ONLINE RETAIL DATA ANALYSIS PROJECT

Exploratory Analysis and Customer Segmentation using RFM and Clustering

Presented by: Ghada Sabry Mahmoud

Date: 20-5-2025

PROJECT WORKFLOW

- **Steps Covered:**
- Data Preparation
- Exploratory Data Analysis (EDA)
- Data Visualization
- Machine Learning - Customer Segmentation
- Summary & Recommendations

IMPORTING & GETTING INSIGHTS ABOUT THE DATA

```python
# import dataset

file_path = r"C:\Users\ghada\Downloads\Online_Retail (1).xlsx"

df = pd.read_excel(file_path)

print(df.head())
print(df.info())
print("\nDescribe values:\n", df.describe())
print("\nMissing Values:\n", df.isnull().sum())
```

✓ 38.6s

```python
print("Shape:", df.shape)
```

✓ 0.0s

```
Shape: (541909, 8)
```

```python
print("\nMissing Values:\n", df.isnull().sum())
```

✓ 0.0s

DATA CLEANING HIGHLIGHTS

- **Steps Covered:**
- Removed rows with missing `CustomerID`.
- Removed transactions with `Quantity <= 0` and `UnitPrice <= 0`.
- Created new column: `TotalPrice = Quantity * UnitPrice`.
- Converted `InvoiceDate` to datetime.
- Converted `CustomerID` to int.

```python
df_cleaned = df.dropna(subset=['CustomerID', 'Description'])
```

✓ 0.0s

```python
df_cleaned = df_cleaned[~df_cleaned['InvoiceNo'].astype(
    str).str.startswith('C')]
```

✓ 0.1s

```python
df_cleaned = df_cleaned[(df_cleaned['Quantity'] > 0) &
                        (df_cleaned['UnitPrice'] > 0)]
```

✓ 0.0s

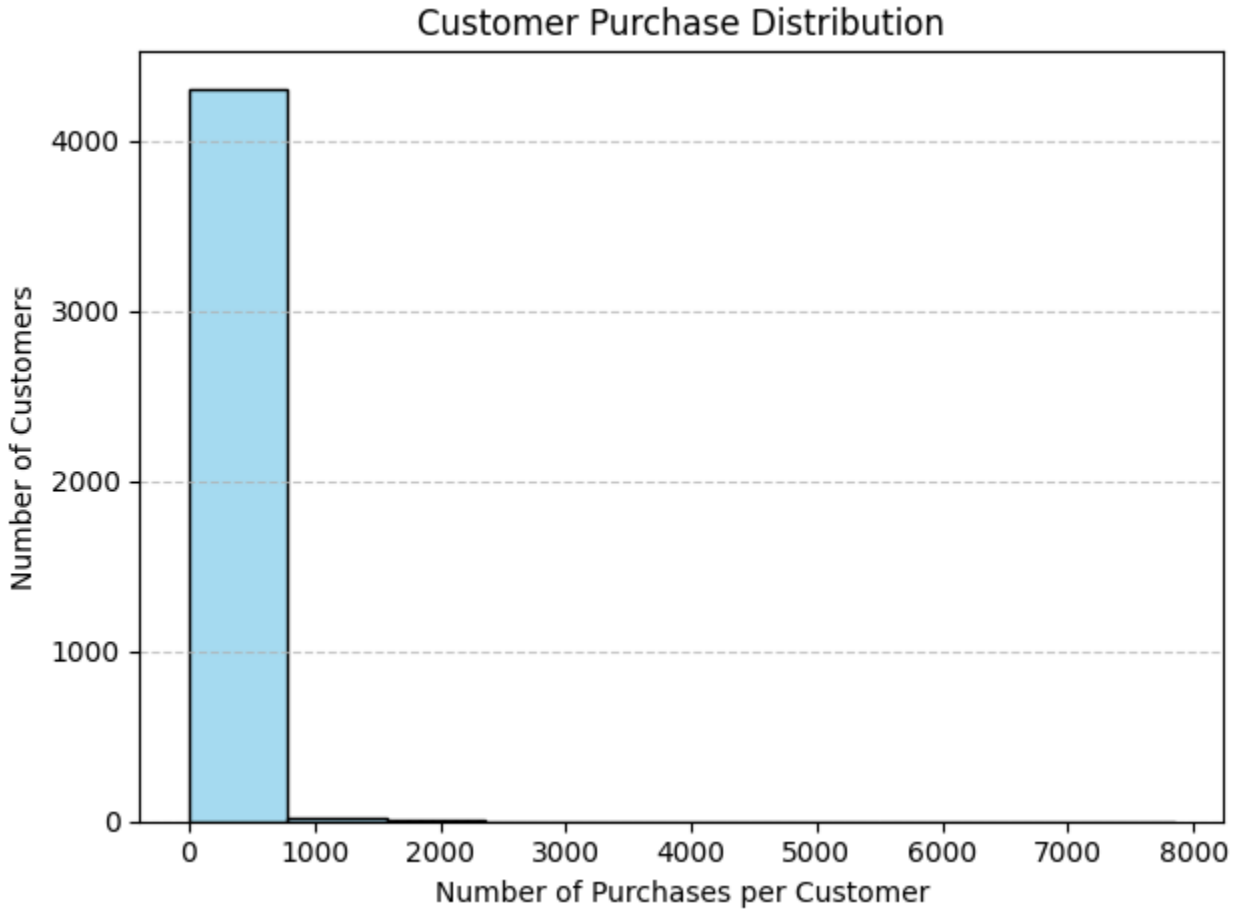
```python
df_cleaned['InvoiceDate'] = pd.to_datetime(df_cleaned['InvoiceDate'])
```

✓ 0.1s

```python
df_cleaned['CustomerID'] = df_cleaned['CustomerID'].astype(int)
```

✓ 0.0s

CUSTOMER PURCHASE DISTRIBUTION

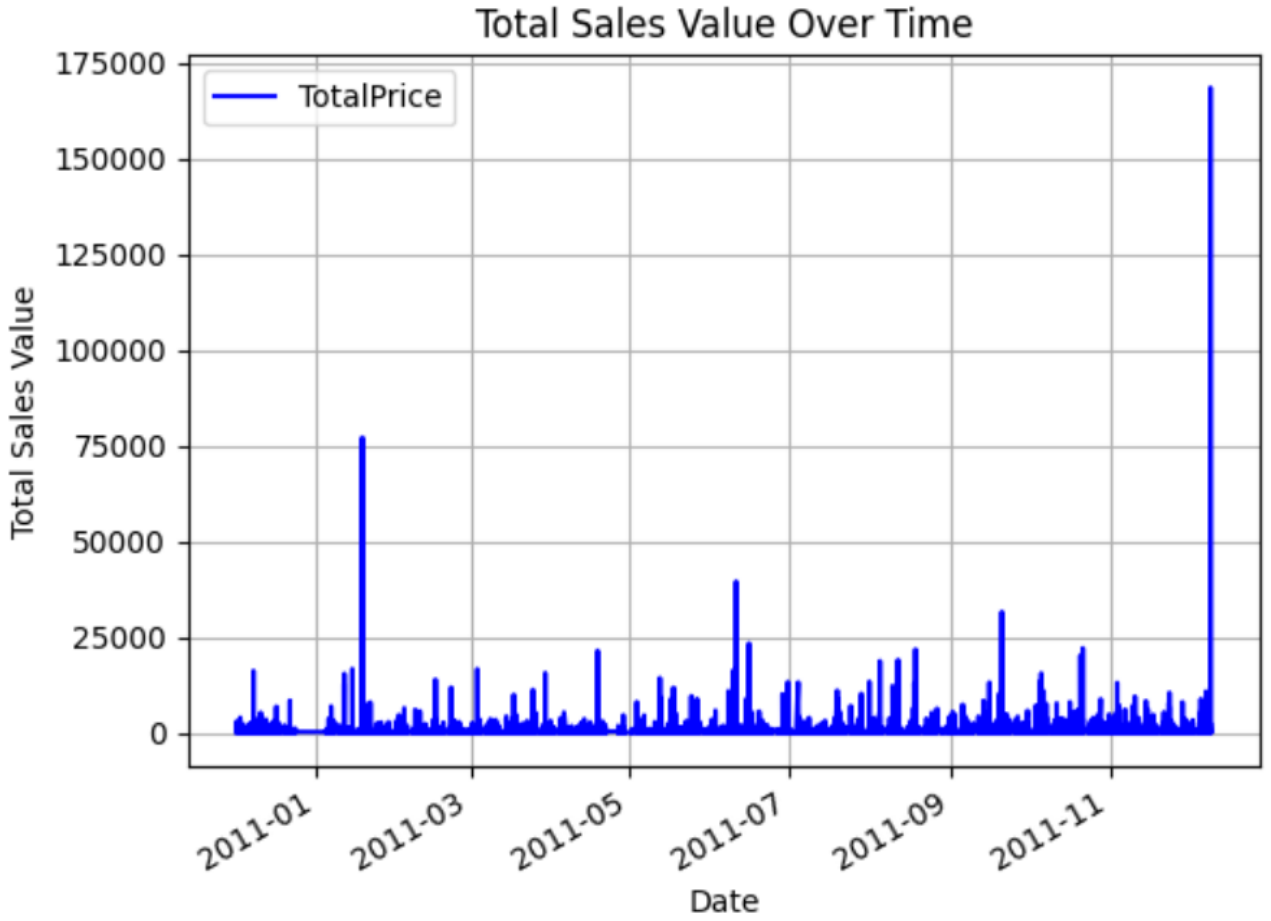


INSIGHT: MOST CUSTOMERS MADE FEWER THAN 50 PURCHASES. A SMALL GROUP MADE SIGNIFICANTLY MORE, SHOWING HIGH-VALUE REPEAT BUYERS.
RECOMMENDATION: CONSIDER LOYALTY PROGRAMS FOR HIGH-FREQUENCY CUSTOMERS.

CUSTOMER PURCHASE DISTRIBUTION

```
# 1. **Customer Distribution:**

customer_purchase_counts = df_cleaned['CustomerID'].value_counts()

sns.histplot(customer_purchase_counts, bins=10, color='skyblue')
plt.title('Customer Purchase Distribution')
plt.xlabel('Number of Purchases per Customer')
plt.ylabel('Number of Customers')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

SALES OVER TIME



INSIGHT: SALES SHOW SEASONAL SPIKES, ESPECIALLY AROUND THE END OF THE YEAR.

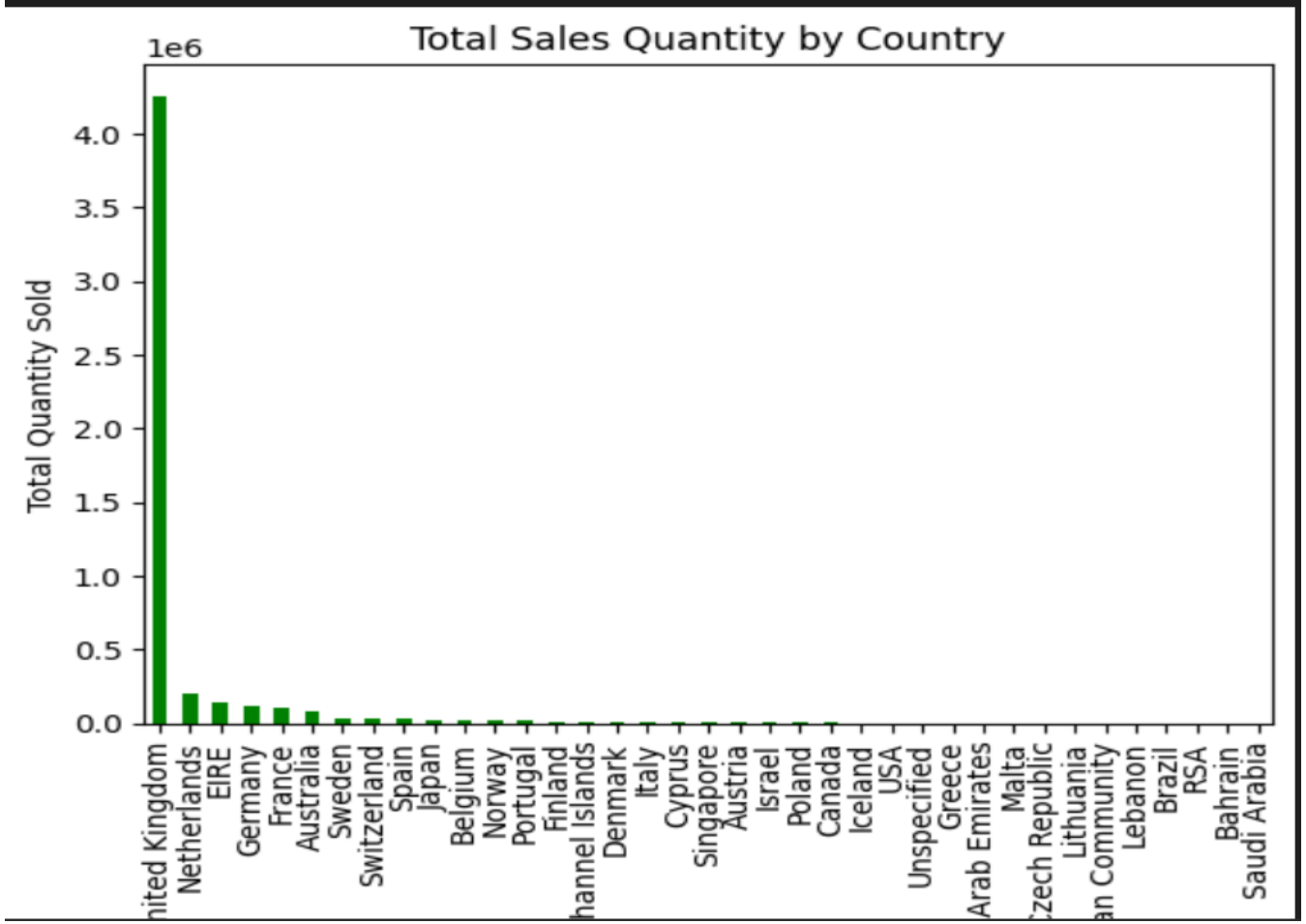
RECOMMENDATION: PREPARE FOR STOCK AND PROMOTIONS DURING PEAK MONTHS (E.G., NOVEMBER–DECEMBER).

SALES OVER TIME

```
# 2. **Sales Over Time:**

sales_over_time = df_cleaned.groupby(
    'InvoiceDate')['TotalPrice'].sum().reset_index()
sales_over_time.plot(x='InvoiceDate', y='TotalPrice',
                     kind='line', color='blue')

plt.title('Total Sales Value Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales Value')
plt.grid(True)
plt.show()
```

✓ 0.2s

COUNTRY-WISE SALES



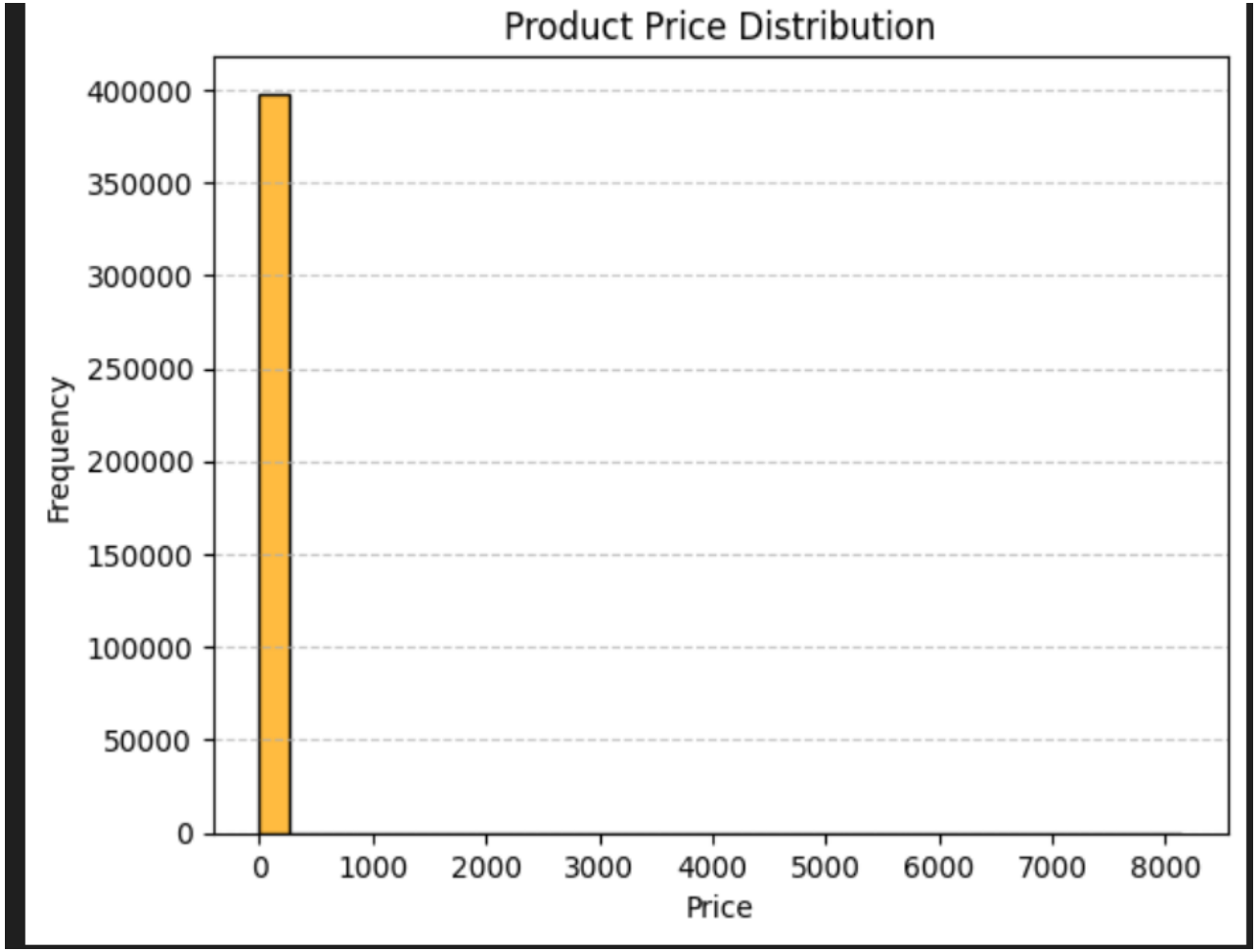
INSIGHT: THE UK IS BY FAR THE LARGEST CONTRIBUTOR TO SALES, FOLLOWED BY NETHERLANDS AND EIRE.
RECOMMENDATION: FOCUS MARKETING EFFORTS ON UK WHILE EXPLORING GROWTH OPPORTUNITIES IN SMALLER MARKETS.

COUNTRY-WISE SALES

```
# 3. **Country-wise Sales**:

# Group by 'Country' and sum 'Quantity'
country_sales = df_cleaned.groupby(
    'Country')['Quantity'].sum().sort_values(ascending=False)


country_sales.plot(kind='bar', color='green')
plt.title('Total Sales Quantity by Country')
plt.xlabel('Country')
plt.ylabel('Total Quantity Sold')
plt.show()
```

✓ 0.8s

PRODUCT PRICE DISTRIBUTION



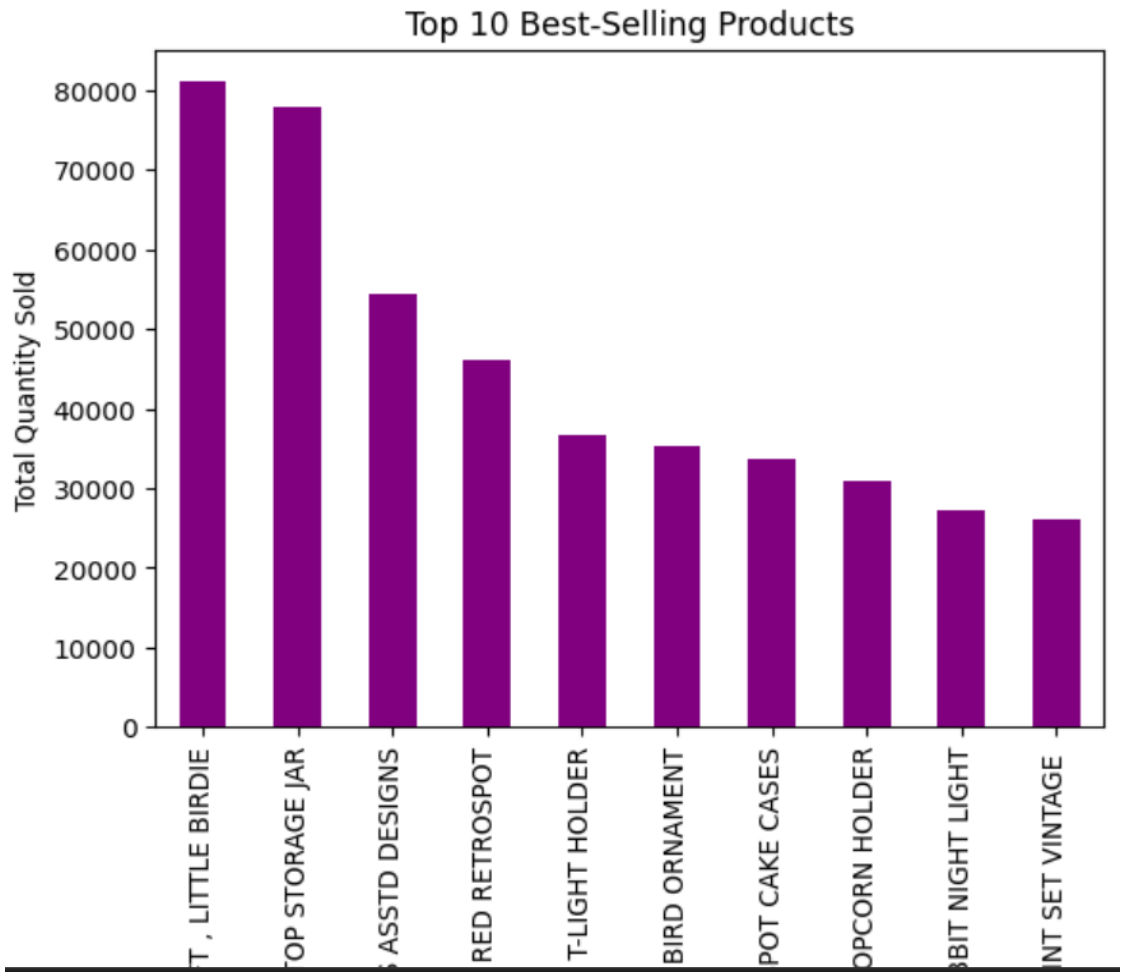
INSIGHT: MOST PRODUCTS ARE PRICED BELOW 20 UNITS. THERE ARE RARE BUT VERY EXPENSIVE OUTLIERS.
RECOMMENDATION: KEEP THE CORE OFFERING AFFORDABLE WHILE PROMOTING PREMIUM LINES FOR NICHE MARKETS.

PRODUCT PRICE DISTRIBUTIO

```
# 4. **Product Price Distribution:**

sns.histplot(df_cleaned['UnitPrice'], bins=30, color='orange')

plt.title('Product Price Distribution')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

✓ 1.0s

TOP SELLING PRODUCTS



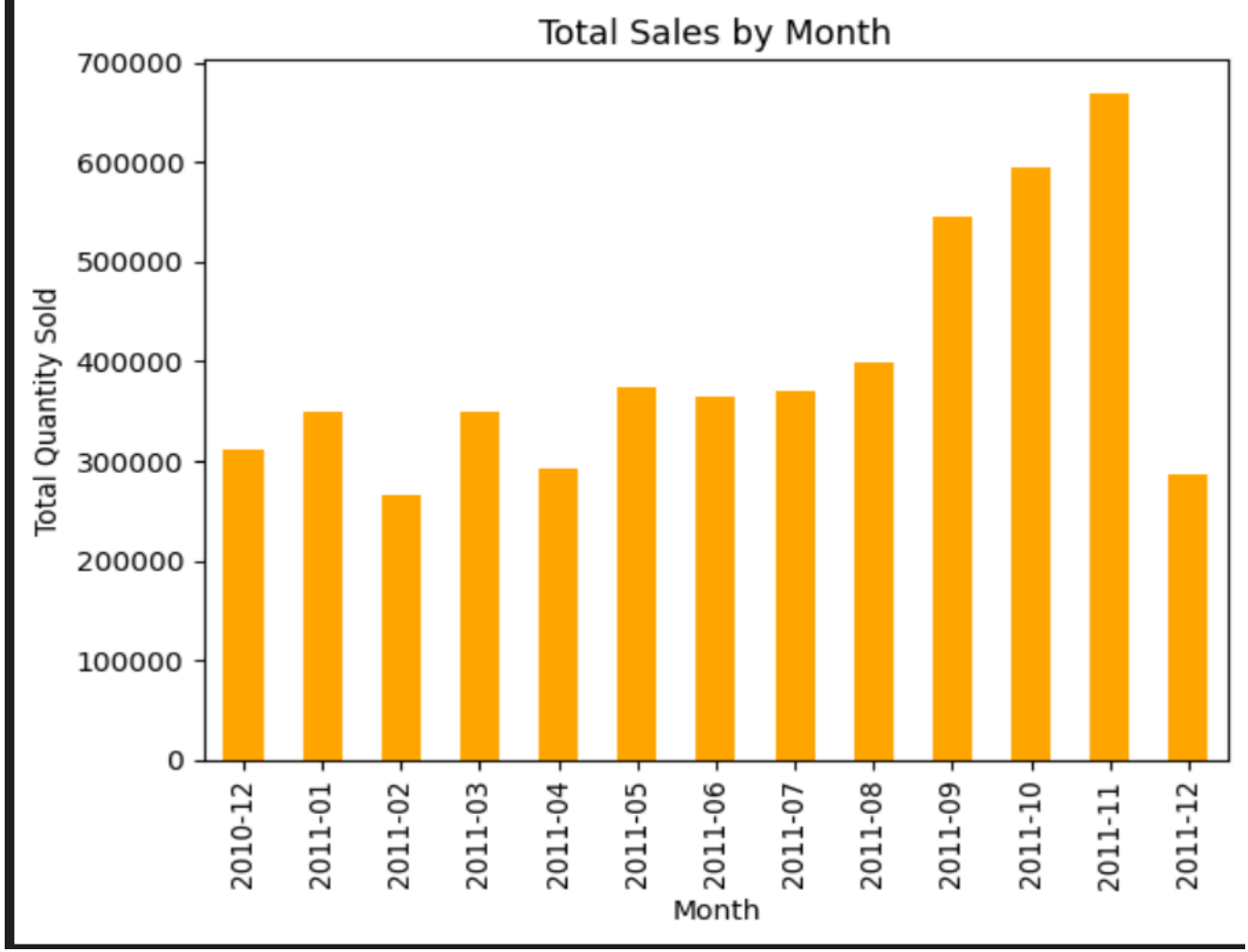
INSIGHT: GIFT-RELATED AND DECORATION ITEMS ARE BEST SELLERS.
RECOMMENDATION: STOCK MORE OF TOP PERFORMERS AND CROSS-SELL THEM WITH COMPLEMENTARY PRODUCTS.

TOP SELLING PRODUCTS

```
# 5. **Top Selling Products:**

# Group by 'Description' (product name) and sum quantities
top_products = df_cleaned.groupby('Description')['Quantity'].sum().nlargest(10)

top_products.plot(kind='bar', color='purple')

plt.title('Top 10 Best-Selling Products')
plt.xlabel('Product Description')
plt.ylabel('Total Quantity Sold')
plt.show()
```

✓ 0.4s

SALES TRENDS BY MONTH



MONTHLY INSIGHT: NOVEMBER IS THE BUSIEST SALES MONTH.

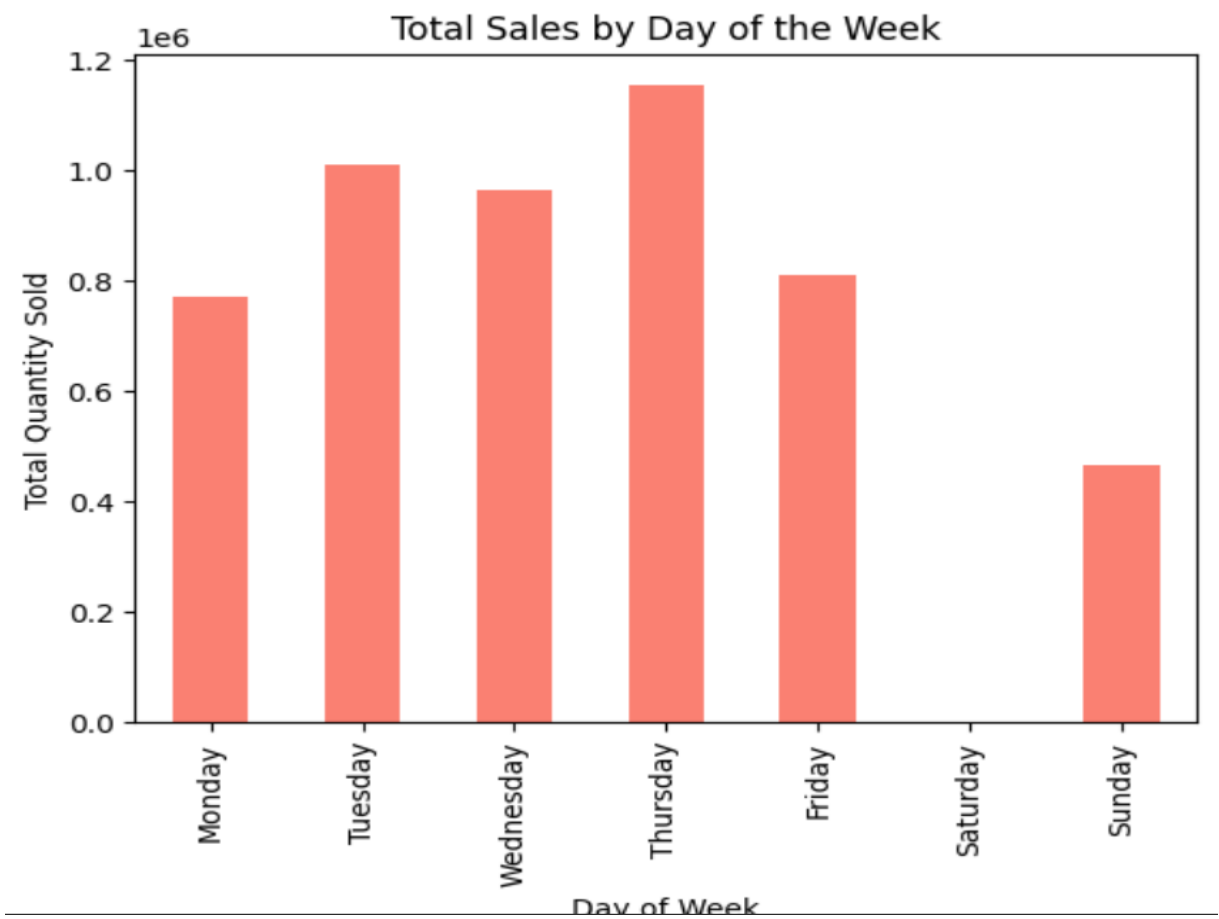
RECOMMENDATION: RUN SPECIAL OFFERS MID-WEEK AND PREPARE STOCK IN Q4.

SALES TRENDS BY MONTH

```
# 5. Analyze the sales trends over time. Identify the busiest months and days of the week in terms of sales.

# Identify the busiest months in terms of sales:

df_cleaned['Month'] = df_cleaned['InvoiceDate'].dt.to_period('M')

monthly_sales = df_cleaned.groupby('Month')['Quantity'].sum()



monthly_sales.plot(kind='bar', color='orange')
plt.title('Total Sales by Month')
plt.xlabel('Month')
plt.ylabel('Total Quantity Sold')
plt.show()
```

✓ 0.4s

SALES TRENDS BY DAY



WEEKLY INSIGHT: SALES PEAK MID-WEEK (TUESDAY–THURSDAY).

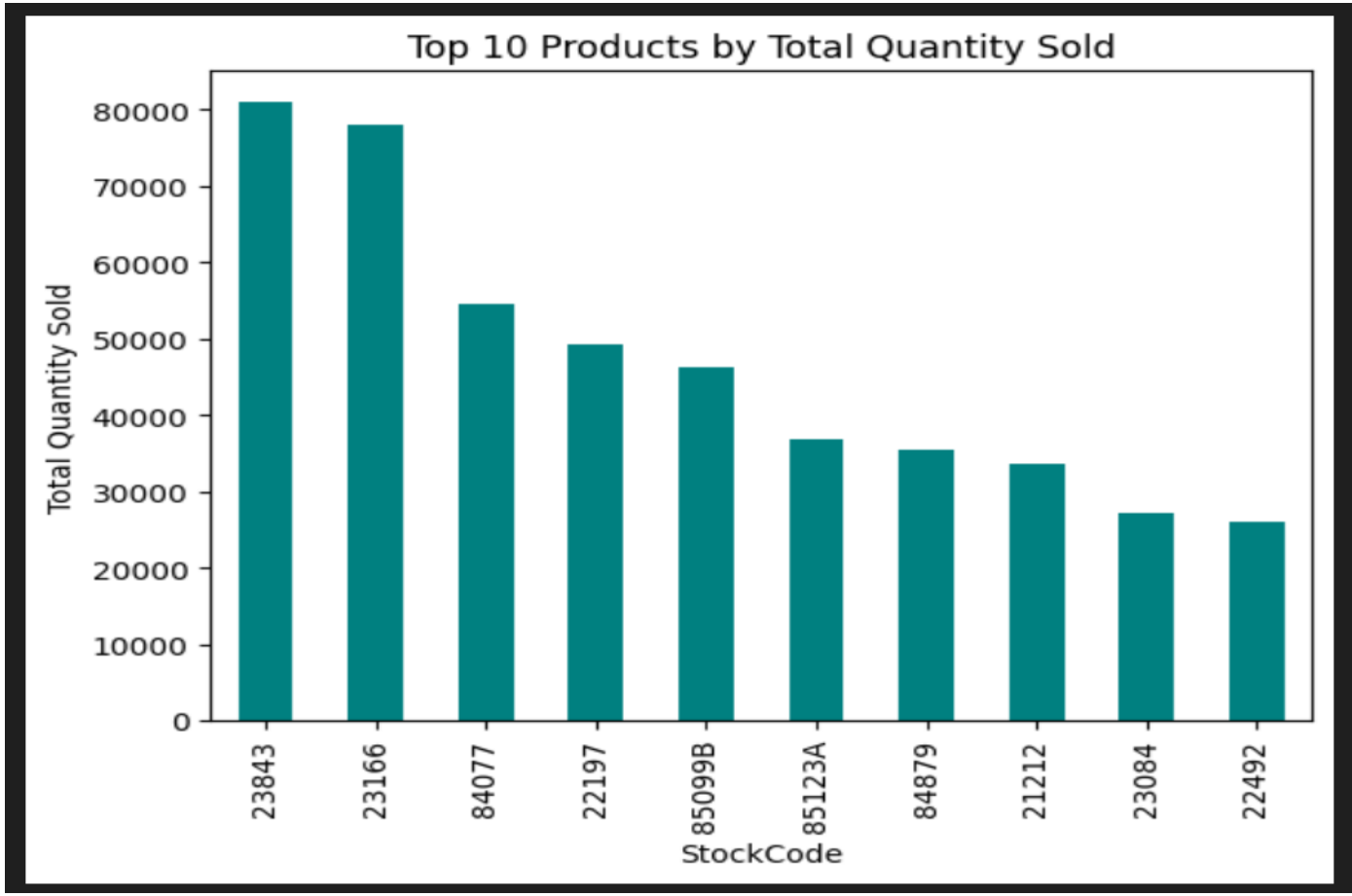
RECOMMENDATION: RUN SPECIAL OFFERS MID-WEEK

SALES TRENDS BY DAY

```
# 5. Analyze the sales trends over time. Identify the busiest months and days of the week in terms of sales.

# Identify the busiest  days of the week in terms of sales:

df_cleaned['DayOfWeek'] = df_cleaned['InvoiceDate'].dt.day_name()


day_of_week_sales = df_cleaned.groupby('DayOfWeek')['Quantity'].sum().reindex(
    ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], fill_value=0
)

day_of_week_sales.plot(kind='bar', color='salmon',)
plt.title('Total Sales by Day of the Week')
plt.xlabel('Day of Week')
plt.ylabel('Total Quantity Sold')
plt.show()
```

✓ 0.5s

STOCKCODE QUANTITY SOLD



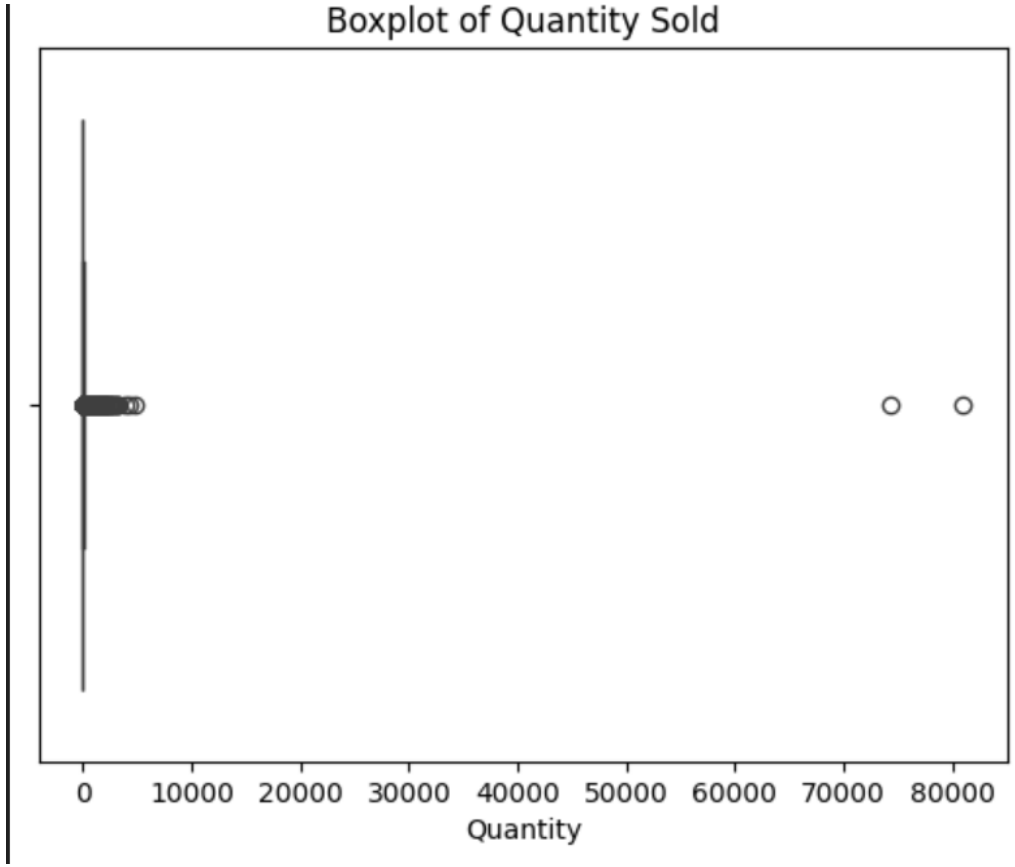
INSIGHT: FEW STOCKCODES DOMINATE IN VOLUME SOLD.
RECOMMENDATION: KEEP TOP STOCKCODES WELL-STOCKED AND MONITOR TRENDS FOR EMERGING PRODUCTS.

STOCKCODE QUANTITY SOLD

```
# 6. create a bar chart to visualize the total quantity of each product (StockCode)
# sold in the provided sample dataset? Additionally, label the x-axis with the StockCodes
# and provide a meaningful title and axis labels for clarity in interpretation.

# Group by 'StockCode' and sum 'Quantity'
product_sales = df_cleaned.groupby('StockCode')['Quantity'].sum()

top_products = product_sales.nlargest(10)



top_products.plot(kind='bar', color='teal')

plt.title('Top 10 Products by Total Quantity Sold')
plt.xlabel('StockCode')
plt.ylabel('Total Quantity Sold')
plt.show()
```

✓ 0.3s

OUTLIER DETECTION



OUTLIER DETECTION

INSIGHT: SOME EXTREMELY HIGH PRICES AND QUANTITIES INDICATE POSSIBLE DATA ERRORS OR SPECIAL CASES.
RECOMMENDATION: FLAG THESE FOR REVIEW; OPTIONALLY EXCLUDE FROM FINANCIAL KPIS.

OUTLIER DETECTION

```
# 7. Identify any outliers or anomalies in the dataset and discuss their potential impact on the analysis.

# Boxplot for UnitPrice:


sns.boxplot(x=df_cleaned['UnitPrice'])
plt.title('Boxplot of Product Unit Prices')
plt.xlabel('Unit Price')
plt.show()

# . Boxplot for Quantity:

sns.boxplot(x=df_cleaned['Quantity'])
plt.title('Boxplot of Quantity Sold')
plt.xlabel('Quantity')
plt.show()
```

✓ 2.7s

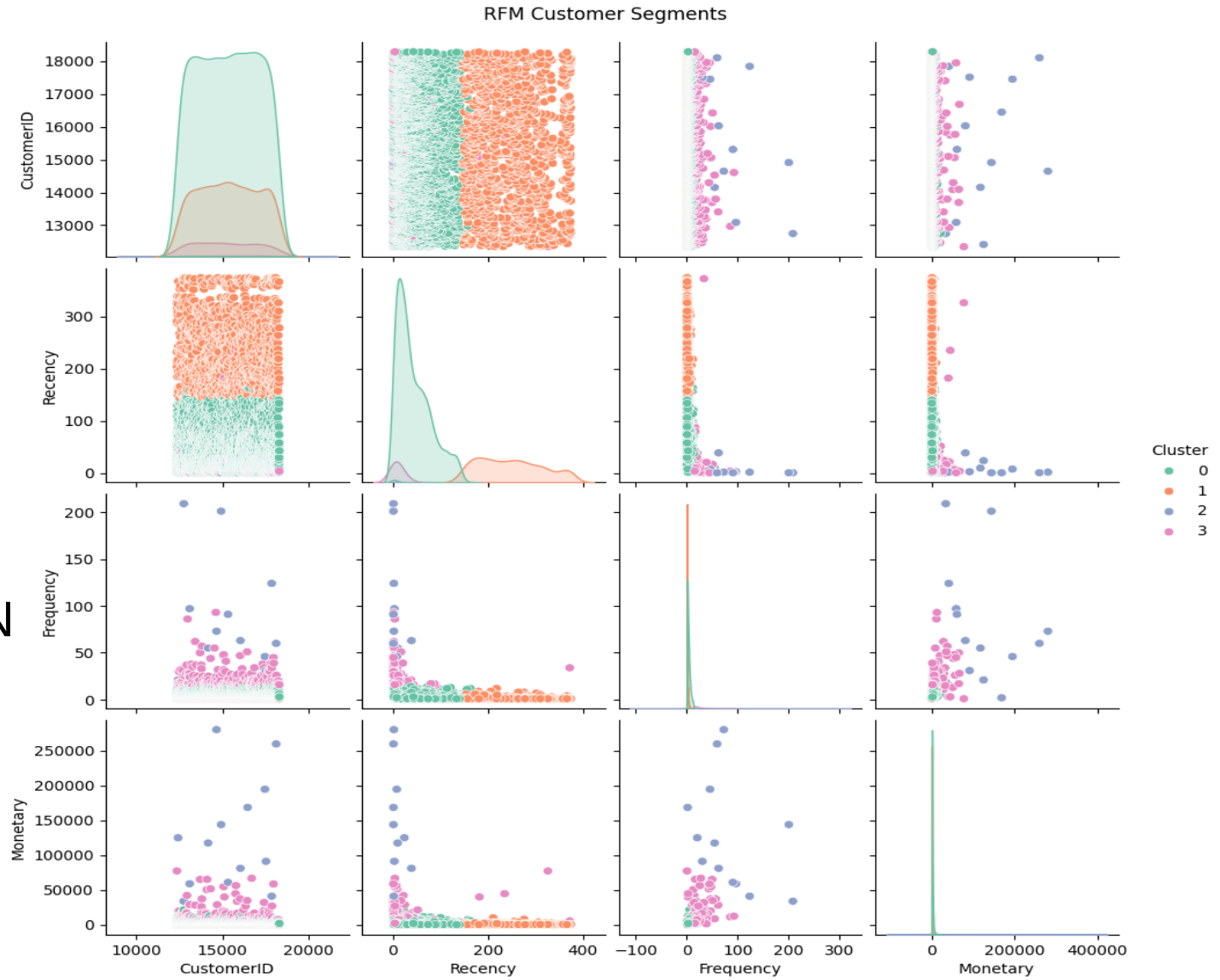
MACHINE LEARNING: RFM CLUSTERING

- **Steps Covered:**
- RFM = Recency, Frequency, Monetary
- Used KMeans clustering (4 segments)

- Segment Examples:
  - **Cluster 0**: Loyal, frequent, high spenders
  - **Cluster 2**: New or dormant customers
  - **Use Case:** Target campaigns based on cluster behavior (e.g., reward loyal, re-engage dormant).

RFM CLUSTER VISUALIZATION

INSIGHT: CLUSTER SEPARATION SHOWS DIVERSE SPENDING AND BEHAVIOR.
RECOMMENDATION: USE CLUSTERS FOR PERSONALIZATION, SUCH AS EMAILS, DISCOUNTS, OR REACTIVATION CAMPAIGNS.



- CLUSTER 0 (E.G., GREEN):
  - LIKELY CUSTOMERS WITH **LOW RECENCY, LOW FREQUENCY, LOW MONETARY**
  - TRANSLATION: *"GHOSTED YOU MONTHS AGO AND BARELY SPENT ANYTHING."*
- CLUSTER 1 (E.G., ORANGE):
  - LIKELY **RECENT** BUYERS, MAYBE NOT FREQUENT, BUT
- CLUSTER 2 (E.G., BLUE):
  - **VERY HIGH FREQUENCY AND MONETARY** VALUES
  - TRANSLATION: *"YOUR VIPS — ROLL OUT THE RED CARPET AND OFFER LOYALTY PERKS!"*
- CLUSTER 3 (E.G., PINK):
  - SOMEWHERE IN BETWEEN — MAYBE REGULARS WHO DON'T SPEND A LOT
  - TRANSLATION: *"THEY LIKE YOU, BUT DON'T*

SUMMARY & RECOMMENDATIONS

SUMMARY:

- CLEANED AND EXPLORED 390K+ TRANSACTIONS.
- IDENTIFIED SALES TRENDS, CUSTOMER BEHAVIOR, AND PRODUCT PERFORMANCE.
- APPLIED RFM SEGMENTATION FOR TARGETED MARKETING.

RECOMMENDATIONS:

- INVEST IN TOP PRODUCTS AND CUSTOMER LOYALTY.
- MONITOR SEASONAL PEAKS.
- LEVERAGE ML SEGMENTATION FOR PERSONALIZED MARKETING.

THANK YOU !