# 3. 예측 유지보수: 오토인코더(Autoencoder) 기반 이상 탐지

## 프로젝트 개요

이 프로젝트는 시계열 데이터에서 \*\*이상 패턴(Anomalies)\*\*을 탐지하기 위해 **오토인코더(Autoencoder, AE)** 딥러닝 모델을 사용했습니다. 고장 샘플(Anomalies)이 희귀하다는 점을 활용하여, 모델이 정상 데이터만을 학습하도록 훈련시킨 후, 정상 패턴에서 벗어나는 데이터를 이상으로 분류하는 기법을 구현했습니다.

## ⚙️ 주요 기술 및 역량

* **기술:** Autoencoder(인코더/디코더), 이상 탐지(Anomaly Detection), 재구성 오류(Reconstruction Error).
* **핵심 역할:**
  1. **이상 탐지 방법론:** 오토인코더가 **정상 데이터만을 학습**하도록 훈련 데이터셋을 구성하고, 복원된 데이터와 원본 데이터 간의 **재구성 오류(Reconstruction Error)** 임계치를 설정하여 이상 여부를 판단하는 논리를 구현했습니다.
  2. **GPU 훈련 최적화:** GPU를 사용하여 대용량 데이터 기반의 오토인코더 모델을 효율적으로 훈련했습니다.
  3. **데이터 라벨링:** 테스트 데이터셋의 고장 라벨을 NumPy 배열 형식으로 추출하여 모델 평가 및 시각화에 활용했습니다.

## 🛠️ 주요 구현 내용 (Lab3-AE-For-Anomaly-Detection 기반)

1. **훈련 데이터 구성:** 훈련 데이터셋에서 **정상(Non-defective) 샘플**만을 필터링하여 오토인코더 입력으로 사용했습니다.
2. **모델 구조:** 입력층, 인코더 레이어, 잠재 공간(Latent Space), 디코더 레이어, 출력층으로 구성된 오토인코더 네트워크를 설계했습니다.
3. **이상 탐지:** 훈련된 모델에 테스트 데이터를 입력했을 때 발생하는 \*\*재구성 오류(MSE)\*\*가 미리 설정된 임계치를 초과할 경우, 해당 하드 드라이브를 \*\*'이상(Anomaly)'\*\*으로 분류했습니다.
4. **평가 데이터셋 정의:** 오토인코더의 훈련은 x\_train 데이터를 사용하여 진행하지만, 유효성 검사 데이터(validation\_data)를 (x\_test, x\_test) 형태로 설정하여 모델의 일반화 능력을 평가했습니다.

## 📌 주요 코드 로직 (오토인코더 훈련 데이터 준비 추상화)

# 'failure' 라벨이 0인 정상 샘플만 추출하여 훈련 데이터셋 구성  
x\_train\_normal = x\_train[df\_train['failure'] == 0]  
  
# 오토인코더는 입력(X)을 출력(X)으로 복원하도록 학습  
# 따라서 입력 데이터와 목표 데이터가 동일합니다.  
history = autoencoder.fit(  
 x\_train\_normal,   
 x\_train\_normal,   
 epochs=EPOCHS,  
 batch\_size=BATCH\_SIZE,  
 validation\_data=(x\_test, x\_test) # 유효성 검사도 원본-복원 형태로 진행  
)