DS Interview Home Challenge

Background

The dataset represents historical job and scheduling information from a manufacturing or production environment. Each row corresponds to a unique job, containing details about scheduling, raw material requirements, inventory, production plans, and resource flags. The task is to predict the `TARGET` variable — a binary outcome (0 or 1), whose meaning is **intentionally undisclosed** to simulate real-world ambiguity.

This setup encourages not only accurate modeling but **critical thinking, domain reflection, and rigorous reasoning** to avoid false correlations and ensure model trustworthiness.

Objective

Build a **high-performing and explainable model** that predicts the `TARGET` variable. Alongside performance, the **reasoning** behind every decision — from feature engineering to modeling choices — is considered **more important** than pure accuracy.

This includes reflecting on:

- Why a feature might be relevant (or misleading),
- Why a modeling choice aligns with the task and data structure,
- Why one metric or validation strategy was chosen over another.

Argue for or against each modeling decision with clarity, grounded in evidence, domain logic, or statistical justification.

Dataset Overview

- **Observations**: 33,860 jobs
- **Features**: 50 columns (dates, quantities, flags, codes, inventory levels, etc.)
- **TARGET**: Binary variable (0/1), meaning undisclosed

Scope of Work

1. **Data Understanding & Preparation**
2. **Exploratory Data Analysis (EDA)**
3. **Feature Engineering & Selection**
4. **Modeling**
    - Please explore multiple models
5. **Model Interpretation**

Software Engineering Considerations

- Modular, clean, and reproducible code (scripts or notebooks)
- Use `.env` for configuration, and Git for version control
- Optional logging for pipelines, versioned datasets/models
- **External libraries and tools** (e.g., SHAP, Optuna, Featuretools, ChatGPT, etc.) are allowed and encouraged

---

Discussion Questions

> **Note**
>
> These questions will be part of a live discussion, please think about them in advance.

Integration Considerations

- **Usage point**: Dashboards, alerts?
- **Prediction frequency**: Real-time or batch?
- **Data integration**: Can live job data be used? Is feature freshness a concern?
- **Interface**: REST API, database output, event triggers?

Risk Management

- **Data drift**: How could factory shifts affect prediction stability?
- **Model degradation**: What's the retraining schedule and monitoring plan?
- **Auditability**: Can we explain and trace each prediction?
- **Fallbacks**: What happens if the model fails or confidence is low?

Deliverables

- ✅ **Notebook or scripts** with modular code and clear documentation
- ✅ **Presentation** covering:
  - Data understanding and EDA
  - Modeling pipeline and decisions
  - Feature importance and insights
- ✅ **Write-up** (*can be in slides*):
  - Integration points (how predictions fit in ops)
  - Architecture overview:
    - Model packaging (`.pkl`, ONNX, etc.)
    - Deployment method (batch, REST API)
    - Tools used (FastAPI, Docker, etc.)
    - Model lifecycle & retraining plans

Final Notes

The ultimate goal is not just a high ROC-AUC, f1-score, or log-loss — it's to **demonstrate thoughtful, reasoned decision-making** in the presence of uncertainty. Every modeling step should be questioned and justified, because in the real world, outcomes are rarely as clean as our datasets suggest...

Use **any tools or libraries** that help illuminate insights or speed up development — this includes **external assistants like ChatGPT** or domain-specific software.

Deadline

- You have **1 week** to complete the project.
- At the end of the week, submit your **code** and **presentation** (including your write-up).