

## **Scenario 1:**

# **Predictive Maintenance Pipeline for Industrial Equipment**

### **Context**

A manufacturing company deploys multiple predictive maintenance models to anticipate machine failures in its production lines.

Each model processes sensor data from IoT devices to predict anomalies and schedule maintenance interventions.

The company uses an automated **MLOps pipeline** involving data collection, preprocessing, model training, validation, and deployment into production monitoring systems.

### **Problem**

After several months of operation, maintenance alerts became unreliable, some models trigger too many false positives, while others fail to detect true anomalies. Engineers suspect that **data drift** and **inconsistent retraining processes** are causing these issues. However, the team lacks a **clear, process-level view** of when and how retraining, validation, and deployment occur across models and machines.

### **Objective**

Students must design an event log schema and data mining approach to:

- Trace the end-to-end retraining process of each model.
- Identify where models skip validation or reuse outdated data.
- Detect bottlenecks or non-compliant steps in retraining cycles.

## **Scenario 2:**

# **Fraud Detection System for Financial Transactions**

### **Context**

A fintech company maintains multiple fraud detection models deployed in real time to classify transactions. The MLOps environment includes stages for **data ingestion, model training, evaluation, and deployment**. Models are continuously updated based on new fraud patterns.

### **Problem**

A recent audit revealed that several fraud models were retrained using inconsistent versions of labeled data, and some were promoted to production **without full compliance testing**. This led to unexpected drops in recall rates and inconsistent fraud alerts between model versions.

### **Objective**

Students must propose a data mining view over MLOps logs to:

- Identify missing or skipped compliance steps before model deployment.
- Monitor versioning consistency between data, models, and deployment configurations.
- Establish process conformance metrics to ensure that all models follow the approved lifecycle path before production release.

## Scenario 3:

# Environmental Forecasting Platform (Satellite + Sensor Data)

### Context

An environmental agency runs several machine learning models that forecast pollution levels and water quality.

These models are trained from heterogeneous data sources — satellite images, weather stations, and water sensors.

Each model has a separate data ingestion pipeline but shares a common **validation and deployment orchestration** managed through MLflow and Airflow.

### Problem

Some forecasts arrive with significant delays or are based on outdated data. Investigations suggest asynchronous execution and **timestamp misalignment** between different data ingestion processes.

The lack of synchronization across pipelines makes it difficult to understand how updates propagate from data ingestion to deployed predictions.

### Objective

Students must create an integrated event log with:

- Synchronized global timestamps across multiple data streams.  
Case correlation between data batches, models, and forecast outputs.
- Process mining techniques to identify temporal misalignments and optimize end-to-end latency.

# Scenario 4: Healthcare Diagnosis Model Lifecycle

## Context

A hospital uses machine learning models to assist in patient diagnosis based on medical imaging and lab results.

Due to regulatory requirements, every model update must go through **data validation**, **clinical review**, **testing**, and **ethical approval** before deployment.

All steps are managed within an internal MLOps platform integrated with machine learning pipelines.

## Problem

During a recent quality audit, discrepancies were found: some models were deployed **before clinical review completion**, while others remained **stuck in approval queues** for weeks.

Moreover, no unified trace exists to explain how many times a model was retrained, reviewed, and redeployed — making **traceability and compliance auditing** nearly impossible.

## Objective

Students should design an event log structure and process discovery method to:

- Trace the full approval and deployment cycle for each diagnostic model.
- Detect cases where compliance steps were skipped or delayed.
- Visualize model lifecycles for auditing and continuous monitoring.