<u>ECommerce Transactions Data Analysis and Modeling Report</u>

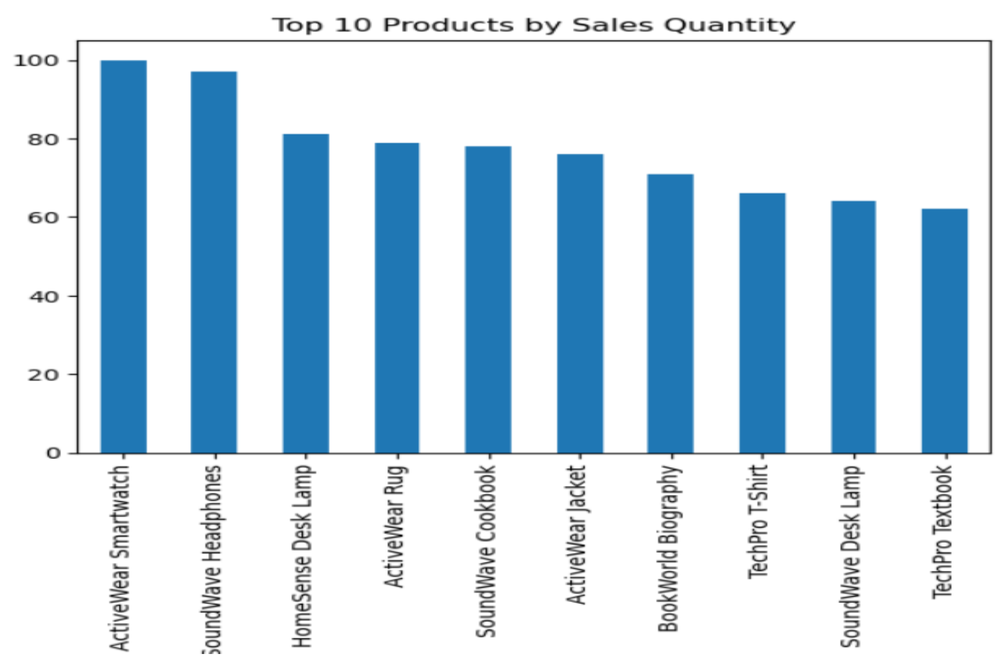
Overview

This report provides insights derived from the analysis of eCommerce data, covering customer behavior, product trends, and transaction patterns. It also includes predictive modeling and customer segmentation using clustering techniques.

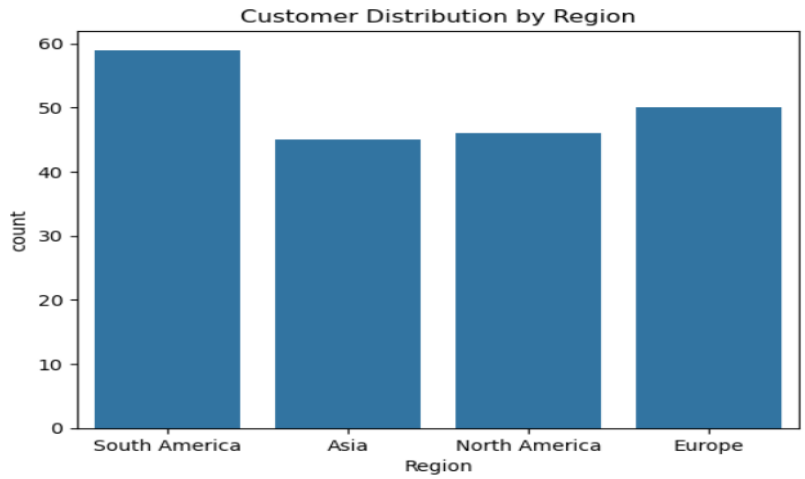
Task 1: Exploratory Data Analysis (EDA)

Key Insights

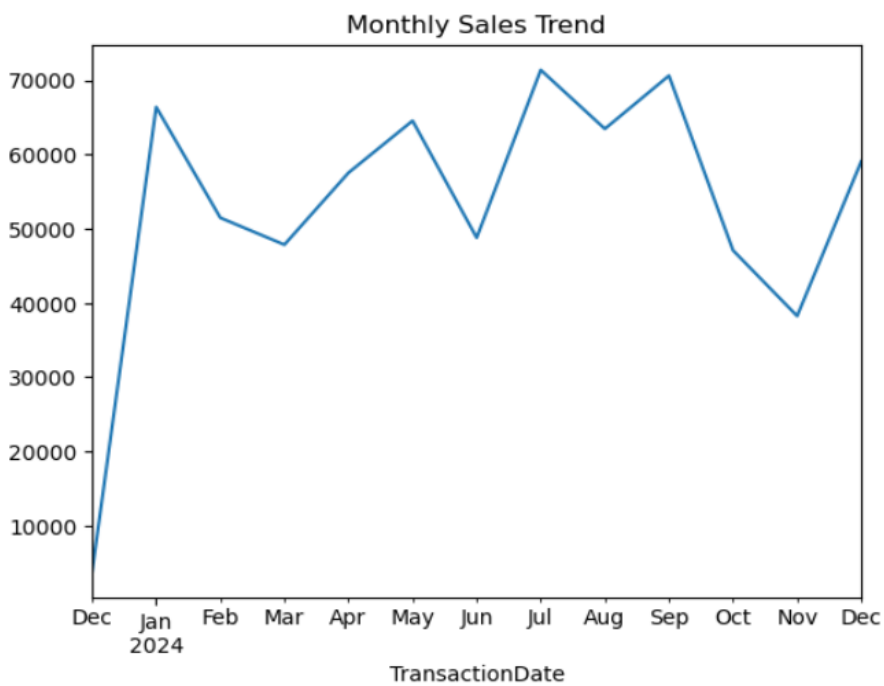
1. **Top 10 Performing Products:**
   ActiveWear Smartwatch has highest sales followed by **Soundwave Headphones** and **HomeSense Desk Lamp**.



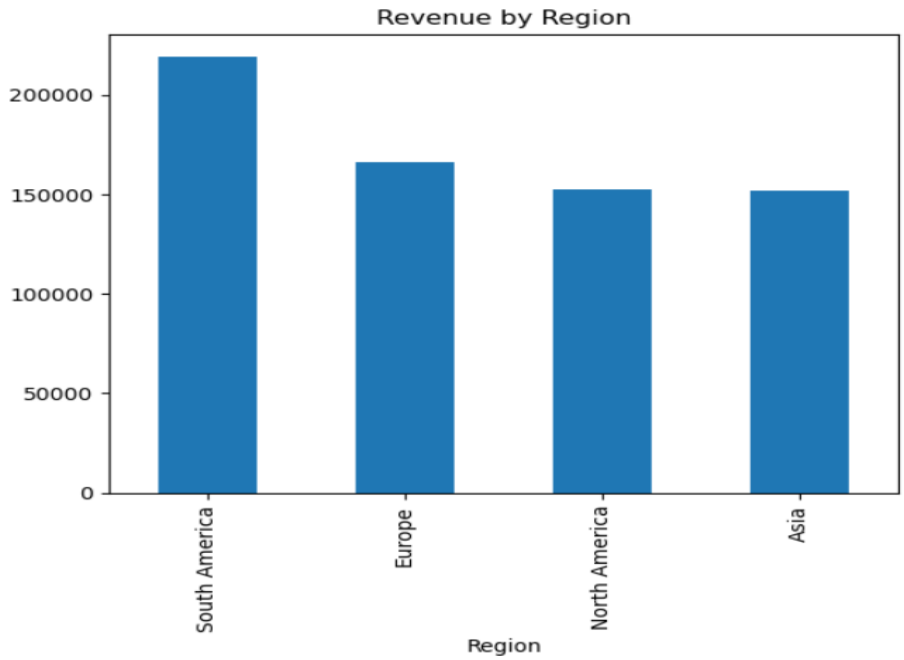
2. **Customer Distribution By Region:**
   Customers in **South America** accounted for the highest customer followed by **Europe and North America**.



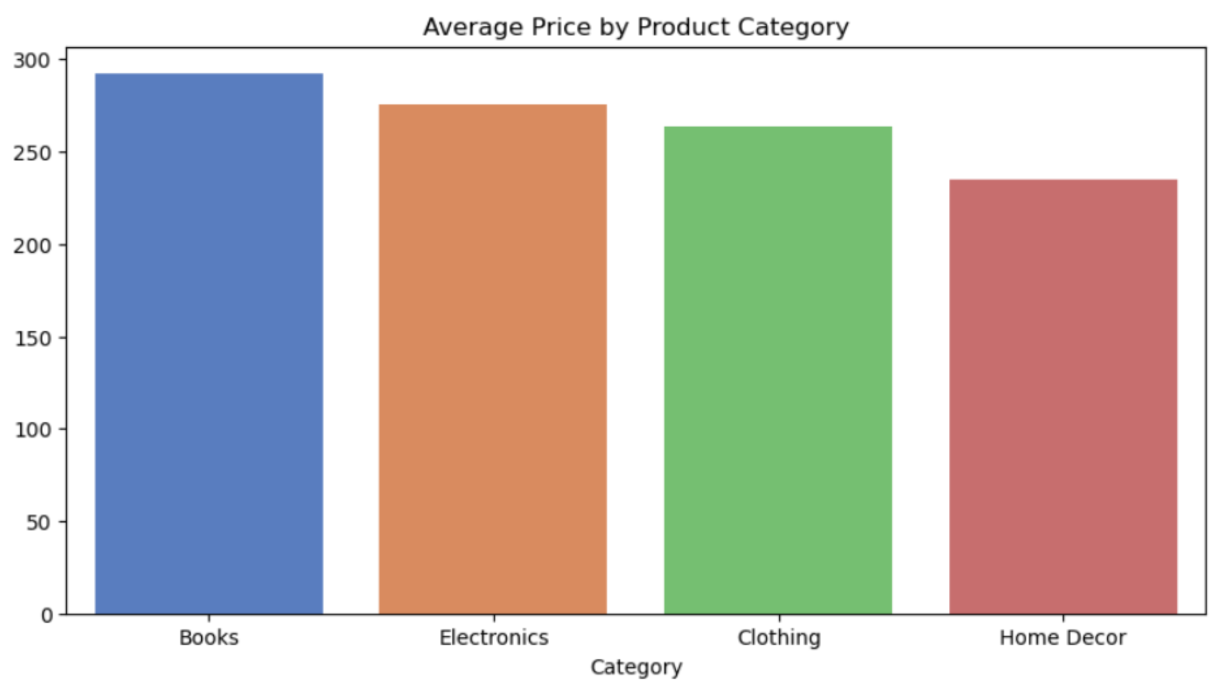
3. **Monthly Sales Trend:**
   July and **September** marked as highest sales month in year in 2024 followed by January.



4. **Revenue by Region:**
   Revenue collected in **South America** is highest compared to other Regions.



5. **Average Price by Product Category:**
   Average price per product quantity is highest for books followed by electronics.



Task 2: Lookalike Model

Objective

The Lookalike Model recommends 3 similar customers based on their transaction behavior and profile data.

Sample Output

CustomerID	Recommended Customers (Similarity Scores)	
C0001	C0137	1.0000
C0001	C0152	1.0000
C0001	C0056	0.9996

Key Takeaways

- The model effectively identifies similarities using a combination of customer profiles and transaction data.
- This approach can aid in targeted marketing and personalized recommendations.

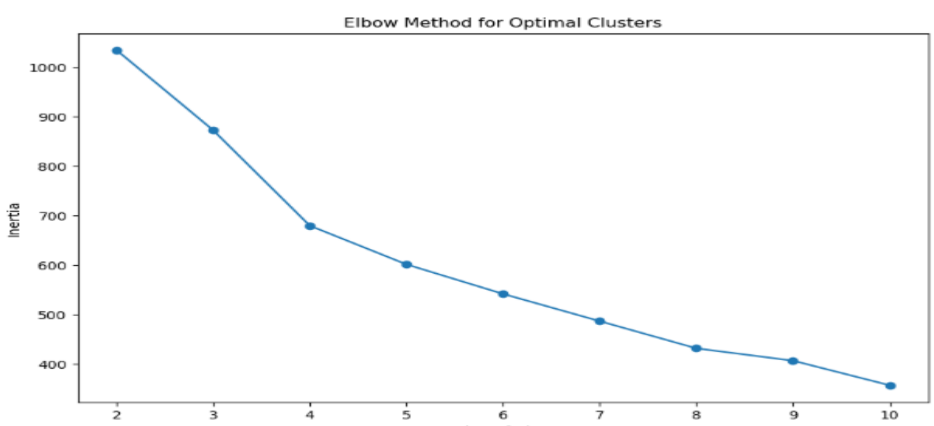
Task 3: Customer Segmentation

Clustering Results

To determine the best clustering approach between KMeans and DBSCAN, we need to evaluate their results based on the task requirements, dataset characteristics, and clustering metrics.

Key Observations:

KMeans Clustering:



KMeans Davies-Bouldin Index: 1.2241425724411819

Clusters Formed: 4

Davies-Bouldin Index (DBI): 1.224

KMeans assigns all data points to clusters, which may result in forced clustering of noise or outliers.

DBI indicates the compactness and separation of clusters; lower DBI is better. Here, the DBI is higher compared to DBSCAN.


```
)BSCAN Davies-Bouldin Index: 0.8912230470947684
```

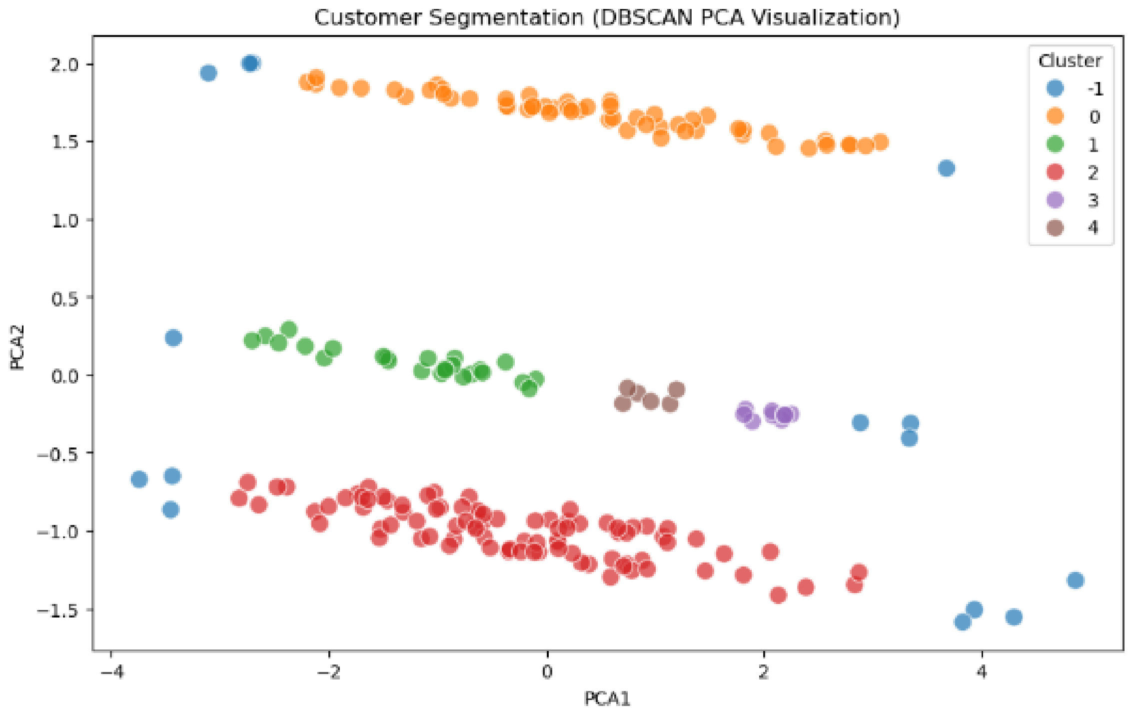
DBSCAN Clustering:

Clusters Formed: 5 (excluding noise points)

Davies-Bouldin Index (DBI): 0.891

DBSCAN handles noise better, marking outliers as -1 instead of forcing them into clusters.

The DBI is lower, indicating better compactness and separation of clusters compared to KMeans.




```
Clustering Results:
KMeans - Number of clusters formed: 4, Davies-Bouldin Index: 1.2241425724411819
DBSCAN - Number of clusters formed: 5, Davies-Bouldin Index: 0.8912230470947684
```

Recommendation:

DBSCAN appears to be the better approach in this scenario, for the following reasons:

Lower DBI Value:

DBSCAN achieves a DBI of 0.891, which is better than the 1.224 from KMeans. This indicates better-defined clusters in terms of compactness and separation.

Handling Noise and Outliers:

DBSCAN is well-suited for datasets with noise or outliers, as it can identify and exclude them from clusters (marked as -1).

Non-Globular Clusters:

DBSCAN can capture clusters of arbitrary shapes, unlike KMeans, which assumes spherical clusters.

Automatic Cluster Count:

DBSCAN does not require a predefined number of clusters (k), making it more flexible. It identified 5 clusters naturally, compared to the manually set 4 for KMeans.

Situations Where KMeans Might Be Better:

If your dataset is large and dense, as DBSCAN can struggle with high memory usage and execution time for large datasets.

If clusters are clearly spherical and the dataset has minimal noise or outliers.