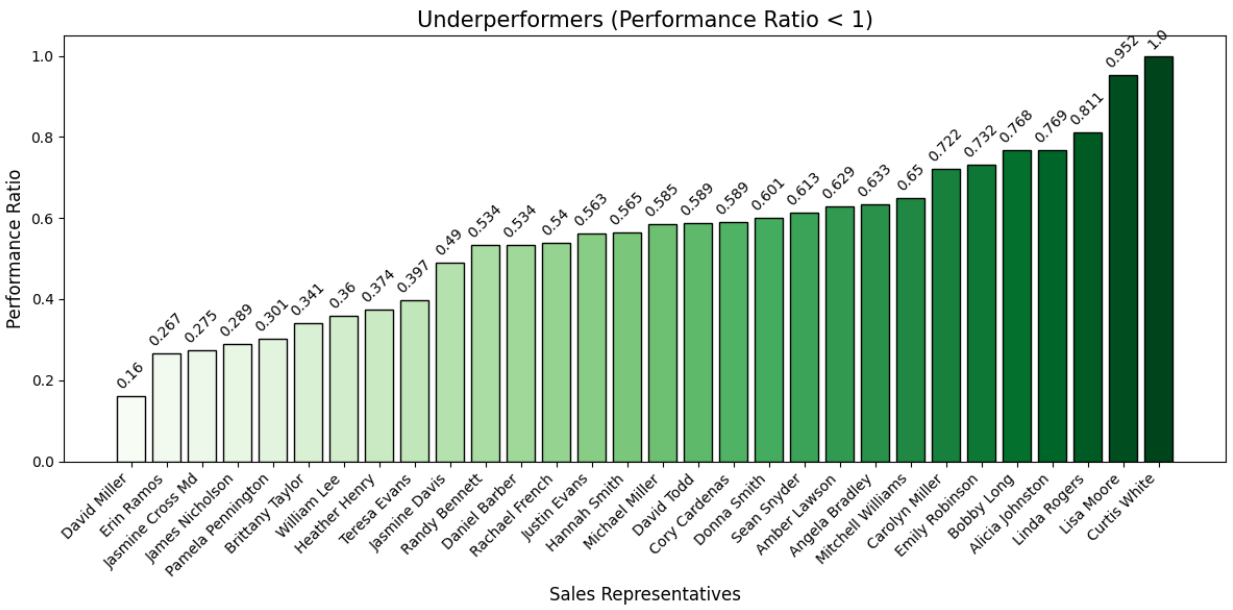


**Fig 3.5.1.1 Sales Target VS Sales Achieved**

1. **Top Performers and Underperformers Identification:**
   * Segregated representatives into top performers and underperformers based on Performance Ratio.
   * Created separate visualizations for both categories.

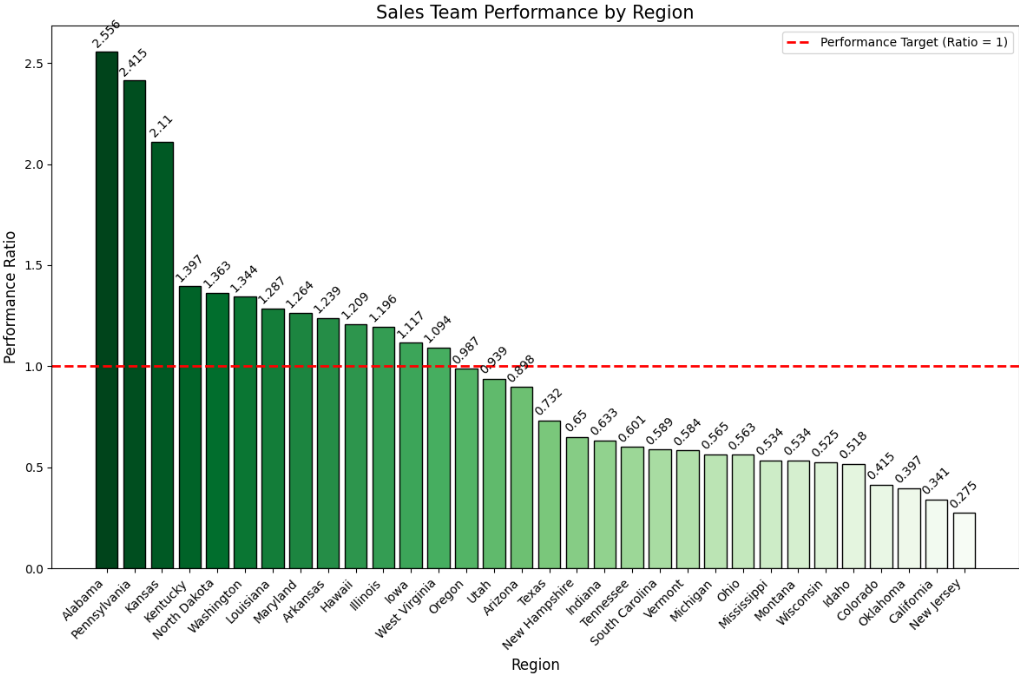


**Fig 3.5.1.2 Top Performers**



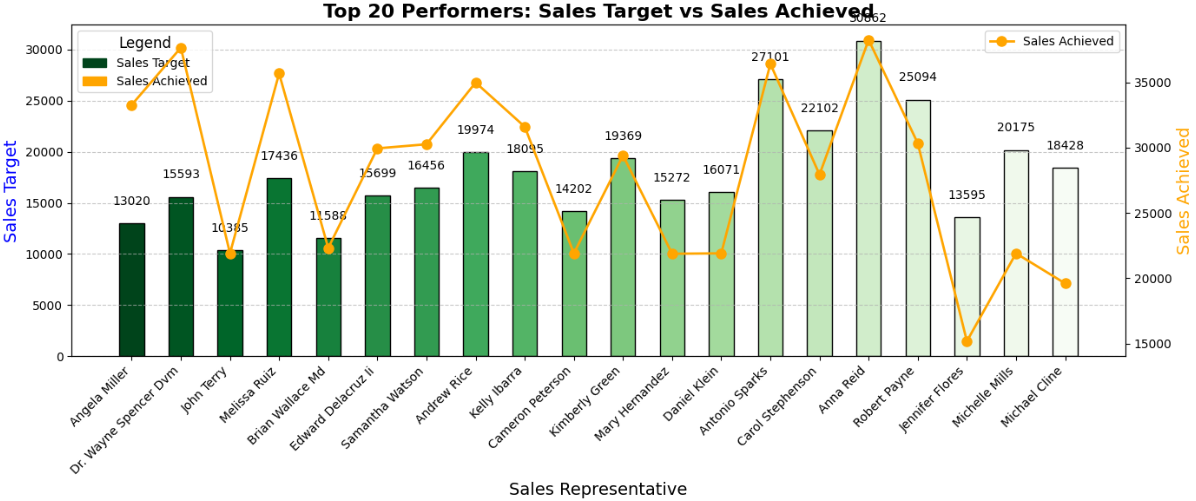
**Fig 3.5.1.3 Underperformers**

1. **Regional Performance Assessment:**
   * Aggregated Sales\_Achieved and Sales\_Target by region.
   * Calculated the regional **Performance Ratio** and classified regions as high-performing or underperforming.



**Fig 3.5.1.4** **Regional Performance**

1. **Top 20 Sales Representatives Performance Report:**
   * Sorted representatives by **Achievement Rate**.
   * Selected the top 20 performers and visualized their performance using a dual-axis bar-line chart.



**Fig 3.5.1.5 Top 20 Sales Representatives**

# **Chapter - 4 Conclusion**

This project successfully implements a robust data cleaning and preprocessing workflow to prepare multiple datasets related to sales and customer interactions for further analysis. By systematically addressing issues such as missing values, duplicates, inconsistencies in formatting, and anomalies in the data, the project ensures that the datasets are accurate, consistent, and ready for insightful analysis.

Handling missing numeric values with calculated averages, standardizing text fields, and validating date and boolean fields contribute to the integrity and usability of the data. Additionally, resolving anomalies like negative sales values and duplicate records prevents skewed results and ensures more reliable analytical outcomes.

The cleaned and preprocessed data now serves as a solid foundation for analysis, supporting key business processes like sales performance evaluation and customer relationship management. This structured approach not only enhances the quality of the datasets but also sets the stage for more accurate and actionable insights, contributing to better strategic decision-making.