Capstone Project (Python): Segmenting Customers Based on Purchasing Behavior

BY

THARAKH GEORGE CHACKO

A Project Report Submitted

in

Partial Fulfillment of the

Requirements for the Data Science and Artificial Intelligence (DSAI) Certification Course

at

Intellipaat

CONTENTS

FIGURES

PROBLEM STATEMENT:

In today's competitive retail environment, particularly in the e-commerce sector, understanding customer behavior is not just beneficial; it is crucial for survival and growth. An online retail store seeks to deepen its understanding of how customers interact with its platform, what drives their purchasing decisions, and ultimately, how these insights can be leveraged to enhance the customer experience and boost business performance.

The core challenge addressed by this project revolves around the multifaceted nature of customer purchase patterns which are influenced by an array of factors including product preferences, seasonal trends, pricing, marketing effectiveness, and economic conditions. Each customer's journey through the online store is a complex dataset of interactions, choices, and decisions that, when aggregated and analyzed, reveal patterns that are critical to business strategy and operational adjustments.

For this online retailer, the problem is twofold:

1. **Lack of Insight into Customer Segmentation and Behavior:** Without a clear understanding of the different types of customers and their purchasing behaviors, the retailer struggles to offer targeted marketing and personalized experiences. This can lead to inefficient marketing spend, lower customer satisfaction, and ultimately, reduced loyalty and revenue.
2. **Inefficient Utilization of Available Data:** The retailer possesses extensive data on customer interactions and transactions. However, the challenge lies in transforming this data into actionable insights. The current lack of sophisticated data analysis means potential insights and opportunities for improvement are left undiscovered.

The implications of these challenges are broad and significantly impact various aspects of the business. For instance, inventory management may suffer from inefficiencies due to a lack of understanding of buying patterns, leading to either overstock or stockouts. Marketing campaigns might not be effectively tailored to the desires and needs of different customer segments, resulting in lower conversion rates and wasted resources.

Additionally, customer retention could be at risk if the retailer continues to operate without a nuanced understanding of customer behavior. In an era where personalization is key to retaining customers, the absence of a tailored approach can lead customers to competitors who offer more personalized experiences.

Thus, the problem statement for this project is: **"How can the online retail store leverage its existing data to gain a deep understanding of customer purchase patterns and behaviors to improve segmentation, personalize customer interactions, optimize marketing strategies, and enhance overall business performance?"**

This problem statement sets the stage for exploring analytical methods and machine learning techniques to segment customers, predict future purchasing behaviors, and identify the key drivers of customer engagement and sales.

PROJECT OBJECTIVE:

The primary objective of this project is to harness advanced analytical techniques to decipher complex customer data, enabling the online retailer to enhance strategic decision-making and operational efficiency. By analyzing historical purchase data, the project aims to provide a granular understanding of customer behaviors and preferences, which will drive more informed, data-driven strategies across various business domains, including marketing, sales, customer service, and inventory management.

Key Goals of the Project:

1. **Customer Segmentation:**
    - **Objective:** To classify customers into distinct groups based on similar behaviors, preferences, and purchasing patterns. This segmentation will allow the retailer to tailor marketing messages, predict future buying behaviors, and identify the most profitable customer segments.
    - **Benefit:** Improved targeting in marketing campaigns, enhanced customer engagement through personalized experiences, and increased effectiveness in promotional strategies.
2. **Trend Analysis:**
    - **Objective:** To identify and analyze purchasing trends over time, including seasonality, product preferences, and response to pricing or marketing strategies.
    - **Benefit:** Enables the retailer to optimize stock levels according to predicted demand, plan promotions or discounts more effectively, and adjust pricing strategies to maximize revenue.
3. **Predictive Analytics:**
    - **Objective:** To employ predictive models to forecast future buying behaviors, potential churn rates, and customer lifetime value.
    - **Benefit:** Facilitates proactive business strategies, enhances customer retention efforts, and helps in allocating resources more efficiently to where they will generate the highest return.
4. **Optimization of Marketing Efforts:**
    - **Objective:** To leverage insights gained from customer data to optimize marketing efforts, ensuring that the right products are marketed to the right customers at the right time through the right channels.
    - **Benefit:** Increases conversion rates, reduces marketing waste, and elevates the overall effectiveness of marketing campaigns.
5. **Enhancement of Customer Experience:**
    - **Objective:** To understand the pain points and the drivers of satisfaction among different customer segments, enabling the retailer to enhance the customer experience.
    - **Benefit:** Leads to higher customer satisfaction and loyalty, which are crucial for long-term success in a competitive retail landscape.
6. **Operational Efficiency:**
    - **Objective:** To streamline operations based on insights derived from data analytics, such as optimized inventory management and improved supply chain decisions.

- **Benefit:** Reduces operational costs, minimizes inventory wastage, and ensures product availability aligning with customer demands.

The culmination of this project will be the development and implementation of a dashboard or a set of reporting tools that provide ongoing insights into customer behaviors, sales trends, and operational efficiency. This tool will allow the retailer to make agile decisions, adapting quickly to changes in customer preferences and market conditions.

By achieving these objectives, the project will not only support the online retailer in enhancing customer satisfaction and loyalty but will also drive significant improvements in profitability and market competitiveness. The insights gained through this project are expected to set a foundation for continuous learning and adaptation, leveraging data analytics as a core component of the retailer's strategic toolkit.

DATA DESCRIPTION:

The dataset used for this project is an extensive collection of transactions from an online retail store. This dataset, referred to as `OnlineRetail.csv`, consists of 5,41,909 entries and is structured into 8 distinct columns, each representing a specific attribute of the transactional records. The data covers a period of activity within the store and includes a wide range of products purchased across various regions. Here is a detailed breakdown of each column in the dataset:

1. **InvoiceNo**:
    - **Description**: A unique identifier for each transaction.
    - **Type**: Alphanumeric.
    - **Relevance**: Crucial for distinguishing individual transactions, which may include multiple product purchases.
2. **StockCode**:
    - **Description**: The product identifier specific to each item available in the inventory.
    - **Type**: Alphanumeric.
    - **Relevance**: Essential for identifying and analyzing the sales performance of each product.
3. **Description**:
    - **Description**: The name or description of the product.
    - **Type**: Text.
    - **Relevance**: Allows for qualitative analysis of product types, grouping similar products, and understanding product range diversity.
4. **Quantity**:
    - **Description**: The quantities of each product purchased per transaction.
    - **Type**: Integer.
    - **Relevance**: Key for assessing sales volume, understanding purchasing patterns, and conducting demand forecasting.
5. **InvoiceDate**:
    - **Description**: The date and time when each transaction occurred.
    - **Type**: DateTime.
    - **Relevance**: Vital for time series analysis, identifying seasonal trends, and evaluating the timing of purchases.
6. **UnitPrice**:
    - **Description**: The price per unit of each product.
    - **Type**: Numeric.
    - **Relevance**: Necessary for revenue calculations, profitability analysis, and pricing strategy assessments.
7. **CustomerID**:
    - **Description**: A unique identifier for each customer.
    - **Type**: Numeric.
    - **Relevance**: Enables customer-specific analysis, segmentation, and behavioral profiling.
8. **Country**:
    - **Description**: The country or region where each customer resides.
    - **Type**: Text.

- **Relevance**: Important for geographical segmentation, market penetration analysis, and regional sales performance evaluation.

```
data.shape
```

```
(541909, 8)
```

```
print(f"There are {data.shape[0]} observations for {data.shape[1]} predictors.")
```

```
There are 541909 observations for 8 predictors.
```

```
print('Total no of unique elements in each predictor')
print('-----------------------------------------------')
for i in data.columns:
    print(f'{i} : {data[i].nunique()} elements')
```

```
Total no of unique elements in each predictor
-----------------------------------------------
InvoiceNo : 25900 elements
StockCode : 4070 elements
Description : 4223 elements
Quantity : 722 elements
InvoiceDate : 23260 elements
UnitPrice : 1630 elements
CustomerID : 4372 elements
Country : 38 elements
```

Figure 1.1: Summary of Dataset

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

Figure 1.2: Summary of Dataset

Data Quality and Structure: The dataset is expected to contain some typical data quality issues such as missing values, duplicate entries, and potential outliers in terms of quantity and

pricing that need to be addressed during preprocessing. Furthermore, the need to standardize descriptions and ensure consistency across similar products with different stock codes or descriptions is anticipated.

Usage in the Project: This dataset provides a comprehensive overview of the transactions processed by the online retailer, making it an asset for extracting actionable insights. Through detailed analysis, the data will help unveil patterns in customer buying behavior, product performance, and sales trends over time. These insights will then inform various strategic decisions such as promotional strategies, inventory management, and customer relationship initiatives designed to enhance customer satisfaction and business profitability.

In summary, the dataset is integral to understanding and optimizing the online retailer's operations and strategic approach, offering a granular view of the commercial dynamics at play.

DATA PREPROCESSING STEPS AND INSPIRATION:

Data preprocessing is a critical phase in any data science project as it ensures the reliability and quality of insights derived from the analysis. For this online retail project, the preprocessing steps are designed to clean and transform the raw transactional data into a format suitable for further analysis and modeling. Here is an extensive overview of the preprocessing activities undertaken:

1. Data Cleaning:

- **Handling Missing Values**: The dataset is scrutinized for missing or null entries, particularly in key columns such as CustomerID and Description. Records with missing CustomerID or Description are removed because they are essential for identifying transactions and analyzing customer purchasing patterns.

```
data.isnull().sum()
```

```
InvoiceNo          0
StockCode          0
Description     1439
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID     127216
Country            0
dtype: int64
```

```
print(f"Duplicated entries before removal: {data.duplicated().sum()}")
data.drop_duplicates(inplace=True)
```

```
Duplicated entries before removal: 4989
```

```
data[data.isin(["NaN","missing","?","??"]).any(axis=1)].shape[0]
```

```
56
```

```
data = data[data.isin(["NaN","missing","?","??"]).any(axis=1) == False]
```

```
data[data.isin(["NaN","missing","?","??"]).any(axis=1)].shape[0]
```

```
0
```

Figure 2: Removing Some Null Values, Missing Values and Duplicates

```
print('Data shape after cleaning the data')
print('---------------------------------------')
data["Description"] = data["Description"].str.lower().str.strip()
incorrect_items = ["amazon fee", "samples", "postage", "packing charge", "manual", "discount",
                   "adjust bad debt", "bank charges", "cruk commission", "next day carriage"]
data = data[~data['Description'].isin(incorrect_items)]
data = data[(data['UnitPrice'] > 0) & (data['Quantity'] > 0)]
print(data.shape)
```

```
Data shape after cleaning the data
---------------------------------------
(498562, 8)
```

Figure 3: Removing Invalid Descriptions

- **Removing Duplicates**: Duplicate entries are identified and eliminated to prevent skewed analysis results. This ensures that each transaction is unique and accurately represented in the dataset.
- **Standardizing Text Data**: The Description field, which contains textual descriptions of products, is standardized by converting to lowercase and stripping any leading or trailing whitespace. This uniformity helps in reducing redundancy and discrepancies in the dataset due to textual case differences or accidental spaces.

2. Data Transformation:

- **Standardization of Product Descriptions**: To ensure consistency across similar products that might be listed under slightly different descriptions or stock codes, a mapping is established. For each unique stock code, a primary description is identified (the mode of descriptions). All entries with the same stock code are then updated to this primary description.
- **Standardization of Stock Codes**: Similarly, each description is mapped to a primary stock code to address any inconsistencies where the same product might have been entered under different codes.

3. Feature Engineering:

- **Total Price Calculation**: A new feature, TotalPrice, is created by multiplying the Quantity of products purchased by the UnitPrice. This feature is crucial for revenue analysis and understanding the monetary impact of transactions.
- **Removing Outliers**: Outliers can skew the results of data analysis, particularly in fields like Quantity, UnitPrice, and TotalPrice. The dataset is filtered to remove outliers using the Interquartile Range (IQR) method, which involves calculating the first (Q1) and third (Q3) quartiles, and then defining acceptable bounds as [Q1 - 1.5*IQR, Q3 + 1.5*IQR]. Transactions falling outside these bounds are considered outliers and are removed.

4. Date Handling:

- **Invoice Date Conversion**: The InvoiceDate column is converted from a string format to DateTime, which facilitates time-series analysis and allows for resampling data based on periods like months or days.

- **Filtering Data by Date**: Entries from specific periods (e.g., excluding transactions from December 2011) are filtered out to focus the analysis on complete months or to prevent biased insights due to incomplete data.

```
print(f'The no of unique items in Description column : {data["Description"].nunique()}')
print(f'The no of unique items in StockCode column : {data["StockCode"].nunique()}')
if data["Description"].nunique() != data["StockCode"].nunique():
    print('These columns totals are not consistent with each other')
else:
    print('These columns totals are consistent with each other')
```

```
The no of unique items in Description column : 4004
The no of unique items in StockCode column : 3910
These columns totals are not consistent with each other
```

```
print('Standardizing Description by StockCode: Initiated')
for stock_code in data['StockCode'].unique():
    primary_description = data[data['StockCode'] == stock_code]['Description'].mode()[0]
    data.loc[data['StockCode'] == stock_code, 'Description'] = primary_description
print('Standardizing Description by StockCode: Completed')
print('----------------------------------------')
print('Standardizing StockCode by Description: Initiated')
for description in data['Description'].unique():
    primary_code = data[data['Description'] == description]['StockCode'].mode()[0]
    data.loc[data['Description'] == description, 'StockCode'] = primary_code
print('Standardizing StockCode by Description: Completed')
```

```
Standardizing Description by StockCode: Initiated
Standardizing Description by StockCode: Completed
----------------------------------------
Standardizing StockCode by Description: Initiated
Standardizing StockCode by Description: Completed
```

Figure 4.1 Standardisation of Stock Code and Description

```
print(f'The no of unique items in Description column : {data["Description"].nunique()}')
print(f'The no of unique items in StockCode column : {data["StockCode"].nunique()}')

if data["Description"].nunique() != data["StockCode"].nunique():
    print('These columns totals are not consistent with each other')
else:
    print('These columns totals are consistent with each other')
```

```
The no of unique items in Description column : 3777
The no of unique items in StockCode column : 3777
These columns totals are consistent with each other
```

Figure 4.2 Standardisation of Stock Code and Description

The meticulous preprocessing of data is inspired by the need to ensure accuracy and reliability in the analysis outcomes. By cleansing the data and creating a robust dataset, the project aims to uncover genuine patterns in customer behavior and sales trends that can drive

strategic business decisions. This approach not only enhances the quality of insights but also aligns the analysis with the business objectives of maximizing revenue, optimizing inventory management, and improving customer satisfaction.

By undertaking these preprocessing steps, the project sets a strong foundation for subsequent analytical tasks, including customer segmentation, trend analysis, and predictive modeling, ensuring that the insights generated are both actionable and aligned with the strategic goals of the online retail firm.

```
fig, axes = plt.subplots(1, 3, figsize=(20, 5))
fig.suptitle("Visualization of Outliers", size=20)
for i, column in enumerate(['UnitPrice', 'Quantity', 'TotalPrice']):
    sns.boxplot(data=data, y=column, ax=axes[i])
```

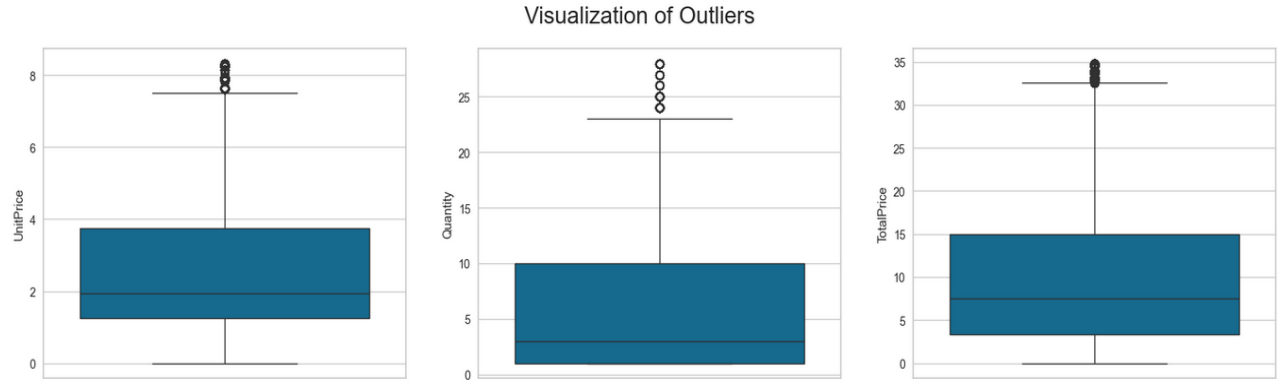

Figure 5: Outliers

EXPLORATORY DATA ANALYSIS(EDA):

- Sales Analysis:

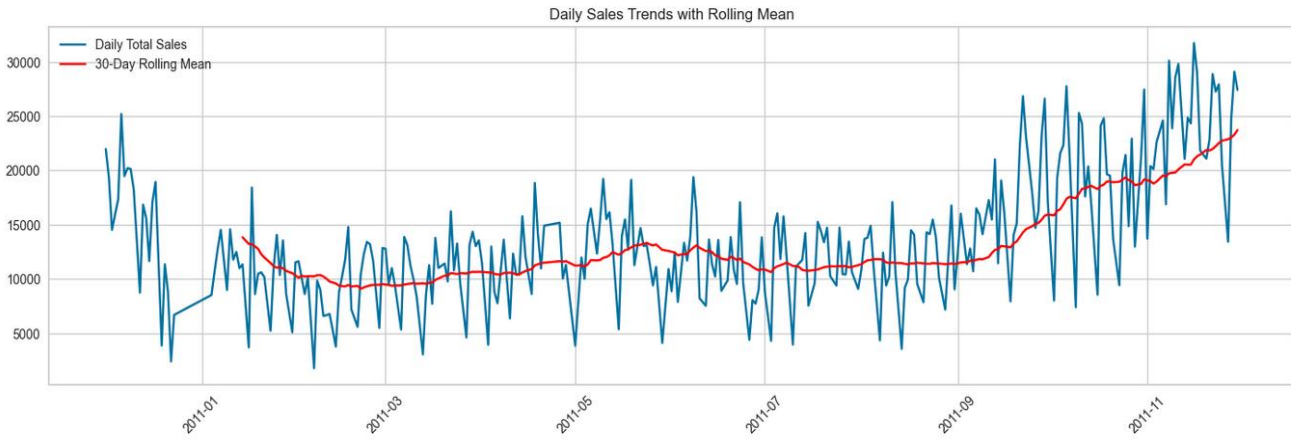

Figure 6: Daily Total Sales with 30 days Rolling Mean

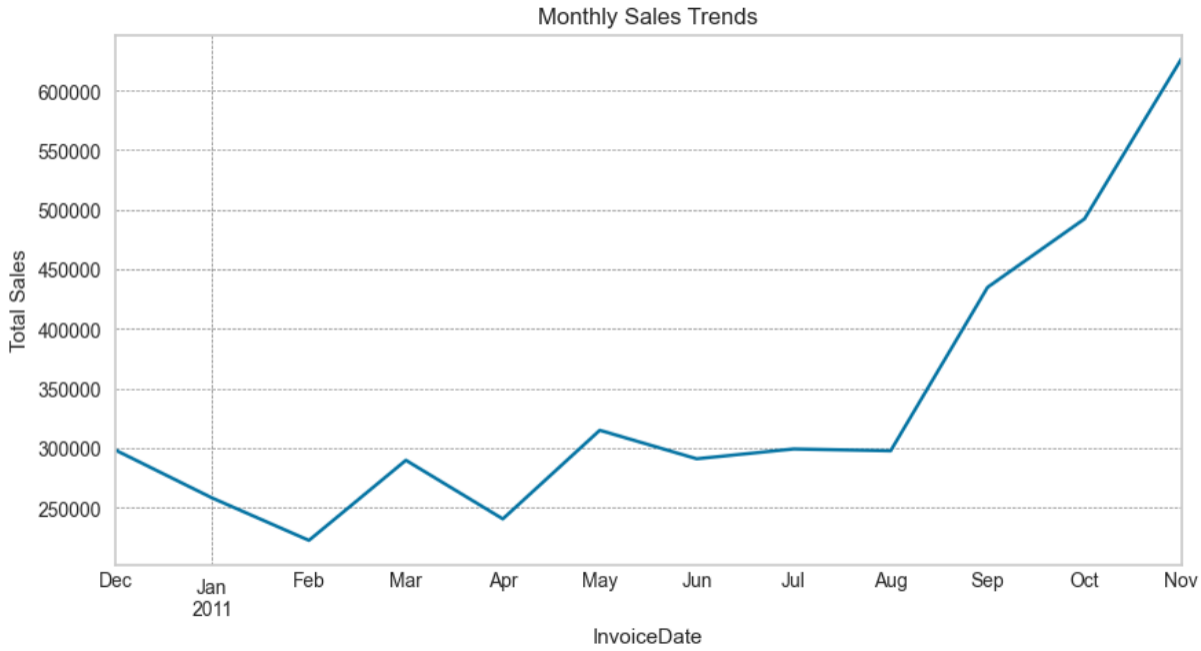

Figure 7: Monthly Sales

- Country Analysis:

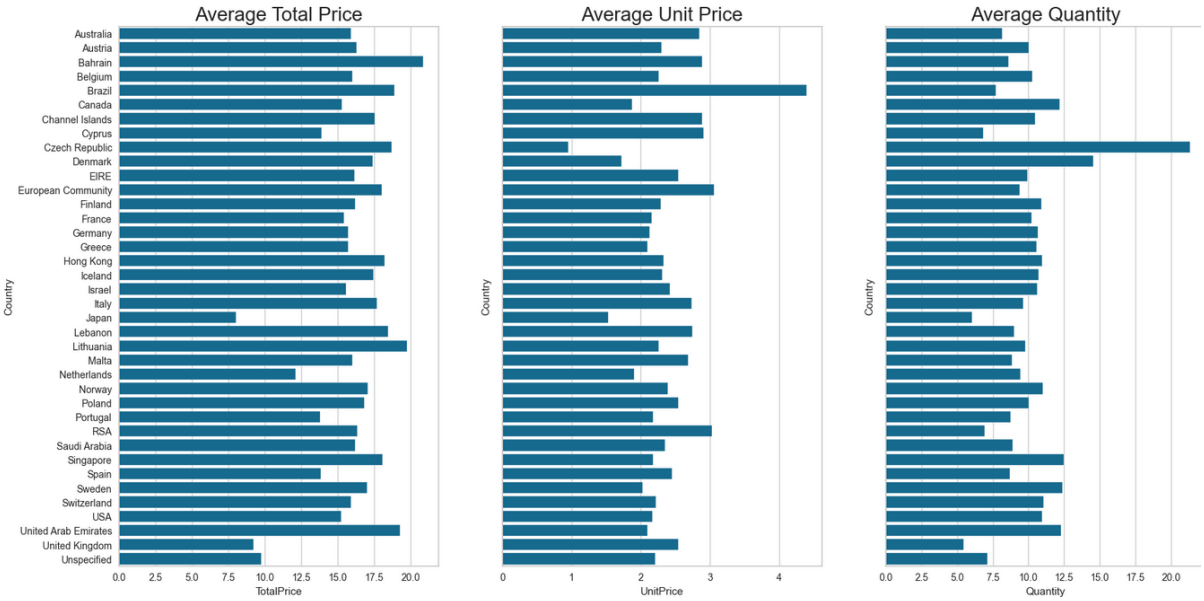

Figure 8: Average Total Price, Average Unit Price, Average Quantity Country-wise

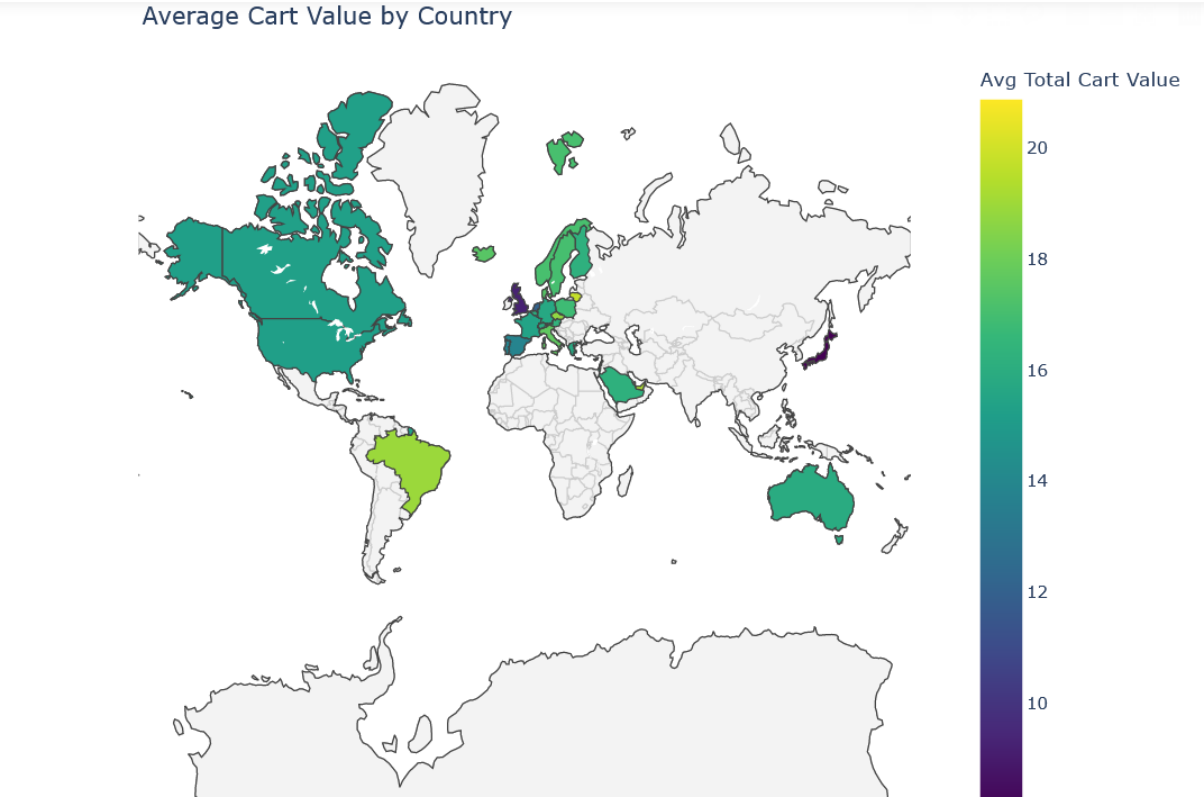

Figure 9: Choropleth World map showing average cart value by country

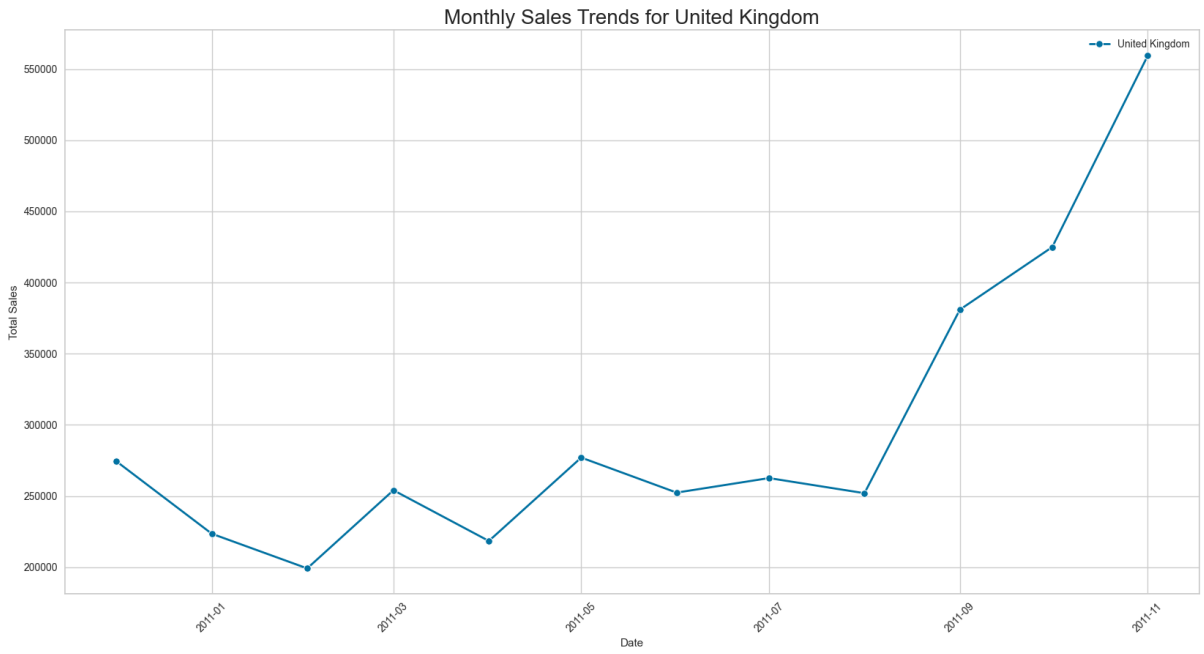

Figure 10: Monthly Sales for United Kingdom

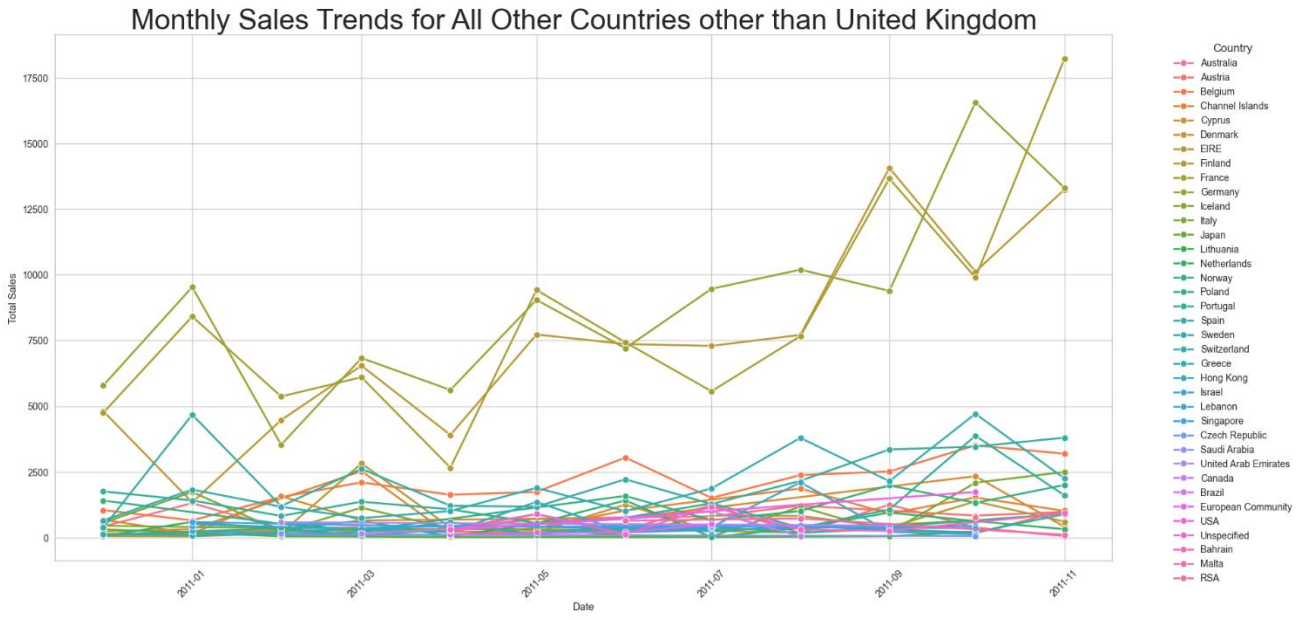

Figure 11: Monthly sales for all countries except United Kingdom

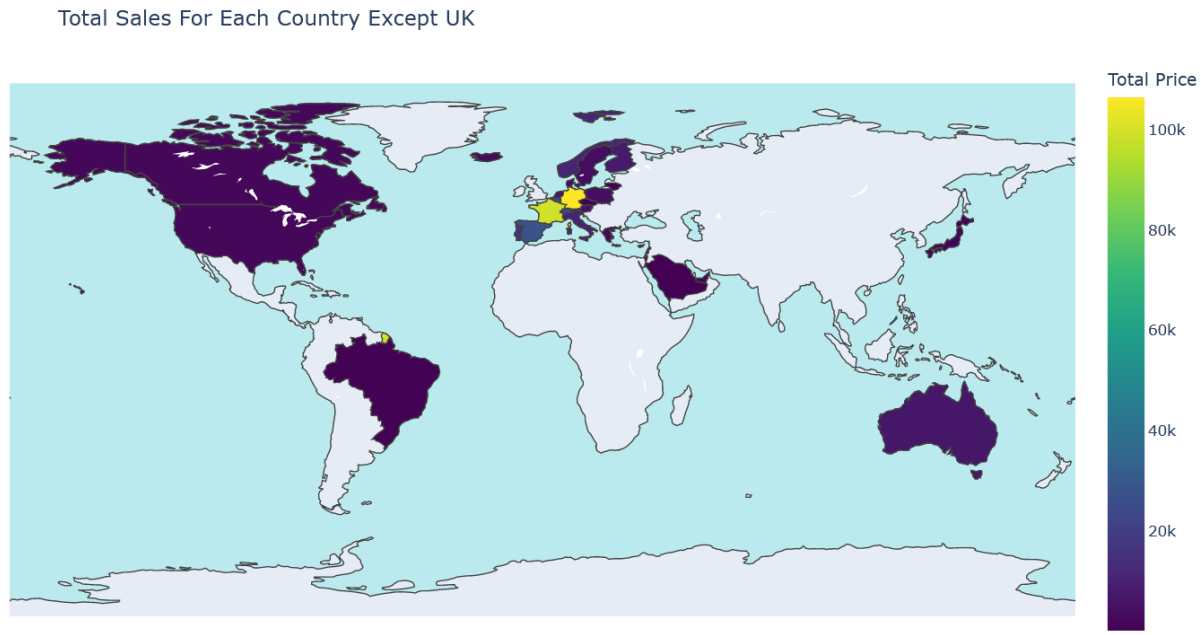

Figure 12: Choropleth World map showing Sales for each country except United Kingdom

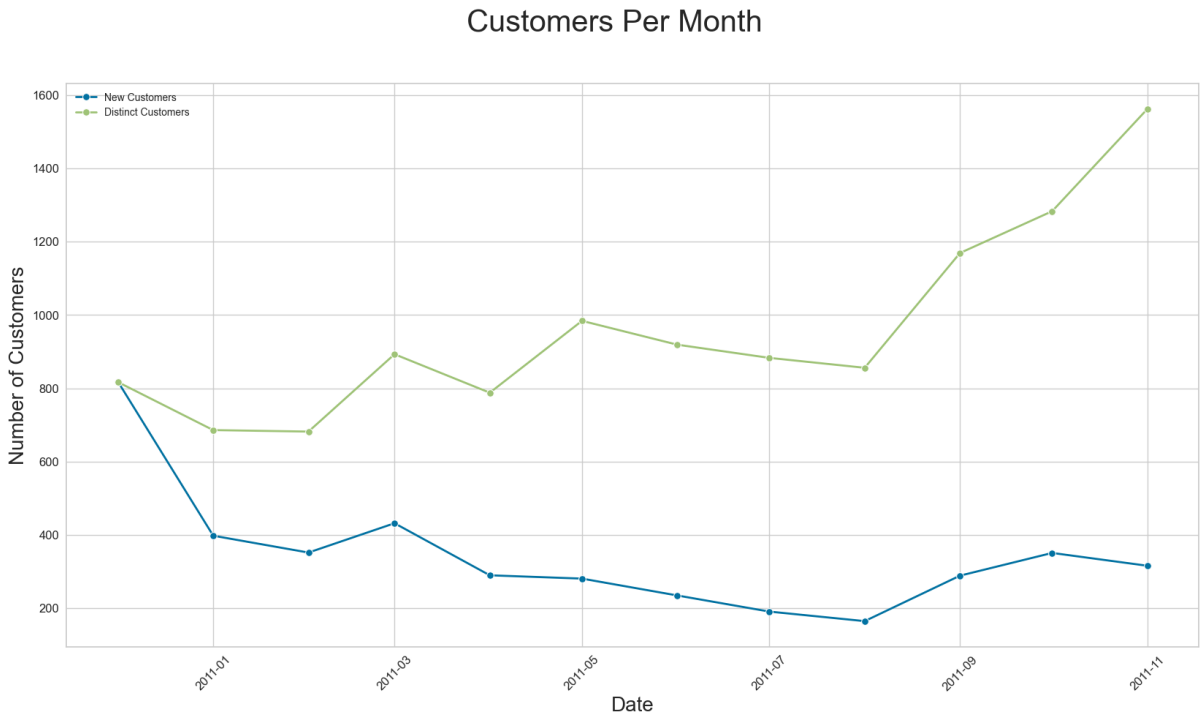

Figure 13: Customers per Month

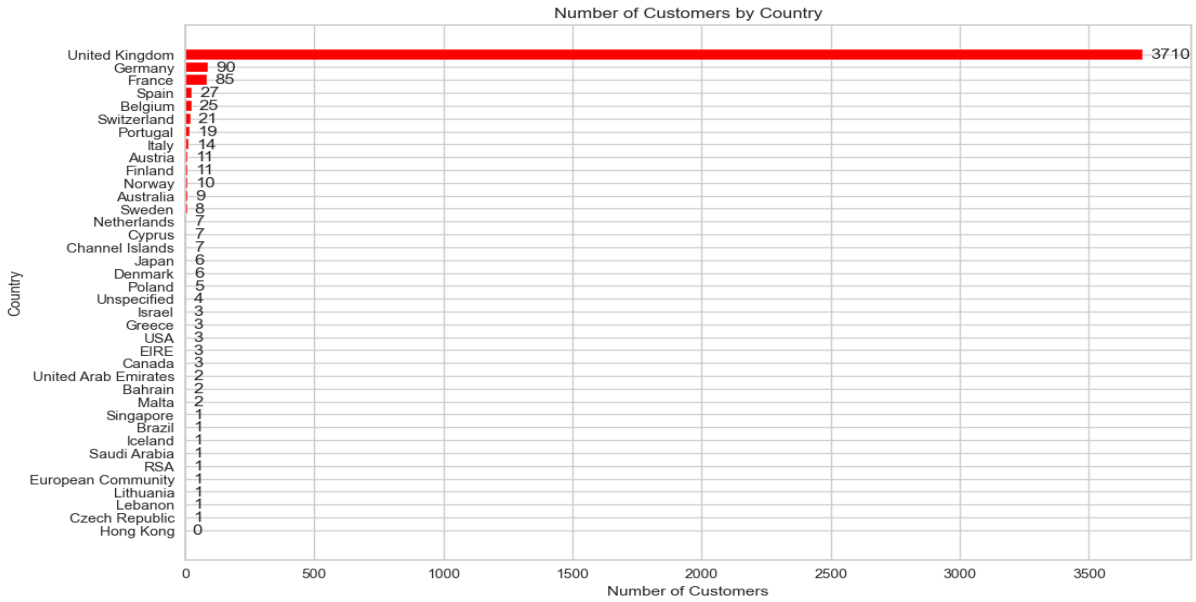

Figure 14: Number of Customers per Country

```
country_sales = data.groupby('Country')['TotalPrice'].sum().reset_index()
top_countries = country_sales.sort_values(by='TotalPrice', ascending=False).head(5).reset_index(drop=True)
top_countries
```

	Country	TotalPrice
0	United Kingdom	3578815.673
1	Germany	106491.180
2	France	99161.100
3	EIRE	88610.310
4	Spain	27511.120

Figure 15: Top 5 Countries in Sales

- Time of Transaction Analysis:

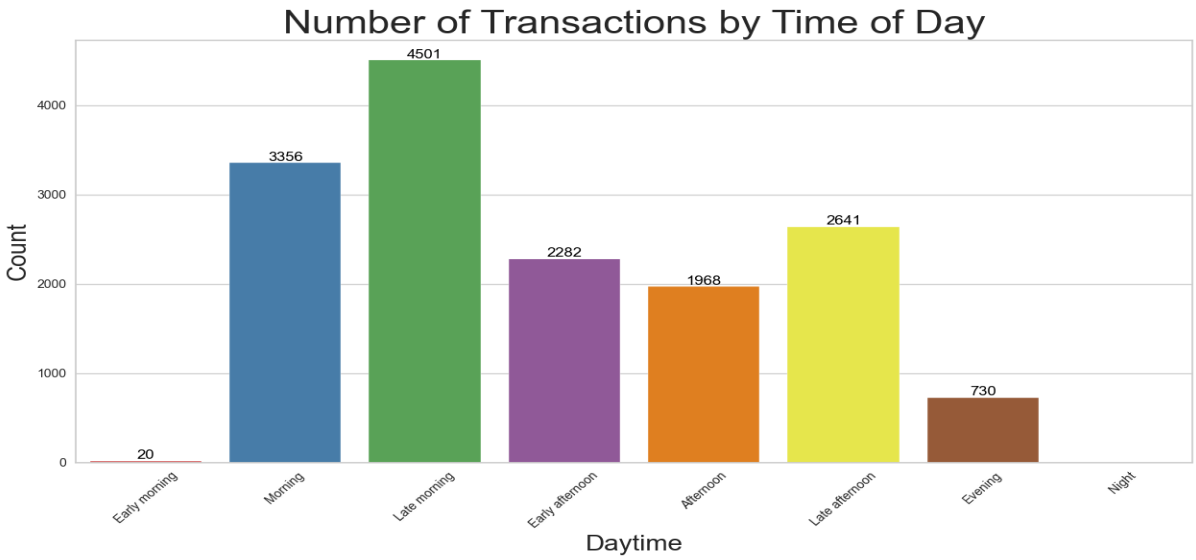

Figure 16: Number of Transactions by Time of Day

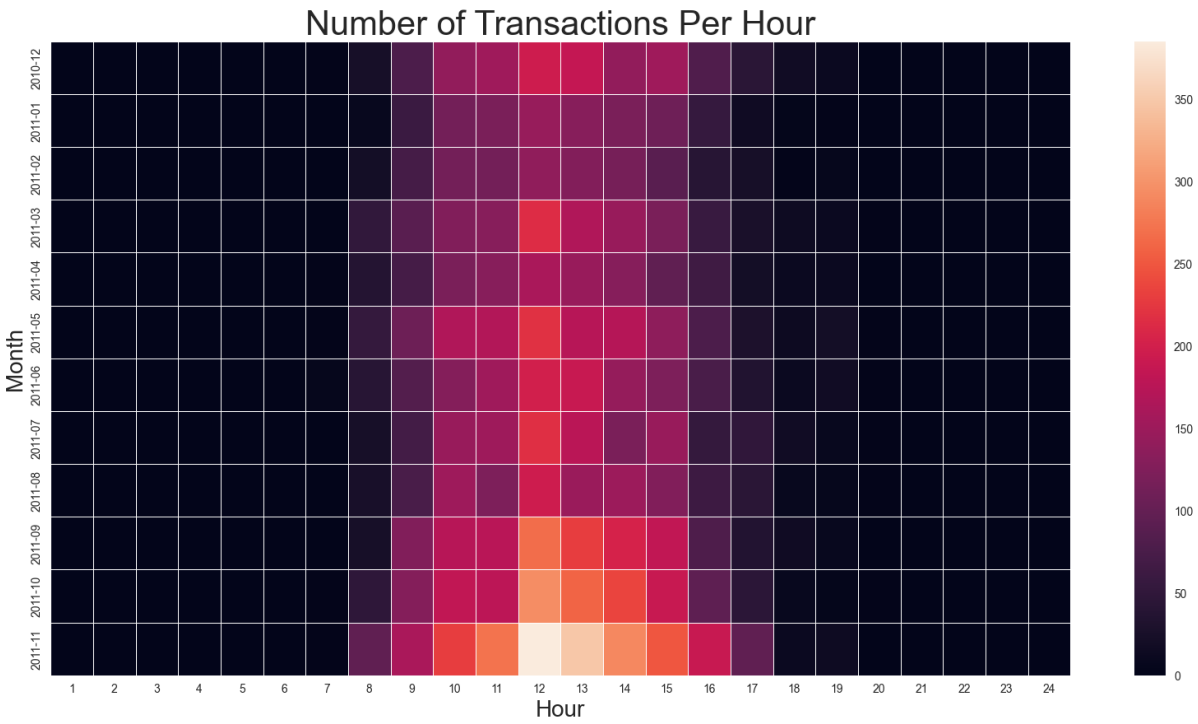

Figure 17: Heat Map showing Number of Transactions per Hour

MOTIVATION AND REASON FOR CHOOSING THE ALGORITHM:

For this online retail project, the K-Means clustering algorithm has been selected to segment customers based on their purchasing behavior. This decision is underpinned by several factors that align with the project objective and the nature of the data available. Here is a detailed exploration of why K-Means was chosen and its appropriateness for the task:

1. Suitability for Customer Segmentation:

- **Simplicity and Efficiency**: K-Means is renowned for its simplicity and computational efficiency, especially beneficial for handling large datasets, which is often the case in retail analytics. This algorithm partitions the customers into distinct groups based on their features, making it an excellent tool for identifying different customer types in a retail setting.
- **Scalability**: The algorithm scales well with the number of transactions and the dataset's dimensionality, a crucial factor given the expansive dataset of customer transactions.

2. Algorithm Mechanics:

- **Feature-Based Clustering**: K-Means works by grouping data points (customers) into clusters that minimize the variance within each cluster. This property is particularly useful in retail, where customers can be segmented into clusters based on features like purchase frequency, monetary value of purchases, and recency of transactions (RFM metrics).
- **Iterative Approach**: The iterative refinement technique used by K-Means ensures that the clusters are optimized to be as distinct and as relevant as possible, which helps in precisely targeting different customer segments.

3. Data Characteristics:

- **Numerical Data Handling**: K-Means is best suited for datasets with numerical attributes. In this project, key customer metrics such as `TotalPrice`, `Frequency`, and `Recency` are numeric and ideal for forming the basis of K-Means clustering.
- **Standardization Ready**: K-Means requires features on a similar scale for optimal performance. The preprocessing steps, including standardization of numerical values, prepare the dataset ideally for this algorithm, preventing skewed results due to differing value ranges.

4. Analytical Goals:

- **Customer Insights**: The project aims to derive actionable insights into customer behavior, such as identifying high-value customers, at-risk customers, or those who might need targeted incentives. K-Means facilitates this by segmenting customers into clear, actionable groups.
- **RFM Analysis:** A critical component of achieving this objective is the application of RFM (Recency, Frequency, Monetary) analysis, a well-established marketing technique used

to quantitatively rank and segment customers based on the timeliness and value of their transactions:

Recency (R): Measures how recently a customer has made a purchase. A lower recency value indicates that the customer bought something very recently, which is a strong indicator that they are still engaged with the brand.

Frequency (F): Measures how often a customer makes a purchase within a defined period. Frequent shoppers are more likely to be loyal and responsive to promotions.

Monetary (M): Measures how much money a customer has spent over a period. Customers who spend more are often more profitable and potentially less sensitive to price changes.

- **Marketing Strategy Development**: By identifying distinct customer segments, the marketing team can tailor strategies to specific groups, enhancing the effectiveness of targeted marketing campaigns.

In this project, RFM analysis serves as the foundation for developing a clustering model using K-Means algorithm to segment customers into distinct groups. Each group represents a different type of customer, from highly engaged and valuable customers (“Top Customers”) to those who rarely shop or spend little ("Dormant" customers). By understanding these segments, the retail store can tailor its marketing efforts, such as through personalized email marketing campaigns, targeted discounts, and optimized product recommendations, to match the specific needs and behaviors of each segment.

5. Validation and Adaptability:

- **Quantitative Metrics for Evaluation**: The use of metrics such as the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index allows for the quantitative evaluation of the clustering results. These metrics help in determining the optimal number of clusters and assessing the quality of clustering, which is critical for validating the effectiveness of K-Means in segmenting customers.
- **Flexibility in Cluster Number**: The ability to choose the number of clusters dynamically, supported by methods like the Elbow Method visualized through the Yellowbrick library’s KElbowVisualizer, provides flexibility and adaptability in segmentation strategy.

The selection of K-Means for this project is justified by its efficacy in handling large datasets, simplicity in execution, suitability for numerical data, and its potential in revealing customer segments that can be targeted with tailored marketing strategies. The choice is further reinforced by the algorithm’s robustness in forming distinct, interpretable clusters that align well with business goals, making it an integral tool for data-driven decision-making in retail analytics.

- **Cluster Characteristics**: Each cluster represents a distinct group of customers with unique purchasing patterns. For example, some clusters might consist of high-value customers with frequent purchases, while others might include occasional shoppers.
- **Customer Segmentation**: The segmentation can help the retail store tailor its marketing strategies, such as personalized promotions and targeted advertisements, to better meet the specific needs and preferences of different customer groups.
- **Purchasing Patterns**: Insights into common purchasing trends within each cluster can aid in inventory management, product placement, and pricing strategies.

ASSUMPTIONS:

In the analysis of the online retail data using the K-Means clustering algorithm, several key assumptions underpin the approach and methodologies employed. These assumptions are essential to the project's design and interpretation of outcomes. Understanding these helps in contextualizing the results and anticipating areas where the model's performance might be constrained by underlying presumptions.

1. Data Distribution and Scale:

- **Numerical Data Normality**: K-Means assumes that the features it uses are numerically scaled and follow a somewhat symmetrical distribution. This assumption affects the preprocessing steps where data normalization and scaling are applied. The project assumes that after applying standardization (mean normalization and variance scaling), the data does not severely violate the normality assumption, which is critical for the effectiveness of distance-based clustering.
- **Equal Variance Across Features**: By standardizing the data, there is an inherent assumption that each feature contributes equally to the cluster analysis. This standardization process assumes that all features should have the same weight or importance in calculating distances between data points, which might not always align with the practical importance of each metric.

2. Cluster Assumptions:

- **Spherical Clusters**: K-Means inherently assumes that the clusters are spherical and isotropic, meaning that the clusters are circular (or hyper-spherical in higher dimensions) around the centroid. This assumption can be limiting in complex datasets where natural clusters might not adhere to this shape, potentially leading to suboptimal clustering performance.
- **Similar Cluster Density**: The algorithm also presumes that clusters have a similar density and that data points are uniformly distributed around the centroids. In real-world scenarios, especially in retail data, clusters might vary significantly in size, density, and dispersion, which could affect the clustering outcome and accuracy.

3. Independence of Observations:

- **Independent Customer Transactions**: The analysis assumes that each transaction or customer record can be treated independently. In reality, transactions from the same customer are not independent, as purchasing behavior is likely influenced by past interactions, seasonal effects, or marketing campaigns.

4. Algorithm-Specific Assumptions:

- **Random Initialization Sensitivity**: The initial placement of centroids in K-Means can affect the final clusters. While the project utilizes multiple initializations (n_init parameter) to mitigate this, it assumes that this approach sufficiently overcomes the potential pitfalls of poor initial centroid placements.
- **Optimal Cluster Number**: Although methods like the silhouette score and the elbow method are used to determine the best number of clusters, there is an assumption that

these quantitative measures align with the most meaningful or useful segmentation from a business perspective.

5. Data Completeness and Accuracy:

- **No Missing or Incorrect Data**: The preprocessing assumes that once missing or anomalous values (like negative prices or quantities) are removed, the remaining data is complete and accurate. This might overlook subtle errors or biases in data collection and processing, which could influence the clustering outcome.

The assumptions in this project form the backbone of the analytical approach and influence every stage from data preprocessing to model evaluation. Awareness of these assumptions is crucial for critically assessing the model's validity and for understanding the boundaries within which the results are applicable. By acknowledging these assumptions, the project not only ensures a transparent analysis but also opens pathways for future improvements and iterations.

MODEL EVALUATION TECHNIQUES:

For evaluating the K-Means clustering model used in this project, several statistical measures were employed to ensure the model's effectiveness and appropriateness in segmenting customer data based on their purchasing behavior:

1. **Silhouette Score**: This metric was used to assess the quality of clusters created by the model. A high silhouette score indicates that clusters are well-separated and cohesive, which implies that the model has effectively captured the underlying patterns in the dataset.
2. **Davies-Bouldin Index**: This index helps measure the average 'similarity' between clusters, where lower values indicate better clustering. A low Davies-Bouldin index signifies that the clusters are well-distributed and distinct from each other.
3. **Calinski-Harabasz Index**: Also known as the Variance Ratio Criterion, this index evaluates the cluster validity based on the ratio of the sum of between-clusters dispersion to within-cluster dispersion. Higher values generally indicate better defined and separated clusters.

These techniques provide a robust framework for evaluating the clustering results, helping validate the effectiveness of the K-Means algorithm in segmenting the customer data.

INFERENCES:

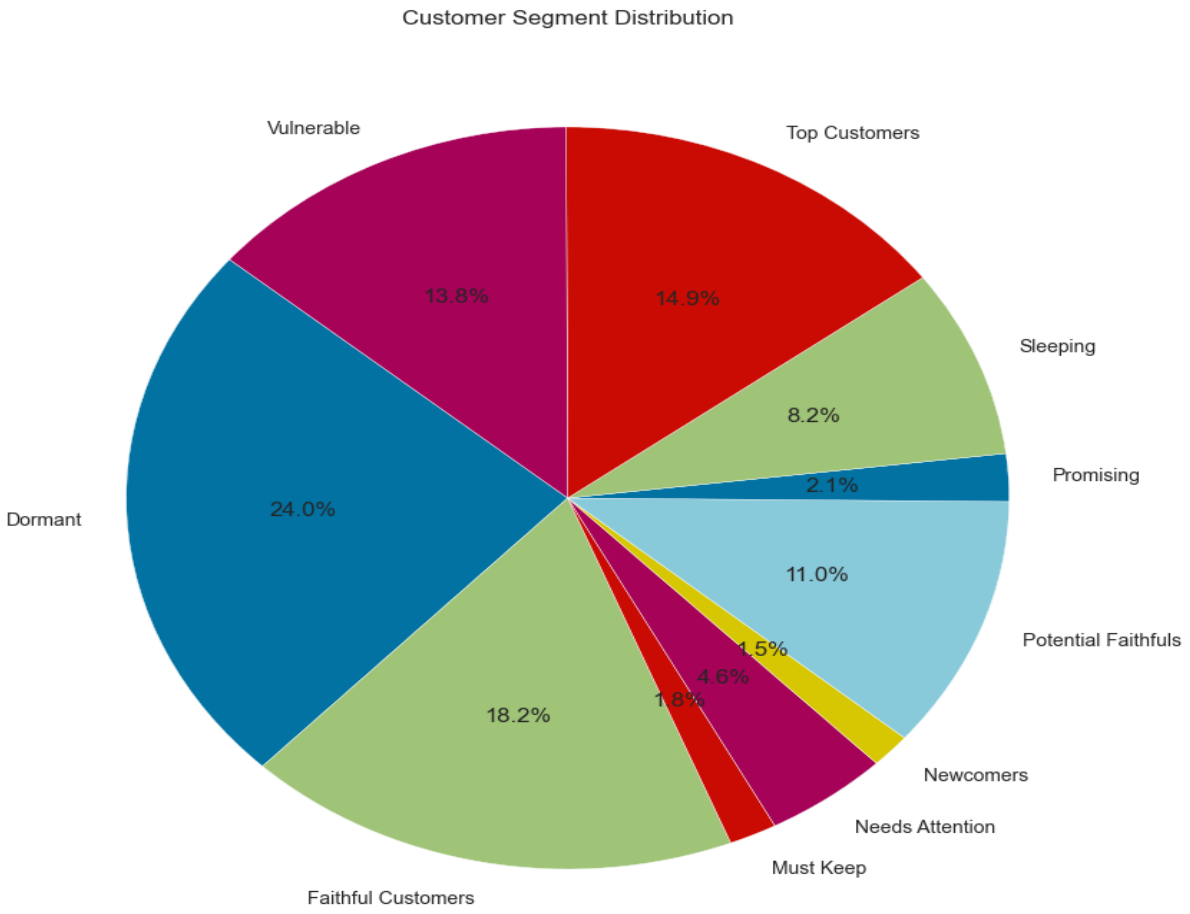

Figure 18: Customer Segment Distribution

```
rfm_summary = rfm.groupby('segment').agg({'recency': 'mean', 'frequency': 'mean', 'monetary': ['mean', 'count']})
rfm_summary
```

segment	recency mean	frequency mean	monetary mean	count
Dormant	218.259146	1.046748	220.341260	984
Faithful Customers	32.961126	6.128686	1407.789666	746
Must Keep	126.541667	7.513889	1290.210833	72
Needs Attention	49.994652	2.213904	541.364545	187
Newcomers	6.918033	1.000000	214.805410	61
Potential Faithfuls	15.046358	1.801325	405.769978	453
Promising	22.611765	1.000000	223.830000	85
Sleeping	50.881306	1.068249	276.789172	337
Top Customers	5.978793	10.047308	2194.136607	613
Vulnerable	154.908451	2.667254	530.566780	568

Figure 19: Customer Segment RFM Analysis

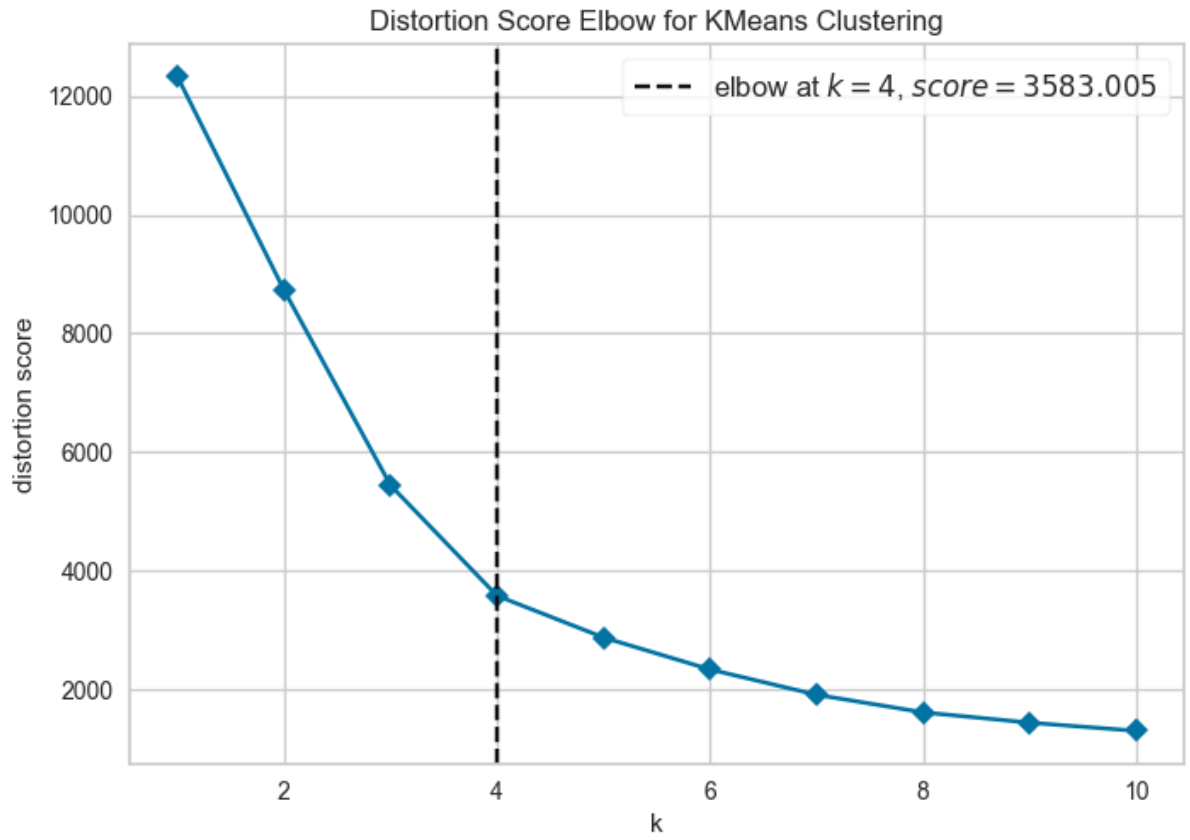

Figure 20: Elbow Method for K-Means Clustering

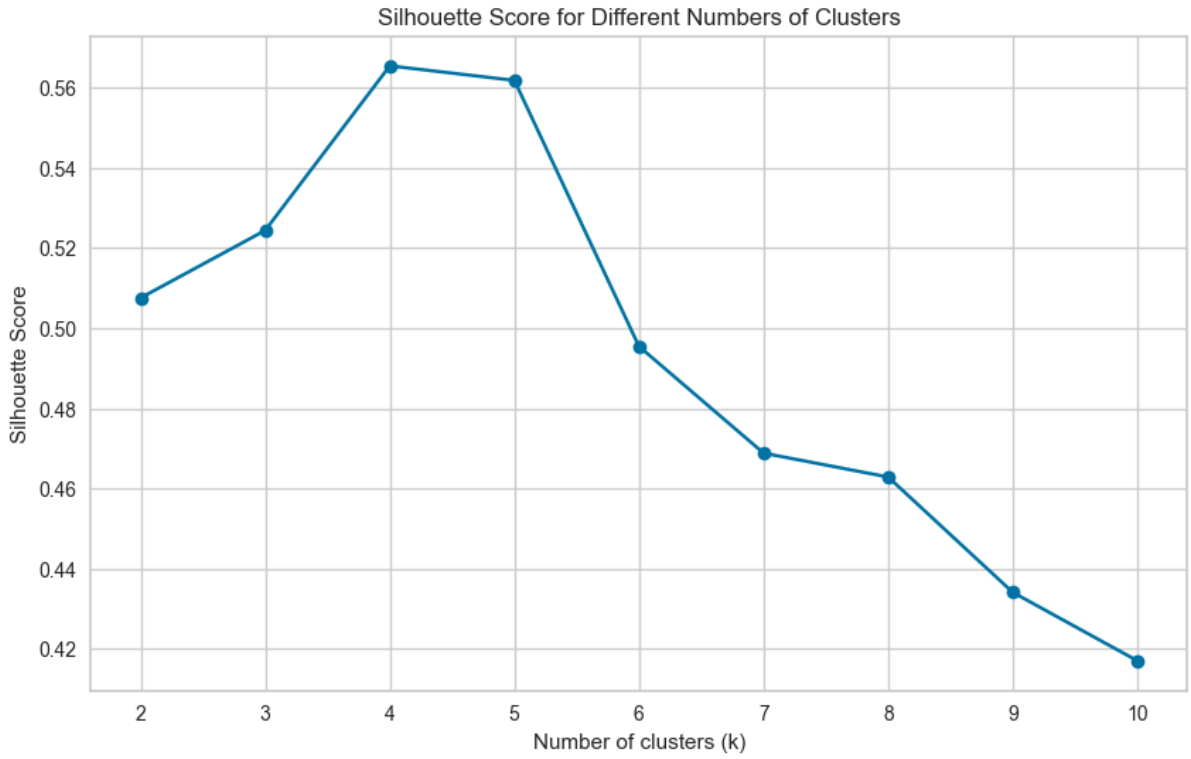

Figure 21: Finding Number of Clusters with Best Silhouette Score

```
cluster_stats = data_kmeans.groupby('labels').agg(['mean', 'count', 'max'])
cluster_stats
```

	recency			frequency			monetary		
	mean	count	max	mean	count	max	mean	count	max
labels									
0	42.586735	2744	160	3.148688	2744	15	682.042581	2744	3856.15
1	240.735099	1057	364	1.497635	1057	12	266.281524	1057	2107.31
2	0.666667	3	1	160.666667	3	186	38324.446667	3	66974.64
3	14.764901	302	363	16.052980	302	88	3803.932417	302	29242.79

Figure 22: Clusters RFM Summary

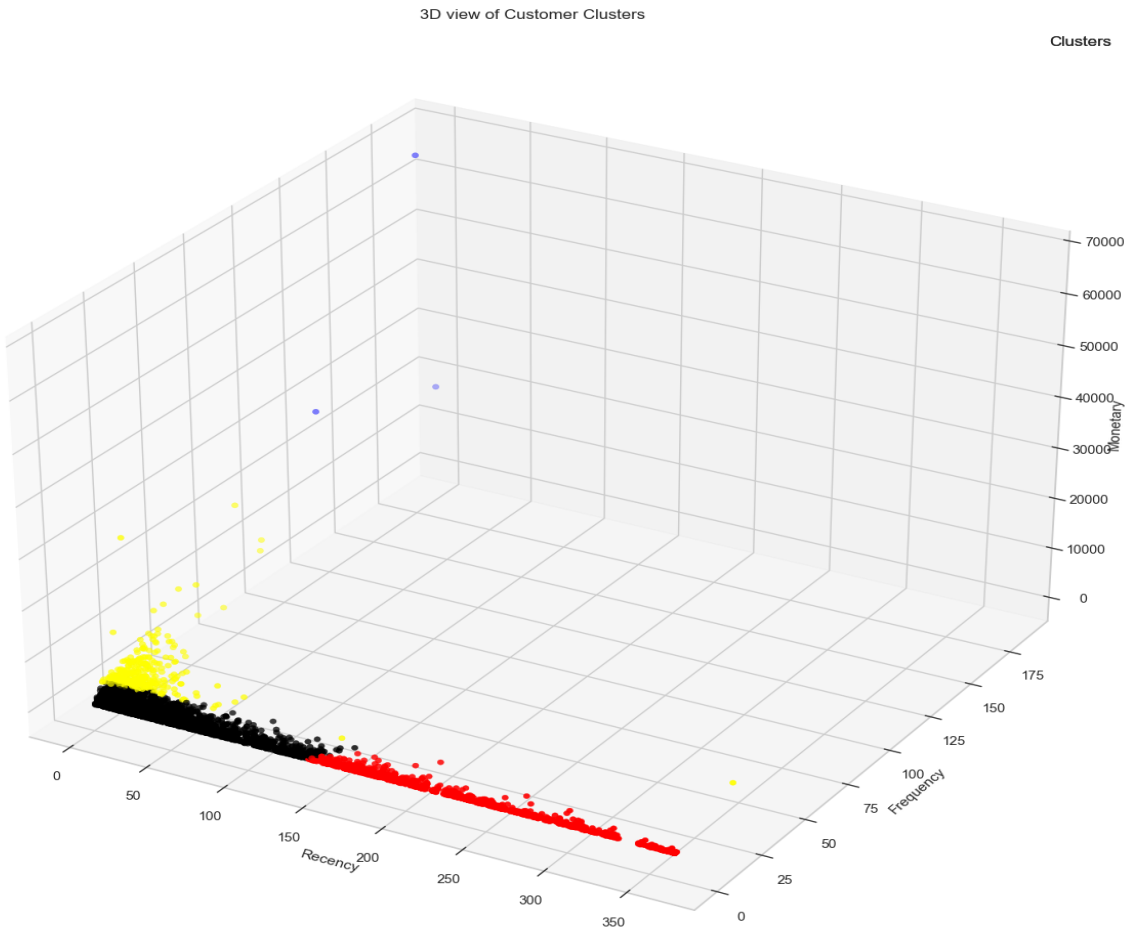

Figure 23: 3D-View of Clusters (RFM)

```
silhouette_avg = silhouette_score(data_scaled, kmeans.labels_)
print(f"Silhouette Score: {silhouette_avg}")
davies_bouldin = davies_bouldin_score(data_scaled, kmeans.labels_)
print(f"Davies-Bouldin Index: {davies_bouldin}")
calinski_harabasz = calinski_harabasz_score(data_scaled, kmeans.labels_)
print(f"Calinski-Harabasz Index: {calinski_harabasz}")
```

```
Silhouette Score: 0.5656164535681453
Davies-Bouldin Index: 0.6394990897875493
Calinski-Harabasz Index: 3333.4166275839725
```

Figure 24: K-Means Metrics

Let us analyze the value of metric in detail:

1. **Silhouette Score (0.565):**
    - The Silhouette Score ranges from -1 to +1, where a higher score indicates that clusters are well separated from each other and tightly grouped internally.
    - A score of approximately 0.57 suggests moderate separation and cohesion within clusters. This indicates that, on average, data points are reasonably well matched to their own cluster and moderately distanced from neighboring clusters. However, there is room for improvement as scores closer to 1 would indicate stronger clustering characteristics.
2. **Davies-Bouldin Index (0.639):**
    - The Davies-Bouldin Index is a measure of clustering validity, where a lower value indicates better clustering. It essentially evaluates the average similarity between each cluster with the cluster most similar to it, with the similarity being a measure that compares the distance between clusters with the size of the clusters themselves.
    - A value of 0.64 is relatively low, suggesting good cluster separation and compactness, which is a positive sign for the clustering result. This indicates that your clusters are distinct from each other and each cluster is tightly packed.
3. **Calinski-Harabasz Index (3333.416):**
    - The Calinski-Harabasz Index is higher when clusters are dense and well-separated, which relates to a model with better-defined clusters.
    - A score of 3333 is quite high, implying that the clusters have a good variance ratio between the inter-cluster and intra-cluster distances. This indicates effective clustering with clusters being significantly more dispersed between each other than within themselves.

Overall Assessment:

- The number of clusters were determined to be 4 as per Elbow method.
- In the customer segmentation distribution, 24% fall in the ‘dormant’ segment; 14.90% are the ‘top customers’ segment and 18.20% fall in the 'Faithful Customers’ segment. Making the most active customers (combining to be 33.10%); a lot of effort and advertising strategies are to be developed to stimulate the customer demand in the other segments.

- The clustering results appear to be quite effective based on these metrics. The Silhouette Score shows good but not excellent cluster cohesion and separation. The Davies-Bouldin Index indicates low intra-cluster similarity (which is desirable), and the Calinski-Harabasz Index suggests a strong differentiation between clusters.
- While these indices suggest a strong performance, it is also important to consider the context of your application and data. Sometimes, even with good scores, the practical usability of clusters depends on the specific needs and domain of your application.
- If precision in specific aspects of clustering is crucial (like maximized separation for anomaly detection or tighter cohesion for customer segmentation), you might want to consider additional tuning of the K-means algorithm parameters, or possibly evaluating other clustering algorithms.

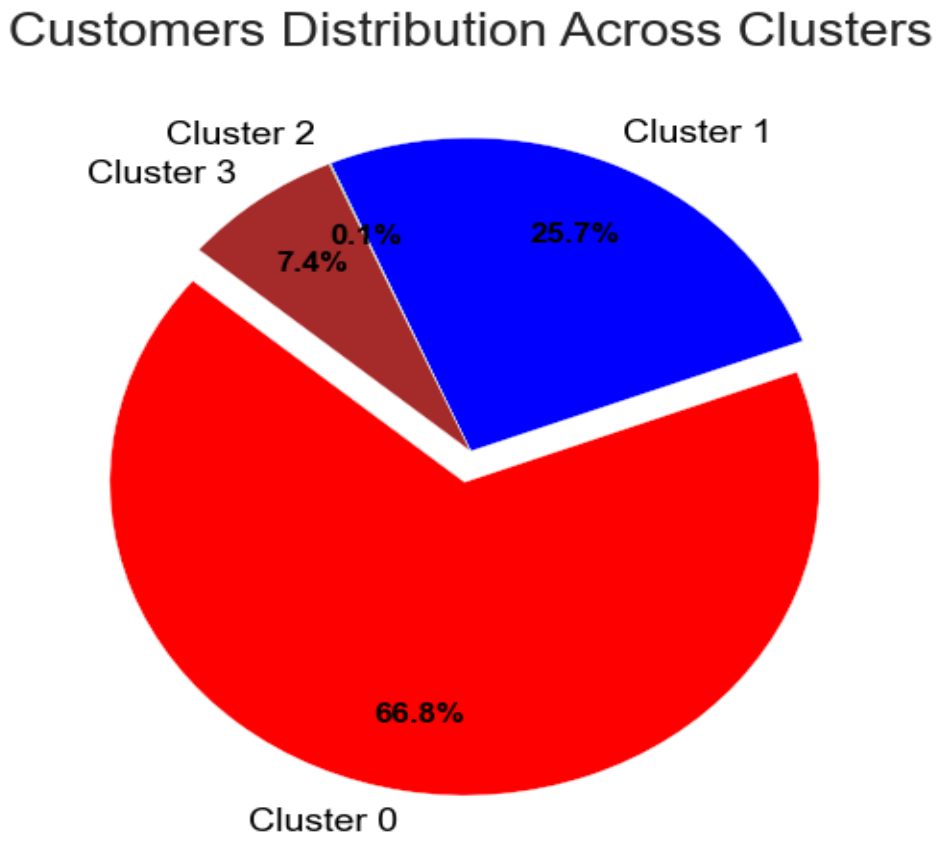

Figure 25: Customer Distribution Across Clusters

FUTURE POSSIBILITIES OF THE PROJECT:

The current project opens several avenues for future enhancement and application:

1. **Integration with Predictive Analytics**: The clustering model could be combined with predictive analytics to forecast future purchasing behaviors of different customer segments.
2. **Dynamic Clustering**: Implementing a dynamic clustering mechanism where the model automatically adjusts to new data can help in real-time customer segmentation, making the insights more actionable.
3. **Enhanced Personalization**: The insights from the clustering can be used to develop more personalized engagement strategies to enhance customer satisfaction and loyalty.
4. **Cross-Channel Marketing**: Understanding customer segments can improve cross-channel marketing efforts, aligning offers and promotions across different sales channels tailored to each segment.

CONCLUSION:

This project has effectively demonstrated how K-Means clustering can be applied to segment customers based on their transactional data. By identifying distinct customer groups, the retail store can better understand its customer base and refine its marketing strategies to cater to the specific needs of each segment. The clustering approach not only provided valuable insights into customer behavior but also highlighted the potential for more targeted and efficient marketing practices.

REFERENCES:

- James, Gareth, et al. "An Introduction to Statistical Learning." Springer Texts in Statistics, 2013.
- Scikit-Learn Documentation: KMeans, Silhouette Score, Davies-Bouldin Index, Calinski-Harabasz Index.
- "Python Data Science Handbook" by Jake VanderPlas; O'Reilly Media, 2016.
- Kaggle Notebooks : https://www.kaggle.com/datasets/carrie1/ecommerce-data/code?datasetId=1985&sortBy=voteCount&searchQuery=RFM