

E-Commerce Sales & Profitability Analysis

Tools Used: Python | SQL | Power BI | MySQL

Prepared by: Anu Kumari

1. Introduction

This project focuses on analyzing an e-commerce dataset to understand sales performance, customer behavior, revenue trends, and overall profitability. The project demonstrates an end-to-end data analytics workflow where Python is used for data cleaning and exploratory analysis, SQL is used for business-level querying, and Power BI is used to build an interactive dashboard. The main objective is to transform raw transactional data into actionable business insights.

2. Problem Statement

The business requires clear visibility into its revenue and profit drivers. Key questions addressed in this project include: - What is the total revenue and total profit generated? - Which products and categories are profitable or loss-making? - How does revenue and profit change over time? - Which customer segments contribute the most to revenue? - How do order status and payment modes impact business performance?

3. Dataset Description

The analysis is based on multiple relational tables representing an e-commerce system:

- **customers:** customer_id, customer_segment, state
- **orders:** order_id, customer_id, order_date, order_status
- **order_items:** order_item_id, order_id, product_id, quantity, final_price
- **products:** product_id, category, cost
- **payments:** order_id, payment_mode, amount_paid, transaction_fee

Table Relationships

- customers.customer_id → orders.customer_id
- orders.order_id → order_items.order_id
- orders.order_id → payments.order_id
- products.product_id → order_items.product_id

These primary and foreign key relationships enable accurate joins and business calculations.

4. Python Implementation (Data Preparation & EDA)

Python was used for data ingestion, cleaning, and exploratory data analysis.

Key Activities:

- Loaded CSV files using Pandas
- Handled missing values and corrected data types
- Merged multiple tables into a single analytical dataset
- Calculated business metrics such as revenue, profit, and customer lifetime value
- Performed exploratory analysis to identify trends and outliers

Python helped validate the correctness of calculations later used in SQL and Power BI.

5. SQL Analysis (Business Logic)

SQL was used to perform intermediate to advanced analytics directly on normalized tables using joins, aggregations, and CASE expressions.

Key Analyses Performed:

- Total Revenue and Net Revenue calculation
- Profit calculation considering product cost and transaction fees
- Profit/Loss classification using CASE WHEN
- Customer value segmentation (High / Medium / Low)
- Category-wise and product-wise performance analysis
- Monthly revenue and profit trends

These queries convert transactional data into meaningful business metrics suitable for reporting.

6. Power BI Dashboard (Interactive Visualization)

Power BI was used to create an interactive dashboard that visually communicates key insights.

Dashboard Features:

- KPI cards for Total Customers, Total Revenue, Total Profit, and Return Rate
- Revenue by Order Status
- Payment Mode Distribution
- Monthly Revenue Trend
- State-wise Profit Analysis
- Customer Rating Distribution
- Category-based slicers for interactivity

How the Dashboard is Interactive

- **Category slicer:** Filters all visuals by selected product category
- **Cross-filtering:** Clicking on a chart element updates other visuals automatically
- **Time analysis:** Monthly trend charts allow easy comparison across periods
- **Drill-down ready visuals:** Category → Product level analysis

The dashboard allows users to explore data dynamically and supports data-driven decision-making.

7. Key Insights & Findings

- A small number of products contribute a large portion of total profit.
 - Certain categories generate high revenue but lower profit margins.
 - Some orders result in losses due to high product cost or transaction fees.
 - Revenue shows clear monthly trends, indicating seasonality.
 - High-value customers contribute disproportionately to total revenue.
-

8. Business Recommendations

- Re-evaluate pricing strategies for loss-making products.
 - Focus marketing efforts on high-value customer segments.
 - Encourage cost-efficient payment modes to reduce transaction fees.
 - Improve inventory and category-level margin management.
-

9. Conclusion

This project demonstrates a complete data analytics lifecycle using Python, SQL, and Power BI. By integrating data engineering, business querying, and interactive visualization, the project delivers meaningful insights that can help improve profitability, customer targeting, and operational efficiency in an e-commerce environment.

10. Tools & Technologies

- Python (Pandas, NumPy, Matplotlib, Seaborn)
- SQL (Joins, CASE WHEN, Aggregations)
- MySQL
- Power BI (DAX, Interactive Dashboards)