Olist E-commerce Full-Stack Marketing Analytics with Product Segmentation

Project Overview

This comprehensive end-to-end marketing analytics project illustrates my ability to design, implement, and deploy a full-stack data solution transforming raw e-commerce data into business-critical recommendations. The focus is on product performance segmentation within a simulated online retail platform, culminating in an interactive dashboard that delivers actionable insights to stakeholders.

1. Introduction: What Is Olist?

Olist is a Brazilian company that empowers small and medium-sized retailers to list and sell their products across major online marketplaces, including Amazon, Mercado Libre, and Magalu. Rather than managing individual accounts on each platform, sellers register through Olist, which:

- Connects their product listings to multiple marketplaces automatically
- Manages logistics, customer service, and order fulfillment
- Helps businesses increase visibility, improve sales efficiency, and scale operations

Olist serves as a commercial enabler, bridging independent sellers and large marketplaces through a unified infrastructure that simplifies digital commerce.

2. Database Schema Design and Implementation

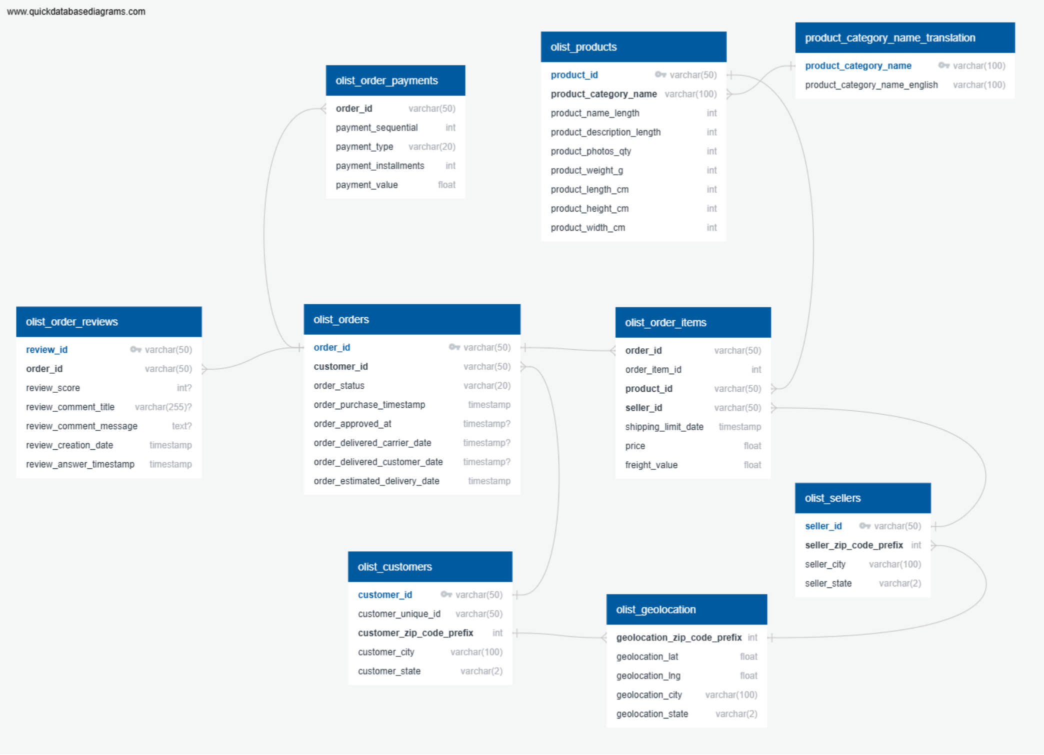
To replicate a real-world analytics ecosystem, the raw CSV data provided by Olist was transformed into a normalized relational database schema. The schema was modeled using **QuickDB** (an ERD tool) and implemented in **PostgreSQL**, enabling robust data governance and complex querying.

Key motivations for using a relational database included:

- **Referential integrity** (e.g., foreign keys linking orders to valid customers, products, and sellers)
- **Query efficiency** for multi-table joins and aggregations
- **Replicability** of e-commerce transactional architecture

This structured approach ensured clean, validated, and scalable storage to support all downstream analytics.

Entity Relationship Diagram (ERD):



www.quickdatabasediagrams.com
olist_order_payments
order_id varchar(50)
payment_sequential int
payment_type varchar(20)
payment_installments int
payment_value float
olist_products
product_id varchar(50)
product_category_name varchar(100)
product_name_length int
product_description_length int
product_photos_qty int
product_weight_g int
product_length_cm int
product_height_cm int
product_width_cm int
product_category_name_translation
product_category_name varchar(100)
product_category_name_english varchar(100)
olist_order_reviews
review_id varchar(50)
order_id varchar(50)
review_score int?
review_comment_title varchar(255)?
review_comment_message text?
review_creation_date timestamp
review_answer_timestamp timestamp
olist_orders
order_id varchar(50)
customer_id varchar(50)
order_status varchar(20)
order_purchase_timestamp timestamp
order_approved_at timestamp?
order_delivered_carrier_date timestamp?
order_delivered_customer_date timestamp?
order_estimated_delivery_date timestamp
olist_order_items
order_id varchar(50)
order_item_id int
product_id varchar(50)
seller_id varchar(50)
shipping_limit_date timestamp
price float
freight_value float
olist_sellers
seller_id varchar(50)
seller_zip_code_prefix int
seller_city varchar(100)
seller_state varchar(2)
olist_customers
customer_id varchar(50)
customer_unique_id varchar(50)
customer_zip_code_prefix int
customer_city varchar(100)
customer_state varchar(2)
olist_geolocation
geolocation_zip_code_prefix int
geolocation_lat float
geolocation_lng float
geolocation_city varchar(100)
geolocation_state varchar(2)

3. Data Extraction and Processing (Python + SQL)

Once the database was established, the next step was data validation and processing. Using **psycopg2** and **SQLAlchemy**, data was extracted from PostgreSQL and transformed using Python and Pandas. Key processing steps included:

- Multi-table joins (orders, reviews, products, sellers)
- Calculating KPIs such as sales volume, revenue, review scores, and delivery times
- Creating derived features (e.g., product performance scores, satisfaction indexes)

The resulting DataFrame served as the cleaned dataset for the segmentation pipeline.

4. Product Segmentation Analysis

4.1 Feature Selection and KPI Aggregation

To enable meaningful product clustering, KPIs were aggregated at the product_category_name level:

Feature	**Description**
total_sales_volume	Total units sold per category
total_revenue	Total revenue per category
avg_review_score	Average review rating (1 to 5)
avg_delivery_delay	Mean delay in delivery (days)

These metrics capture both commercial and operational performance.

4.2 Data Cleaning and Transformation

To prepare for modeling:

- **Log Transformation** was applied to skewed features (revenue, sales)
- **Outlier Removal** excluded the top 5% values (95th percentile)
- **Standardization** used z-score scaling to normalize all variables

4.3 Dimensionality Reduction (PCA)

To reduce dimensionality and improve cluster visualization, **Principal Component Analysis (PCA)** was used. The first two principal components captured most of the dataset's variance and were used to plot and interpret cluster formations.

4.4 Optimal Clustering with Elbow Method

The **Elbow Method** was applied to determine the optimal number of clusters (k), examining inertia across different values of k. A clear inflection point at **k = 4** suggested the best balance between simplicity and explanatory power.

4.5 K-Means Clustering and Labeling

Using **K-Means**, product categories were grouped into four strategic clusters:

Segment	**Strategic Meaning**
Top Sellers	High sales and revenue; prioritize inventory and logistics
Reliable Performers	Moderate sales, excellent reviews; enhance exposure
Niche Favorites	Low sales, high reviews; target loyal customer bases with promotions
Underperformers	Low across KPIs; investigate issues in pricing, fulfillment, or quality

---

5. Dashboard Design and Insights

5.1 Dashboard Purpose

A dashboard was developed using **Streamlit** and **Plotly** to bridge technical insights with executive decision-making. It enables stakeholders to:

- Select a product cluster via dropdown
- View KPIs for each cluster
- Explore top-selling categories
- Download filtered datasets as CSV

5.2 Technical Stack

Component	**Purpose**
Python	Data extraction, transformation, clustering
PostgreSQL (Docker)	Structured data backend
Streamlit	Dashboard UI and logic
Plotly	Interactive visualizations
Docker	Environment reproducibility and portability
QuickDB	Schema modeling and ERD visualization

5.3 Strategic Use of Clusters

Each cluster comes with embedded business recommendations, enabling marketing, product, and logistics teams to take targeted actions immediately.

5.4 Deliverables

- Relational database in PostgreSQL
- Python notebooks for preprocessing, clustering, and modeling
- Streamlit dashboard with cluster exploration tools
- CSV download functionality for segment-specific insights
- Dockerized deployment instructions

6. Conclusion and Future Directions

This project showcases the full lifecycle of a data-driven marketing analytics solution from structured database design and data transformation to clustering-based product segmentation and interactive dashboard deployment. Through the use of PostgreSQL, Python, and Streamlit, we translated raw e-commerce data into strategic insights that support product positioning and business decision-making.

Looking ahead, this system provides a solid foundation for expanding into deeper, business-critical analyses. Planned next steps include:

- **Customer Lifetime Value (LTV/CLV)** modeling to support strategic targeting
- **Churn prediction** to proactively retain high-risk customers
- **Customer Acquisition Cost (CAC)** estimation per channel or campaign
- **Sentiment analysis** from reviews to inform product and service improvements
- **Target segment identification** using behavioral clustering

To further elevate the system, we aim to integrate advanced capabilities such as:

- **A/B testing frameworks** for experimentation and campaign evaluation
- **Customer personas** combining behavioral and demographic clustering
- **Revenue forecasting** using time-series models like SARIMA and LSTM
- **Inventory optimization** based on sales velocity and demand predictions

These enhancements will transform the project into a comprehensive marketing intelligence platform positioned to drive smarter decisions in acquisition, retention, and operational efficiency.