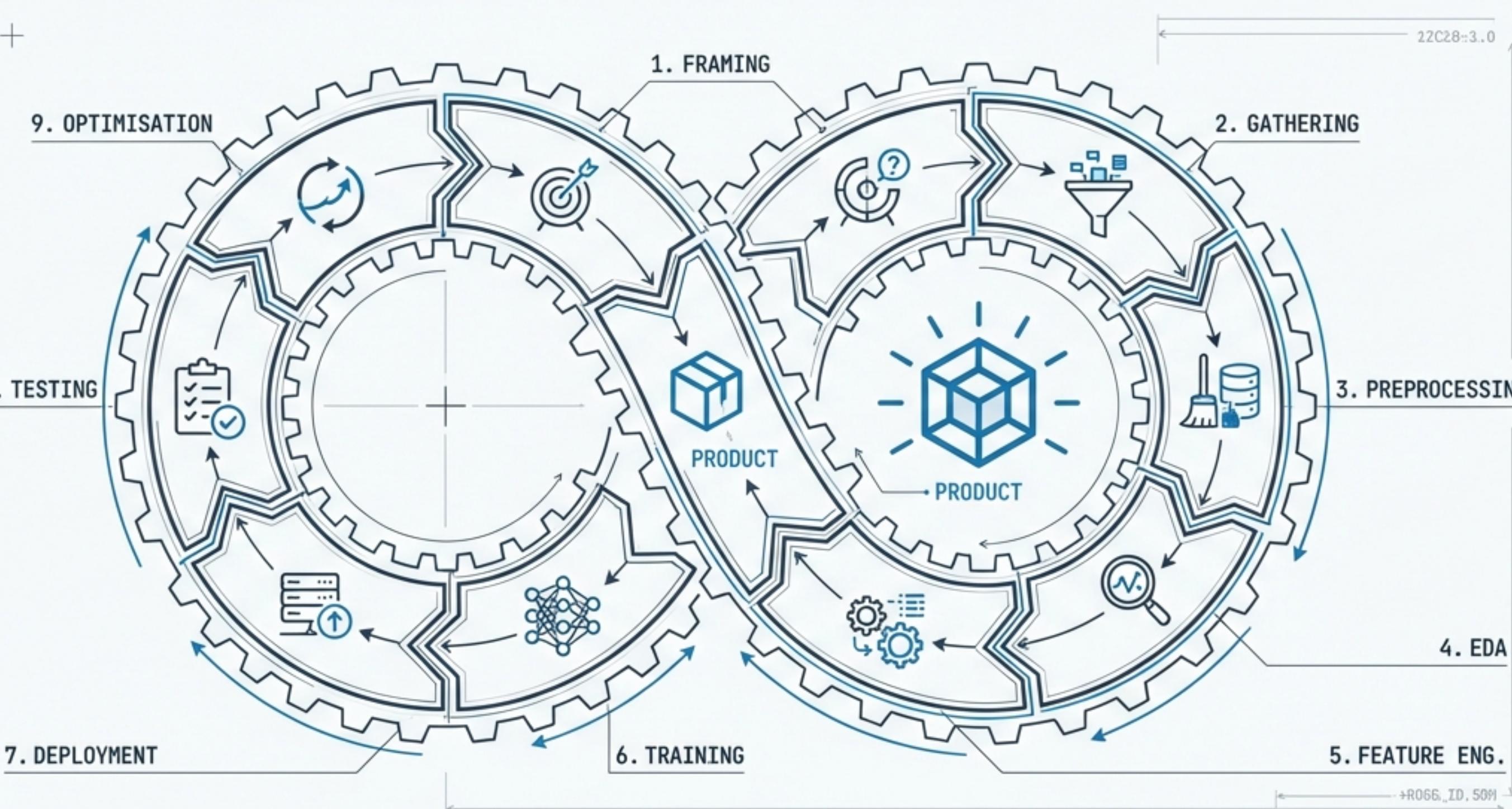


# The Machine Learning Development Life Cycle (MLDLC)

A 9-Step Roadmap from Raw Idea to Deployed Product

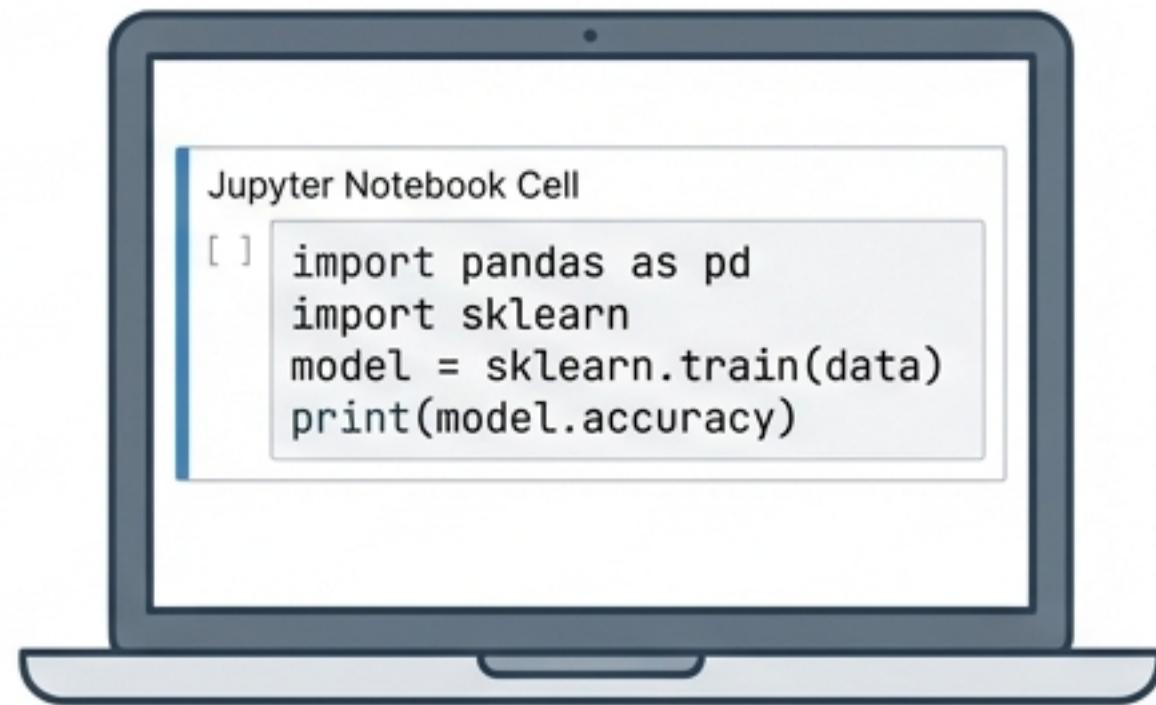


This deck outlines the transition from simple model-building to full-stack product development. Based on industry standards derived from the Software Development Life Cycle (SDLC), this roadmap details the exact process required to build end-to-end ML systems that solve real-world problems.

**NOTE: CONTINUOUS ITERATION REQUIRED FOR OPTIMAL PERFORMANCE**

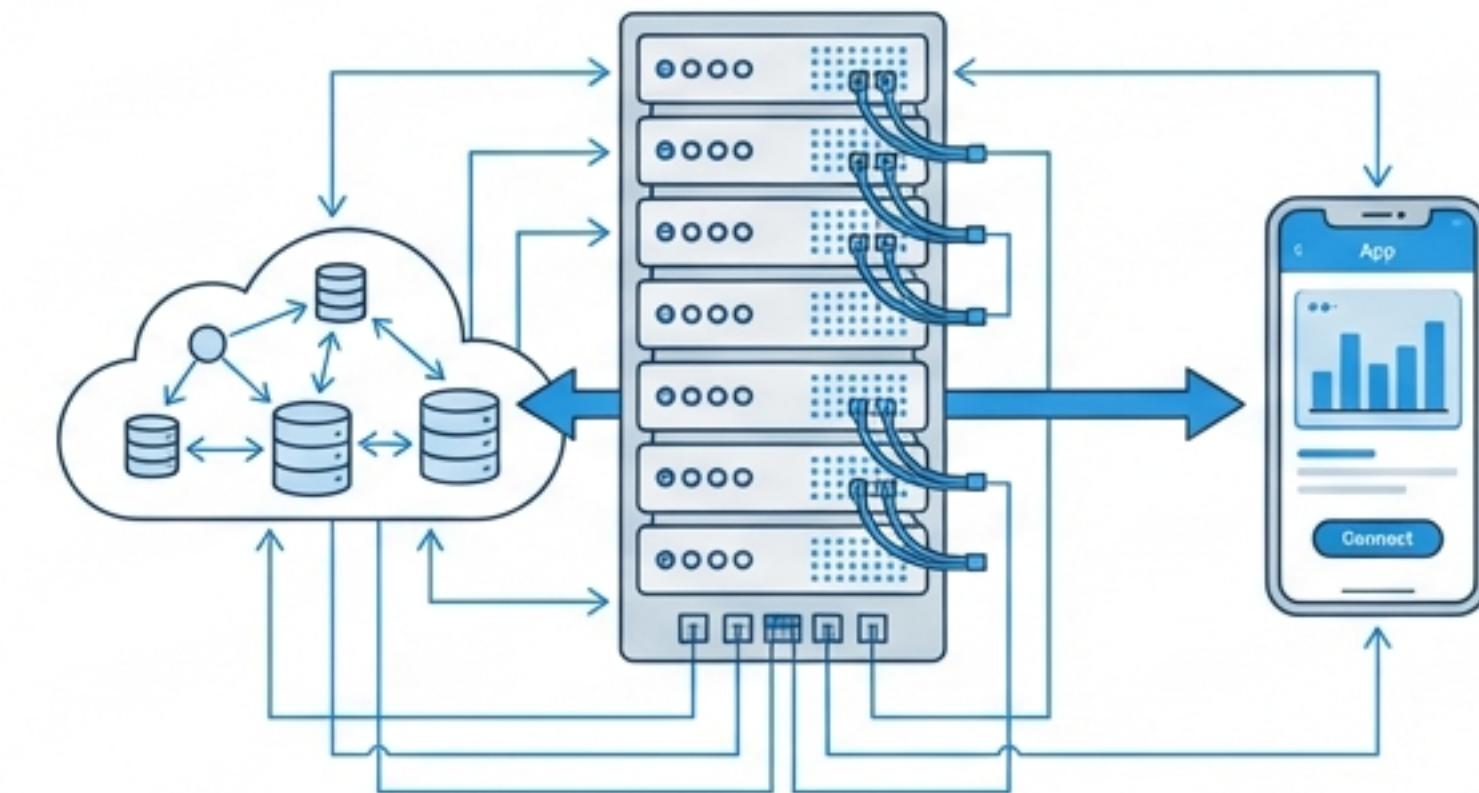
# THE INDUSTRY DEMANDS PRODUCTS, NOT JUST MODELS

## THE STUDENT MINDSET



- **Focus:** The 'What' (Algorithm)
- **Goal:** 99% Accuracy
- **Output:** A Static .ipynb File
- **Metric:** Kaggle Score

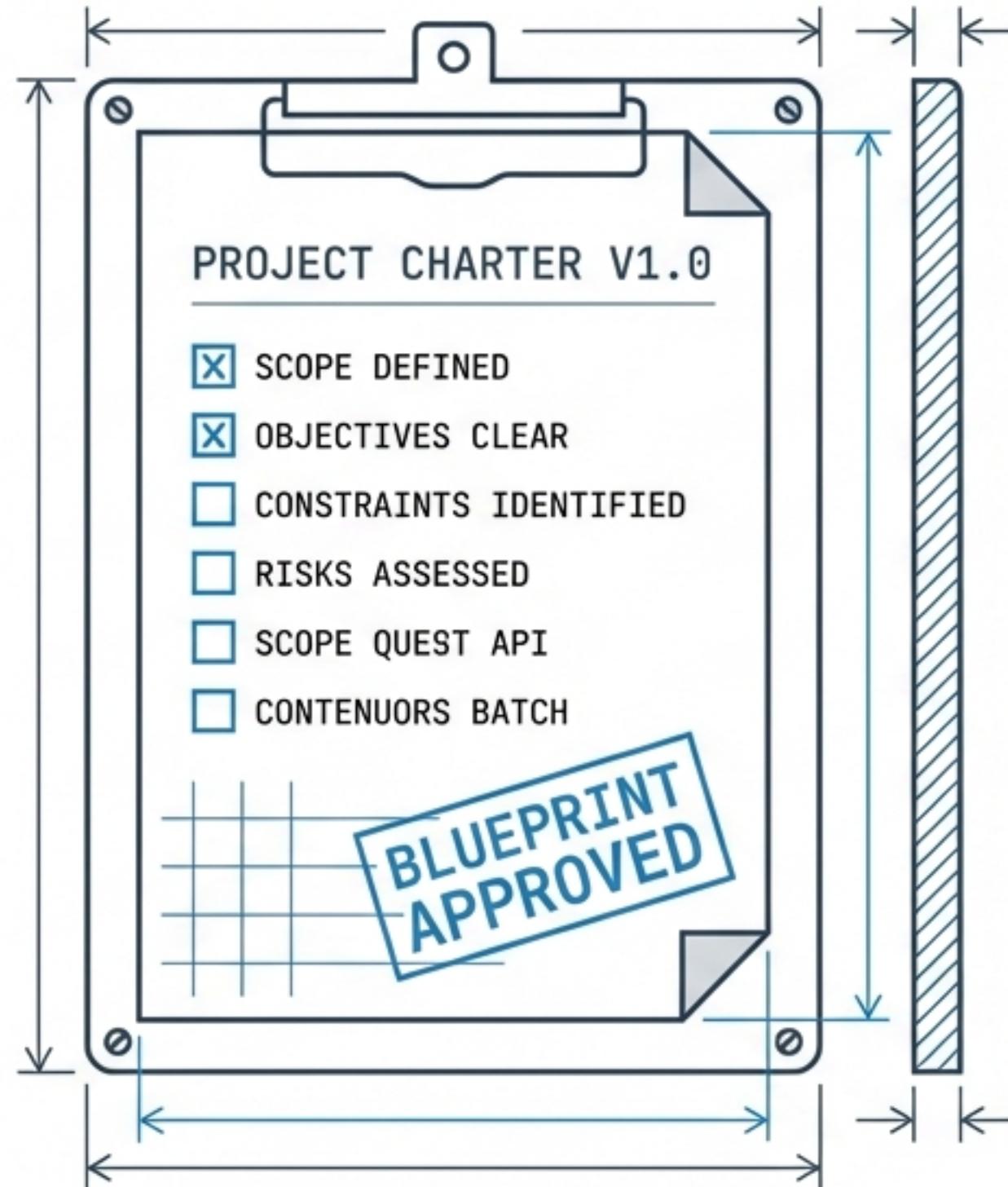
## THE ENGINEER MINDSET



- **Focus:** The 'How' (System)
- **Goal:** Utility & Scalability
- **Output:** Deployed API Endpoint
- **Metric:** User Satisfaction / Latency

Key Insight: In interviews, you are not judged solely on your ability to import a library. You are judged on your ability to build a sustainable, end-to-end software product.

# Step 1: Framing the Problem



You cannot build a house without blueprints. Before execution, scope must be defined to prevent wasted resources.



## Who is the client?

Company, direct customers, or internal team?



## What is the objective?

What exact problem are we solving?



## Commercial Constraints

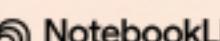
Budget, team size, and timeline.



## Technical Scope

Supervised vs. Unsupervised?  
Batch vs. API?



**CRITICAL: SCOPE CREEP ALERT**  
Undefined scope leads to resource drain. Lock it down early. 

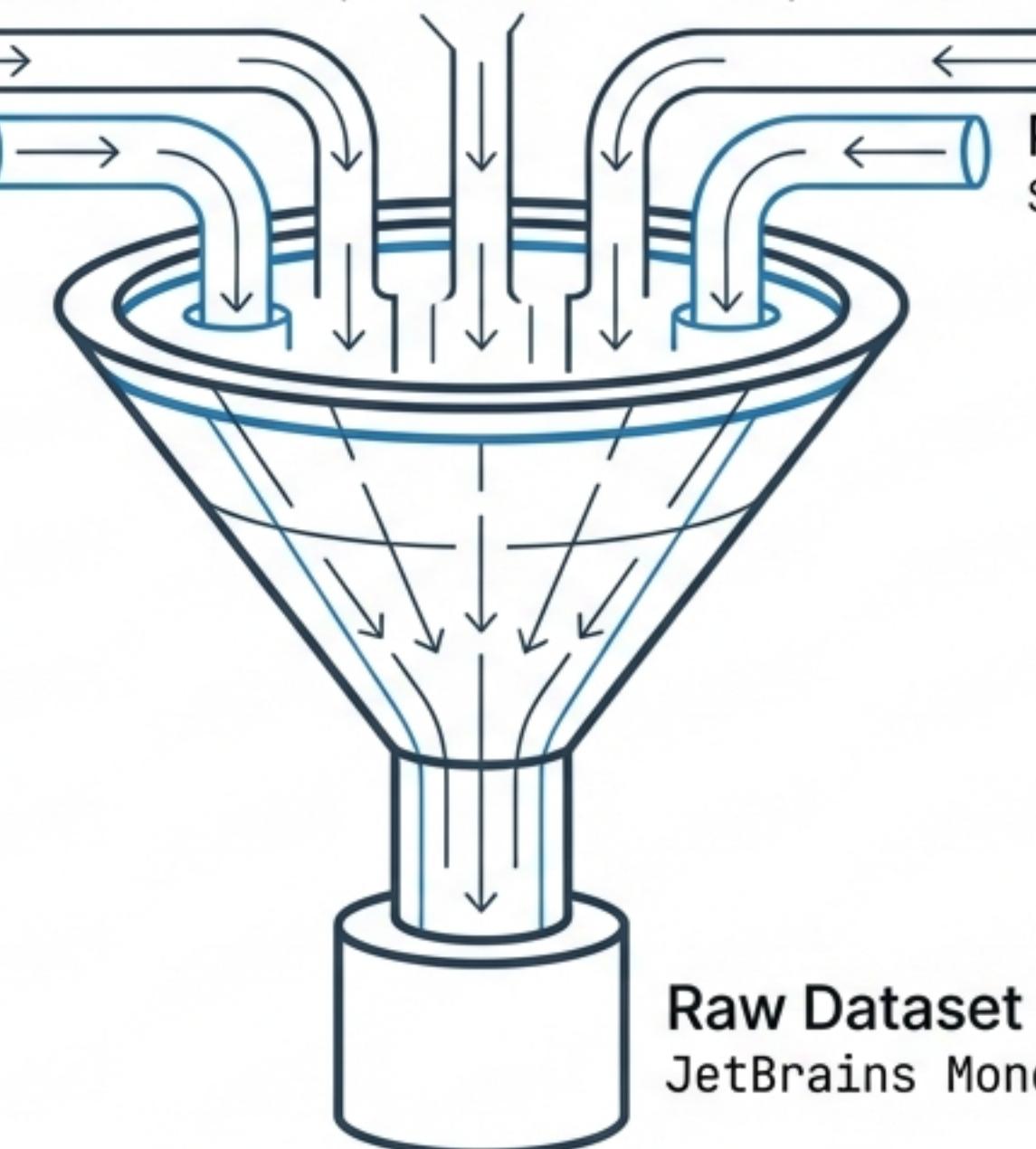
# Step 2: Gathering the Data

STEP  
Inter  
JetBrains Mono



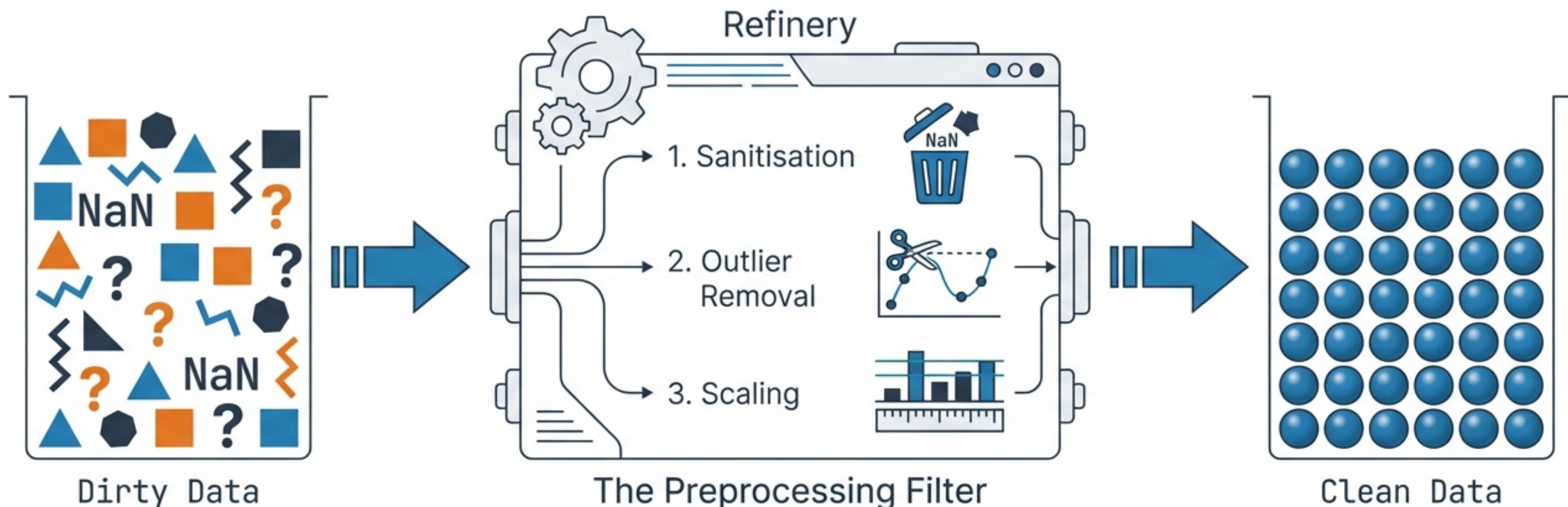
Pipe 1: Static Files    Pipe 2: APIs    Pipe 3: Web Scraping    Pipe 4: Data Warehouses  
CSVs, Excel    JSON, REST Endpoints    BeautifulSoup, HTML Parsing    SQL, ETL Pipelines

Pipe 1: Static Files    Pipe 5: Big Data Clusters  
STM/iVs, Excel    Spark, Distributed Storage



Goal: Secure the raw fuel required  
required for the factory. data, ML  
Without data, ML is impossible. In  
rarely handed to you; it must be  
you; it must be extracted.

# Step 3: Data Preprocessing



The goal is to convert disparate data types into a format compatible with the mathematical assumptions of the algorithm.

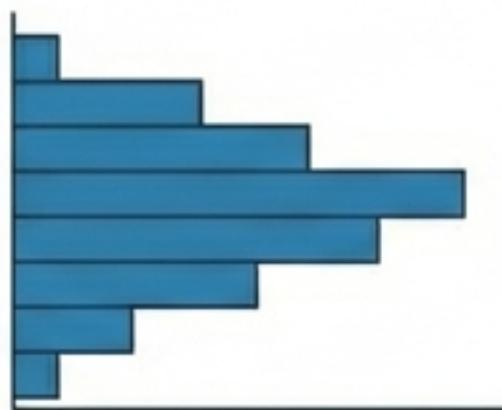
# Step 4: Exploratory Data Analysis (EDA)



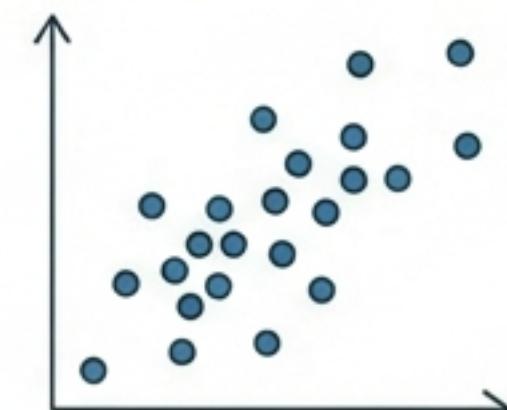
If I had six hours to chop down a tree, I'd spend the first four sharpening the axe.

## The Toolset

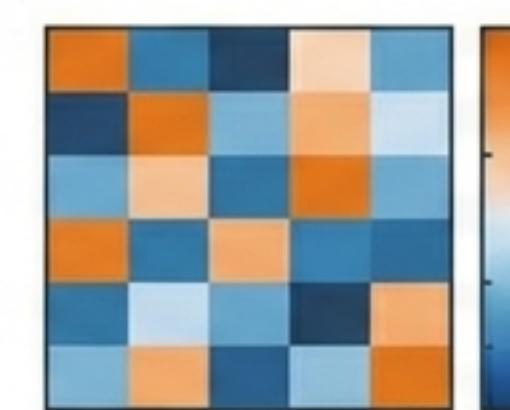
### ① Visualisation Strategy



Univariate  
A single histogram



Bivariate  
A scatter plot



Multivariate  
A heatmap

### ② Handling Imbalance



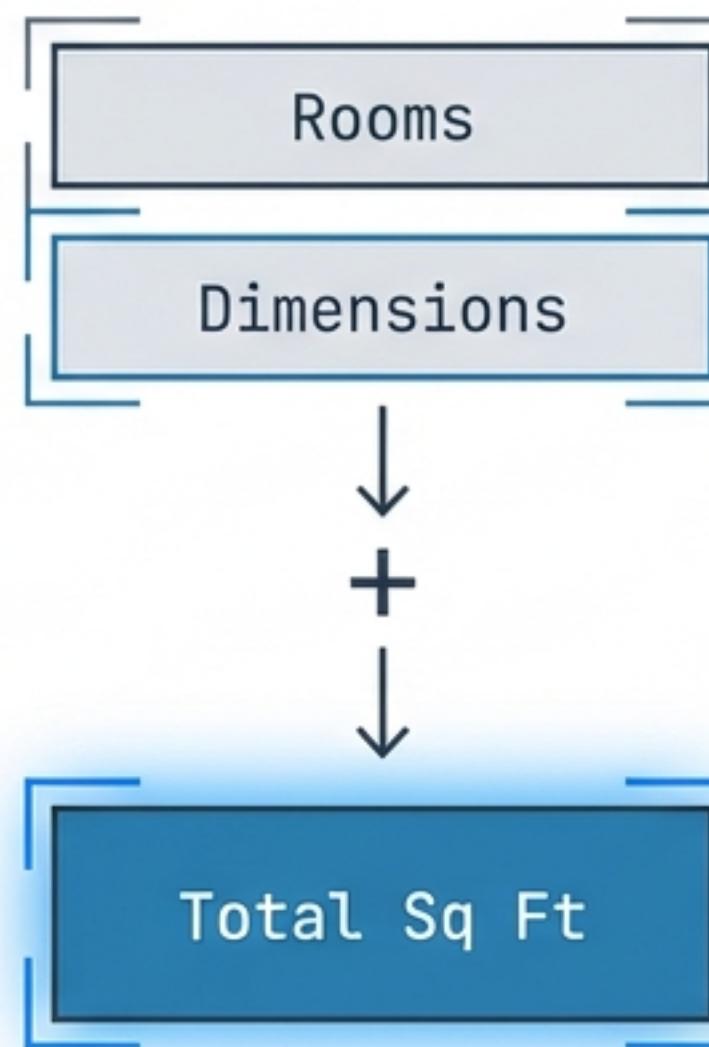
Upsampling

EDA is about gaining intuition. Understanding the relationship between input and output simplifies decision-making later.

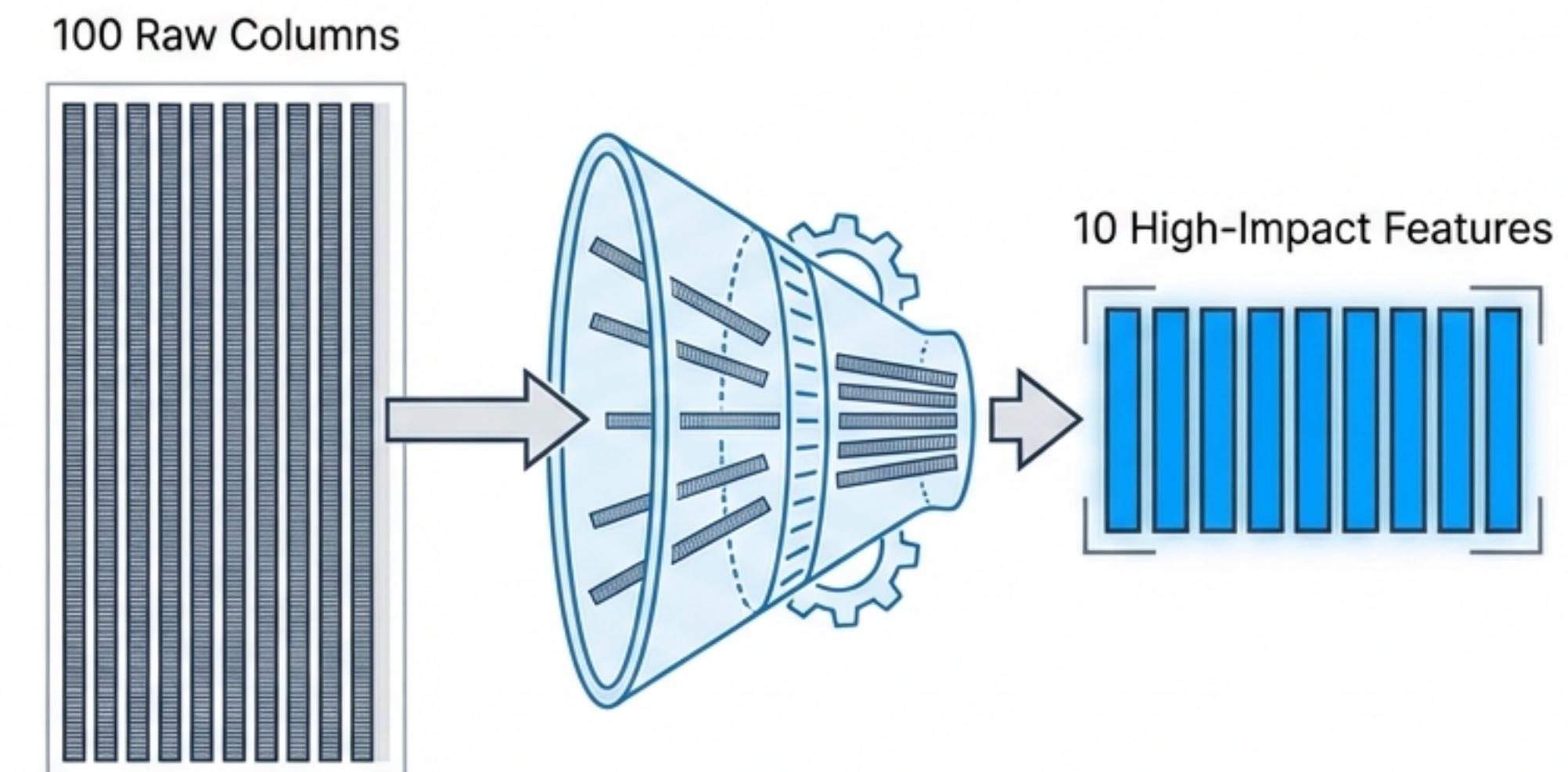
# Step 5: Feature Engineering & Selection



## Engineering (Creation)



## Selection (Reduction)

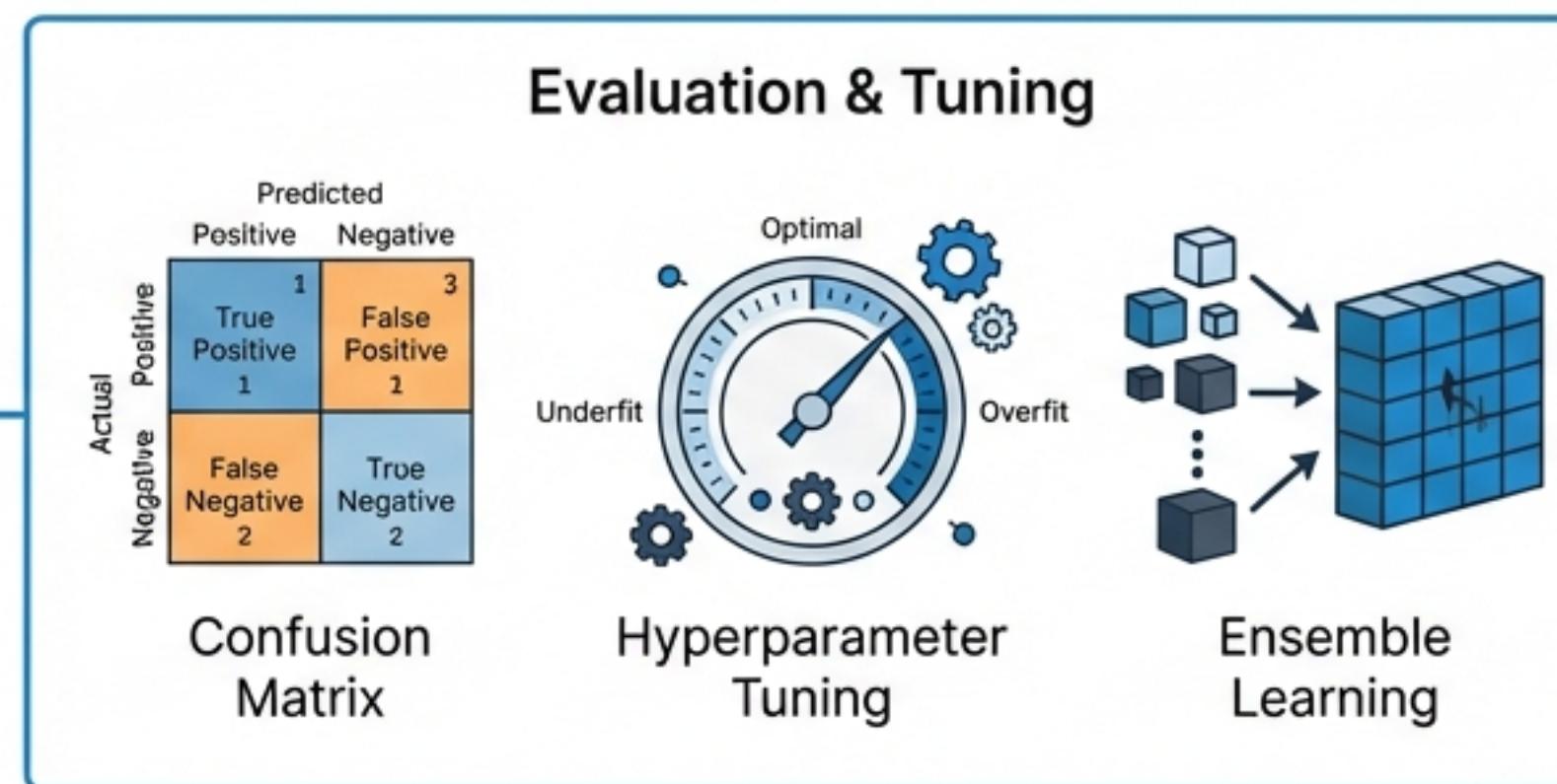
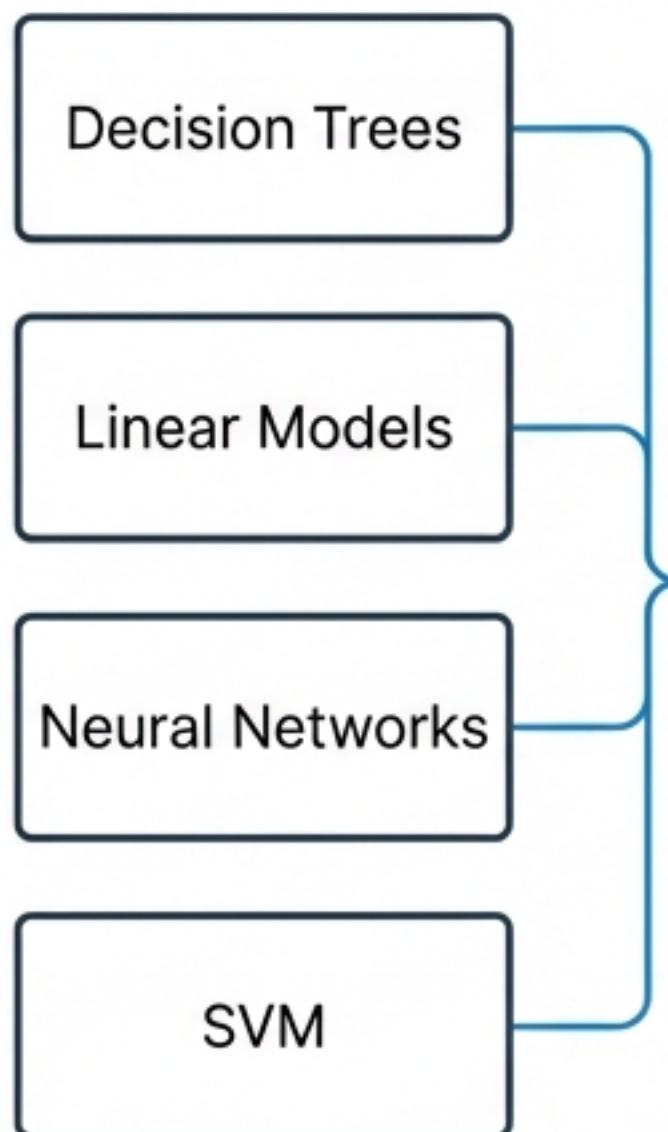


Creating new intelligence  
from existing signals.

⚠️ **Avoiding the Curse of Dimensionality.** Removing  
noise to improve speed and generalisation.

# Step 6: Model Training & Evaluation

## Contenders



Helvetica Now Display

## Champion Model

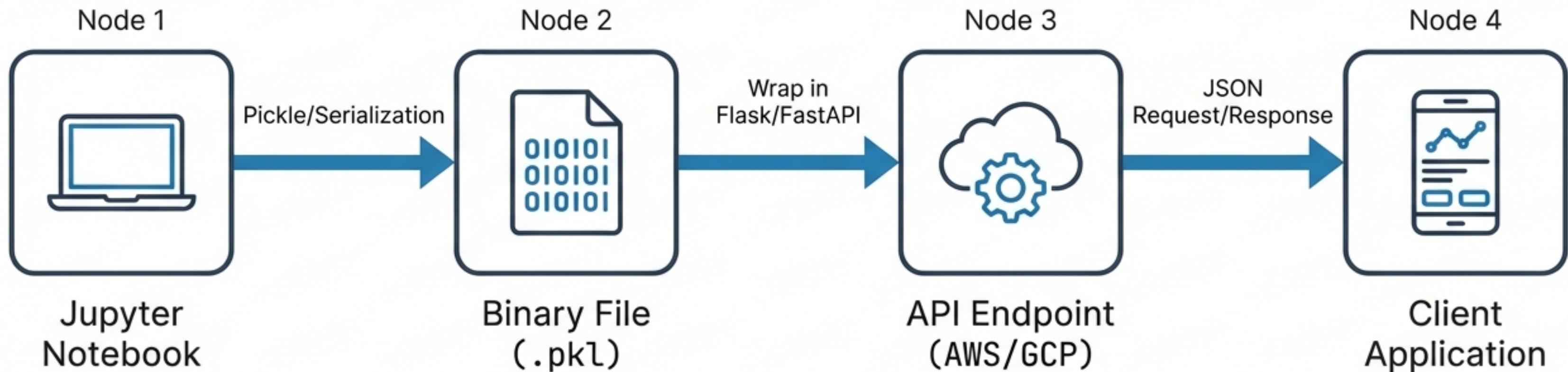
Optimized for accuracy and generalization.

The Goal: Finding the mathematically optimal solution, not just the first one that works.

# Step 7: Deployment



STEP 07/09



Jupyter  
Notebook

Binary File  
.pkl

API Endpoint  
(AWS/GCP)

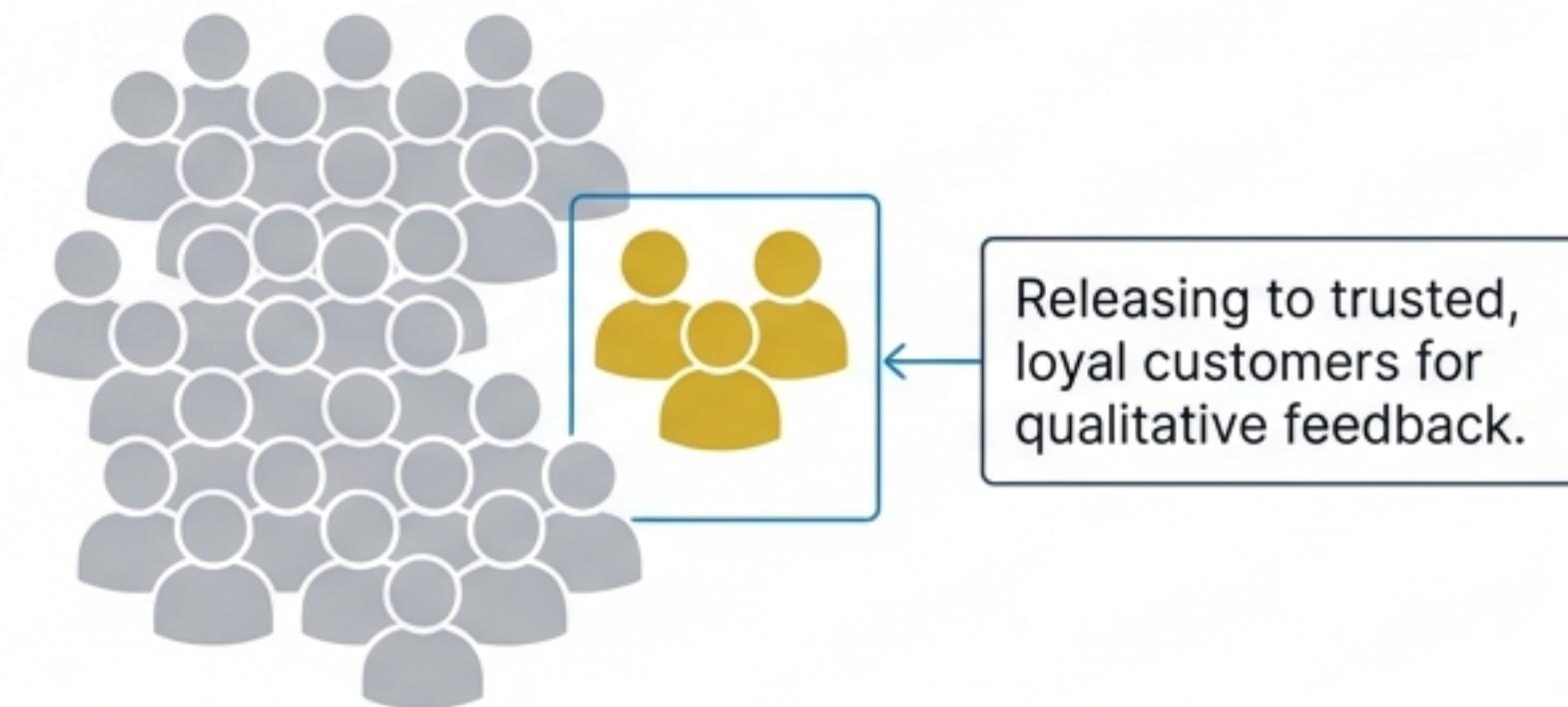
Client  
Application

The Bridge: Moving from a local experiment to a real-world server accessible 24/7.  
Inter: JetBrains Mono

# Step 8: Testing the Solution

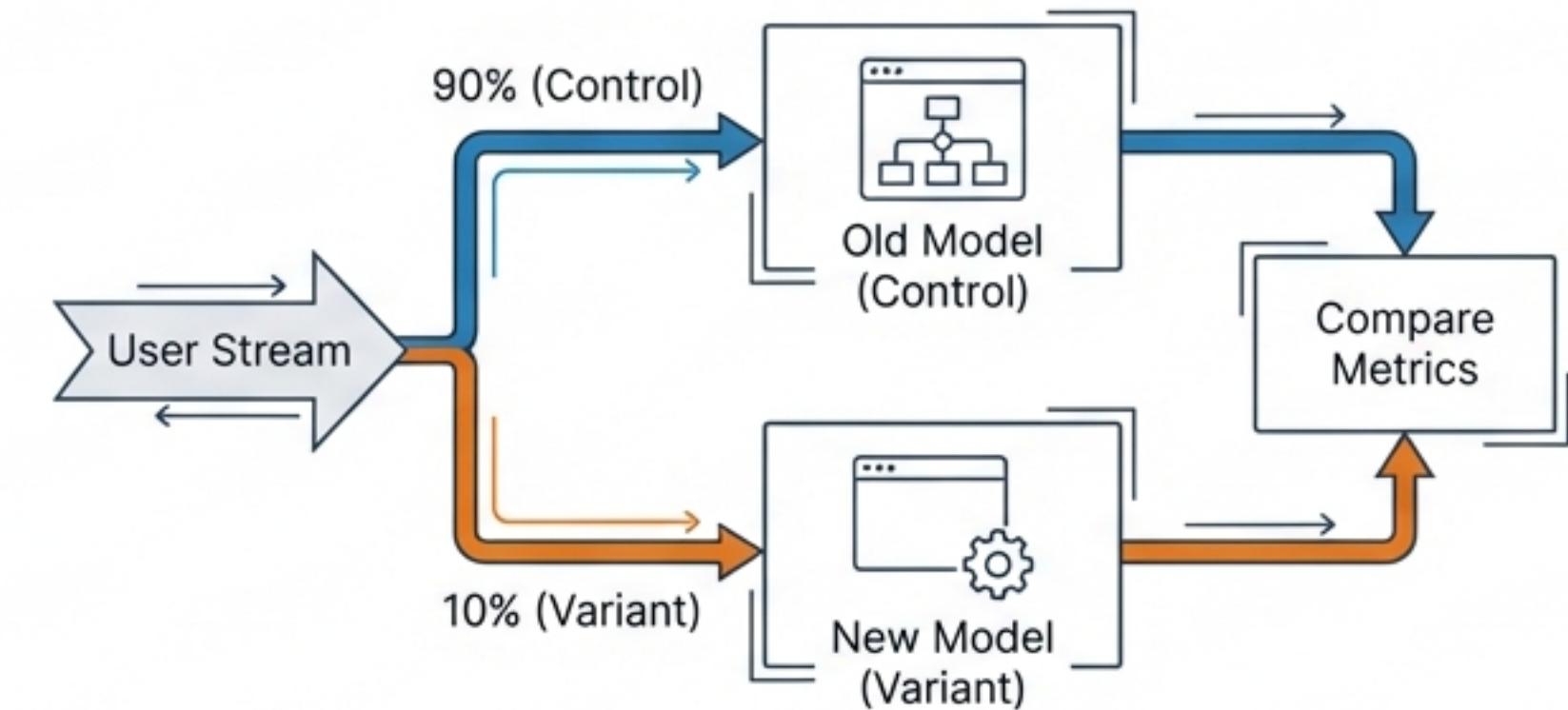


## Beta Testing

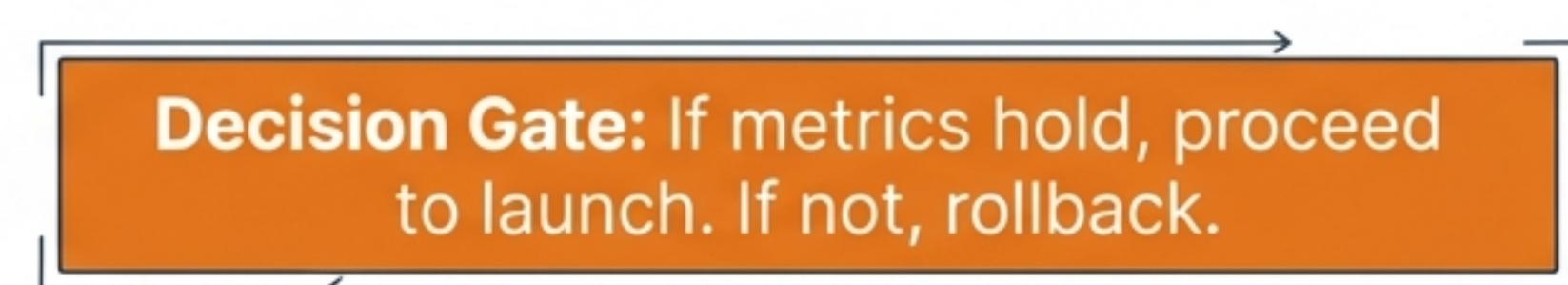


Releasing to trusted, loyal customers for qualitative feedback.

## A/B Testing



Rolling out to live traffic to compare quantitative performance metrics.



# Step 9: Optimisation & Maintenance



## Dashboard Monitor

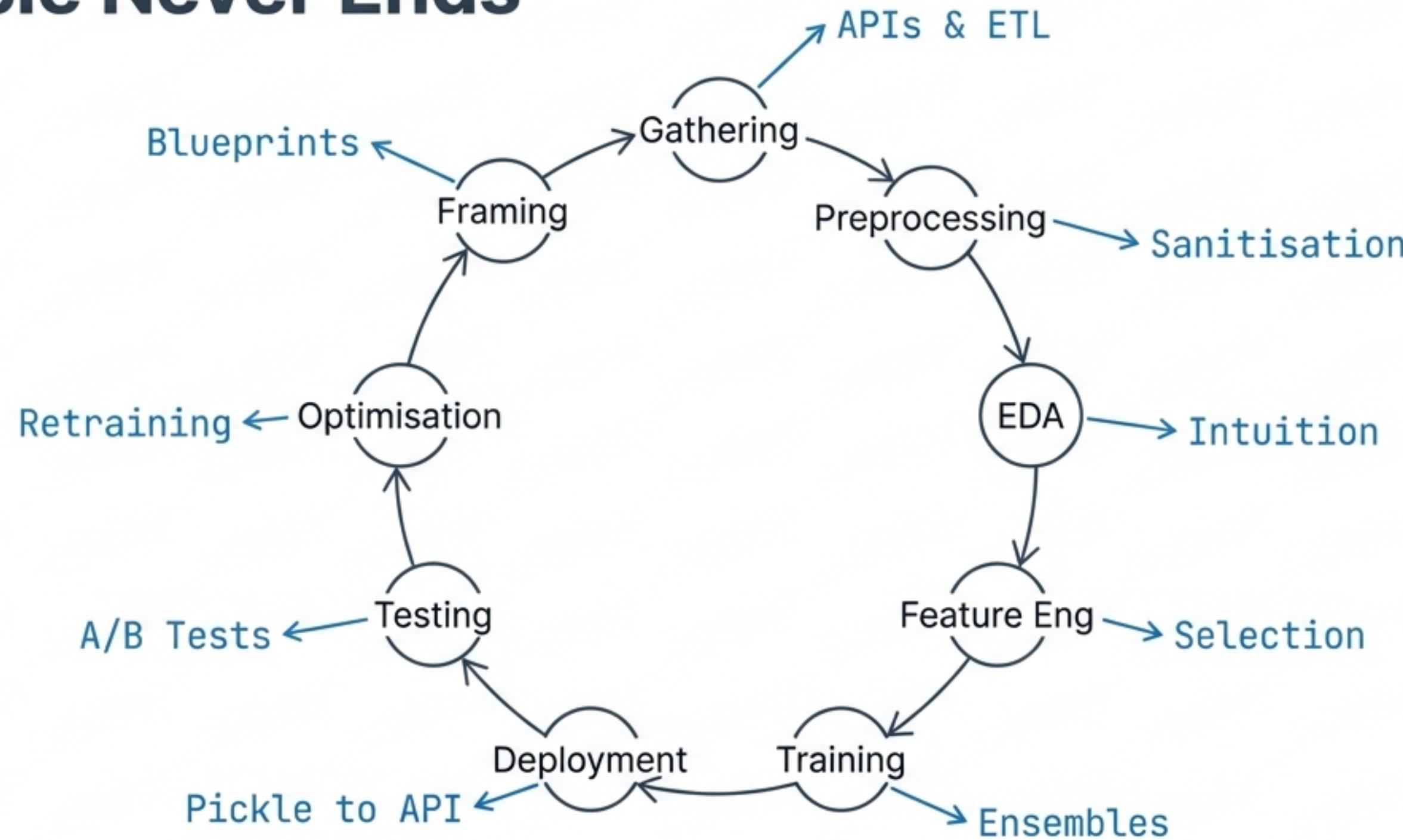
### Warning: Model Rotting

Data changes over time (e.g., evolving trends), causing performance to degrade.

### Maintenance Protocols

- Automated Retraining Schedules (Inter and JetBrains Mono)
- Data & Model Versioning/Backups
- Load Balancing for Traffic Spikes

# The Cycle Never Ends



Final Takeaway: A true Data Scientist manages the lifecycle, not just the algorithm. Stop stopping at the notebook. Start building end-to-end projects that solve problems, handle dirty data, and live on a server.