

# UCI SECOM 공정 불량 분석

(개인 프로젝트)

## 1. 프로젝트 개요

본 프로젝트는 UCI SECOM 공정 센서 데이터를 활용하여, 제조 공정에서 발생하는 불량 제품의 패턴을 분석하고 불량 발생 가능성을 사전에 예측하는 모델을 구축하는 것을 목표로 했습니다.

단일 센서 값만 보는 것이 아니라, 여러 센서와 공정 변수 간의 관계를 분석하여 불량 원인을 추적하고, 이를 예측 모델에 반영하는 데 초점을 맞췄습니다.

본 프로젝트는 개인 프로젝트로 진행되었으며, 프로젝트의 방향 설정, 데이터 전처리, 분석 구조 설계를 주도했습니다.

## 2. 문제 정의

대규모 제조 공정 데이터에서는 다음과 같은 문제가 존재합니다.

- 불량 제품이 어떤 센서/공정 변수에서 발생하는지 명확히 파악하기 어려움
- 변수 수가 많아 고차원 데이터 분석 필요
- 불량 발생을 사전에 예측하거나 예방할 기준 부족

핵심 문제:

“공정 센서 데이터만으로 불량 제품을 사전에 예측하고, 문제 공정 변수를 추적할 수 있을까?”

## 3. 데이터 및 입력 정보

- 데이터셋: UCI SECOM (Kaggle)
- 샘플 수: 1,567

- 센서 변수: 590+
- 타겟: Pass/Fail (-1: 정상, 1: 불량)

데이터는 결측치가 존재하며, 고차원 변수들로 구성되어 있어 전처리 및 Feature Selection 과정이 필수적이었습니다.

## 4. 해결 접근 방식

### 4-1. 데이터 전처리 및 변수 선택

- 결측치 70% 이상 변수 제거
- 저분산 변수 제거
- 상관관계 높은 변수 제거
- 최종 사용 변수 수: 202개

### 4-2. 모델링

- Logistic Regression
  - class\_weight 적용, StandardScaler 사용
  - ROC-AUC: 0.68, 불량 Recall 낮음
- XGBoost
  - scale\_pos\_weight 적용, Threshold 0.3
  - ROC-AUC: 0.80
  - Feature Importance 상위 20개 센서 도출

## 5. 문제 발생 및 해결 과정

## 문제 1. 고차원 데이터와 결측치 문제

문제

상황:

센서 수가 많고 결측치 존재로 분석 어려움

해결 과정:

- 결측치 70% 이상 변수 제거
- 저분산/상관 변수 제거, 최종 202개 변수 사용

결과:

- 분석 가능 구조 확보
- 주요 변수 중심으로 모델 학습 가능

## 문제 2. 클래스 불균형

문제

상황:

불량 제품 비율 약 6.6% → 예측 모델이 정상 제품 위주 학습

해결 과정:

- Logistic Regression: class\_weight 적용
- XGBoost: scale\_pos\_weight 적용, Threshold 0.3 조정

결과:

- 불량 Recall 개선
- ROC-AUC 0.80 달성

## 문제 3. 불량 원인 추적

문제

상황:

단순 예측만으로는 문제 공정 변수 파악 불가

해결 과정:

- XGBoost Feature Importance로 상위 센서 도출

- Threshold 조정을 통한 불량 탐지 민감도 개선

결과:

- 불량 발생 영향 변수 추적 가능
- 실제 공정 개선 및 모니터링에 활용 가능

## 6. 주요 성과

- 대규모 공정 데이터 분석 경험 확보
- 불량 예측 모델 구축 및 중요 변수 추적
- Threshold 기반 불량 탐지 개선
- 단순 모델링을 넘어 데이터 기반 의사결정 지원 가능성 확보

## 7. 사용 기술 및 개발 환경

- Python
  - Pandas, NumPy, Matplotlib, Seaborn
  - scikit-learn, XGBoost
- 분석 기법: 데이터 전처리, Feature Selection, Logistic Regression, XGBoost, Threshold 조정, Feature Importance

## 8. GitHub Repository

<https://github.com/rjaekawpxm1-netizen/portfolio/tree/main/Projects/secom>