End-to-End E-Commerce Analytics: Delivery Performance & Customer Satisfaction

1. Business Understanding

The business aims to improve customer satisfaction while maintaining revenue growth. Delivery performance was suspected to be a key driver of customer sentiment, but required quantitative validation.

Stakeholder priorities:

- Improve customer satisfaction
- Reduce operational inefficiencies
- Support long-term revenue growth

2. Data Overview

Dataset used: **OLIST Brazilian E-Commerce Dataset**

Tables leveraged:

- Orders analyzed: 99,441
- Order items: 112,650
- Customers: 99,441
- Reviews: 99,224
- Products: 32,951
- Sellers: 3,095
- Time span: Sep 2016 – Oct 2018
- Analysis window: Most recent 12 months of available data
-

The dataset represents a multi-seller marketplace with realistic operational complexity.

3. Data Preparation

- Loaded raw CSV data into PostgreSQL
- Validated row counts and date ranges
- Created an analytics view scoped to the most recent 12 months
- Engineered delivery delay and late delivery indicators

```
olist_analytics=#
olist_analytics=# DROP VIEW IF EXISTS v_orders_12m;
NOTICE:  view "v_orders_12m" does not exist, skipping
DROP VIEW
olist_analytics=#
olist_analytics=# CREATE VIEW v_orders_12m AS
olist_analytics-# WITH max_date AS (
olist_analytics(#   SELECT MAX(order_purchase_timestamp) AS max_purchase
olist_analytics(#   FROM orders
olist_analytics(# ),
olist_analytics-# recent_orders AS (
olist_analytics(#   SELECT o.*
olist_analytics(#   FROM orders o
olist_analytics(#   CROSS JOIN max_date m
olist_analytics(#   WHERE o.order_purchase_timestamp >= (m.max_purchase - INTERVAL '12 months')
olist_analytics(# )
olist_analytics-# SELECT
olist_analytics-#   o.order_id,
olist_analytics-#   o.customer_id,
olist_analytics-#   o.order_status,
olist_analytics-#   o.order_purchase_timestamp::date AS purchase_date,
olist_analytics-#   DATE_TRUNC('month', o.order_purchase_timestamp)::date AS purchase_month,
olist_analytics-#   o.order_delivered_customer_date,
olist_analytics-#   o.order_estimated_delivery_date,
olist_analytics-#   CASE
olist_analytics-#     WHEN o.order_delivered_customer_date IS NULL OR o.order_estimated_delivery_date IS
NULL THEN NULL
olist_analytics-#     WHEN o.order_delivered_customer_date::date > o.order_estimated_delivery_date::date
THEN 1
olist_analytics-#     ELSE 0
olist_analytics-#   END AS is_late,
olist_analytics-#   CASE
olist_analytics-#     WHEN o.order_delivered_customer_date IS NULL OR o.order_estimated_delivery_date IS
NULL THEN NULL
olist_analytics-#     ELSE (o.order_delivered_customer_date::date - o.order_estimated_delivery_date::date
)
olist_analytics-#   END AS delivery_delay_days,
olist_analytics-#   r.review_score
olist_analytics-# FROM recent_orders o
```

1.1 SQL view creation and row validation queries

4. SQL Analysis

SQL was used to:

- Filter the most recent 12-month period
- Calculate late delivery rates
- Aggregate monthly revenue and AOV
- Join delivery outcomes with customer reviews

Key SQL techniques:

- CTEs
- Date arithmetic
- Conditional logic
- Aggregations and joins

total_orders	orders_with_delivery_info	late_orders	late_delivery_pct
69660	67924	5509	8.11

(1 row)

1.2 late delivery percentage

purchase_month	total_revenue	orders	aov
2017-10-01	346320.66	2084	166.18
2017-11-01	1187779.95	7451	159.41
2017-12-01	866838.54	5624	154.13
2018-01-01	1113929.01	7220	154.28
2018-02-01	998137.75	6694	149.11
2018-03-01	1159663.98	7188	161.33
2018-04-01	1162227.22	6934	167.61
2018-05-01	1150474.33	6853	167.88
2018-06-01	1023674.47	6160	166.18
2018-07-01	1061204.65	6273	169.17
2018-08-01	1003413.17	6452	155.52
2018-09-01	166.46	1	166.46

(12 rows)

1.3 revenue by month

is_late	orders	avg_review_score	pct_low_reviews
0	62095	4.29	9.40
1	5394	2.23	63.40

(2 rows)

1.4 review score aggregation

5. Python Analysis & Visualization

Python (Pandas, Matplotlib, Seaborn) was used to:

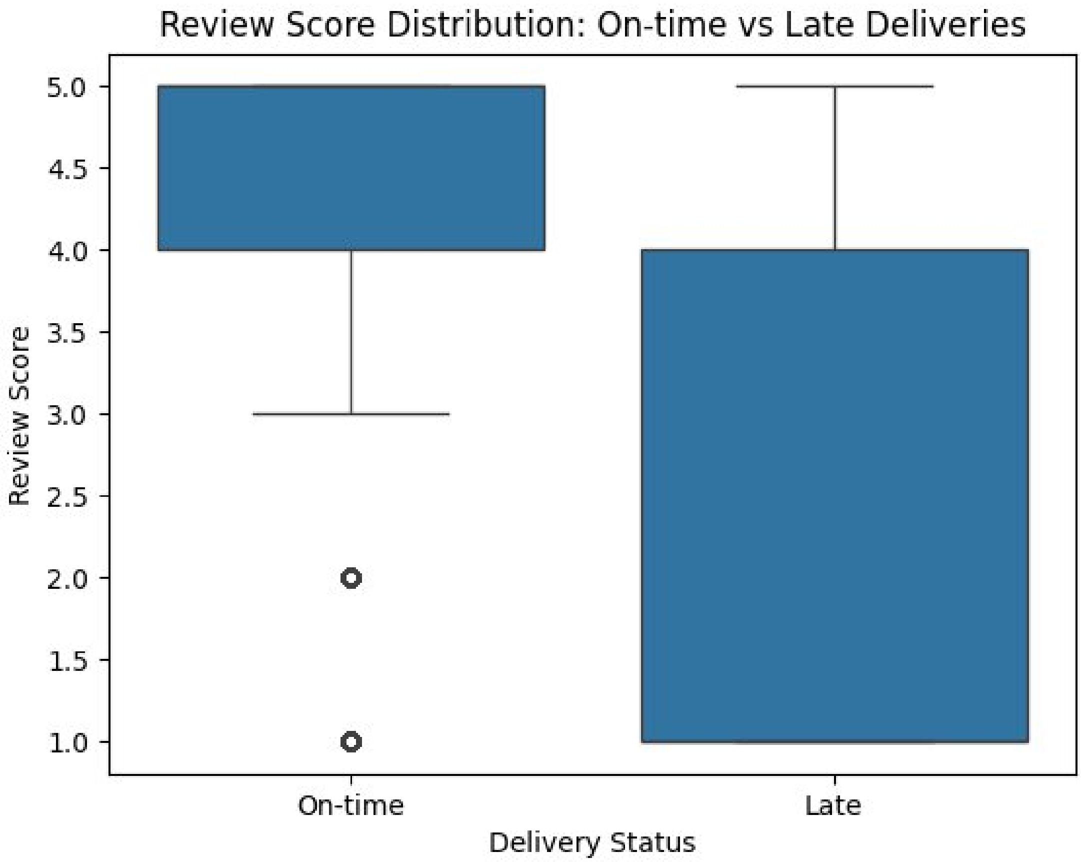
- Explore relationships between delivery delay and review score
- Visualize distributions and trends

- Validate findings from SQL analysis

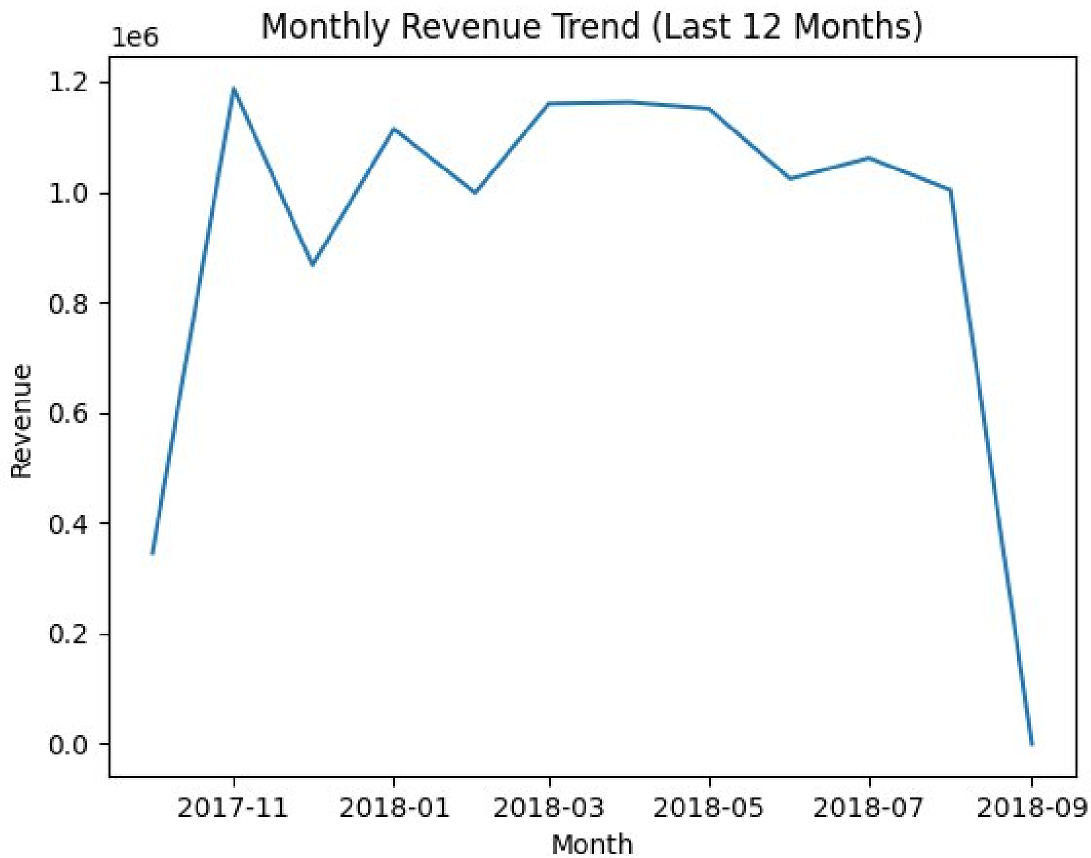
Visuals Generated

- Delivery delay vs review score scatter plot
- Review score distribution (on-time vs late)
- Monthly revenue trend line

1.5 VS Code – Scatter plot: Delivery Delay vs Review Score



1.6 VS Code – Boxplot: On-time vs Late Reviews



1.7 VS Code – Line chart: Monthly Revenue Trend

6. Key Findings

- Late deliveries are consistently associated with **lower review scores**
- On-time deliveries show **tighter, higher review score distributions**
- Even moderate delivery delays increase the likelihood of low ratings
- Revenue shows **seasonal variation**, not consistent growth
- Operational inefficiencies in delivery can indirectly impact revenue through customer dissatisfaction

7. Business Recommendations

1. Prioritize reducing delivery delays to improve customer satisfaction
2. Monitor delivery performance as a leading indicator of customer sentiment
3. Incorporate delivery KPIs into seller performance evaluations
4. Use customer feedback to proactively identify logistics issues
5. Align operational improvements with revenue growth goals

8. Limitations

- Dataset limited to historical data (no real-time tracking)
- Revenue approximated using available pricing fields
- External factors (marketing, promotions) not included

9. Next Steps

- Deeper seller-level performance analysis
- Region-wise logistics optimization
- Executive dashboards for ongoing monitoring