Ecommerce Sales Analysis Project Report

Prepared By :- Vinay Raykar

INDEX

1. Introduction

This report presents the analysis of ecommerce sales data focusing on understanding sales trends, customer behavior, and seller performance across different product categories and geographic regions. The goal is to provide actionable insights to drive business decisions.

This project was made by using connection between **Python** and **SQL** for efficient data extraction, manipulation, and analysis.

2. Data Description

The dataset comprises multiple tables including products, orders, payments, sellers, and customer information. Key variables analyzed include sales amounts, order dates, product categories, geographic locations, payment methods, and seller/customer IDs. These data provide a holistic view of ecommerce activities.

3. Exploratory Analysis & Key Findings

Exploratory analysis showed dominant product categories driving most sales, clear seasonal patterns in monthly order volumes, and geographic concentration in a few top states. Payment methods varied, with credit cards and digital payments favored by most customers. Top customers and sellers accounted for a significant portion of revenues.

DATA EXPLORATION

a. Number of Customers from Each State

Bar chart detailing distribution of customers by state, useful for targeted outreach and expansion strategies.

```
12]:  query = """SELECT customer_state, count(customer_unique_id) FROM customers
      GROUP BY customer_state
      """

      cur.execute(query)

      data = cur.fetchall()

      customer_count = pd.DataFrame(data, columns = ["State", "Count of customers"])

      print('The state SP has 80k+ count of the customers number.')
      print(customer_count)
```

```
The state SP has 80k+ count of the customers number.
     State  Count of customers
0      SP               125238
1      SC                10911
2      MG                34905
3      PR                15135
4      RJ                38556
5      RS                16398
6      PA                 2925
7      GO                 6060
8      ES                 6099
9      BA                10140
10     MA                 2241
11     MS                 2145
12     CE                 4008
13     DF                 6420
14     RN                 1455
15     PE                 4956
16     MT                 2721
17     AM                  444
18     AP                  204
19     AL                 1239
20     RO                  759
21     PB                 1608
22     TO                  840
23     PI                 1485
24     AC                  243
25     SE                 1050
26     RR                  138
```

Count the number of customers from each state.
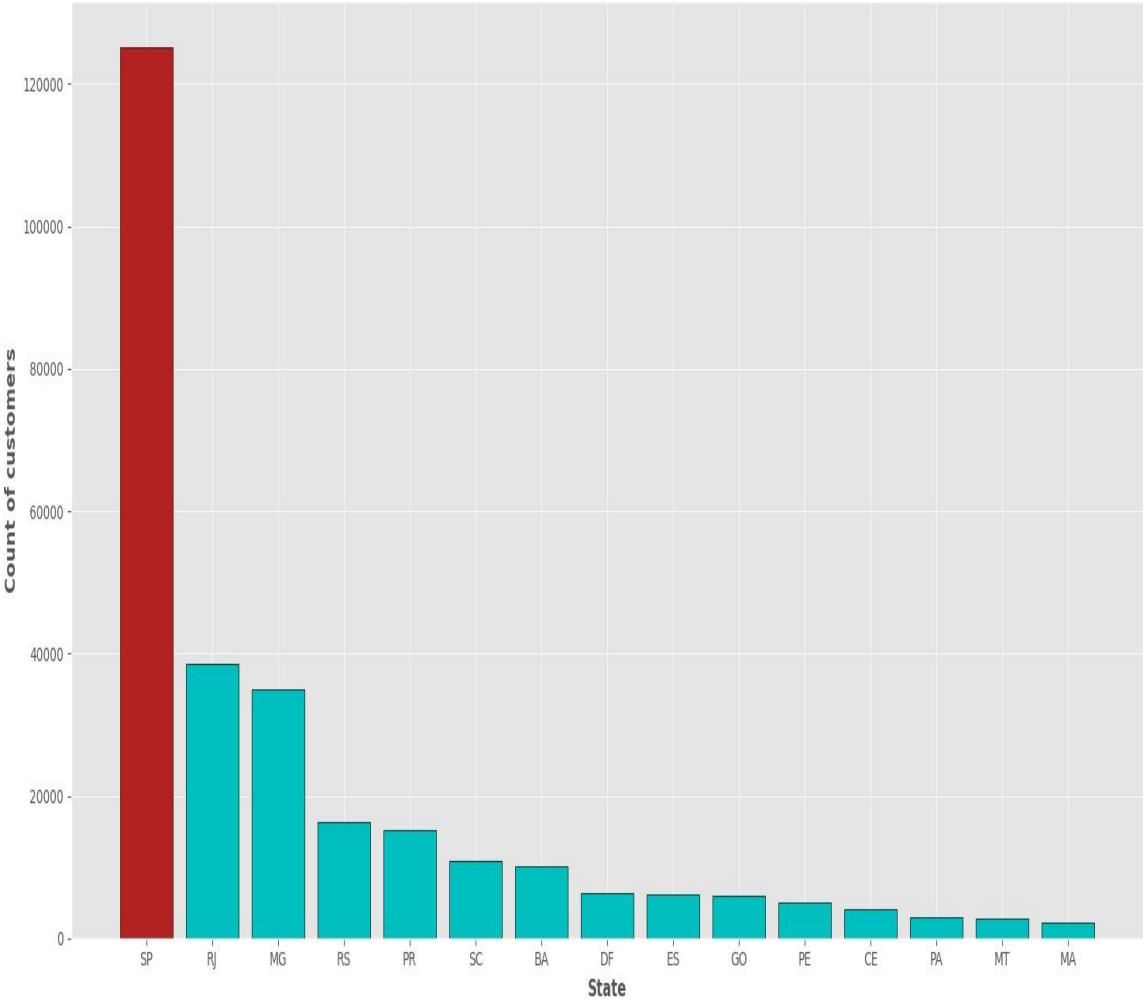
b. Monthly Order Volume

Line plot showing trends in monthly order volumes, revealing seasonal sales peaks and troughs.

```
[219]: query = """SELECT MONTH(order_purchase_timestamp) as Monthly,
       COUNT(order_id) as Order_Count
       FROM orders
       GROUP BY Monthly
       ORDER BY Monthly ASC;"""

       cur.execute(query)

       data = cur.fetchall()
       Monthly_vol= pd.DataFrame(data, columns = ["Month", "Order_Count"])
```
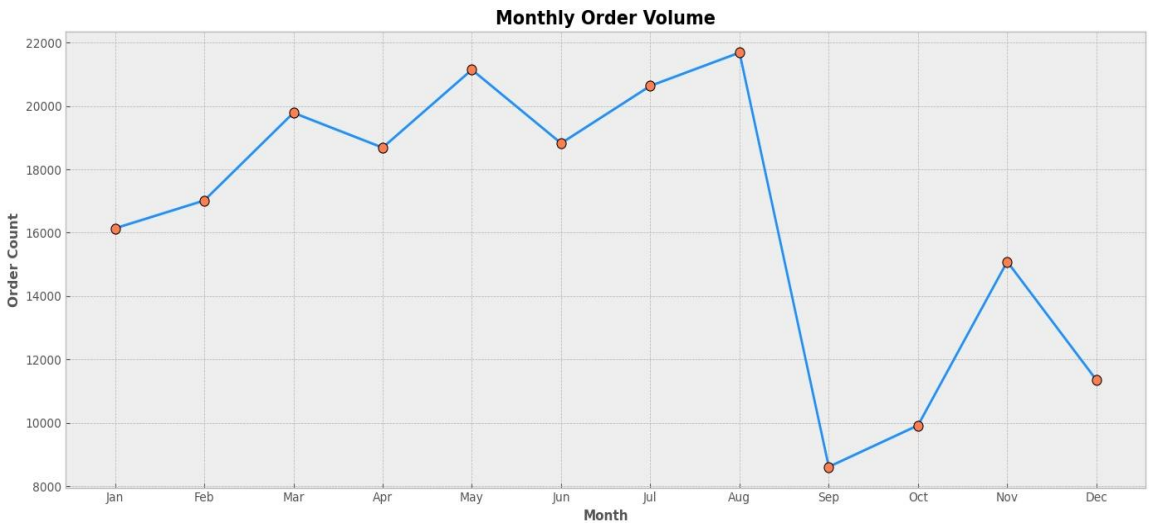
Count of sales of orders by months

```
[221]: Monthly_vol
```

[221]:

	Month	Order_Count
0	1	16138
1	2	17016
2	3	19786
3	4	18686
4	5	21146
5	6	18824
6	7	20636
7	8	21686
8	9	8610
9	10	9918
10	11	15088
11	12	11348

c. Revenue by Seller

Bar chart ranking sellers by total revenue, identifying highest-performing sellers for prioritizing partnerships.

```python
[226]:  # Using Window function.
        query = """SELECT *, dense_rank() over(ORDER BY Revenue DESC) as rank_seller FROM
        (SELECT DISTINCT orders_items.seller_id,
        round(sum(payments.payment_value),2) Revenue
        FROM orders_items JOIN payments
        ON payments.order_id = orders_items.order_id
        GROUP BY orders_items.seller_id) as Seller_group
        LIMIT 10
        """

        cur.execute(query)

        data = cur.fetchall()

        seller_rank = pd.DataFrame(data, columns = ["Seller_id","Revenue", "Rank"])
```

Seller Id :- 7c67e1448b00f6e969d365cea6b010ab is in the First Rank for generating the maximum revenue

```
[227]:  seller_rank
```

[227]:

	Seller_id	Revenue	Rank
0	7c67e1448b00f6e969d365cea6b010ab	2028667.63	1
1	1025f0e2d44d7041d6cf58b6550e0bfa	1232888.16	2
2	4a3ca9315b744ce9f8e9374361493884	1204981.08	3
3	1f50f920176fa81dab994f9023523100	1161013.68	4
4	53243585a1d6dc2643021fd1853d8905	1139612.32	5
5	da8622b14eb17ae2831f4ac5b9dab84a	1088877.28	6
6	4869f7a5dfa277a7dca6462dcf3b52b2	1056664.48	7
7	955fee9216a65b617aa5c0531780ce60	945289.20	8
8	fa1c13f2614d7b5c4749cbc52fecda94	826052.92	9
9	7e93a43ef30c4f03f38b393420bc753a	740536.84	10

Total revenue generated by each seller, and rank them by revenue.
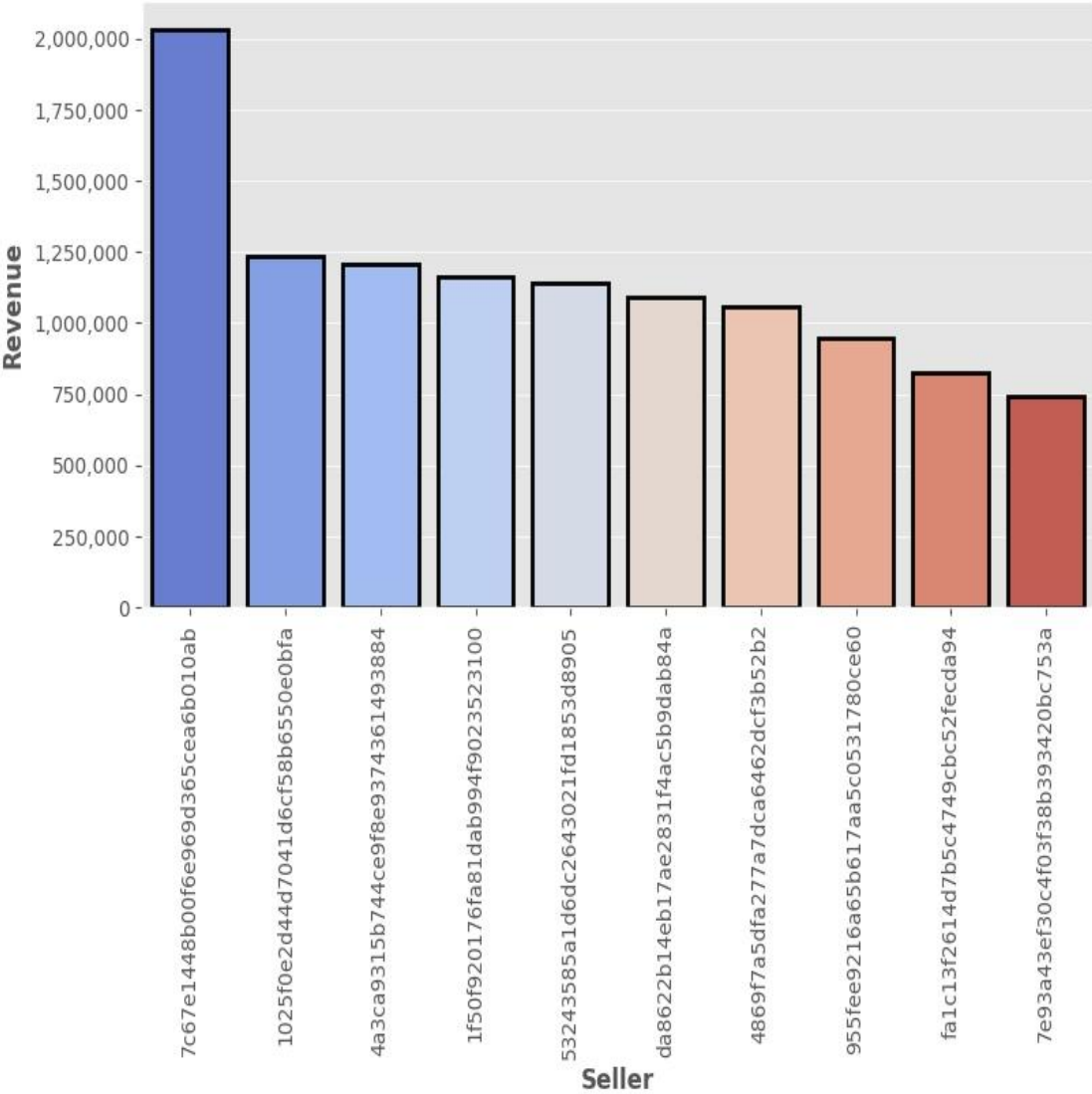
d. Top 3 Customers by Year

This chart highlights the top three customers who spent the most money in each year, showing shifts in high-value customer retention and acquisition.

```
[231]: query = """SELECT years, customer_id, payment, d_rank
       FROM
       (SELECT year(orders.order_purchase_timestamp) as years,
       orders.customer_id,
       sum(payments.payment_value) as payment,
       dense_rank() over (partition by year(orders.order_purchase_timestamp)
       ORDER BY sum(payments.payment_value) desc) d_rank
       FROM orders JOIN payments
       ON payments.order_id = orders.order_id
       GROUP BY years,
       orders.customer_id) as a
       WHERE d_rank <= 3"""


       cur.execute(query)

       data = cur.fetchall()
       top_3 = pd.DataFrame(data, columns = ["Years", "Customer ID", "Payment", "Rank"])
```

The top 3 customers who spent the most money in each year.

```
[232]: top_3
```

[232]:

	Years	Customer ID	Payment	Rank
0	2016	a9dc96b027d1252bbac0a9b72d837fc6	5694.200195	1
1	2016	1d34ed25963d5aae4cf3d7f3a4cda173	5602.959961	2
2	2016	4a06381959b6670756de02e07b83815f	4911.120117	3
3	2017	1617b1357756262bfa56ab541c47bc16	54656.320312	1
4	2017	c6e2731c5b391845f6800c97401a43a9	27717.240234	2
5	2017	3fd6777bbce08a352fddd04e4a7cc8f6	26906.640625	3
6	2018	ec5b2ba62e574342386871631fafd3fc	29099.519531	1
7	2018	f48d464a0baaea338cb25f816991ab1f	27688.839844	2
8	2018	e0a2412720e9ea4f26c1ac985f6a7358	19237.759766	3

Top 3 customers who spent the most money in each year.
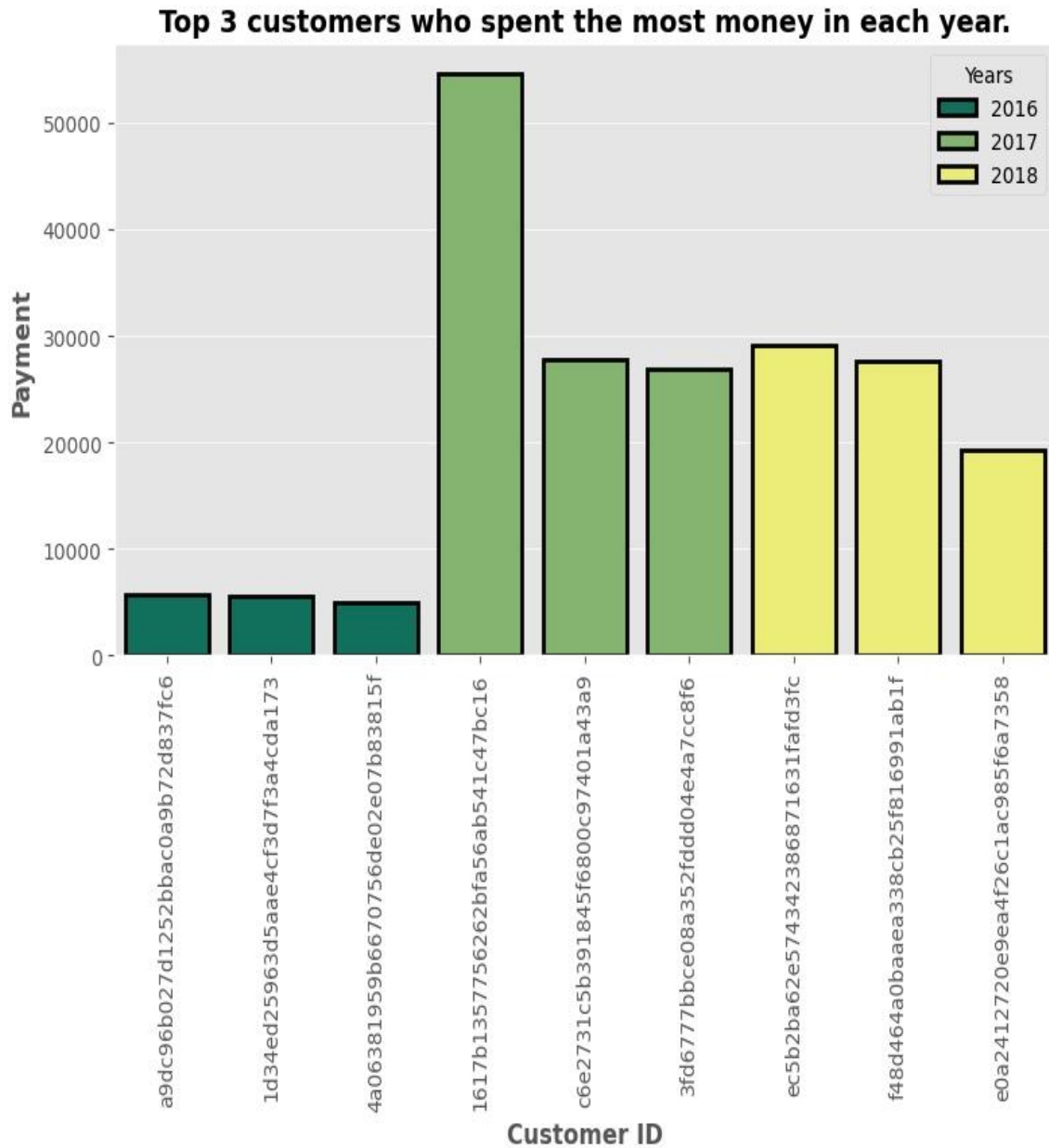
4. Methodology

Data cleaning included missing value imputation and duplicate removal. Analysis involved SQL queries for aggregation and Python (Pandas, Matplotlib, Seaborn) for visualization and deeper exploratory data analysis. The dashboard was created using matplotlib's gridspec for layout efficiency.

5. Dashboard Overview

The dashboard visually summarizes key metrics such as sales by category, monthly order volumes, top customers, sales by state, and top sellers, allowing rapid assessment of business performance and trends.

Ecommerce Sales Dashboard

Comprehensive overview dashboard summarizing key sales metrics for quick business performance assessment.
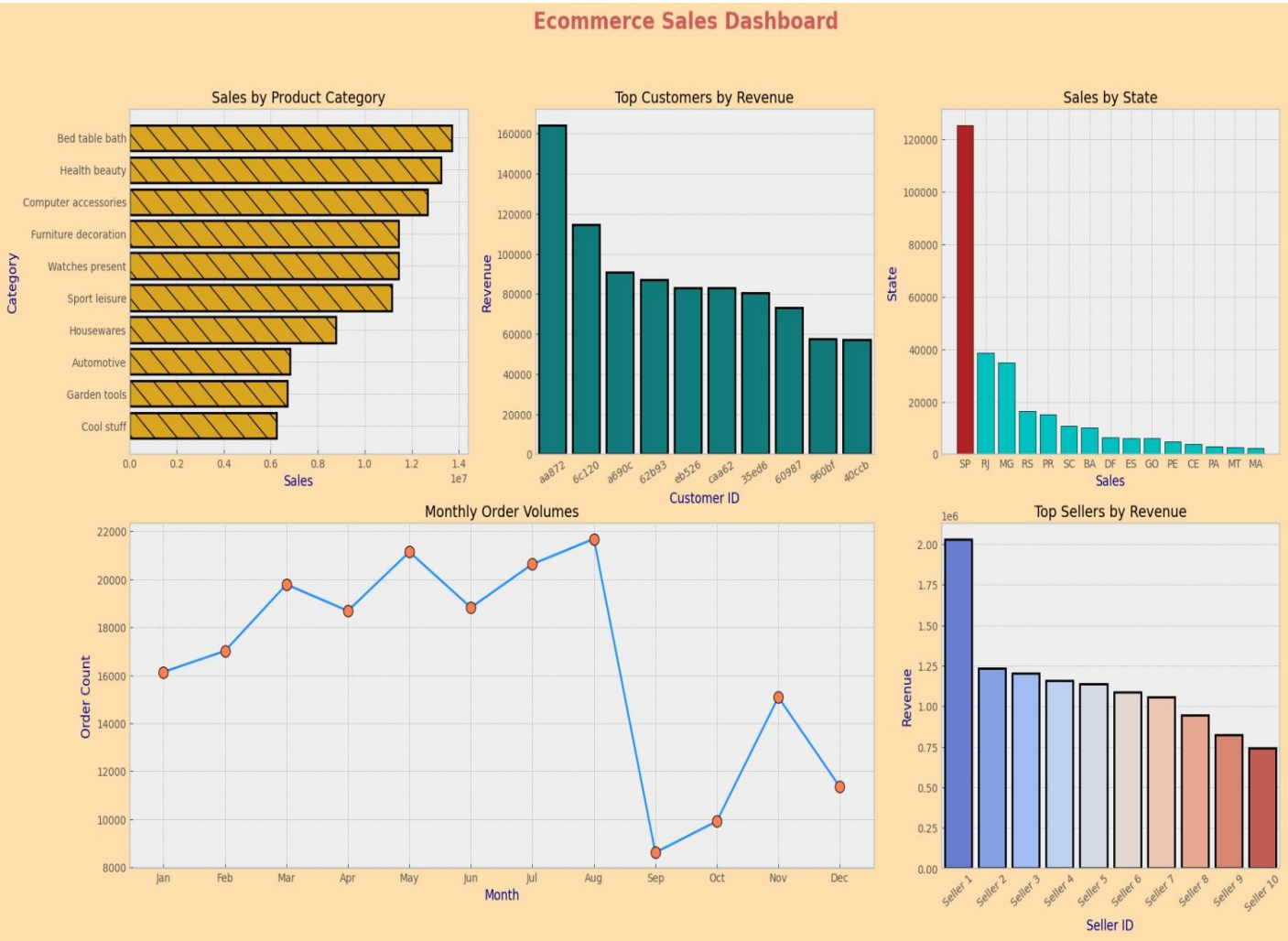
6. Conclusion

This analysis highlights key ecommerce sales drivers including product categories and geographic hotspots. The findings suggest focusing inventory on high-performing categories and targeting marketing in strong sales regions. Future work could incorporate customer segmentation and external market data to enhance decision-making.