Assessment: Sales Data Analysis

Introduction

In this report I provide a complete analysis of the sales dataset. The dataset contains transaction-level details, including invoice number, invoice date, product description, quantity, unit prices, customer identifiers, and the countries where the transactions took place. The aim of this report is to summarize the findings from the data and provide insights into the sales performance.

The dataset was loaded from Google Drive using Google Colab. The data file, named data.csv, was read into a pandas DataFrame. Then I counted the total number of entries (rows) present in the data. In this dataset there are a total of 541909 entries. The entries is from 12-01-2010 to 30-09-2011

Sales Data Overview

The dataset contains the following columns:

1. **InvoiceNo** - Object data type
2. **StockCode** - Object data type
3. **Description** - Object data type
4. **Quantity** - int64 data type
5. **InvoiceDate** - Object data type
6. **UnitPrice** - float64
7. **CustomerID** - float64
8. **Country** - Object data type

Data Exploration and Cleaning:

The dataset contains various columns related to sales transactions. The presence of duplicate and missing values can impact the accuracy and reliability of any analysis or model built using this dataset.

Duplicate Values

Using Python's duplicated function, the analysis calculate a significant number of duplicate entries in the dataset:

- **Total Duplicate Rows**: 5,268

Missing Values

The dataset also contains missing values in certain columns, which are crucial for a comprehensive analysis. Here are the details of the missing values:

- **Description**: 1,454 missing values
- **Customer ID:** 135,080 missing values

```
missing_values = data.isnull().sum()
total_missing_values = missing_values.sum()
duplicate_rows = data[data.duplicated()]
total_duplicates = len(duplicate_rows)
print("Summary of Missing and Duplicate Values")
print(f"Total Missing Values: {total_missing_values}")
print("\nMissing Values by Column:")
print(missing_values)
num_duplicates = len(duplicate_rows)
print("Total number of duplicate rows:", num_duplicates)
```

```
Summary of Missing and Duplicate Values
Total Missing Values: 136534

Missing Values by Column:
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
Total number of duplicate rows: 5268
```

After removing duplicate entries. There are a total of 536641 entries. **Out of these 1364911 missing values.**

Handling Missing Values:

For description of the product: According to my analysis the number of missing values is relatively small. We can either remove these rows or attempt to fill in the missing descriptions to Unknown.

For customer id : There is a large number of missing values, removing these rows might result in significant data loss. So I will decide whether to impute these values or analyze the trend of the data.

I removed the rows that contain missing values in the description using this command:

sales_data_cleaned = sales_data_cleaned.dropna(subset=['Description'])

After analyzing the dataset, I found some quantities in negative and some unit prices as 0, which does not make any sense. Such anomalies create noise and can lead to misleading

insights. Therefore, it was necessary to clean the dataset by removing these entries. So I will do that

All rows with negative quantities and unit price less than and equal to zero were removed to maintain dataset consistency. After removing that, the length of the dataset is 524878.

```
data2 = data2[data2['Quantity'] >= 0]
data2 = data2[data2['UnitPrice'] > 0]
print("Rows with negative quantities and unit prices equal to 0 have been removed.")
print(f"Remaining rows in dataset: {len(data2)}")
```

```
Rows with negative quantities and unit prices equal to 0 have been removed.
Remaining rows in dataset: 524878
```

CUSTOMER ID HANDLING:-

To handle the missing customer id values in the sales data effectively. The initial 132,186 missing values in the customer id column.

Methodology

I used the mode imputation method within specific groups to fill the missing customer Id values. The mode represents the most frequently occurring value, which is suitable for categorical data like Customer Id.

The code below impute the missing values:-

```
def impute_customer_id(df, group_columns):
    for col in group_columns:
        uncleaned_data['CustomerID'] = uncleaned_data.groupby(col)['CustomerID'].transform(lambda x: x.fillna(x.mode()[0] if not x.mode().empty else pd.NA))
    return uncleaned_data

group_columns = ['InvoiceNo', 'Country', 'StockCode']

data_imputed = impute_customer_id(uncleaned_data, group_columns)

remaining_missing = data_imputed['CustomerID'].isnull().sum()
print(f"Remaining missing CustomerID values: {remaining_missing}")

data_imputed.head(10)
```

I choose **Invoice No, Country and Stock Code** as the columns for grouping. This choice is based on the assumption that within each group, the Customer Id is likely to be the same. For instance, a particular invoice no from a specific country and stock code should ideally correspond to a single customer. **Within each group, I used the mode to fill in the missing Customer Id.** I used it because the mode represents the most likely customer in that group, making it a logical choice for categorical imputation.

- The lambda function **lambda x: x.fillna(x.mode()[0] if not x.mode().empty else pd.NA** It fills the missing values with the mode of the group if the mode is not empty. If the mode is empty, it leaves the value as missing **(denoted by pd.NA).**

Summary Statistics of cleaned dataset:

	Quantity	UnitPrice	CustomerID
count	524878.000000	524878.000000	524878.000000
mean	10.616600	3.922573	15922.453585
std	156.280031	36.093028	1851.495344
min	1.000000	0.001000	12346.000000
25%	1.000000	1.250000	14367.000000
50%	4.000000	2.080000	16243.000000
75%	11.000000	4.130000	17841.000000
max	80995.000000	13541.330000	18287.000000

Cleaned Dataset

The dataset has been cleaned and is now ready for further analysis. You can access the cleaned dataset here.

Summary of the Cleaned dataset:-

- I provide an overview of the key metrics derived from the sales dataset. The analysis focuses on calculating the total number of transactions, total sales, and the average transaction value to offer insights into the dataset's overall performance

Key Metric

Metric	Value
Total Number of Transactions	19,960
Total Sales	₹10,642,110.80
Average Transaction Value	₹533.17

- The dataset contains a total of **19,960** unique transactions. **This metric is calculated by counting the unique values in the Invoice Number column.**
- **The total sales amount calculated by the product of quantity and unit price for each transaction. The formula used is:**

  Total Sales=∑(Quantity×UnitPrice)

- **On average, each transaction is derived by dividing the total sales by the total number of transactions. The formula used is:**

  Average Transaction Value = Total Sales / Total Number of Transactions

Date and Time Feature Extraction:-

This section of the report details the process of converting the Invoice Date column to a datetime format and extracting relevant time-based features.

1. Conversion to Datetime Format

   The **invoice date** column, initially stored as strings, was converted to a datetime format. This conversion allows for easy manipulation and extraction of date and time components

2. After the conversion, **I extracted the year, month, day and hour from the Invoice Date column.** These features facilitate granular analysis of temporal trends in the dataset.

The Command I used:-

```
cleaned_data['InvoiceDate'] = pd.to_datetime(cleaned_data['InvoiceDate'], format='%m/%d/%Y %H:%M')
```

It helps identify patterns and trends over different time periods easily.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Year	Month	Day	Hour
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	2010	12	1	8
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	12	1	8
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	2010	12	1	8
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	12	1	8
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	12	1	8

The updated dataset, with these new time-based features, is now ready for further analysis.

Descriptive Analysis:

This section provides an analysis of the sales data, including total revenue generated from sales, identification of the top 10 best-selling products by revenue, and the total

number of unique customers. The insights gained from this analysis can help in understanding the performance of different products and **customer behavior.**

Key Metric:-

Metric	**Value**
Total Revenue Generated from Sales	**₹10,642,110.80**
Total Number of Unique Customers	**4,338**

1. **The total revenue generated from sales is ₹10,642,110.80**. This is calculated by summing the total price of all items sold. The total price for each item is obtained by multiplying the quantity by the unit price.
2. **The top 10 best-selling products by revenue are identified by grouping the sales data by stock code and summing the total price for each product.** The following table lists these products:

```
Top 10 Best-Selling Products by Revenue:
StockCode
DOT       206248.77
22423     174156.54
23843     168469.60
85123A    104462.75
47566      99445.23
85099B     94159.81
23166      81700.92
POST       78101.88
M          77750.27
23084      66870.03
```

3. The total number of unique customers who made purchases is **4,338.** This is calculated by counting the unique values in the **customer Id** column.

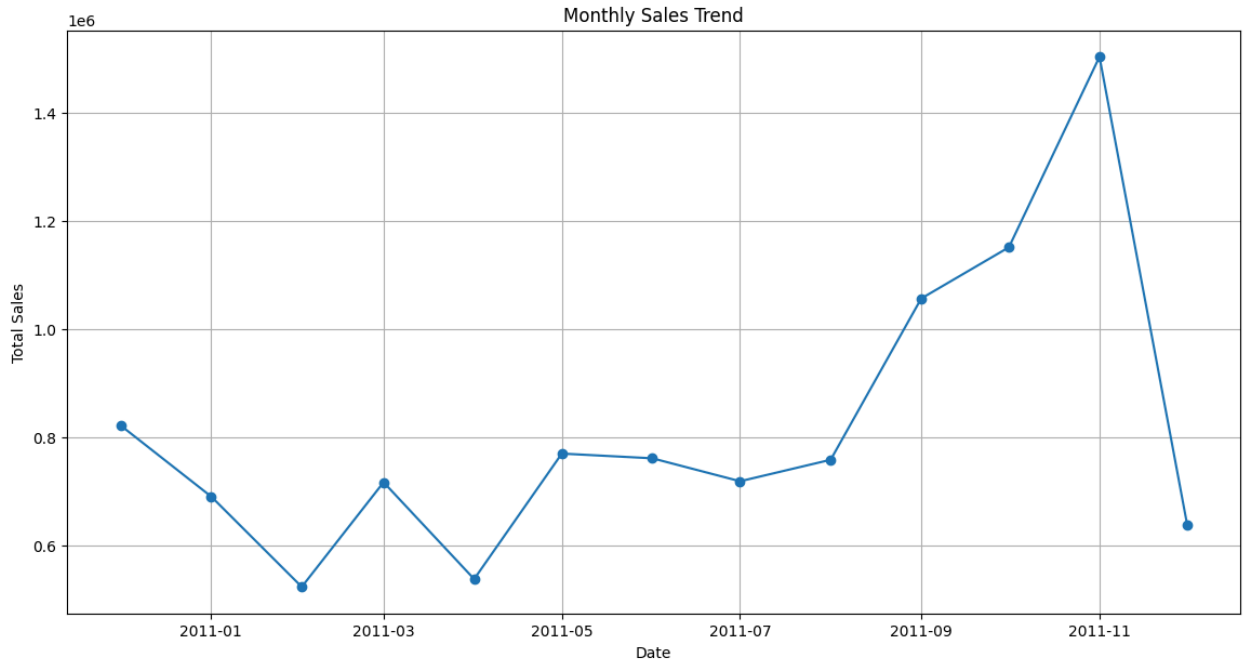
This analysis provides valuable insights into the sales data, highlighting the total revenue generated, the best-selling products, and the number of unique customers. These metrics are crucial for understanding the sales performance.

Sales Trends:

To analyze sales trends, we first converted the invoice date column to a datetime format and extracted relevant time-based features such as year, month, week, and day. I have already done it.

Monthly Sales Trend

To visualize the monthly sales trend, I have aggregated the sales data by year and month, then plotted the results. Monthly Sales Trend Graph Below:



Monthly Sales Change Analysis:

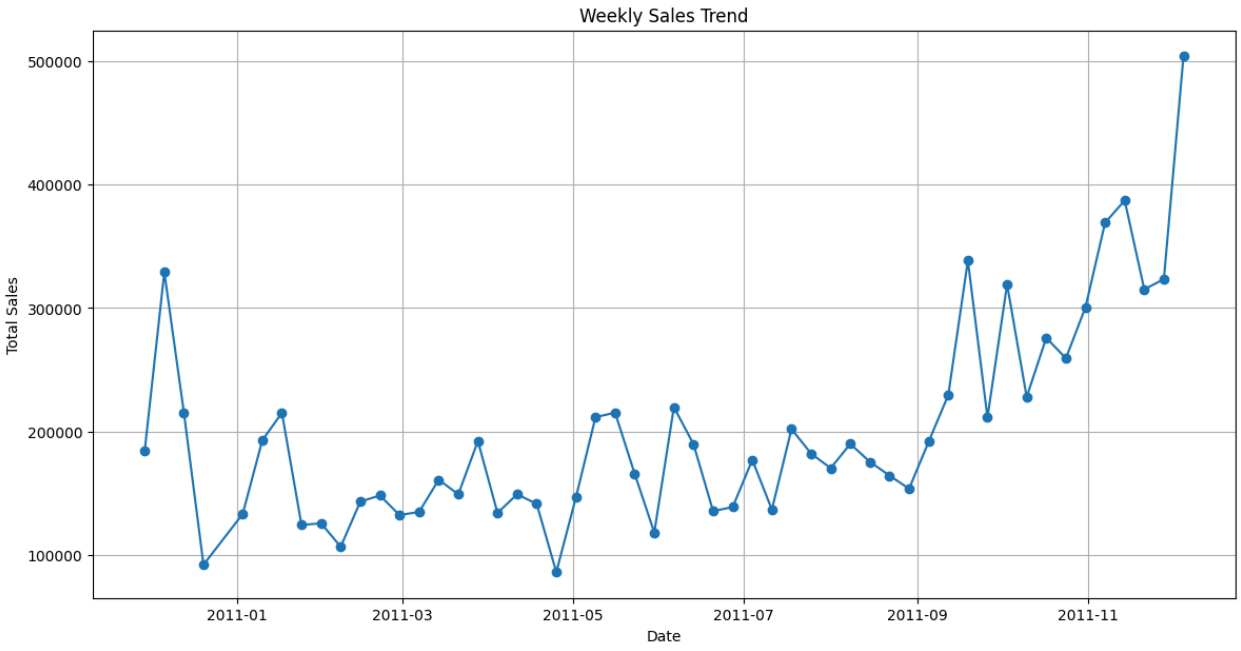
	Year	Month	TotalPrice	Date	SalesChange
0	2010	12	821452.730	2010-12-01	NaN
1	2011	1	689811.610	2011-01-01	-131641.120
2	2011	2	522545.560	2011-02-01	-167266.050
3	2011	3	716215.260	2011-03-01	193669.700
4	2011	4	536968.491	2011-04-01	-179246.769
5	2011	5	769296.610	2011-05-01	232328.119
6	2011	6	760547.010	2011-06-01	-8749.600
7	2011	7	718076.121	2011-07-01	-42470.889
8	2011	8	757841.380	2011-08-01	39765.259
9	2011	9	1056435.192	2011-09-01	298593.812
10	2011	10	1151263.730	2011-10-01	94828.538
11	2011	11	1503866.780	2011-11-01	352603.050
12	2011	12	637790.330	2011-12-01	-866076.450

Observations:-

- Sales in **January started at around ₹700,000** but declined steadily, reaching a low in **february with about ₹600,000 in sales.**
- There was a recovery in **April, with sales fluctuating around ₹536,968 to ₹757841 through the summer months.**
- **A significant surge in sales is evident in November,** with sales peaking at over **₹1.5 million.** This spike is likely due to holiday shopping and major sales events.
- In December, there is a significant drop in sales, because the data stops at the end of the year.

Weekly Sales Trend

To visualize the weekly sales trend, I have aggregated the sales data by year and week, then plotted the results.



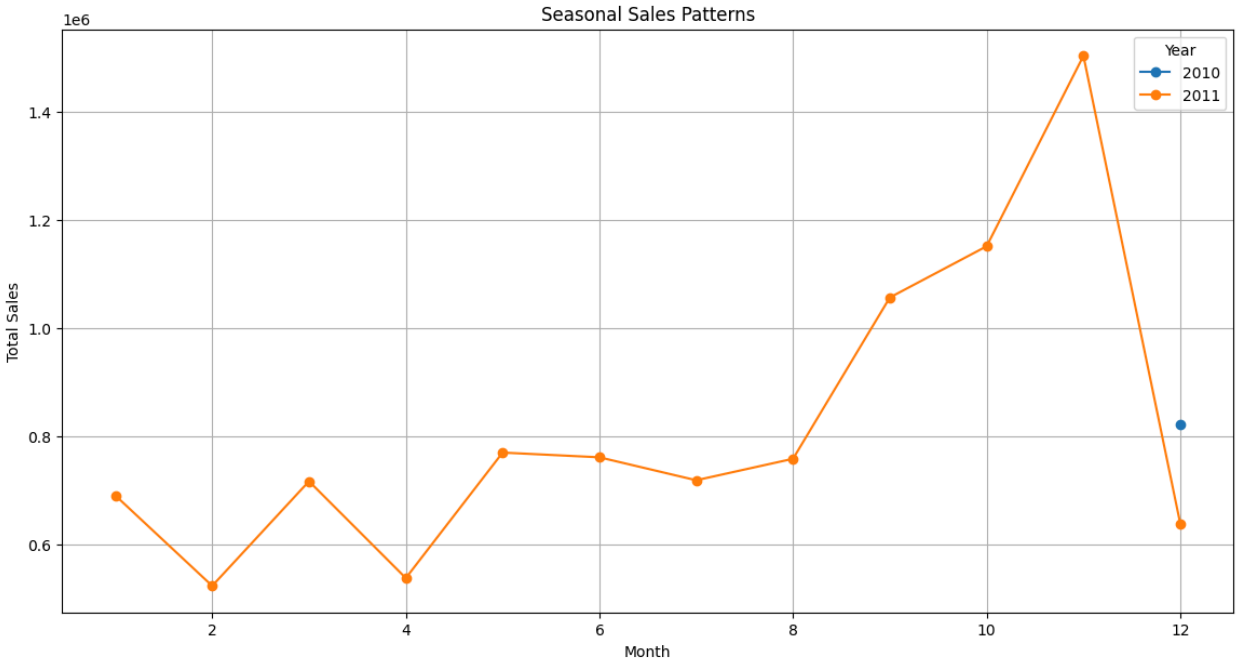
Weekly Sales Change Analysis:

	Year	Week	TotalPrice	Date	SalesChange
0	2010	48	184669.470	2010-11-29	NaN
1	2010	49	329108.220	2010-12-06	144438.750
2	2010	50	215357.040	2010-12-13	-113751.180
3	2010	51	92318.000	2010-12-20	-123039.040
4	2011	1	133429.720	2011-01-03	41111.720
5	2011	2	192991.950	2011-01-10	59562.230
6	2011	3	215227.010	2011-01-17	22235.060
7	2011	4	124534.220	2011-01-24	-90692.790
8	2011	5	125847.360	2011-01-31	1313.140

Observations:-

- **Early 2011:** Sales started strong, peaking in the third week of January with over ₹300,000 in sales. This was followed by a sharp decline in the subsequent weeks.
- I realized that February to August, sales fluctuated between ₹100,000 and ₹250,000.
- In September, there is a steady upward trend, culminating in the highest weekly sales of over ₹500,000 in December. This indicates a strong seasonal demand, likely driven by holiday shopping and sales events.

Seasonal Patterns in Sales



- The graph shows that sales decreased after the holiday season in December, **reaching the lowest point in February and March.**
- The mid-year months show fluctuations, likely driven by various promotions or external market conditions according to my opinion.

- **There is a noticeable rise in sales at the end of the year, especially in November. This suggests that people tend to shop more during this time of year.**

Customer Analysis

The main focus of this analysis is to understand customer behavior by examining the average number of transactions per customer, the average spend per customer, and identifying the most valuable customers based on total spending.

- The dataset shows an average of approximately 4.60 transactions per customer.

```
cleaned_data['Revenue'] = cleaned_data['Quantity'] * cleaned_data['UnitPrice']

transactions_per_customer = cleaned_data.groupby('CustomerID')['InvoiceNo'].nunique()
average_transactions_per_customer = transactions_per_customer.mean()
print(f"Average number of transactions per customer: {average_transactions_per_customer}")
```

```
Average number of transactions per customer: 4.603503918856616
```

- On average, each customer spends approximately ₹2453.22.

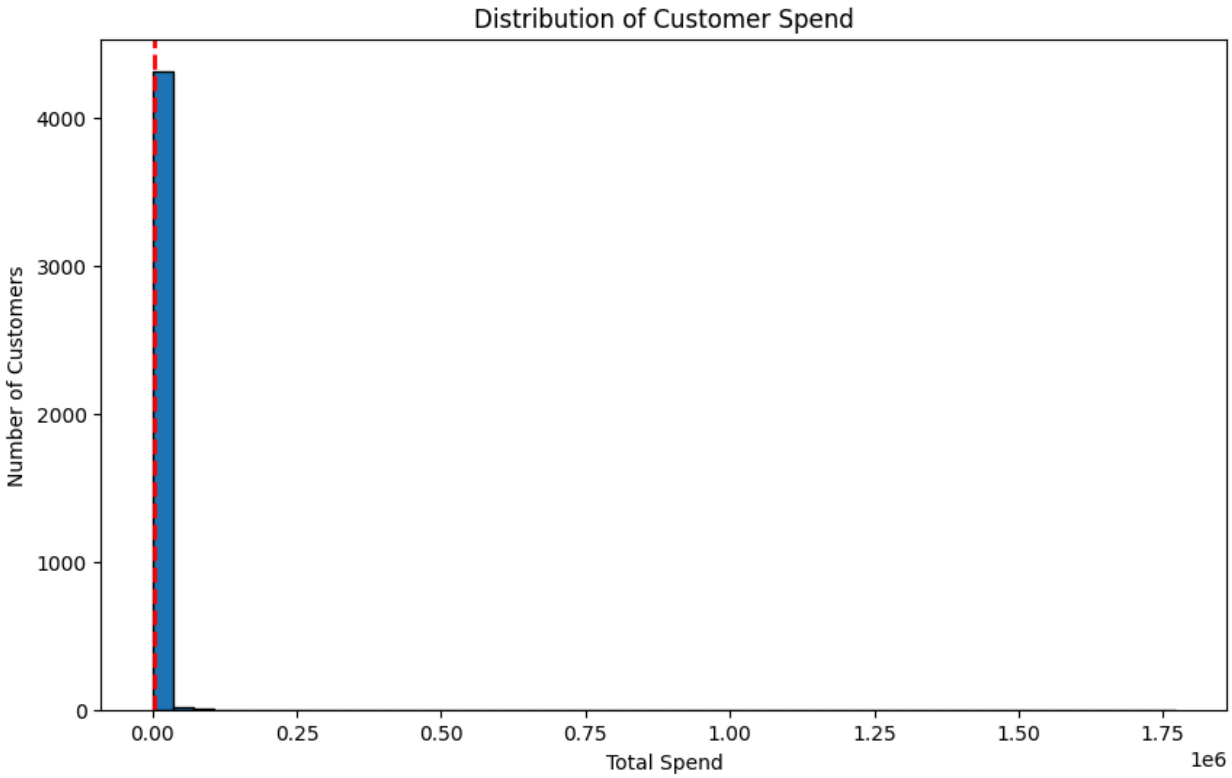
Identification of Most Valuable Customers

To identify the most valuable customers, I considered the top 10% of customers based on total spend.

To identify the top 10% of customers based on their spending, I set a threshold amount. Customers who spent more than this threshold are considered the most valuable. The top 10% of customers spend more than this amount, making them particularly valuable to the business. And this threshold is determined by looking at the total spend of all customers and finding the amount that marks the 90th percentile.

A total of 434 customers fall into the top 10% category

```
Top 10% of customers by total spend:
CustomerID
12346.0    77183.60
12347.0     4310.00
12357.0     6207.67
12359.0     6310.03
12362.0     5226.23
             ...
18223.0     6484.54
18225.0     5504.96
18226.0     5228.05
18229.0     7276.90
18251.0     4314.72
```



- I used histogram to visualize the distribution of total spend per customer. The plot shows a highly skewed distribution with the majority of customers spending low amounts, while a small number of customers spend significantly more.
- In my opinion this distribution is typical in retail data, where a small percentage of customers often contribute a large portion of the revenue.
- It's recommended to further analyze the purchasing patterns and preferences of these high-value customers to tailor marketing strategies, improve customer retention, and increase overall sales.

Geographical Analysis:

In this section I provide an analysis of sales performance by country, highlighting total revenue, year-over-year growth, and identifying countries with significant changes in sales.

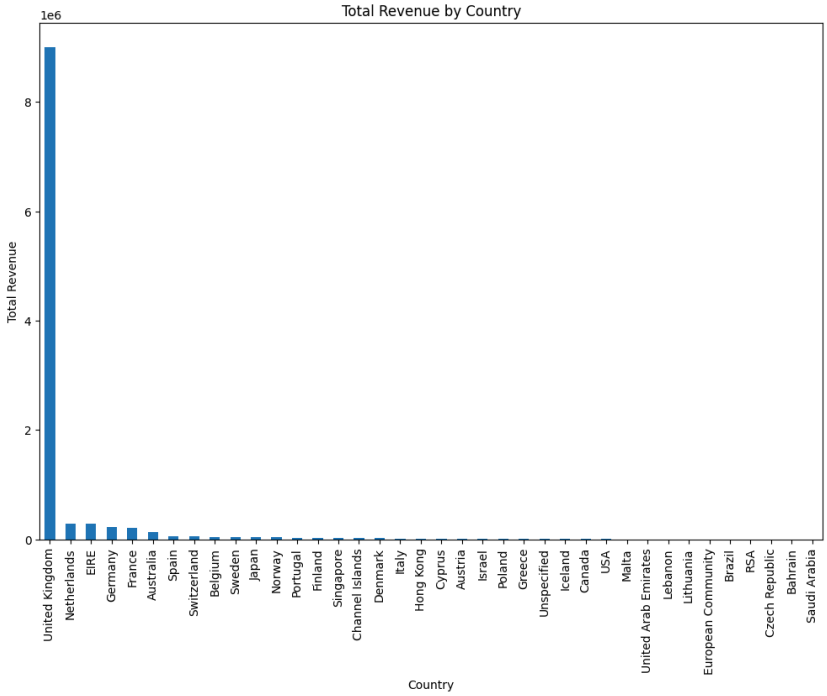
Total Revenue by Country:-

During my analysis I found that the United Kingdom leads in total revenue, followed by the Netherlands, EIRE, Germany, and France.

The total revenue generated by each country is summarized below:

Total Revenue by Country:

Country	
United Kingdom	9001744.094
Netherlands	285446.340
EIRE	283140.520
Germany	228678.400
France	209625.370
Australia	138453.810
Spain	61558.560
Switzerland	57067.600
Belgium	41196.340
Sweden	38367.830
Japan	37416.370
Norway	36165.440
Portugal	33683.050
Finland	22546.080
Singapore	21279.290
Channel Islands	20440.540
Denmark	18955.340
Italy	17483.240
Hong Kong	15483.000
Cyprus	13502.850
Austria	10198.680
Israel	8129.410
Poland	7334.650
Greece	4760.520
Unspecified	4740.940
Iceland	4310.000
Canada	3666.380
USA	3580.390
Malta	2725.590
United Arab Emirates	1902.280
Lebanon	1693.880
Lithuania	1661.060
European Community	1300.250
Brazil	1143.600
RSA	1002.310
Czech Republic	826.740
Bahrain	754.140
Saudi Arabia	145.920



Year-over-Year Sales Analysis:-

To understand sales trends over time, I analyzed the total revenue for 2010 and 2011:
Here I mention only top 5 country sales analysis:-

Country	2010 Revenue	2011 Revenue	Growth Rate
United Kingdom	₹746,082.22	₹8,255,661.87	10.07%
Netherlands	₹8,784.48	₹276,661.86	30.49%
EIRE	₹10,033.26	₹273,107.26	26.22%
Germany	₹15,205.74	₹213,472.66	13.04%
France	₹9,616.31	₹200,009.06	19.80%

Significant Sales Growth and Decline

Countries with notable changes in sales growth or decline are mentioned below:

Sales Growth:-

Country	**Growth**
Australia	141.42%
Channel Islands	54.23%
Netherlands	30.49%
Switzerland	41.73%

Sales Decline:

Lithuania: -100% (indicating a complete drop in revenue)

The analysis reveals that while some countries like the United Kingdom and Netherlands show strong sales growth, others such as Lithuania have seen significant declines. This information is crucial for strategic planning and resource allocation.

Recommendations:- Countries with high growth rates should be prioritized for further investment and marketing efforts and a deeper analysis is needed to understand the reasons behind the decline in countries like Lithuania.

Recommendations:

- **Target High-Value Customers**: Focus on the top 10% of customers who contribute significantly to the total revenue. Tailoring marketing strategies to the preferences and purchasing patterns of these high-value customers can enhance customer retention and increase overall sales.
- **Expand in High-Growth Markets**: Countries like Australia, the Netherlands, and Switzerland have shown strong sales growth. Prioritizing these markets for further investment and marketing efforts can help capture additional revenue.
- **Capitalize on Seasonal Trends**: The data shows a significant surge in sales during November. To leverage this trend, consider implementing targeted promotions, discounts, and marketing campaigns during this period to maximize sales and implement targeted promotional activities during early months(January to March).

Machine Learning Model

I developed a predictive model for determining the likelihood of a customer making a purchase in the next month. Based on the task, a **Logistic Regression model and Naive Bayes Classifier** is a suitable choice because the target variable (whether a customer will make a purchase or not) is binary. Let's Calculate the accuracy of both the models.

Logistic Regression:

Logistic Regression is effective for binary classification problems because it estimates the probability of a binary outcome using a logistic function.

In this the 'InvoiceDate' column was converted to datetime format to facilitate time-based calculations. I created a target variable, next month purchase, which is 1 if a purchase occurred in the next month and 0 otherwise. Categorical features, such as 'Country', were encoded using label encoder to convert them into numeric format suitable for modeling.

Feature Engineering:-

Engineered several engineered features are:

- **Recency**: Days since the last purchase.
- **Frequency**: Number of unique purchases.
- **Monetary**: Total spending.
- **Average Order Value**: Average amount spent per order.
- **Revenue**: Total revenue generated.

The data was split into training and testing sets using train_test_split, with 70% of the data for training and 30% for testing. I used SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance in the training set. This technique generates synthetic samples for the minority class to balance the dataset. Standardization was performed using standard scaler to ensure all features have a mean of 0 and a standard deviation of 1.

The best logistic regression model was selected based on the results of GridSearchCV.

Predicting Next Month's Purchase Using Gaussian Naive Bayes:

I used the Gaussian Naive Bayes model for its simplicity and effectiveness in classification problems. I have created a target variable, Next month purchase, which is 1 if a purchase occurred in the next month and 0 otherwise.

Here also I used SMOTE (Synthetic Minority Over-sampling Technique). This technique generates synthetic samples for the minority class to balance the dataset.

The Gaussian Naive Bayes model was trained on the standardized training data.

RESULTS:-

Model	Accuracy	Precision	Recall	F1- Score	ROC-AUC SCORE
Logistic Regression	0.46	0.22	0.85	0.35	0.67
Naive Bayes	0.63	0.58	0.91	0.71	0.62

SUMMARY:-

Based on the evaluation metrics, the Naive Bayes classifier demonstrated superior performance in predicting the likelihood of a customer making a purchase in the next month compared to the logistic regression model. With an accuracy of 0.63, a precision of 0.58, a recall of 0.91, an F1 score of 0.71, and an ROC-AUC score of 0.62, the Naive Bayes model effectively balances precision and recall, making it the best choice for this prediction task. Therefore, in my opinion, the Naive Bayes classifier is the most suitable model for predicting future customer purchases.

Additional Resources

For more details on the analysis, including the code and further analysis, please visit the GitHub repository. This repository contains all the scripts, data, and documentation necessary to reproduce the results and gain a deeper understanding of the methodology and findings.