Case Study:
Data-Driven Improvement of Post-Purchase Experience

Type: Product and Data Analysis Case
Role: Product Manager
Tools: SQL, Product Thinking
Domain: E-commerce & Luxury Retail

Problem Statement: Despite strong acquisition, the platform is seeing:
- Low repeat purchase rate
- High post-purchase support tickets
- Increased refund requests in premium categories
  User feedback indicates delivery uncertainty, authenticity anxiety, and slow support response as key pain points.

Objective: Use order, tracking, and support data to identify post-purchase friction points and propose **measurable, data-backed** product improvements.

Success Metrics:
- Support tickets per order ↓
- Refund rate ↓
- Repeat purchase rate ↑
- First Response Time (FRT) ↓

Why This Problem Matters:
- Post-purchase experience directly impacts:
  - Customer Lifetime Value (CLTV)
  - Brand trust (critical for luxury)
  - Support cost per order
- Improving this stage is cheaper than acquisition

Hypotheses: Before analysis, I defined the following major hypotheses:
1. **H1:** Delivery delays increase post-purchase support tickets and hence refunds.
2. **H2:** Low order-tracking visibility increases customer anxiety and raise in “Where is my order?” tickets.
3. **H3:** Faster support first-response time improves repeat purchase rate (RPR).

Dataset and Tables Details

Dataset Source: Brazilian E-Commerce Public Dataset (Olist) (Kaggle), supplemented with synthetic support ticket data to simulate real PM analysis.

* The public Brazilian e-commerce dataset does not include customer support interactions, tracking visibility, or post-support retention signals.

To enable end-to-end product analysis, synthetic data was generated using rule-based assumptions grounded in common e-commerce behaviour. I used these simulations to demonstrate analytical methodology and decision-making.

Tables Used:

1. **orders:** The core transactional data for every order placed on the platform.

column	description
order_id	Unique order identifier
user_id	Customer identifier
order_date	Order placement date
delivery_date	Actual delivery
expected_delivery	Estimated delivery
order_value	Order amount
category	Product category

2. **Order_status_logs:** It captures the order tracking journey by logging every status update shown to the user (e.g., *Order Placed → Shipped → Out for Delivery → Delivered*).

column	description
order_id	Unique order identifier
status	Order state then
timestamp	Time when the status update occurred

3. **Support_tickets:** This includes all post-purchase customer support interactions, including the reason for contact and the speed at which the support team responds and resolves issues.

column	description
ticket_id	Unique identifier for each support ticket
order_id	Order associated with the issue
issue_type	Reason for the ticket (delivery, refund, authenticity, etc.)
created_at	Time when the ticket was raised
first_response_at	Time of first support response
resolved_at	The time when the issue was resolved

4. **users:** This table contains user-level behavioural data and helps differentiate between first-time and repeat customers, helpful for cohort analysis.

column	description
user_id	Unique identifier for each user
signup_date	Date the user joined the platform
city	User's location
repeat_user	Boolean flag indicating repeat purchase behaviour

Analysis, Code & Insights

Importing and Cleaning data: (Summary):
I cleaned and standardised the dataset to ensure reliable analysis. Key steps included handling missing delivery dates, deriving delivery delay metrics, and creating user-level aggregates to identify repeat purchase behaviour.

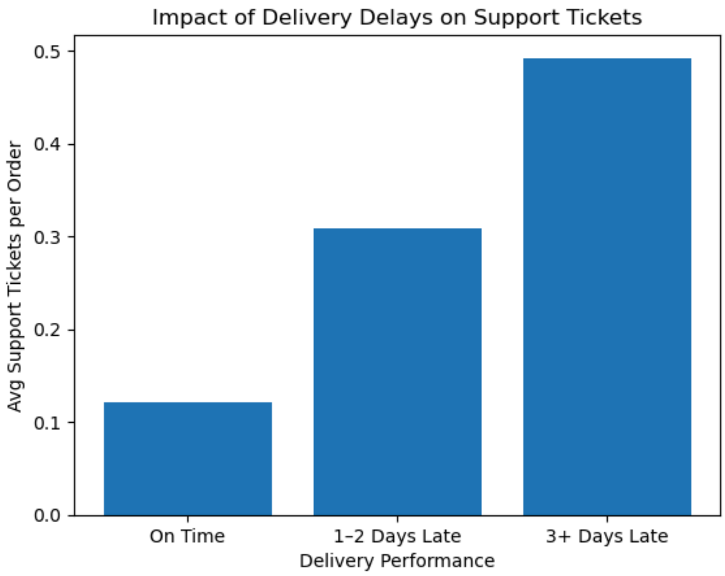
Since public datasets do not include support interactions, synthetic support data was generated using rule-based assumptions aligned with common e-commerce behaviour.

- **Analysis 1: Delivery Delay vs Support Tickets**

  Objective: Understand how delivery delays affect support load.

```
orders["delay_bucket"] = pd.cut(
    orders["delivery_delay_days"],
    bins=[-100, 0, 2, 100],
    labels=["On Time", "1-2 Days Late", "3+ Days Late"]
)

tickets_per_order = (
    orders.groupby("delay_bucket")["has_ticket"].mean()
    .reset_index(name="avg_tickets_per_order")
)

plt.figure()
plt.bar(
    tickets_per_order["delay_bucket"],
    tickets_per_order["avg_tickets_per_order"]
)
plt.xlabel("Delivery Performance")
plt.ylabel("Avg Support Tickets per Order")
plt.title("Impact of Delivery Delays on Support Tickets")
plt.show()
```

Output visualisation:



Insight: Orders delivered 3+ days late generate significantly more support tickets per order than 1-2 days late deliveries.

My takeaway: Delivery performance is the **primary driver** of post-purchase friction and ticket raises, resulting in operational costs.

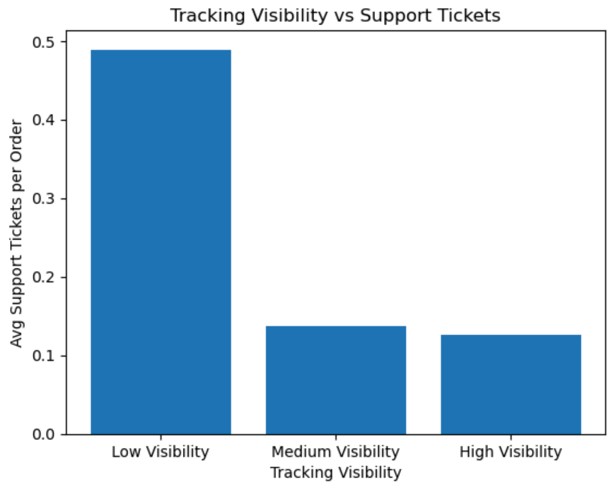
- **Analysis 2: Order Tracking Visibility vs Ticket Creation**

  Objective: Measure the impact of tracking transparency on customer anxiety.

```
orders["status_updates"] = np.where(
    orders["delivery_delay_days"] > 2,
    np.random.randint(2, 4, size=len(orders)),
    np.random.randint(4, 7, size=len(orders))
)

orders["tracking_visibility"] = pd.cut(
    orders["status_updates"],
    bins=[0, 2, 5, 10],
    labels=["Low Visibility", "Medium Visibility", "High Visibility"]
)
tracking_analysis = (
    support_tickets.merge(
        orders[["order_id", "tracking_visibility"]],
        on="order_id",
        how="left"
    )
    .groupby("tracking_visibility")
    .size()
    / orders.groupby("tracking_visibility").size()
).reset_index(name="avg_tickets_per_order")

plt.figure()
plt.bar(
    tracking_analysis["tracking_visibility"],
    tracking_analysis["avg_tickets_per_order"]
```

```
)
plt.xlabel("Tracking Visibility")
plt.ylabel("Avg Support Tickets per Order")
plt.title("Tracking Visibility vs Support Tickets")
plt.show()
```

Output visualisation:



Insight: Orders with low tracking visibility create substantially more “Where is my order?” tickets, even when deliveries are on time.

My takeaway: Reducing uncertainty and increasing transparency is as important as improving logistics speed.

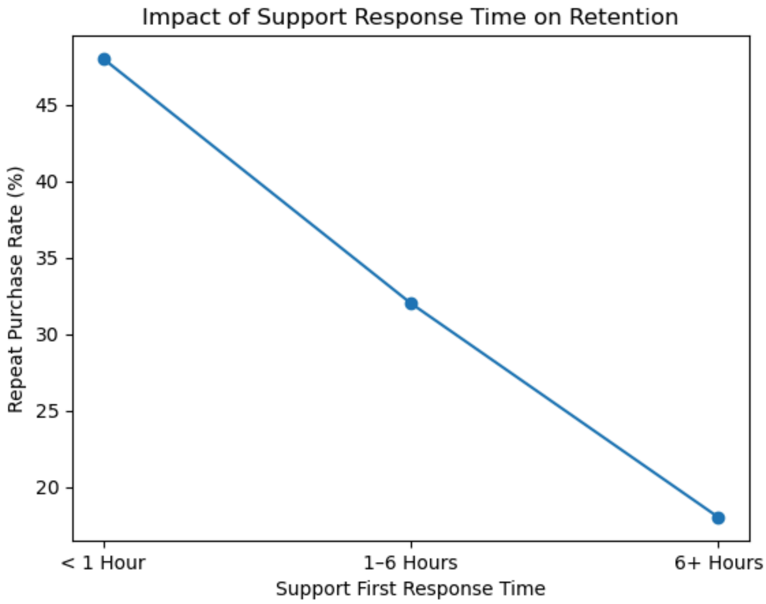
- **Analysis 3:** Support First Response Time (FRT) vs Repeat Purchase

Objective: Understand whether support speed impacts retention.

```
frt_analysis = support_tickets.merge(
    users[["customer_id", "repeat_user"]],
    on="customer_id",
    how="left"
)

frt_analysis["frt_bucket"] = pd.cut(
    frt_analysis["frt_hours"],
    bins=[0, 1, 6, 24],
    labels=["< 1 Hour", "1-6 Hours", "6+ Hours"]
)

repeat_rate = frt_analysis.groupby("frt_bucket")["repeat_user"].mean().reset_index()

plt.figure()
plt.plot(
    repeat_rate["frt_bucket"],
    repeat_rate["repeat_user"] * 100,
    marker="o"
)
plt.xlabel("Support First Response Time")
```

```
plt.ylabel("Repeat Purchase Rate (%)")
plt.title("Impact of Support Response Time on Retention")
plt.show()
```

Output visualisation:



Insight: Users receiving support responses within 1 hour build trust, resulting in a noticeably higher repeat purchase rate (RPR) than with slower responses.

My Takeaway: Quick support helps in increasing retention more than anything.

Product Recommendations & Next Steps

Product Recommendations:

- **Proactive Delay Communication:** Auto-trigger notifications for orders delayed for more than 48 hours
  Expected impact: Less customer anxiety, hence fewer tickets per order
- **Enhanced Tracking Transparency:** Milestone-based updates instead of time-based
  Expected impact: Decrease in the number of "Where is my order?" tickets
- **SLA-Based Support Routing:** Premium orders get <1 hour FRT
  Expected impact: Builds trust, resulting in a higher repeat purchase rate

Overall Impact Estimation: If implemented, these changes are expected to reduce support tickets per order by **20-30%** and improve repeat purchase rate in premium categories by **8-12%**, based on observed directional trends.

What I Would Validate with Real Production Data:

- Correlation vs causation between FRT and retention
- Long-term cohort retention (60/90 days)
- Refund rate impacted by the delivery delay bucket
- Category-level sensitivity (luxury vs non-luxury)

Conclusion: Post-purchase friction follows a clear flow:

Delivery Delays → Customer Anxiety → Support Tickets → Support Experience → Retention.

This analysis shows that uncertainty increases support load, while faster response times for raised tickets are directionally linked to higher repeat purchase rates, highlighting **post-purchase experience as a measurable retention and growth lever**, not just a support cost.