# 2. 예측 유지보수: Keras+TensorFlow LSTM을 활용한 시계열 예측

## 프로젝트 개요

이전 Lab에서 다룬 XGBoost와 달리, \*\*시계열 데이터의 시간적 흐름(Time Sequence)\*\*을 분석하여 미래의 하드 드라이브 고장 가능성을 예측하는 **순환 신경망(RNN) 모델**인 LSTM(Long Short-Term Memory)을 구축했습니다. 이는 장기간의 경향성을 파악하는 예측 유지보수 분야의 핵심 딥러닝 기법입니다.

## ⚙️ 주요 기술 및 역량

* **기술:** Keras, TensorFlow, LSTM 레이어, 시퀀스 데이터 리쉐이핑(Reshaping).
* **핵심 역할:**
  1. **시퀀스 데이터 변환:** 시간 단위 데이터(df\_train)를 LSTM 모델이 요구하는 **3차원 시퀀스 데이터** 형태로 변환하는 전처리 로직(예: (samples, timesteps, features))을 구현했습니다.
  2. **LSTM 모델 설계:** Keras와 TensorFlow를 사용하여 GPU 환경에 최적화된 LSTM 모델 구조(레이어 수, 유닛 수)를 설계하고 훈련했습니다.
  3. **추세 분석:** 현재 데이터뿐만 아니라 **과거 일정 기간 동안의 데이터 추세**를 기반으로 미래 고장 여부를 예측하는 모델의 성능을 평가했습니다.

## 🛠️ 주요 구현 내용

1. **시퀀스 길이 정의:** sequence\_length를 정의하여 LSTM이 몇 개의 과거 시점을 고려할지 결정했습니다.
2. **데이터 Reshaping:** NumPy의 reshape 기능을 사용하여 2D 테이블 형태의 훈련 데이터를 LSTM 입력 형식인 3D 텐서로 변환했습니다.
3. **모델 훈련 및 평가:** GPU를 활용하여 훈련 시간을 단축하고, 훈련된 모델의 예측 결과와 실제 고장 기록을 비교하여 모델의 예측력을 검증했습니다.

## 📌 주요 코드 로직 (시퀀스 데이터 변환 추상화)

import numpy as np  
  
# 데이터프레임을 NumPy 배열로 변환  
x\_train = x\_train.to\_numpy()   
x\_test = x\_test.to\_numpy()   
  
# 시퀀스 길이 (예: 10일간의 데이터를 시퀀스로 사용)  
sequence\_length = 10   
  
# LSTM 입력 형태로 데이터 Reshaping (3차원: 샘플 수, 시퀀스 길이, 특징 수)  
# X\_train.shape[0] / sequence\_length : 새로운 샘플(시퀀스)의 개수  
x\_train = x\_train.reshape(int(x\_train.shape[0] / sequence\_length),   
 sequence\_length,   
 x\_train.shape[1])  
# x\_train의 최종 형태: (num\_sequences, sequence\_length, num\_features)