Brazilian E-commerce Data Analytics Reporting

Presented By: Ganna Hassan

INTRODUCTION

This project focuses on analyzing raw Brazilian e-commerce data to generate actionable insights. The main goal is to create a smart visualization dashboard that helps businesses understand online retail dynamics in Brazil. The project addresses challenges like unclear data patterns and regional disparities in sales and customer behavior.

Objectives of the Analysis

- *Understand customer purchasing behavior.*
- *Identify top-selling product categories and regions.*
- *Analyze payment preferences and delivery performance.*
- *Calculate Customer Lifetime Value (CLV).*
- *Detect high-return product categories.*
- *Offer insights to improve decision-making for e-commerce stakeholders*

Dataset Description

- Data Source: Brazilian E-commerce Public Dataset
- Tables Involved:
- Customers: Customer IDs, location info.

- 2. Orders: Order lifecycle details.

- 3. Order Items: Product, seller, price, shipping info.

- 4. Order Payments: Payment method and value.

- 5. Order Reviews: Customer satisfaction scores.

- 6. Products: Product category details.

- 7. Sellers: Seller location and IDs.

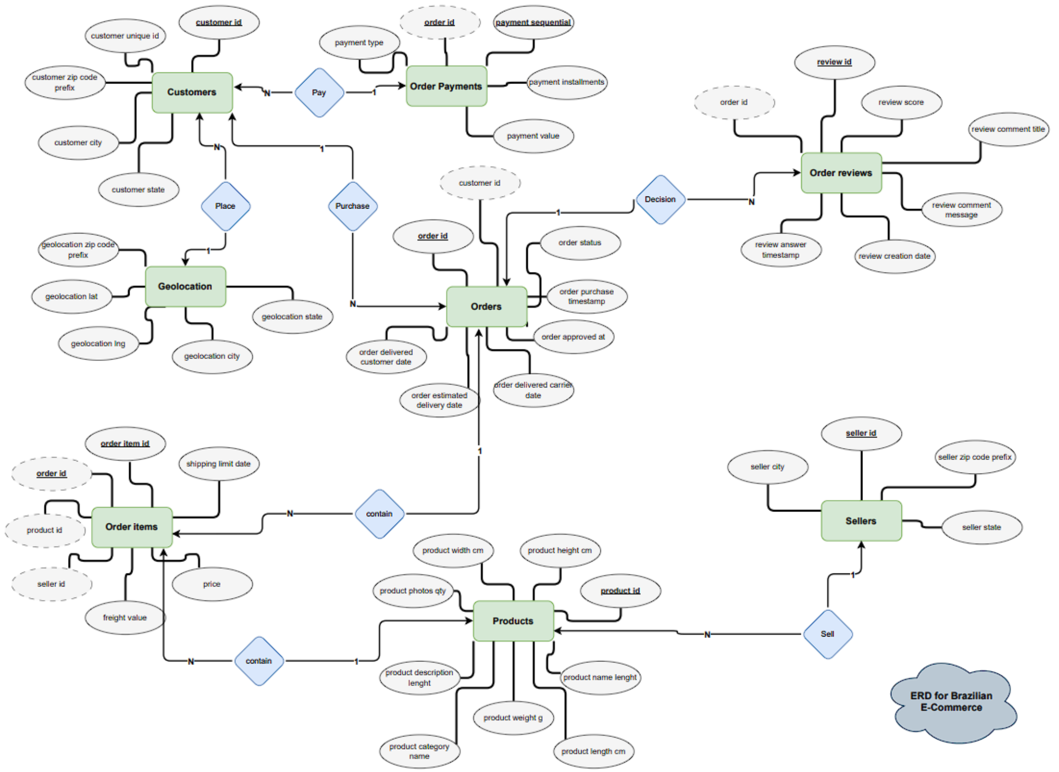
- 8. Geolocation: Mapping ZIP codes to cities/states.

- Data Type: Relational, structured in CSV
- Format Storage: PostgreSQL.
- Access: REST API (Flask) with React frontend.

Data Cleaning & Preparation

Steps Taken
✓ Removed duplicate ✓ Rows Handled missing values (e.g., imputing ZIP codes using city averages).
✓ Standardized formats (e.g., date fields, category name normalization from Portuguese to English). ✓ Converted data types (e.g., strings to datetime).
✓ Corrected inaccurate or inconsistent entries. ✓ Detected and managed outliers.
✓ Applied normalization/scaling for analysis. ✓ Tools used: Python (Pandas), SQL.

Relations Between Data Points



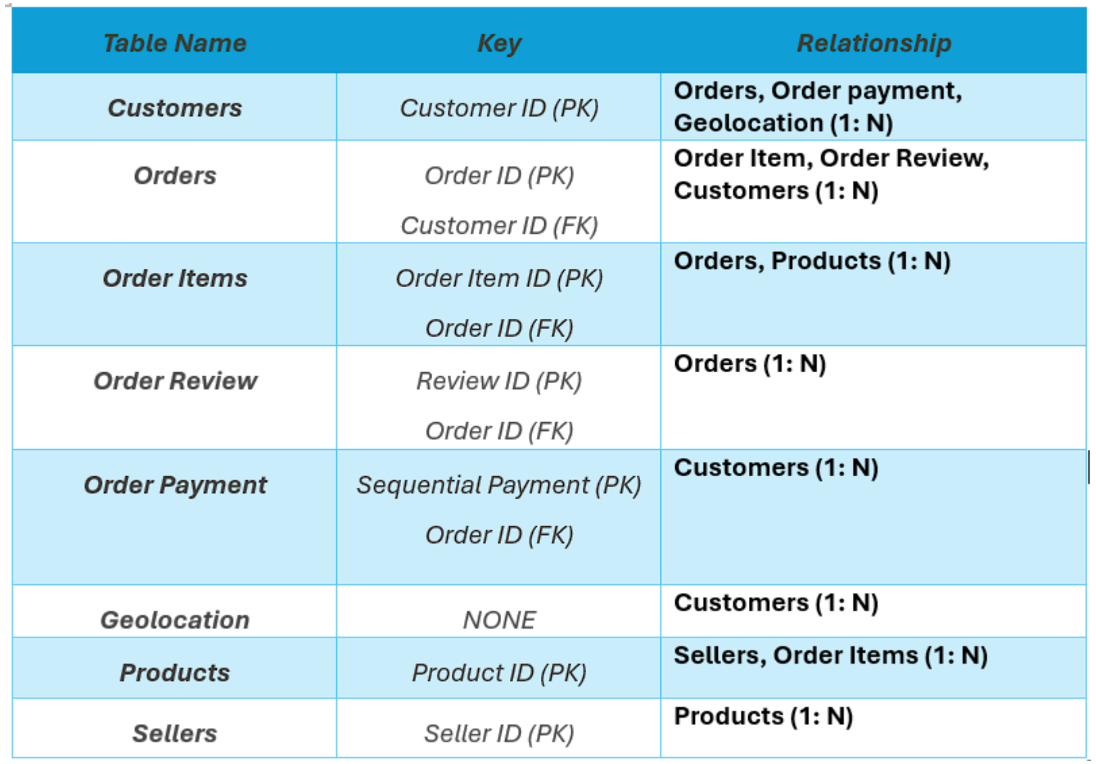
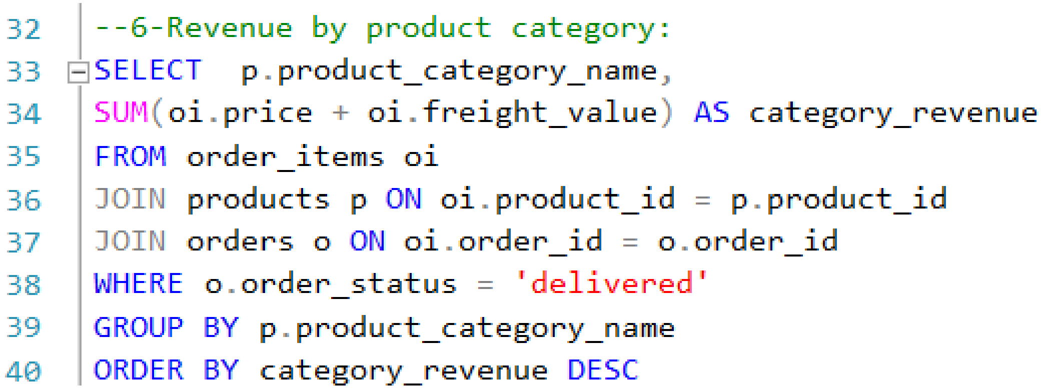


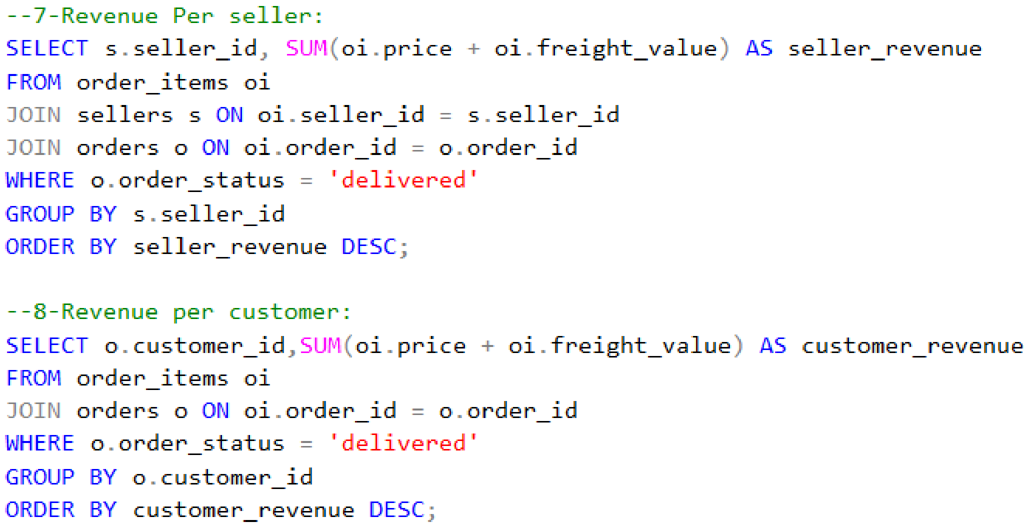
Table Name	Key	Relationship
Customers	Customer ID (PK)	Orders, Order payment, Geolocation (1: N)
Orders	Order ID (PK) Customer ID (FK)	Order Item, Order Review, Customers (1: N)
Order Items	Order Item ID (PK) Order ID (FK)	Orders, Products (1: N)
Order Review	Review ID (PK) Order ID (FK)	Orders (1: N)
Order Payment	Sequential Payment (PK) Order ID (FK)	Customers (1: N)
Geolocation	NONE	Customers (1: N)
Products	Product ID (PK)	Sellers, Order Items (1: N)
Sellers	Seller ID (PK)	Products (1: N)

A
B
Exploratory Data Analysis

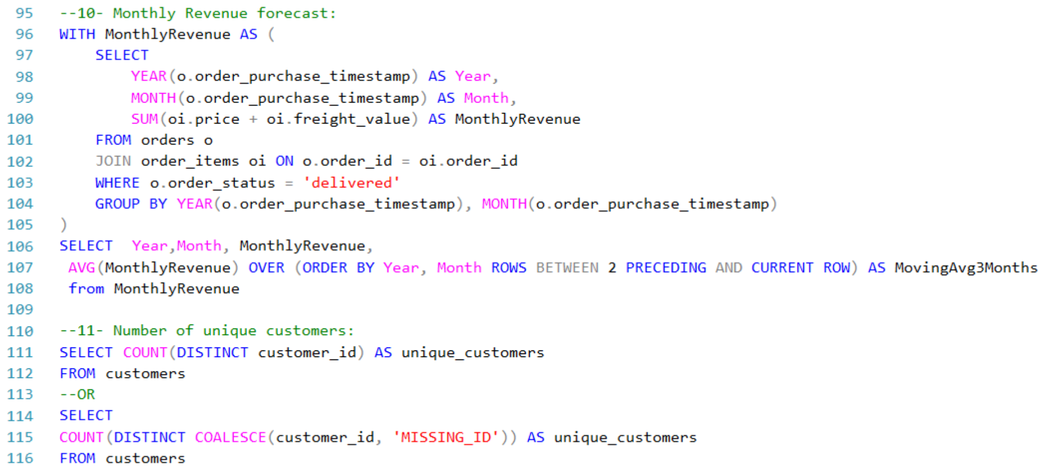
Exploratory Data Analysis (EDA)



```sql
--6-Revenue by product category:
SELECT  p.product_category_name,
SUM(oi.price + oi.freight_value) AS category_revenue
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
JOIN orders o ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY p.product_category_name
ORDER BY category_revenue DESC
```



```sql
--7-Revenue Per seller:
SELECT s.seller_id, SUM(oi.price + oi.freight_value) AS seller_revenue
FROM order_items oi
JOIN sellers s ON oi.seller_id = s.seller_id
JOIN orders o ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY s.seller_id
ORDER BY seller_revenue DESC;

--8-Revenue per customer:
SELECT o.customer_id,SUM(oi.price + oi.freight_value) AS customer_revenue
FROM order_items oi
JOIN orders o ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY o.customer_id
ORDER BY customer_revenue DESC;
```

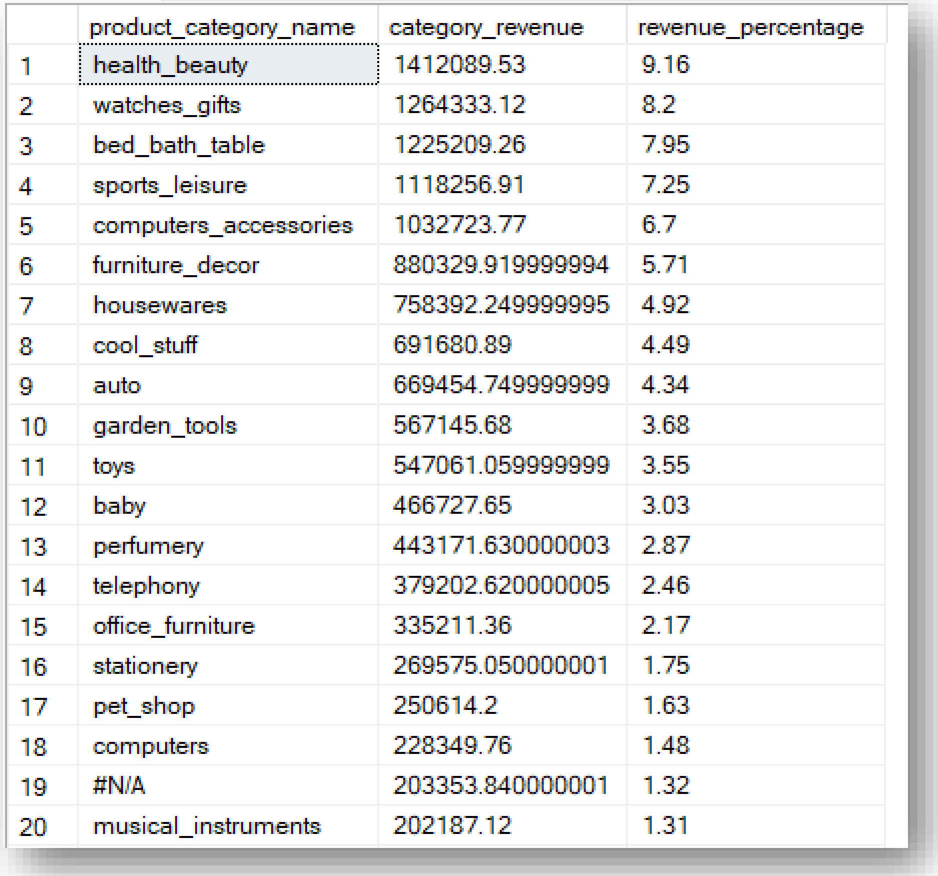


```sql
--10- Monthly Revenue forecast:
WITH MonthlyRevenue AS (
    SELECT
        YEAR(o.order_purchase_timestamp) AS Year,
        MONTH(o.order_purchase_timestamp) AS Month,
        SUM(oi.price + oi.freight_value) AS MonthlyRevenue
    FROM orders o
    JOIN order_items oi ON o.order_id = oi.order_id
    WHERE o.order_status = 'delivered'
    GROUP BY YEAR(o.order_purchase_timestamp), MONTH(o.order_purchase_timestamp)
)
SELECT  Year,Month, MonthlyRevenue,
 AVG(MonthlyRevenue) OVER (ORDER BY Year, Month ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS MovingAvg3Months
 from MonthlyRevenue

--11- Number of unique customers:
SELECT COUNT(DISTINCT customer_id) AS unique_customers
FROM customers
--OR
SELECT
COUNT(DISTINCT COALESCE(customer_id, 'MISSING_ID')) AS unique_customers
FROM customers
```

Exploratory Data Analysis (EDA)

Performed using SQL and Python :Overviewed distributions of sales, payments, and customer locations.
Plotted frequency charts for: Order statuses.
Payment types Review scores.
Regional order distributions.
Time series plots for peak purchase times.

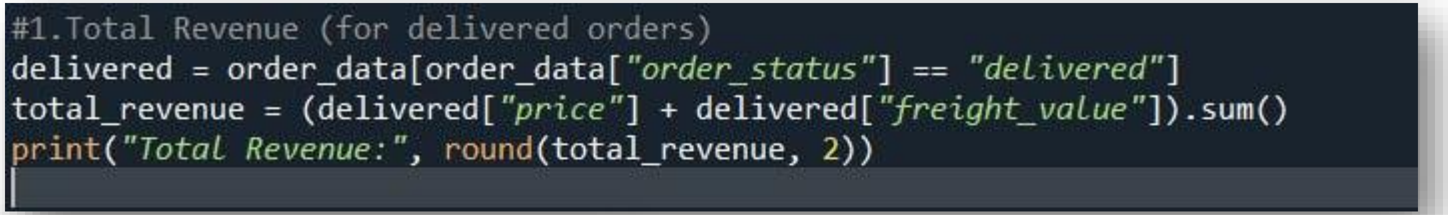
Sample Of KPIs : Revenue per product category By SQL



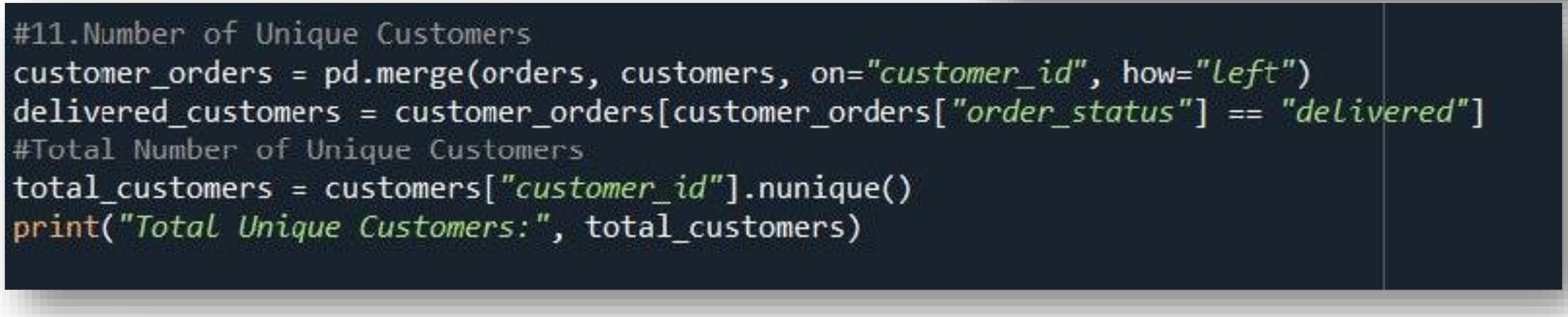
	product_category_name	category_revenue	revenue_percentage
1	health_beauty	1412089.53	9.16
2	watches_gifts	1264333.12	8.2
3	bed_bath_table	1225209.26	7.95
4	sports_leisure	1118256.91	7.25
5	computers_accessories	1032723.77	6.7
6	furniture_decor	880329.919999994	5.71
7	housewares	758392.249999995	4.92
8	cool_stuff	691680.89	4.49
9	auto	669454.749999999	4.34
10	garden_tools	567145.68	3.68
11	toys	547061.059999999	3.55
12	baby	466727.65	3.03
13	perfumery	443171.630000003	2.87
14	telephony	379202.620000005	2.46
15	office_furniture	335211.36	2.17
16	stationery	269575.050000001	1.75
17	pet_shop	250614.2	1.63
18	computers	228349.76	1.48
19	#N/A	203353.840000001	1.32
20	musical_instruments	202187.12	1.31

Exploratory Data Analysis (EDA)

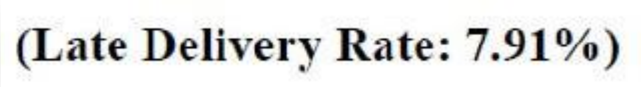
Using Python



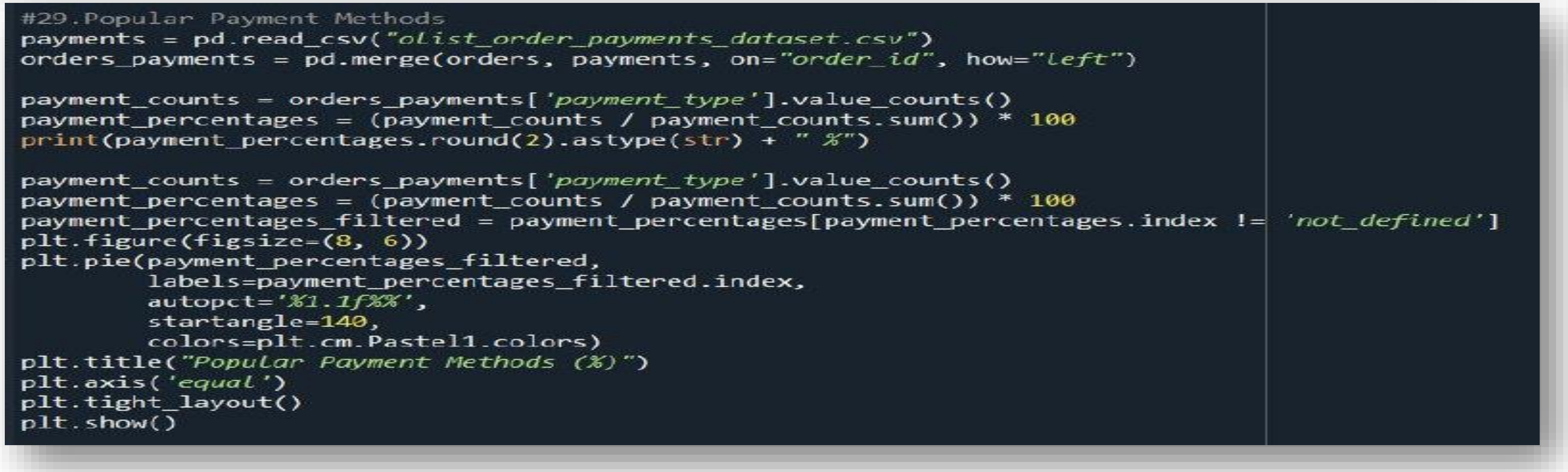
```
#1.Total Revenue (for delivered orders)
delivered = order_data[order_data["order_status"] == "delivered"]
total_revenue = (delivered["price"] + delivered["freight_value"]).sum()
print("Total Revenue:", round(total_revenue, 2))
```



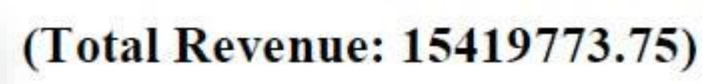
```
#11.Number of Unique Customers
customer_orders = pd.merge(orders, customers, on="customer_id", how="left")
delivered_customers = customer_orders[customer_orders["order_status"] == "delivered"]
#Total Number of Unique Customers
total_customers = customers["customer_id"].nunique()
print("Total Unique Customers:", total_customers)
```



(Late Delivery Rate: 7.91%)



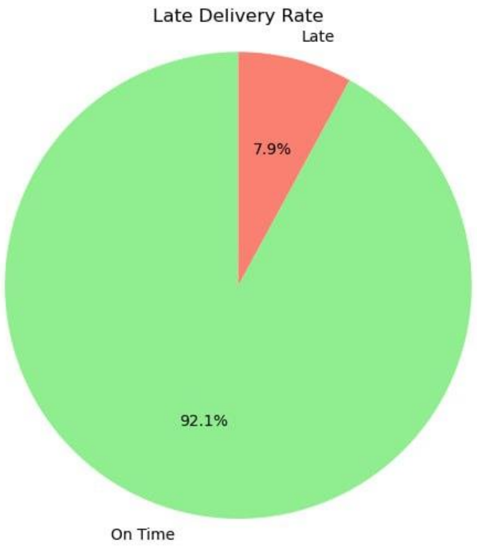
```
#29.Popular Payment Methods
payments = pd.read_csv("olist_order_payments_dataset.csv")
orders_payments = pd.merge(orders, payments, on="order_id", how="left")

payment_counts = orders_payments['payment_type'].value_counts()
payment_percentages = (payment_counts / payment_counts.sum()) * 100
print(payment_percentages.round(2).astype(str) + " %")

payment_counts = orders_payments['payment_type'].value_counts()
payment_percentages = (payment_counts / payment_counts.sum()) * 100
payment_percentages_filtered = payment_percentages[payment_percentages.index != 'not_defined']
plt.figure(figsize=(8, 6))
plt.pie(payment_percentages_filtered,
        labels=payment_percentages_filtered.index,
        autopct='%1.1f%%',
        startangle=140,
        colors=plt.cm.Pastel1.colors)
plt.title("Popular Payment Methods (%)")
plt.axis('equal')
plt.tight_layout()
plt.show()
```

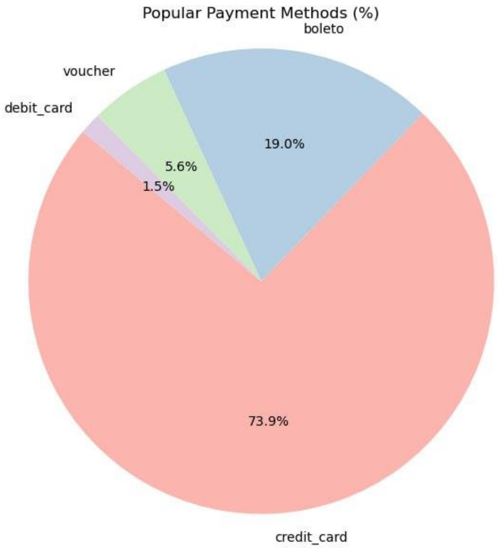


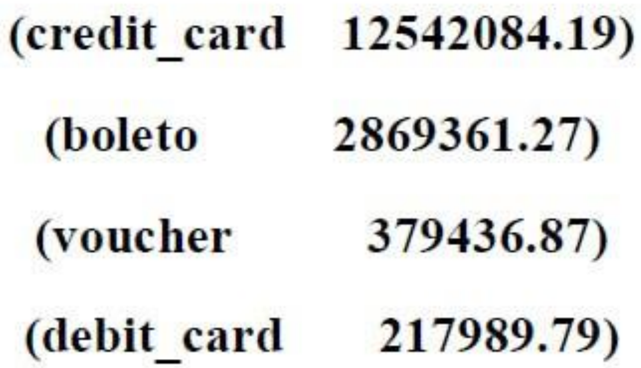
(Total Revenue: 15419773.75)

Exploratory Data Analysis (EDA)

Samples of Visualization Using Python







```
(credit_card   12542084.19)
 (boleto        2869361.27)
 (voucher        379436.87)
 (debit_card     217989.79)
```

Deep-Dive Analysis

- State-Level Sales: Mapped sales by region using geolocation.
- Top Categories: Ranked product categories by revenue.
- Payment Analysis: Identified most common payment methods.
- Delivery Performance: Evaluated shipping delays and average delivery times by state.
- Customer CLV: Analyzed which customers contributed the most revenue.
- Return Rates: Flagged categories with high return frequencies.
- Temporal Trends: Tracked weekly/monthly sales patterns.

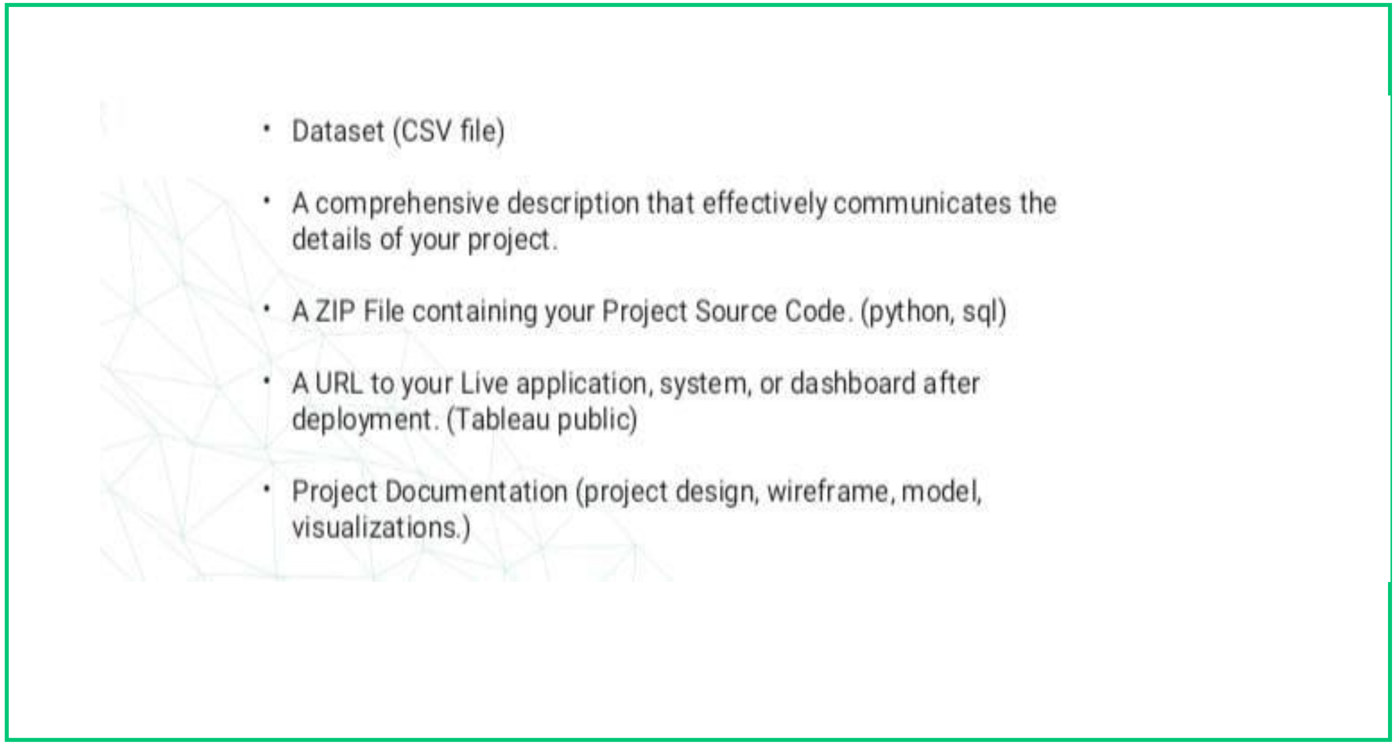
Key Insights

- Southeast Brazil leads in both order volume and revenue.
- Electronics and fashion are top-performing product categories.
- "Credit card" is the dominant payment method.
- Delivery delays occur more frequently in northern regions.
- Customers with high purchase frequency often give higher review scores.
- Return rates were highest in fashion and furniture categories.

Recommendations

- Optimize Logistics in underperforming delivery regions.
- Focus Promotions on high-CLV customers.
- Improve Return Policies in high-return categories.
- Enhance Seller Dashboards with product-level performance alerts.
- Introduce Loyalty Programs for repeat buyers.

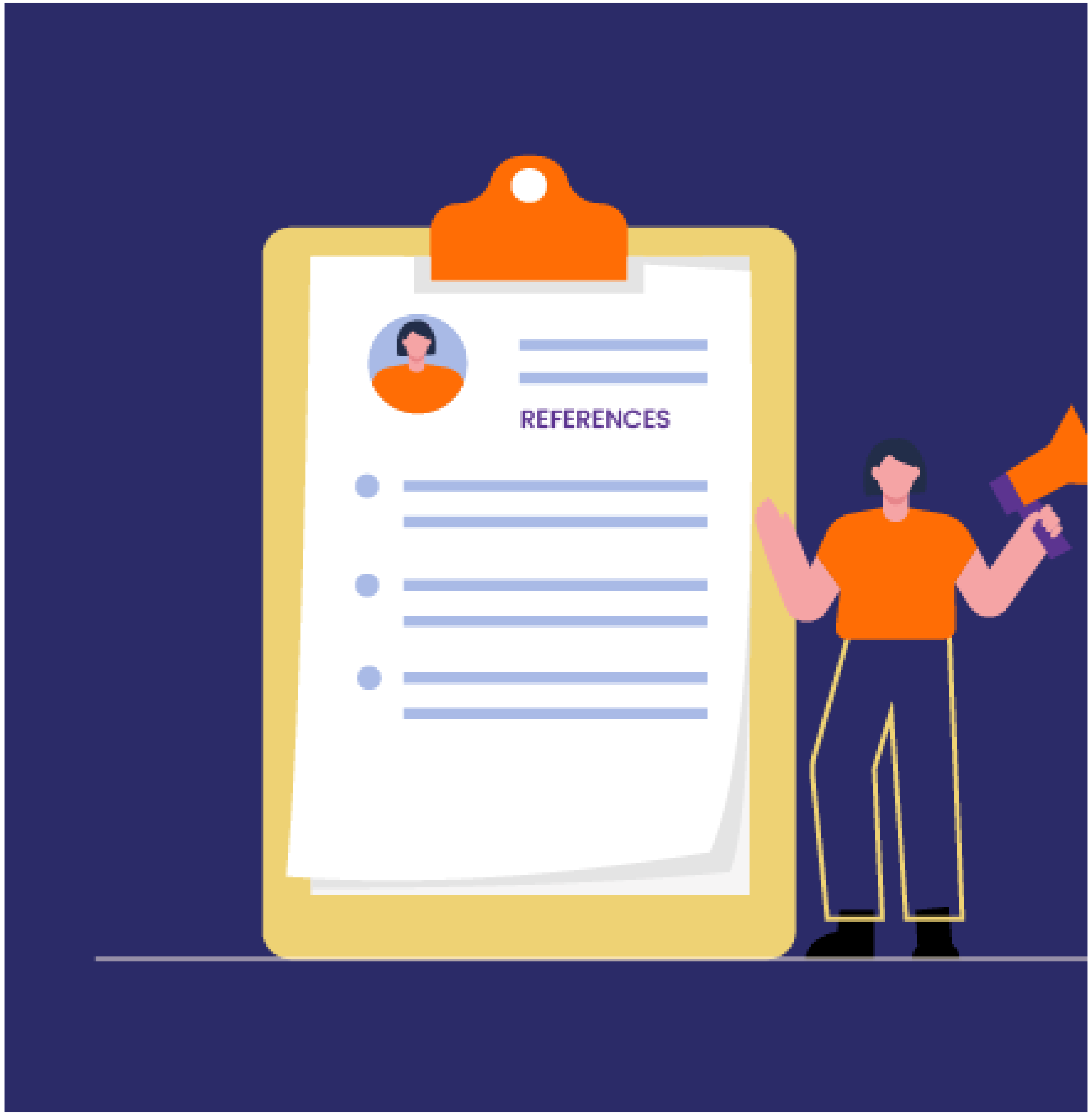
Deliverables



- Dataset (CSV file)
- A comprehensive description that effectively communicates the details of your project.
- A ZIP File containing your Project Source Code. (python, sql)
- A URL to your Live application, system, or dashboard after deployment. (Tableau public)
- Project Documentation (project design, wireframe, model, visualizations.)

Conclusion

This project successfully transformed complex Brazilian e-commerce data into a set of meaningful dashboards and reports. It provides business users with an interactive platform to explore trends, address customer needs, and optimize operations.



References

- Python (Pandas)
- Microsoft SQL Server
- Tableau & Power BI
- Google Cloud Platform
- GitHub for version control
- Collaboration methods (e.g., communication tools, agile methodologies)
- ⍰ Used GitHub for version control.
- ⍰ Held weekly syncs via Google Meet.