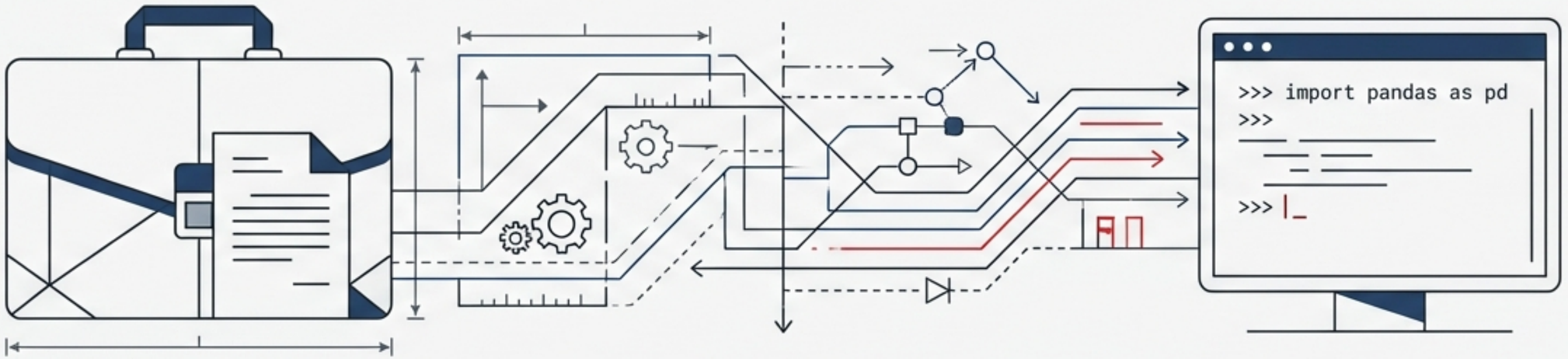
The Art of Problem Framing in Data Science

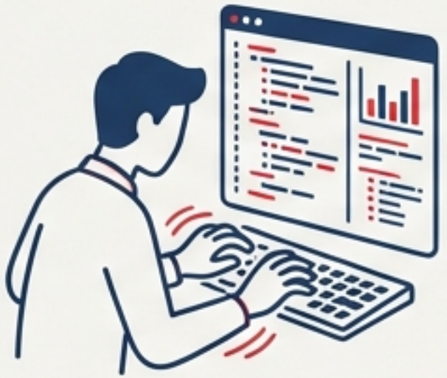


Moving beyond syntax: How to translate ambiguous business requirements into precise Machine Learning roadmaps

Based on the frameworks by CampusX

The deciding factor between a Junior Analyst and a Data Science Lead

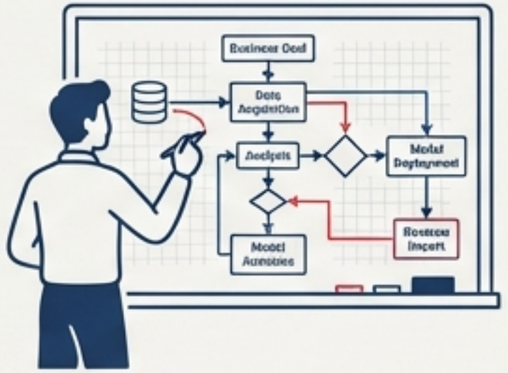
The Task-Doer (Junior)



- Executes black-and-white instructions.
- Focuses on hyperparameter tuning and cleaning provided datasets.
- Measures success by model accuracy on a static test set.

"Here is the dataset; improve the model."

The Problem-Solver (Leader)

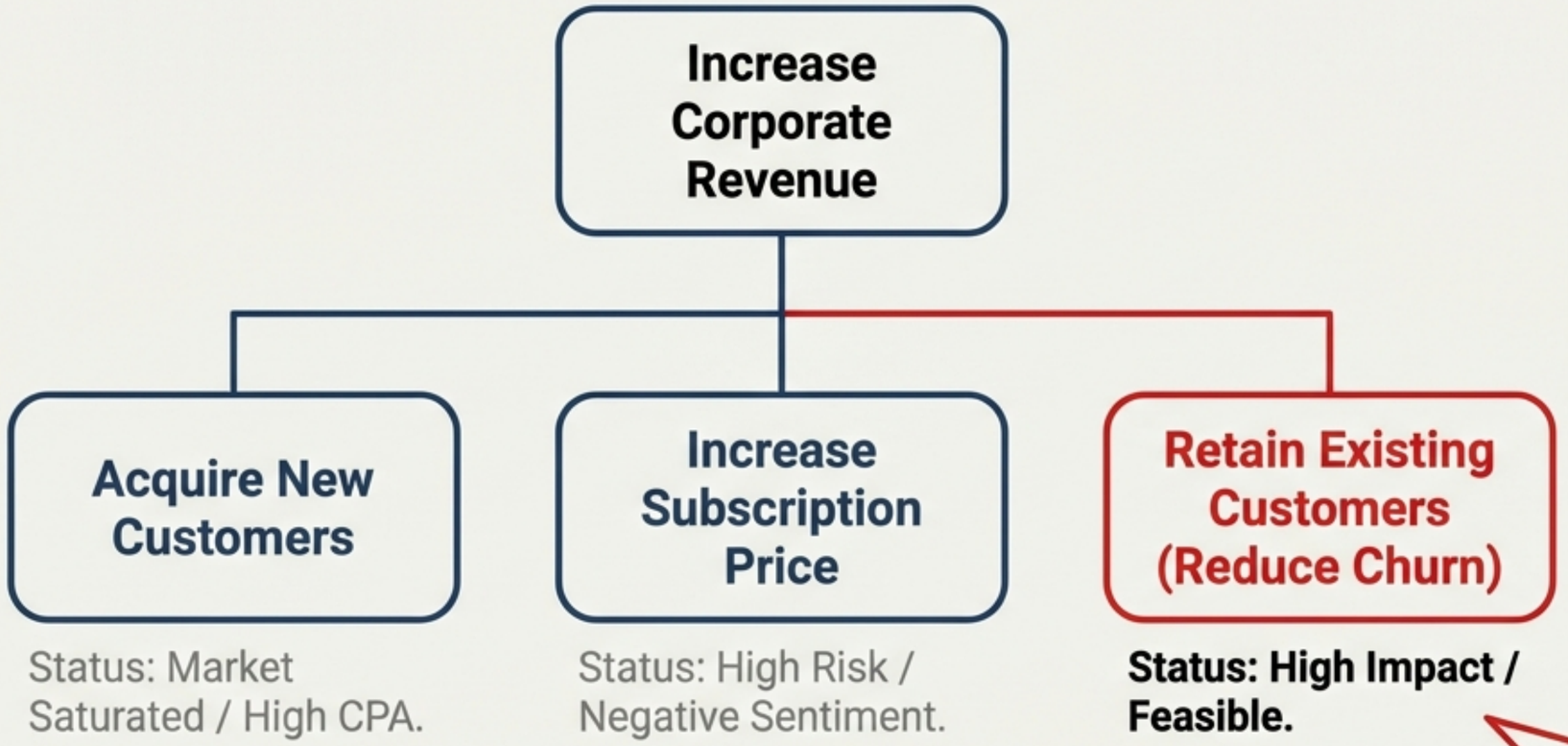


- Navigates the grey areas of business ambiguity.
- Focuses on defining the problem scope and feasibility.
- Measures success by revenue impact and business value.

"We have a revenue problem. How do we solve it using data?"

Key Insight:
Your progression to the C-Suite isn't defined by your coding speed, but by your ability to frame the problem before writing a single line of code.

The Boardroom Scenario: Netflix needs to increase revenue



The Strategic Pivot:

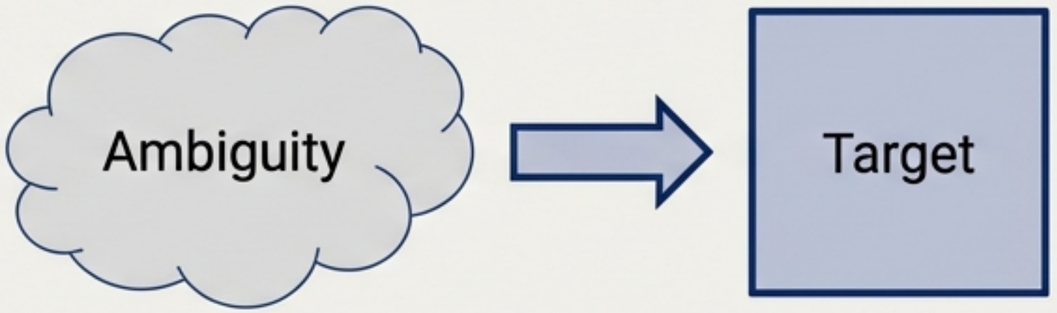
As Lead Data Scientist, the CEO asks you, “How can we stop people from leaving?”

You must translate this qualitative business desire into a quantitative technical roadmap.

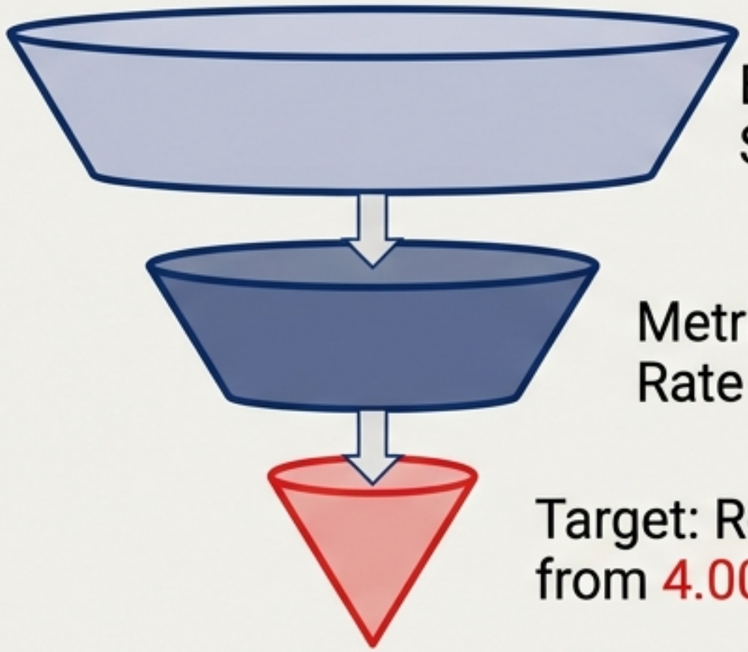
Step 1: Converting Business Intent to Mathematical Targets

General Principle

A Data Scientist's first instinct must be to convert the 'Business Problem' into a solvable 'Mathematical Problem'. Avoid vague goals.



Case Study Application



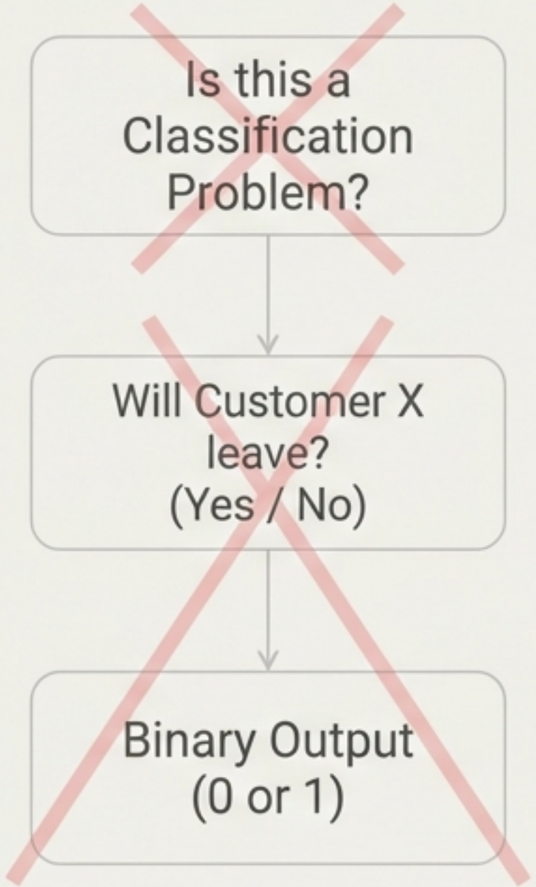
Business Goal: Stop People Leaving

Metric: Monthly Churn Rate

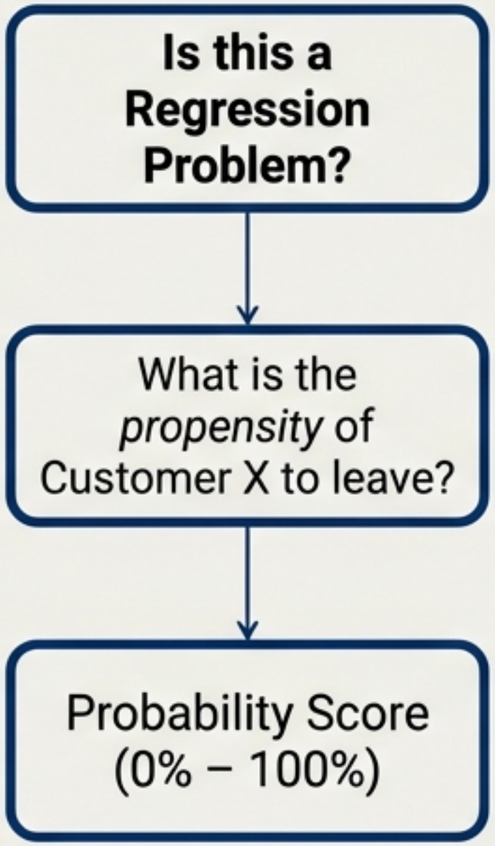
Target: Reduce Churn from 4.00% to 3.75%

I cannot just 'fix churn'. I need a specific number to aim for so the team doesn't burn out chasing an impossible zero. A 0.25% reduction across millions of users is a massive revenue retention win.

Step 2: Identifying the Problem Type



Inefficient. Treats a slightly annoyed user the same as a furious one. Wastes budget on flat discounts.



Allows Tiered Intervention.

Tiered Intervention Strategy

High Risk (90% Score)	→	Aggressive Discount (50% off)
Medium Risk (40% Score)	→	Soft Nudge / Content Recommendation
Low Risk (10% Score)	→	Do Nothing

The **nuances of the business tactic** (targeted discounts) **dictate the technical approach** (regression over classification).

Step 3: The Literature Review

Efficiency over Ego: Don’t reinvent the wheel.

- Has the engineering team built a "v1.0" churn model previously?
- What variables did they use?
- What was the error margin?

- How do competitors handle this?
- Read white papers and tech blogs to establish a baseline.

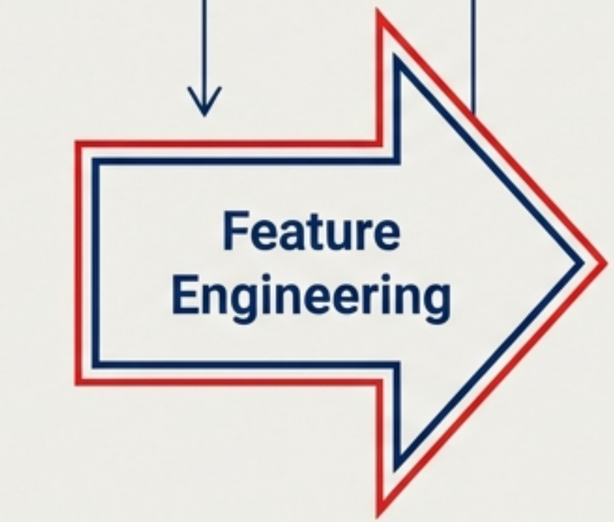
- Speak to Senior Architects.
- Identify existing rules-based systems.

Netflix Context: We discovered an existing rules-based system that predicts churn within a 5% margin of error. We will use their feature list as a baseline feature set rather than starting with a blank canvas.

Step 4: Data Strategy & Feature Engineering

Abstract Behaviours (Signals of Dissatisfaction)

- User stops watching halfway through.
- User browses for 20 mins but plays nothing.
- User searches for a title and gets zero results.
- Usage drops significantly compared to last month.



Concrete Data Features (Schema)

[Column Name]	[Data Type]
abandonment_rate	Float
browsing_vs_watch_ratio	Float
search_failures_count	Integer
watch_time_delta_30d	Float
account_age_months	Integer

The goal is to translate intuitive ‘feelings’ of user unhappiness into hard numerical columns for the model to train on.

The Collaboration: Scientist vs. Engineer

The Data Scientist (Architect)

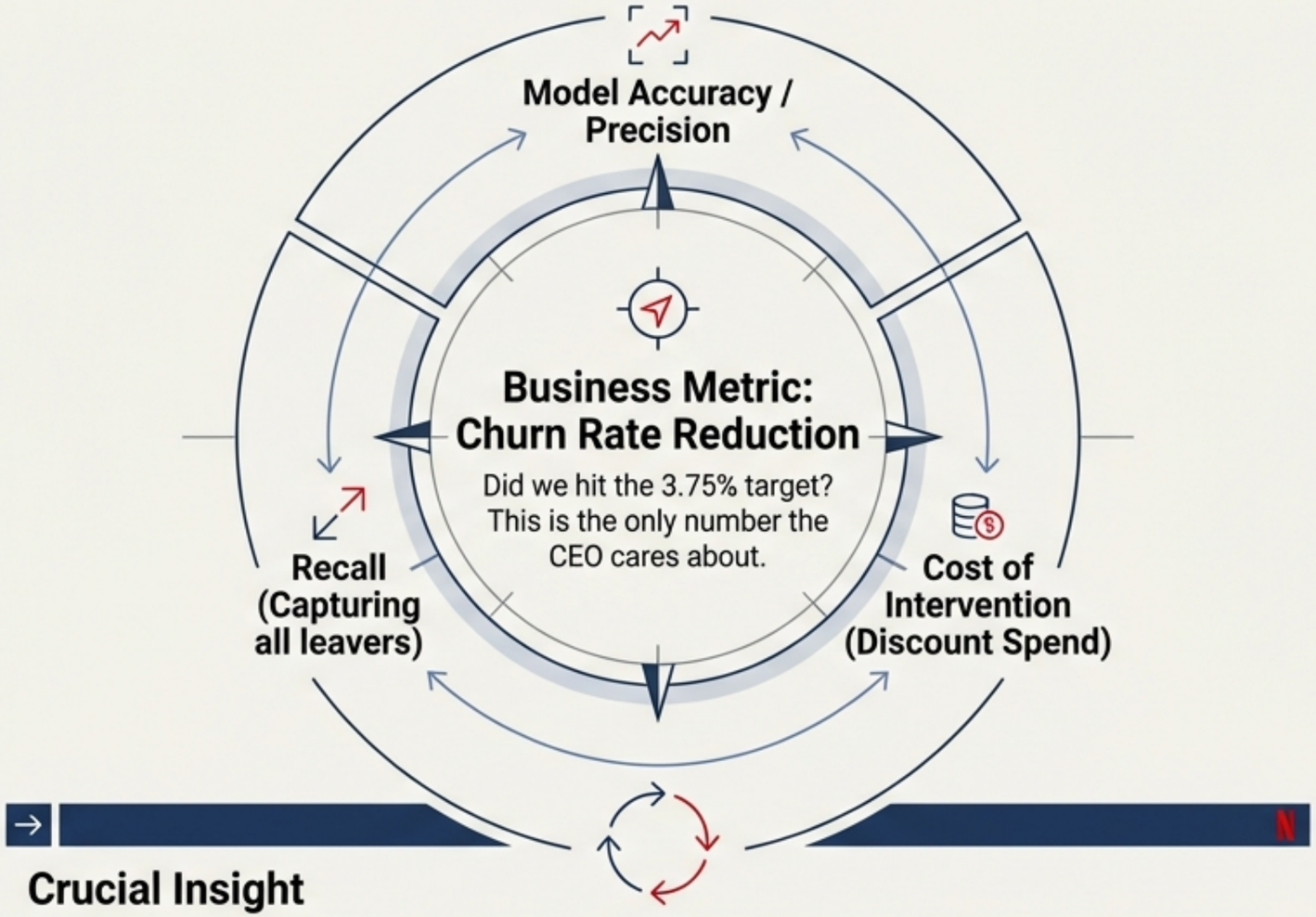
- Defines *WHAT* is needed.
- **Responsibilities:** Feature definition, Logic creation, Metric selection.
- **The Ask:** "I need a table with search failures aggregated by user ID."

Collaboration Point:
You provide the schema requirements; they build the pipeline.

The Data Engineer (Builder)

- Defines *HOW* to retrieve it.
- **Responsibilities:** Pipeline construction, SQL optimization, Data Warehouse maintenance.
- **The Deliverable:** A robust, scalable View in the Data Warehouse (not a one-off CSV).

Step 5: Metric Selection & The ‘North Star’



Crucial Insight

It is not enough to predict who leaves. We must measure if our intervention (the discount) actually changed their behaviour. We are optimizing for ***Retention***, not just ***Prediction***.

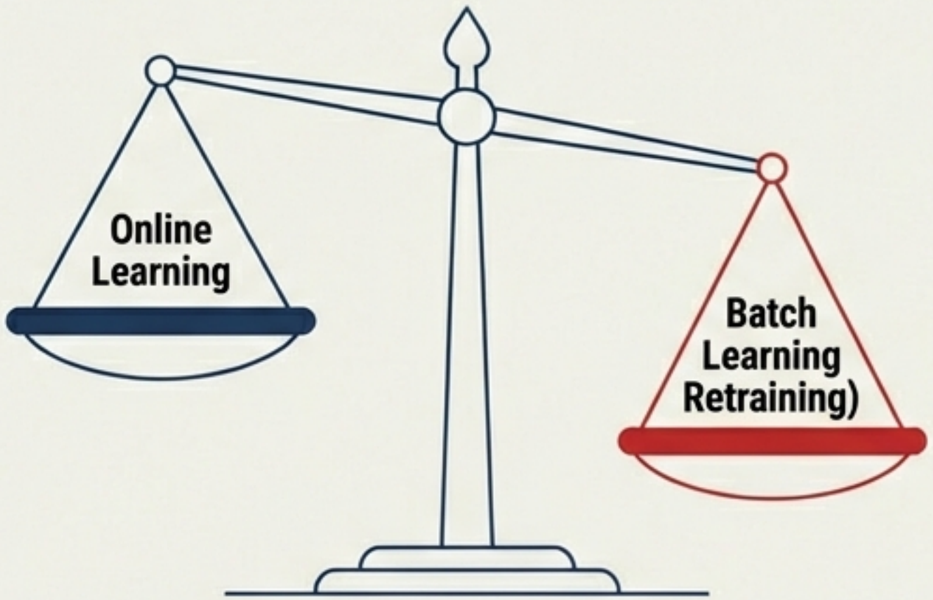
Step 6: Training & Deployment Strategy

Handling Volatility in Consumer Behaviour

Online Learning

Model updates continuously with every new data point.

- **Pros:** Adapts instantly to trends.
- **Cons:** Complex infrastructure, risk of instability.



Batch Learning (Weekly Retraining)

Model is retrained offline on a schedule (e.g., every Sunday).

- **Pros:** Stable, allows for human review, simpler infrastructure.
- **Cons:** Lag in adaptation.

Verdict

Decision: Netflix user behaviour is volatile (e.g., holidays, lockdowns, weekends). A static model will rot, but fully online learning is risky.

Compromise: Frequent Batch Retraining. Train offline every week to capture recent trends (like a new hit show) without the infrastructure overhead of online learning.

Step 7: Sanity Checks & Assumptions

⚠ Geographic Bias

We trained on US data. Does this apply to users in India? (Likely not—different pricing sensitivity and content preferences).

⚠ Data Availability

We assumed we have data on 'Search Failures'. Does the logging system actually capture zero-result searches, or does it only log successful clicks?

⚠ Correlation vs. Causation

Just because watch time drops, doesn't mean they hate the service. They might just be busy (e.g., exam season). Context matters.

→

Verify these assumptions with Product and Engineering teams **before** full-scale training.

The 7-Step Problem Framing Framework

1. **1. Business to Maths** — Convert ambiguous goals (Reduce Churn) to numeric targets (3.75%).
2. **2. Problem Type** — Pivot strategy based on intervention (Regression for Propensity Scoring).
3. **3. Literature Review** — Audit existing internal models and industry standards.
4. **4. Data Strategy** — Translate behaviours into specific feature columns.
5. **5. Metric Selection** — Define the North Star (Business Impact) vs. Proxy Metrics (Accuracy).
6. **6.Deployment** — Analyze volatility to choose between Online vs. Batch learning.
7. **7.Sanity Checks** — Rigorous review of biases, data availability, and causal assumptions.

The Path to Leadership

Thousands of people enter the industry knowing how to write code. Only a fraction advance to management.

The differentiator is the ability to sit in a room, listen to a vague business struggle, and architect the entire solution before touching the keyboard.

Don’t just learn to code the model. Learn to frame the world.