**Team : Fantastic Quartz**

**Project : Brazil Olist E-Commerce**

**Introduction​**

Olist is a prominent player in the Brazilian e-commerce market, providing small and medium enterprises (SMEs) with access to a vast marketplace. The analysis of Olist’s 100k orders dataset from 2016 to 2018 aims to refine strategies for improving customer interactions, optimizing operations, and increasing order volumes. Key insights will focus on delivery performance, regional distribution, and special event preparedness.

**Problem Statement**

* Analyze 8 % late deliveries
* Find relationship between Customer’s Satisfaction and Late Deliveries
* Compare World Cup vs Normal Day delivery performance

**Data Sources**

9 csv files from Kaggle project – Brazil Olist

<https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce>

**olist\_customers\_dataset.csv**

* Contains customer information, including unique IDs, zip codes, and city/state details

**olist\_geolocation\_dataset.csv**

* Provides geolocation data, mapping zip codes to latitude and longitude coordinates

**olist\_order\_items\_dataset.csv**

* Includes details of items within each order, such as product ID, price, freight value, and shipping details

**olist\_order\_payments\_dataset.csv**

* Captures payment-related information, including payment type, number of installments, and total payment amount

**olist\_order\_reviews\_dataset.csv**

* Contains customer reviews, ratings, and timestamps for feedback on orders.

**olist\_orders\_dataset.csv**

* Provides order-level information, including order status, purchase timestamp, delivery times, and customer IDs

**olist\_products\_dataset.csv**

* Includes product details such as category names, dimensions, and product descriptions

**olist\_sellers\_dataset.csv**

* Contains seller information, including unique IDs and location details (city, state, zip code)

**product\_category\_name\_translation.csv**

* Offers translations of product categories from Portuguese to English for better interpretability
* Our project uses Panda dataframe in jupyter notebook to extract CSV files data

**Tools & Libraries**

* **Programming Languages** : python (pandas, NumPy), plotly, matplotlib, scikit
* **Data Visualization** : Power BI, Jupyter notebook,
* **Database Management** : AWS RDS (Relational Database Service)
* **Machine Learning** : Predictive analytics using ML models for clustering

**\*Note : For more coding details please refer to jupyter notebook**  
**ETL Processes /Approach**

**Extract** :

* Extract raw data from 9 CSV files

**Transform** :

* Added new columns (delivery\_delay, delay\_category, month) to the orders table
* Drop duplicates and handle missing values (NA), change data types to numeric
* Create TABLE, Add foreign key constraint
* Configure , set up AWS RDS online
* Standardize data queries across platforms to ensure consistency

**Load** :

* Deploy transformed data to AWS RDS for cloud-based storage and querying

**Architecture Diagram**

**Data Source Layer** : Raw CSV files.

**Staging Area** : Intermediate storage for data cleaning and transformation.

**Database Layer** : AWS RDS hosting the cleaned and structured data.

**Visualization Layer** : Power BI and Jupyter Notebook for reporting and analysis.

**A close-up of several icons

Description automatically generated**

**Schema Diagram**

**A diagram of a computer

Description automatically generated**

* Managed to link relationship among all 8 tables but our project mainly focusing on 4 tables - **orders , order\_items, geolocation, reviews**

The schema includes tables linked by foreign keys:

* orders → order\_items (via order\_id)
* orders → customers (via customer\_id)
* order\_items → products (via product\_id)
* order\_items → sellers (via seller\_id)
* reviews → orders (via order\_id)

**Data Validation** :

* Cross-check data consistency between Power BI and Jupyter Notebook visualizations.
* Validate transformations by comparing aggregated metrics (e.g., total orders, average delivery time) before and after ETL.

**Challenges**

**Foreign Key Implementation**

* Challenge : Database errors occurred when implementing foreign key constraints.
* Solution : Implemented foreign keys separately for each table before loading data into AWS RDS.

**AWS RDS Configuration**

* Challenge : Initial setup of the cloud database was complex.
* Solution : Utilized documentation, online resources, and tutorials for guidance.

**Data Visualization Consistency**

* Challenge : Discrepancies between Power BI and Jupyter Notebook visualizations.
* Solution : Standardized SQL queries across platforms to ensure consistent data inputs.

**\*Note : For more graph details please refer to Jupyter notebook and PowerBI file**

**Findings/ Analysis**

**Query**

Jupyter

**A screenshot of a computer code

Description automatically generated**

PowerBI

**A screenshot of a computer

Description automatically generated**

**A graph of blue bars

Description automatically generated with medium confidence**

**Query**

Jupyter

**A computer screen shot of a code

Description automatically generated**

PowerBI

**A close-up of a text

Description automatically generated**

**A blue pie chart with a number of numbers

Description automatically generatedA graph of a number of people

Description automatically generated with medium confidence**

**Delivery Performance**

* On-Time Deliveries : 92%
* Delayed Deliveries : 8%
* Monthly Trends:
* May: 6,873 orders
* June: 6,167 orders (-10.3% decline)
* July: 6,292 orders (+2.0% increase)

**A black text on a white background

Description automatically generated**

**Query**

Jupyter

A screen shot of a computer code

Description automatically generated

PowerBI

A screenshot of a computer

Description automatically generated

**A screenshot of a computer screen

Description automatically generated**

**A graph with dots and numbers

Description automatically generated with medium confidence**

**Regional Insights**

**Top 3 States by Orders**

* São Paulo (SP)
* Rio de Janeiro (RJ)
* Minas Gerais (MG)
* Average delivery satisfaction scores ranged from 3.6 to 4.2 across states.

**A close-up of a list of information

Description automatically generated**

**Query**

Jupyter

A screenshot of a computer code

Description automatically generated

PowerBI

A screenshot of a computer

Description automatically generated

**A graph of blue lines

Description automatically generated**

**Query**

Jupyter

**A screenshot of a computer program

Description automatically generated**

PowerBI

A screenshot of a computer program

Description automatically generated

**A graph of a customer satisfaction on delivery performance

Description automatically generatedA graph of a number of days to deliver order

Description automatically generated**

**World Cup vs Normal Day Comparison**

**A screenshot of a white background

Description automatically generated**

* Data shows events can cause order to drop
* Order delivery time is shortened, satisfaction improves

**Next Steps/Conclusion**

Suggested Implementation Plan

**Phase 1** : Data Analysis & Delivery Optimization (3 Months)

Implement predictive analytics using ML models to forecast delays.

**Phase 2** : Real-Time Tracking & Updates (3 Months)

Introduce GPS tracking and automated notifications for customers.

**Phase 3** : Regional Distribution Centers (6-8 Months)

Establish hubs in SP, RJ, and MG to reduce delivery times.

**Phase 4** : System Integration & Scalability (3 Months)

Ensure seamless integration of AWS-hosted tools with Olist’s infrastructure.

**Phase 5** : Special Event Preparedness (6-8 Months)

Scale delivery capacity and align inventory with demand during major events.

**Summary**

**A screenshot of a computer

Description automatically generated**

**Conclusion**

This project provides actionable insights to enhance Olist’s delivery performance and customer satisfaction. By leveraging historical data, predictive analytics, and strategic partnerships, Olist can achieve faster deliveries, reduced operational costs, and improved scalability