

DAY 1 / SESSION 4

Virtual Metrology



가상 공간에서의 실험, 현실의 실패를 막다



SCHEDULE

15:00 ~ 17:00 (120min)



KEY TOPICS

Virtual Metrology • Optimization

가상 계측 (Virtual Metrology)

Data-Driven Quality Prediction



개념 및 배경 (Background)

Why Virtual Metrology?

모든 제품을 일일이 파괴해서 검사할 수 없으므로, **센서 데이터**만 보고 품질 수치를 예측하는 기술.

- ⚠ 전수 검사 불가능 (시간/비용 과다)
- 공정 변수(온도, 압력 등)로 품질을 추정
- ✓ "가상의 계측기" 역할 수행



예측 메커니즘 (Mechanism)

Regression Model

품질(Y)과 공정 변수(X) 간의 상관관계를 학습하여 수식화.

$$\text{Quality} = f(\text{Temp}, \text{Pressure}, \text{RPM})$$

- ☒ 실시간 품질 보증 (Real-time QA)
- ▷ 불량 발생 사전 차단 (Proactive)

↔ Process Change

물리적 검사 (PHYSICAL)

샘플링 검사

사후 확인, 불량 유출 위험

vs

가상 계측 (VIRTUAL)

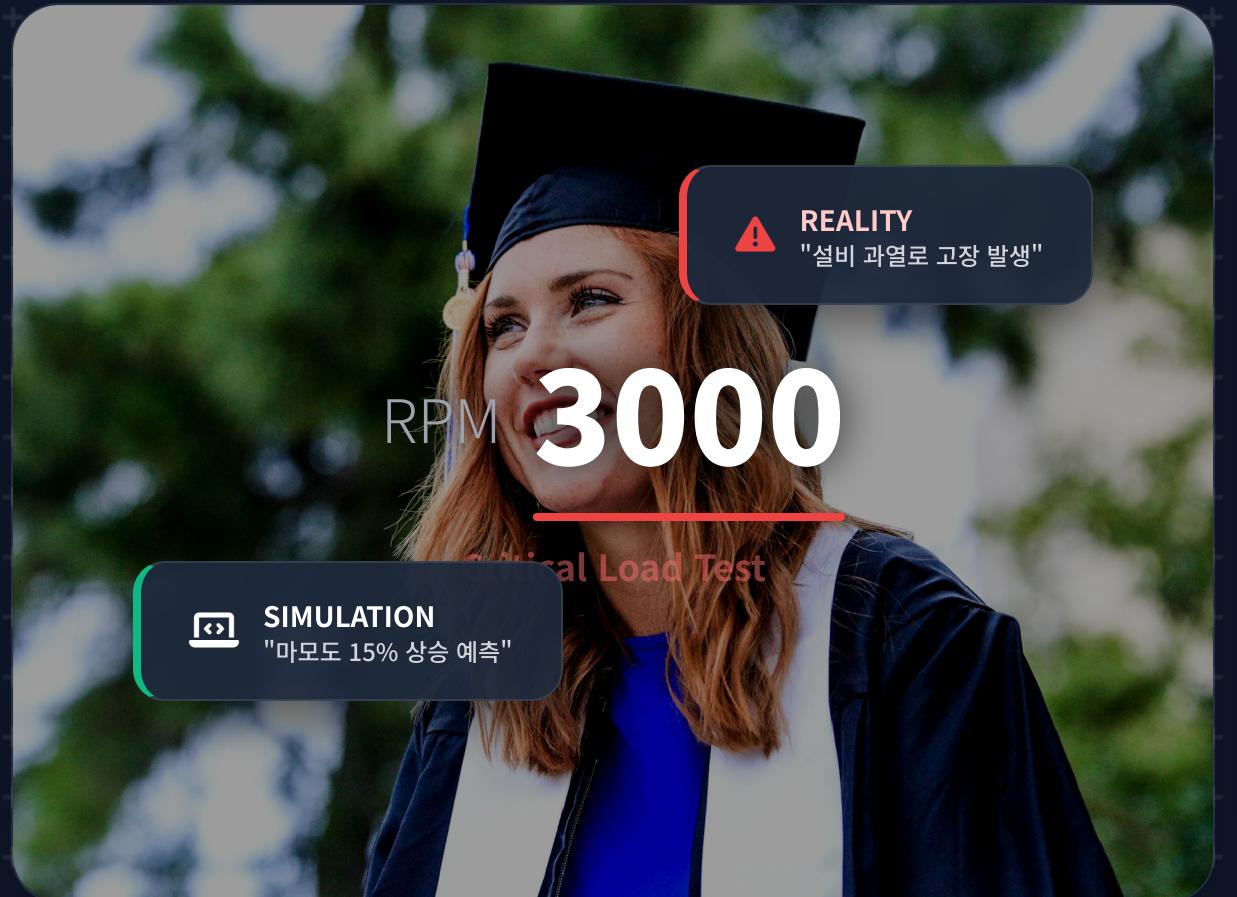
실시간 전수 검사

사전 예측, 100% 품질 보증

"데이터가 곧 품질 검사 장비가 됩니다."

The Power of What-if Simulation

?"생산량을 늘리고 싶은데, RPM을 2000 → 3000으로 올려도 될까?"



REALITY

Trial & Error (시행착오)

* 물리적 파손 위험

실제 기계로 테스트하다 고장나면 돌이킬 수 없음

₩ 막대한 손실 비용

수리비 + 라인 중단(Downtime) 손해 = 수억 원



DIGITAL TWIN

Risk-free Optimization

戴上 부하 테스트

가상 모델에 '3000' 입력 → 온도/마모도 즉시 예측

실패 비용 Zero

수천 번 실패해도 비용 0원, 안전하게 최적점 탐색

최적화(Optimization)의 딜레마 & 목표

속도와 안정성 사이의 줄타기: "어디까지 올려도 안전한가?"



Trade-off

상충 관계의 딜레마

⌚ 생산성 (Speed)

RPM을 높이면 생산량은 증가하지만...

❤️ 품질 & 수명 (Quality & Life)

설비 부하가 커져 마모가 빨라지고 품질 불량 발생.

⚠️ 현실의 위험 (Risk)

최적점을 찾으려다 기계를 고장 낼 수 있음.



ANALOGY

과속 운전: 목적지에 빨리 가려다 엔진 과열이나 사고 발생.



Goal

안전 한계 내 최대 효율

☒ 제약 조건 (Constraint)

"마모도가 위험 수준(200min)을 넘지 않아야 함"

↳ What-if 시뮬레이션

가상 공간에서 RPM을 단계별로 올려보며 한계점 탐색.

▣ 최적점 도출 (Optimal Point)

안전 마진을 확보한 상태에서의 최대 RPM 결정.



ANALOGY

레드존 직전: 엔진이 터지지 않는 한계까지 성능을 끌어올림.

최종 산출물 (Optimization Output)

단순한 예측을 넘어, 현장에 적용 가능한 구체적인 운전 레시피를 제공합니다.

RECOMMENDED
2,150 RPM

EST. WEAR
195 min

What-if Simulation Result

RPM 변화에 따른 공구 마모도(Tool Wear) 예측 결과 분석

Simulation Config

RANGE (RPM)

1,200 ~ 2,800

STEP

50 RPM



Exponential Growth

RPM이 증가할수록 마모도가 **비선형적(급격)**으로
상승하는 패턴 확인



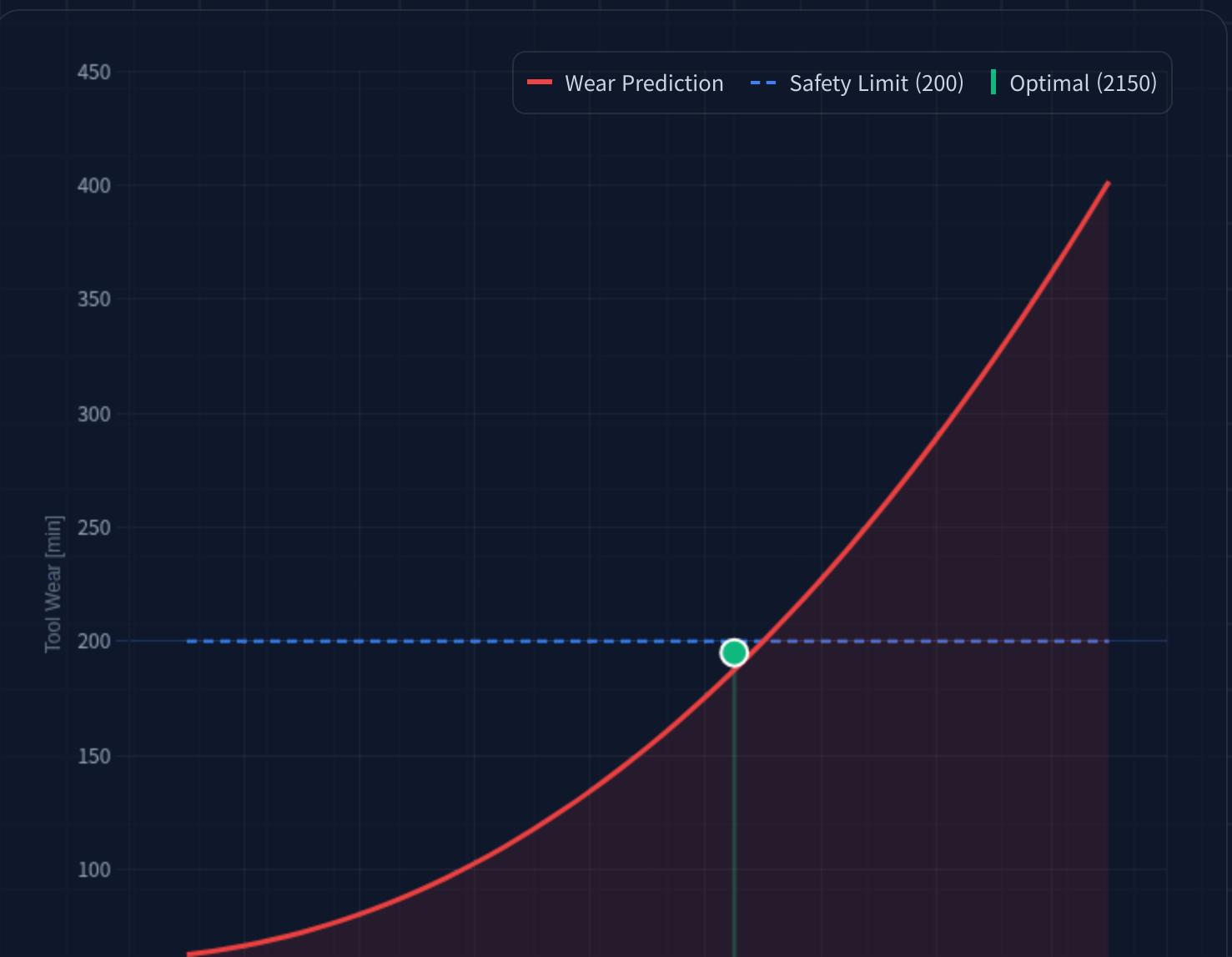
Safety Threshold

공구 파손 방지를 위한 안전 한계선 **200 min** 설정



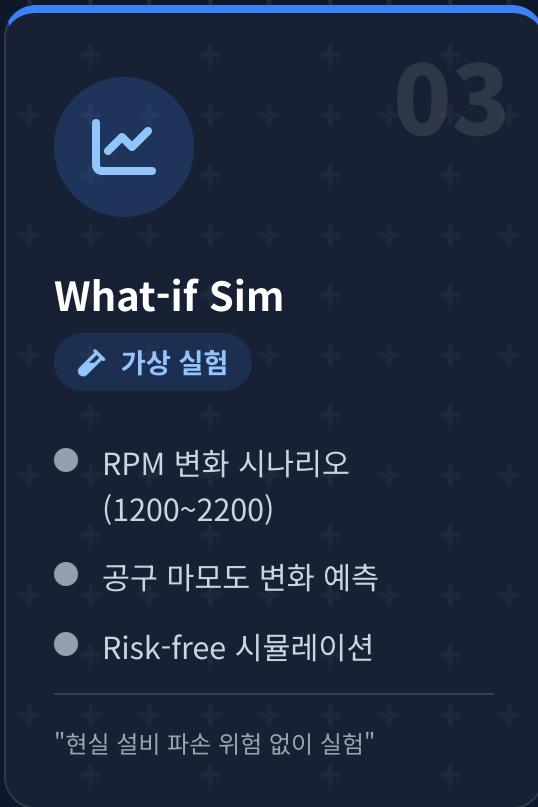
Optimal Point

안전 한계를 넘지 않는 최대 효율 지점 **2,150 RPM** 도출



실습 개요: 가상 공정 시뮬레이터 만들기

G Google Colab (Python 3) &  pandas, scikit-learn 환경



Goal: 센서 데이터가 AI 모델을 거쳐 **최적의 제어 파라미터**로 변환되는 과정을 경험합니다.

 Virtual Metrology  What-if Analysis

데이터 로드 & AI 모델 학습

센서 데이터를 불러와 전처리를 거친 후, Random Forest Regressor 모델을 학습시킵니다.

● ● ● train_model.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

1. 데이터 로드 (UCI AI4I 2020)

```
url = "https://archive.ics.uci.edu/ml/machine-learning-
databases/00601/ai4i2020.csv"
df = pd.read_csv(url)
```

2. 입력 변수(X) 및 출력 변수(y) 설정

```
features = ['Air temperature [K]', 'Process temperature [K]',
'Rotational speed [rpm]', 'Torque [Nm]']
X = df[features]
y = df['Tool wear [min]']
```

3. 학습/테스트 데이터 분리 (8:2)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Training Result

R2 Score

0.9987

High Accuracy

MSE (Mean Squared Error)

1.24

Low Error Rate



↳ What-if 시뮬레이션 실행

학습된 AI 모델을 활용해 RPM을 가상으로 증가시키며 공구 마모도 변화를 예측합니다.



whatif_sim.py

```
# 1. 시뮬레이션 대상 설비 선택 (100번째 데이터)
target_machine = X_test.iloc[100].copy()
print("== Current Machine Settings ==")
print(target_machine)

# 2. RPM 시뮬레이션 범위 설정 (1200 ~ 2200, 50단위)
simulation_rpms = np.arange(1200, 2201, 50)
predicted_wears = []

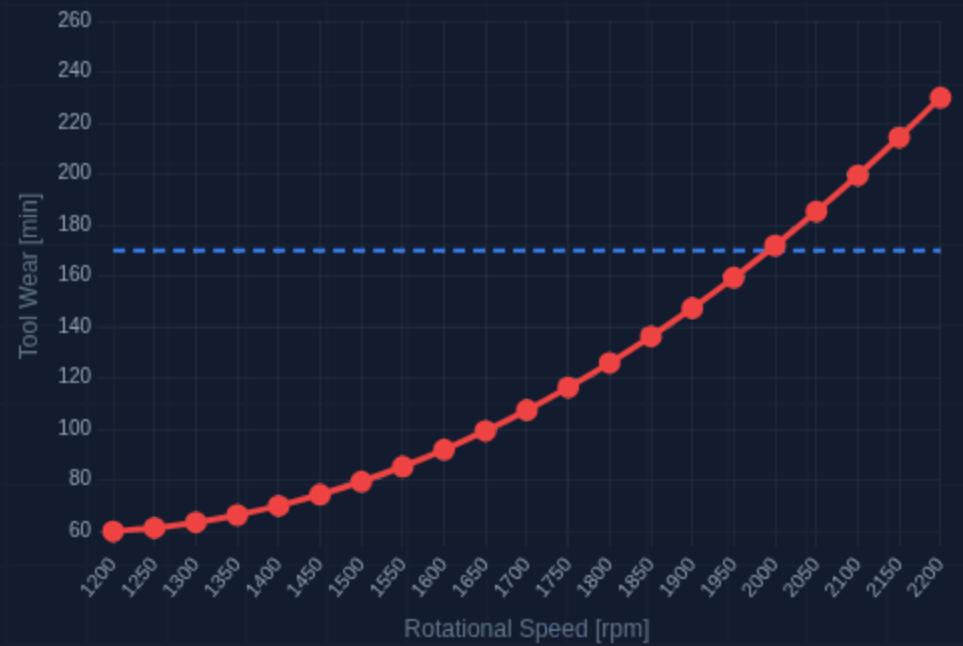
# 3. 반복문을 통한 What-if 예측 수행
for rpm in simulation_rpms:
    # 가상 입력 데이터 생성 (RPM만 변경)
    virtual_input = target_machine.copy()
    virtual_input['Rotational speed [rpm]'] = rpm

    # AI 모델로 마모도 예측
    pred_wear = rf_regressor.predict([virtual_input])[0]
    predicted_wears.append(pred_wear)

# 4. 결과 시각화 (Matplotlib)
plt.figure(figsize=(10, 6))
```

Simulation Output

— Predicted Wear — Safety Limit (170)



💡 Scenario Setting

"현재 설정에서 **RPM**만 1200에서 2200까지 강제로 올린다면?"

→ 실제 설비에서는 위험해서 못하는 실험을 가상 공간(Digital Twin)에서 수행

최적화 해석 & Full Source

안전 한계선 이하에서 최대 생산성을 내는 최적 RPM을 도출하는 전체 파이프라인 코드입니다.



full_pipeline.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split

# 1. 데이터 로드 & 모델 학습
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00601/ai4i2020.csv"
df = pd.read_csv(url)
features = ['Air temperature [K]', 'Process temperature [K]', 'Rotational speed [rpm]',
            'Torque [Nm]']
X = df[features]; y = df['Tool wear [min]']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
rf = RandomForestRegressor(n_estimators=100, random_state=42).fit(X_train, y_train)

# 2. What-if 시뮬레이션 설정
base_config = X_test.iloc[0].copy()
rpm_range = np.arange(1300, 2801, 50)
sim_results = []

for r in rpm_range:
```

Optimization Result

● Wear — Limit ● Optimal

Optimal RPM

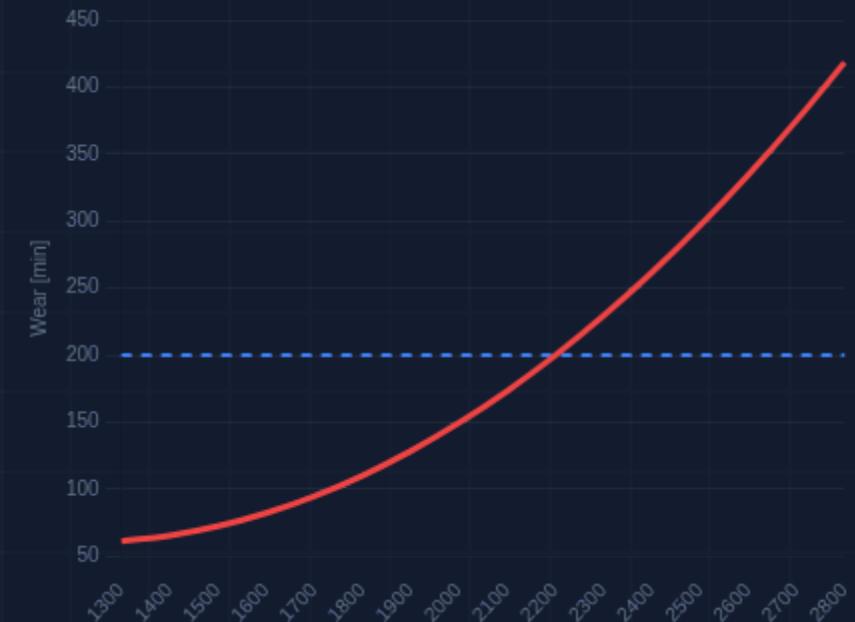
2,150

Max Productivity

Est. Tool Wear

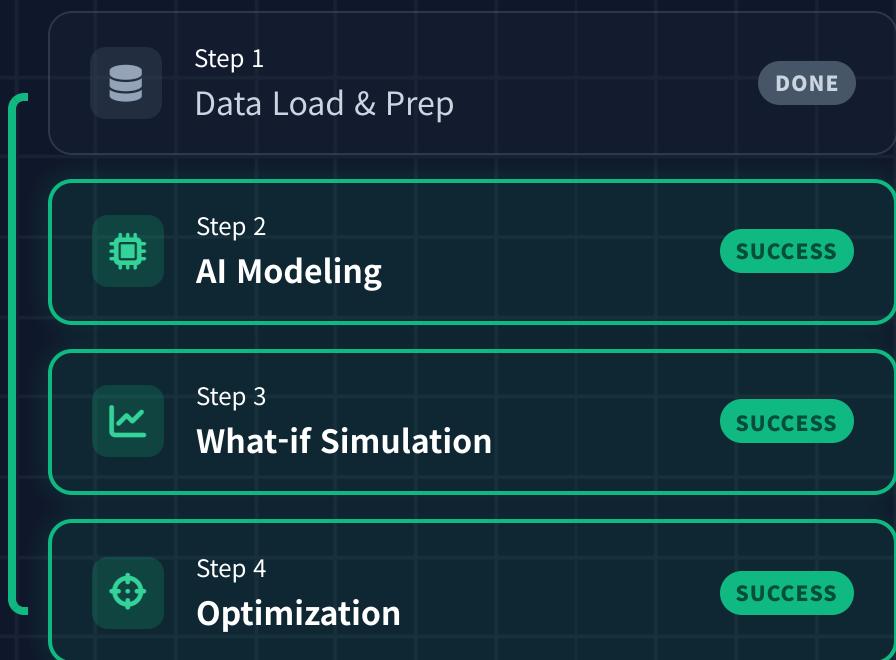
195 min

Safe Range (< 200)



실습 목표 점검 (Checklist)

☑ 가상 공정 시뮬레이터 구축 완료: 자가 진단



☰ Core Objectives (핵심 목표)

회귀 모델 학습 성공

Random Forest 모델을 학습시켜 테스트 데이터셋에서 **R2 Score > 0.95** 이상의 정확도를 달성했습니까?

What-if 그래프 생성

RPM 변화(1200~2200)에 따른 공구 마모도의 **비선형적 증가 패턴(곡선)**을 시각화했습니다?

최적 RPM 도출

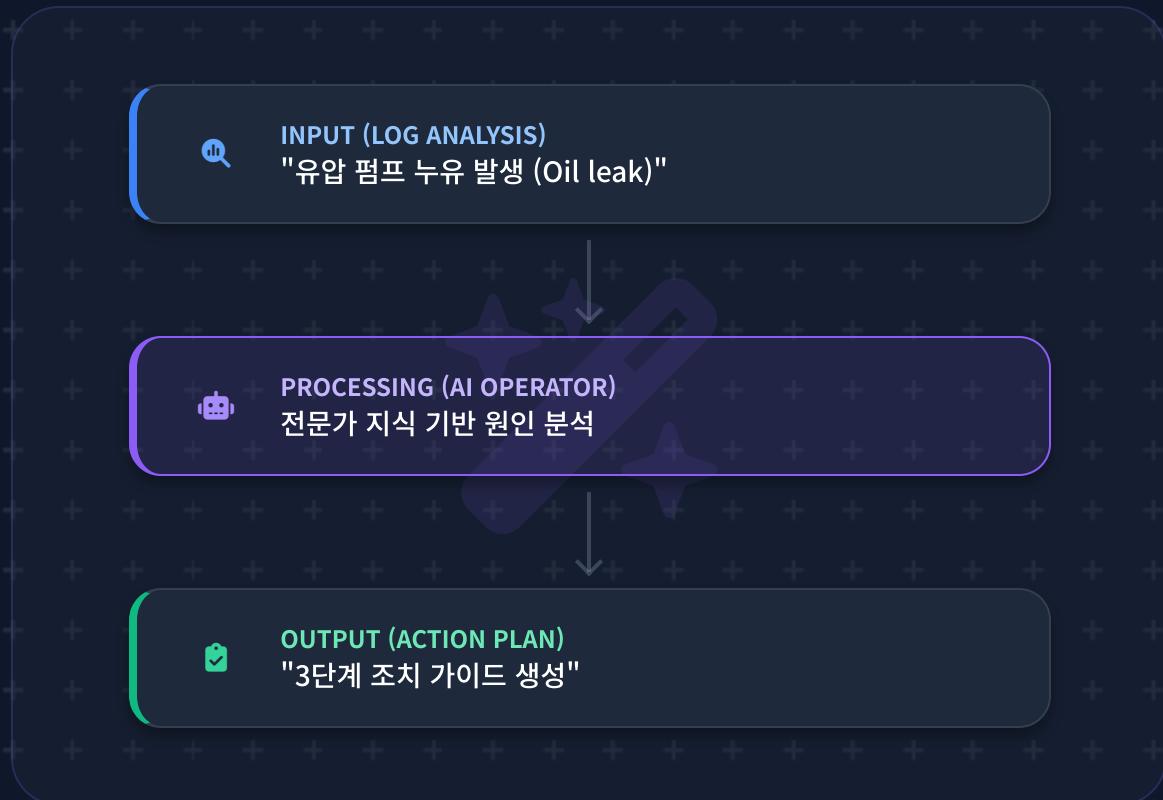
안전 한계선(200min)을 넘지 않는 범위 내에서 **최적의 운전 속도(약 2,150 RPM)**를 찾아냈습니다?

코드 이해도 (Python)

데이터 로드부터 모델 학습, 예측 시뮬레이션까지의 **전체 파이프라인 코드** 흐름을 이해했습니다?

Bridge Session: From Data to Insight

☞ LDA(분석)에서 LLM(처방)으로 연결되는 디지털 트윈의 진화



THE GAP

What vs How의 단절

기존 분석(LDA)은 "무슨 문제가 발생했는지(What)"는 알려주지만,
"어떻게 고쳐야 하는지(How)"에 대한 답은 주지 못합니다.

THE BRIDGE

LLM as an Engineer

LLM에게 '20년 차 설비 보전 전문가' 페르소나를 부여하여,
로그 데이터에 대한 전문적인 해석과 해결책을 요청합니다.

VALUE

Actionable Insight

단순 진단을 넘어 현장 작업자가 즉시 실행 가능한
표준 운영 절차(SOP)를 자동으로 생성합니다.

🤖 LLM 연동: 고장 로그 → 조치 가이드 생성

GPT에게 '20년 차 전문가' 페르소나를 부여하여, 로그에 대한 구체적인 수리 절차(SOP)를 처방받습니다.

The screenshot shows a code editor interface with a dark theme. At the top, there are three colored circular icons (red, yellow, green) and a file tab labeled "llm_operator.py". The code itself is as follows:

```

# !pip install openai
import openai

# 1. API Key 설정 (실습 시 본인 키 입력)
# openai.api_key = "sk-..."

# 2. 1일차 LDA 분석에서 발견된 로그 예시
log_message = "Oil leak in hydraulic pump system"

# 3. 시스템 프롬프트 (전문가 페르소나 부여)
system_prompt = """
당신은 제조 현장의 20년 차 베테랑 설비 보전 엔지니어입니다.
작업자가 입력한 '설비 고장 로그'를 분석하여 다음을 수행하세요:
1. 고장의 가장 유력한 기술적 원인 1가지
2. 현장에서 즉시 수행해야 할 조치 사항 3단계 (Step-by-step)
답변은 한국어로 간결하고 전문적으로 작성하세요.
"""

# 4. LLM 호출 (API 호출 시뮬레이션 코드)
"""
response = openai.chat_completions.create(
    model="gpt-3.5-turbo",
    messages=[{"role": "user", "content": log_message + "\n\n" + system_prompt}]
)
print(response['choices'][0]['message']['content'])
"""

```

>_ Output Console

Simulated

⚠ DETECTED LOG (Input)

"Oil leak in hydraulic pump system"

↓

👤 AI Maintenance Engineer

🔍 1. 유력 원인 (Root Cause)

유압 시스템 내 O-ring 경화(Hardening) 및 손상으로 인한 밀폐력 상실 가능성이 높음.

🛠 2. 긴급 조치 사항 (Action Plan)

- | **Step 1 (Safety):** 설비 전원 차단 및 밸브 폐쇄로 잔류 유압 제거
- | **Step 2 (Inspect):** 형광 염료 투입 후 UV 램프로 미세 누유 지점 식별
- | **Step 3 (Repair):** 규격 O-ring(Viton 재질) 교체 후 규정 토크로 재조립