

DAY 1 / CLASS 2

Manufacturing Data Preprocessing Lab

제조 시계열 데이터 처리 기초



TIME

11:00 ~ 12:00 (60min)



TOPIC

Colab, Preprocessing, Correlation

Google Colab 접속 가이드

| 실습 환경 준비: Step 1 & Step 2



Step 1. 접속 및 검색

구글 검색창에 '**Google Colab**' 입력 후 최상단 링크 클릭
URL: colab.research.google.com



Step 2. 구글 계정 로그인

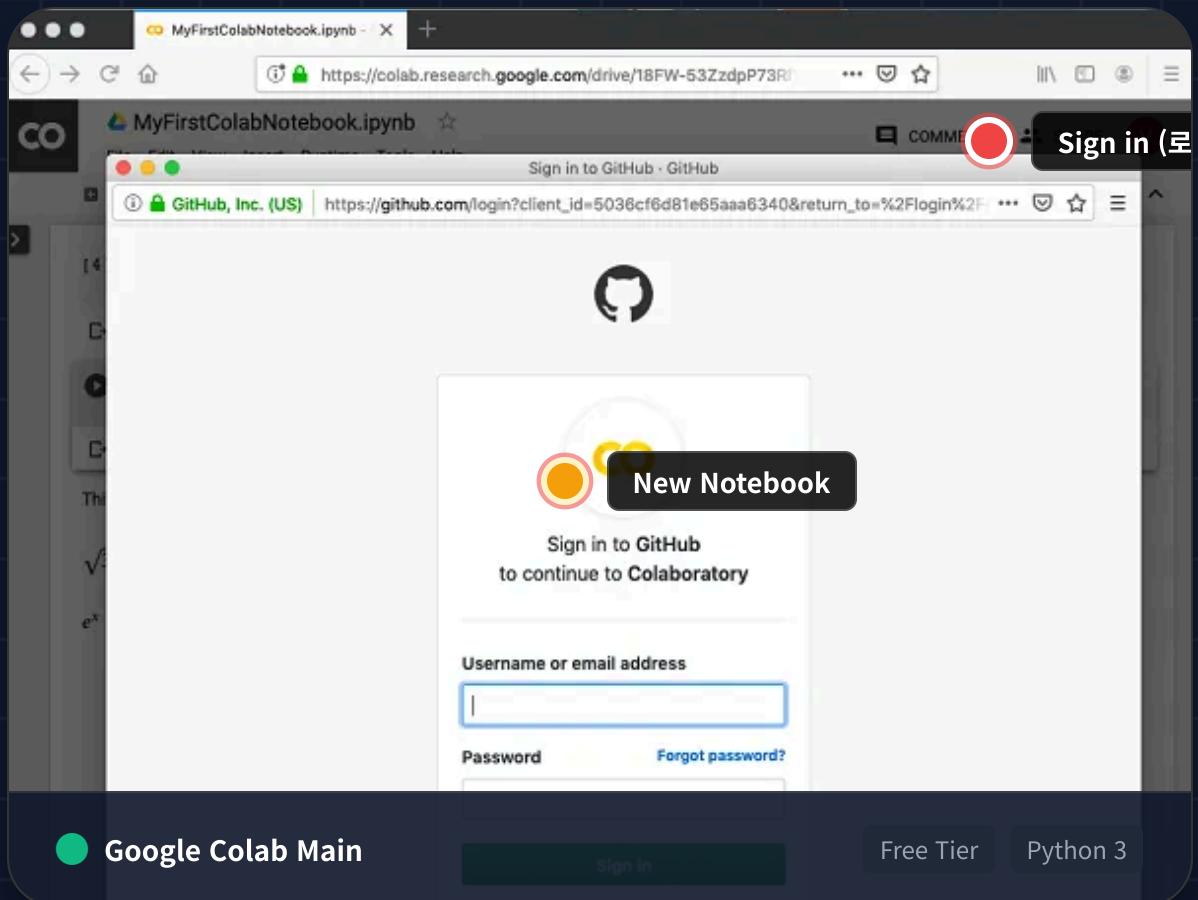
우측 상단 '로그인' 버튼 클릭 (Google 계정 필수)
기존 Gmail 계정 그대로 사용 가능



Browser Recommendation

최적의 호환성을 위해 **Chrome** 또는 **Firefox** 최신 버전을 사용해주세요.

* 팝업 차단이 설정되어 있다면 해제해야 새 창이 열립니다.



Step 3 Workspace Setup

| 노트북 생성 및 드라이브 연동하기



1. 새 노트 생성 (Create Notebook)

팝업창 하단의 '새 노트(New Notebook)' 버튼 클릭

또는 상단 메뉴: 파일(File) > 새 노트



2. 파일명 변경 (Rename)

좌측 상단 'Untitled0.ipynb'를 클릭하여 변경

파일명: Day1_DigitalTwin_Practice.ipynb



3. 드라이브 마운트 (Mount Drive)

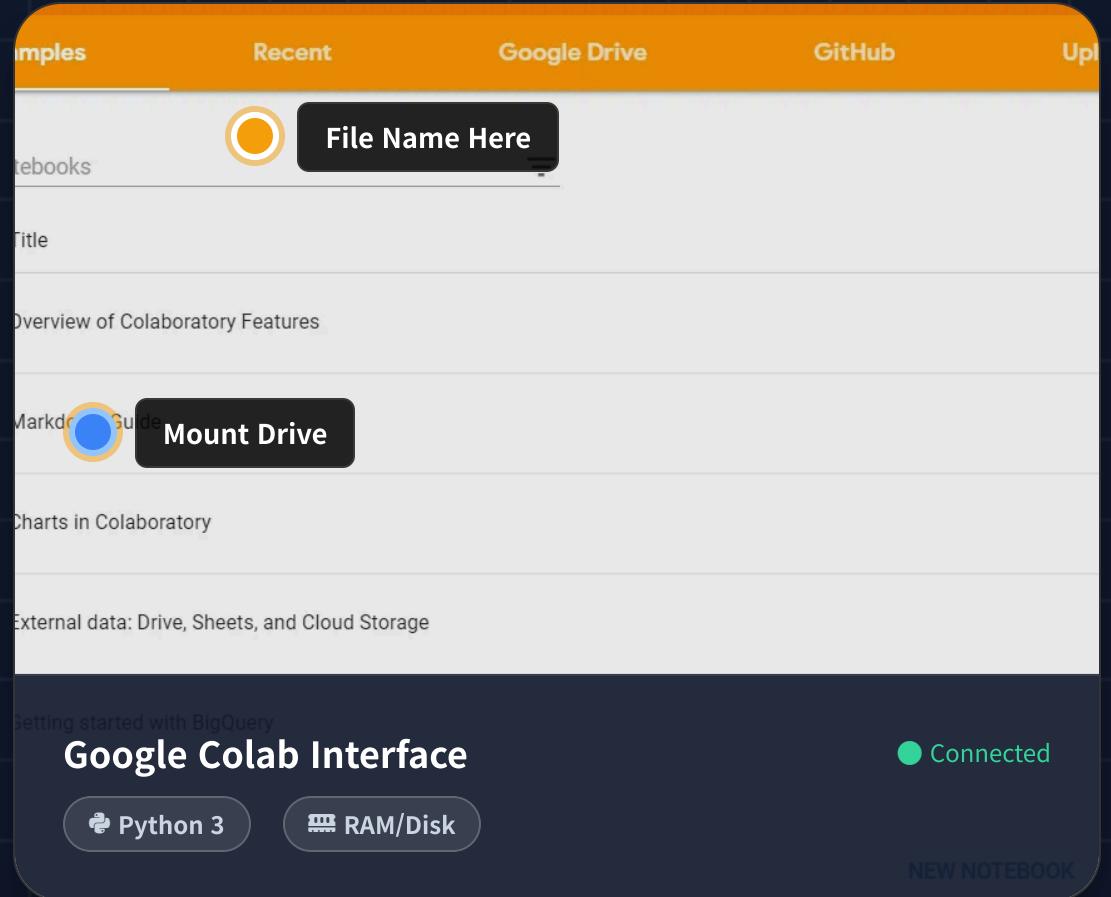
좌측 폴더 아이콘 클릭 후 드라이브 아이콘 선택

Google Drive 파일에 접근하기 위한 권한 허용 필요



Setup Tip: Runtime Type

이번 데이터 전처리 실습은 연산량이 크지 않으므로, **GPU 가속기 없이 기본(CPU)** 런타임으로 진행해도 충분합니다.



Target Milling Machine

| 분석 대상 설비: 5가지 핵심 센서 변수 구조



Temperature Kelvin [K]

- Air temperature: 공장 내부 대기 온도
- Process temperature: 실제 가공 부위 온도



Kinematics rpm, Nm

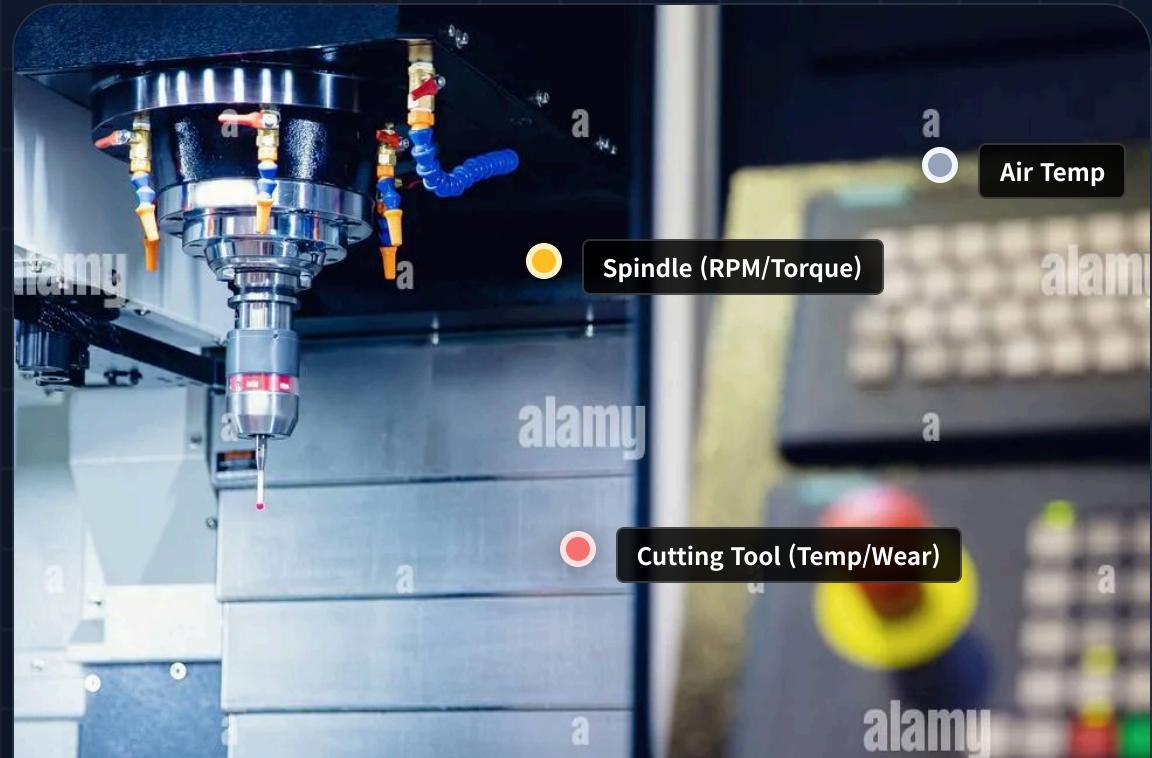
- Rotational speed: 스팬들 모터 회전 속도
- Torque: 모터가 축에 가하는 회전력(부하)



Tool Condition min

- Tool wear: 공구 사용 누적 시간

*마모도가 높을수록 고장 확률 증가 (핵심 예측 변수)



Data Source

UCI Machine Learning Repository

AI4I 2020 Predictive Maintenance Dataset

CNC Milling Center

● Active Monitoring

CSV 10,000 Rows

14 Features

Load Dataset with pandas

UCI Machine Learning Repository에서 AI4I 2020 예측 정비 데이터셋을 직접 불러옵니다.


load_data.py

```
import pandas as pd

# 1. UCI 데이터셋 URL 정의
url = "https://archive.ics.uci.edu/ml/machine-learning-
databases/00601/ai4i2020.csv"

# 2. CSV 파일 로드 (DataFrame 생성)
df = pd.read_csv(url)

# 3. 데이터 구조 확인
print("== Dataset Shape ==")
print(df.shape)

# 4. 상위 5개 행 미리보기
print("\n== First 5 Rows ==")
print(df.head())
```

Execution Result

== Dataset Shape ==

(10000, 14)

== First 5 Rows ==

UDI	Type	Air temp [K]	Process temp [K]	RPM
1	M	298.1	308.6	1551
2	L	298.2	308.7	1408
3	L	298.1	308.5	1498
4	L	298.2	308.6	1433
5	L	298.2	308.7	1408

... 9 more columns hidden



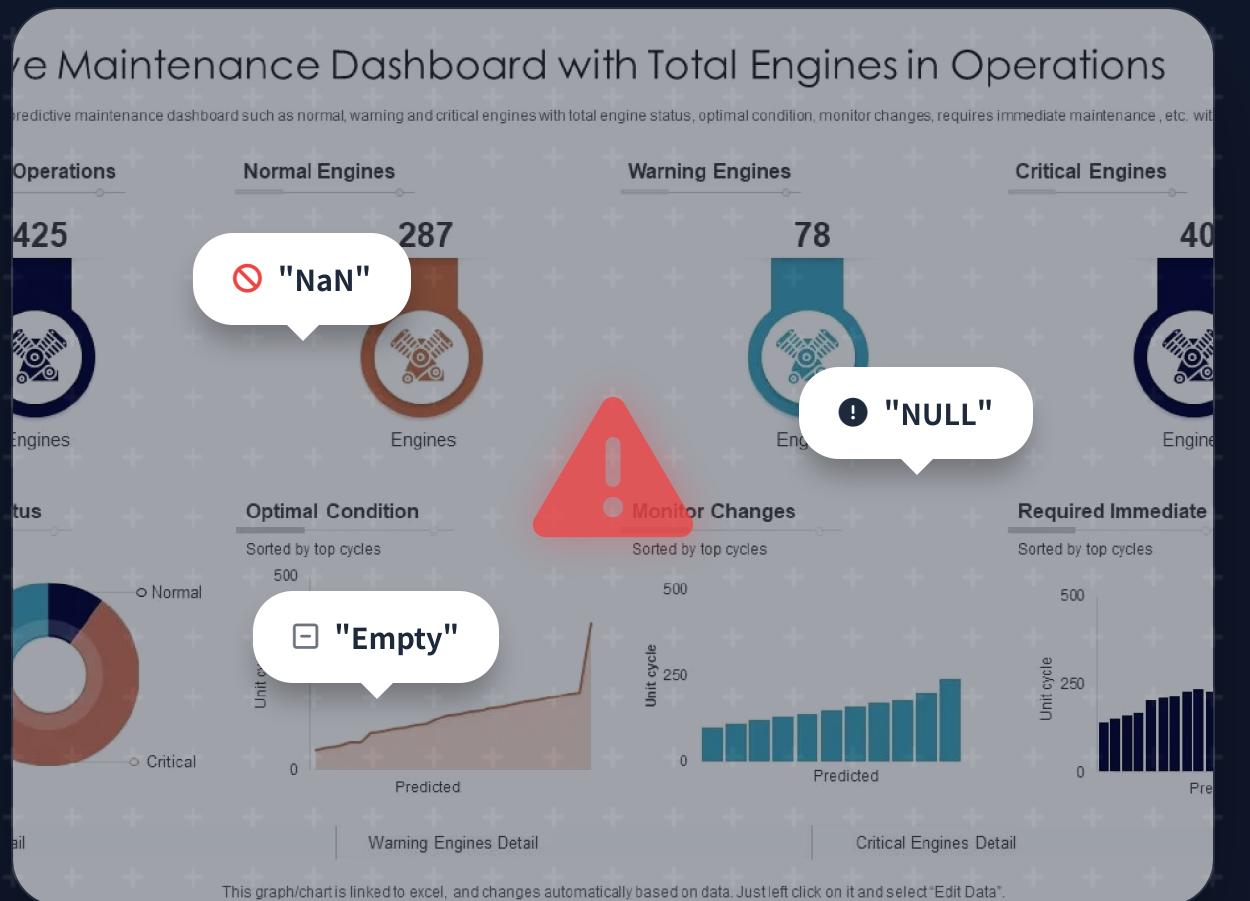
Data Overview

총 10,000개의 센서 샘플이 로드되었습니다.

주요 변수: 온도(Temp), 회전속도(RPM), 토크(Torque), 마모(Wear)

결측치(Missing Value) 처리 전략

데이터의 흐름이 끊기면 분석도 멈춘다



PROBLEM SITUATION

Data Discontinuity

- ⚡ **발생 원인:** 센서 고장, 네트워크 패킷 손실, 전원 차단 등 물리적 이슈
- ✖ **분석 방해:** 대부분의 머신러닝 알고리즘은 NaN 값을 입력받으면 에러 발생
- ⌚ **시계열 특성:** 단순 삭제 시 시간 간격(Time Step)이 깨져 패턴 학습 불가



STRATEGY

Handling Methods

상황에 따라 삭제(Drop)하거나 값을 채워넣는 대치(Imputation)를 선택합니다.



Drop

데이터 충분할 때



Mean/Median

정적 데이터용



Interpolate

시계열 추천

Noise Filtering 노이즈 필터링

Moving Average (이동 평균) 기법의 원리와 적용



Problem: Signal Noise

전기적 간섭이나 미세 진동으로 데이터가 지글거리는 현상.

순간적인 변화와 실제 추세를 구분하기 어렵게 만듦



Solution: Moving Average

최근 N개의 데이터를 평균 내어 부드럽게 만듦.

Formula: $MA_t = (x_t + \dots + x_{t-n}) / n$

※ Window Size (구간 크기) 설정 Tip

Window가 클수록 그래프는 부드러워지지만, 실제 신호보다 반응이 늦어지는
지연(Lag) 현상이 발생합니다.

* 보통 5~10 구간부터 테스트하며 최적값을 찾습니다.



🐍 [Code] 전처리 실습: 결측치 & 이동평균

fillna()와 rolling()을 활용하여 데이터의 구멍을 메우고 노이즈를 제거합니다.

● ● ● ➡ preprocessing_practice.py

```
import numpy as np

# 1. Create artificial missing values (연습용 결측치 생성)
df.loc[10:20, 'Air temperature [K]'] = np.nan
print("Missing Values:", df.isnull().sum())

# 2. Imputation (Forward Fill)
# 시계열 데이터는 직전 값으로 채우는 ffill이 유리함
df_filled = df.fillna(method='ffill')

# 3. Noise Filtering (Moving Average)
# 공정 온도 데이터를 10개 단위로 평균내어 스무딩
df_filled['Process temperature_smooth'] = (
    df_filled['Process temperature [K]'].rolling(window=10).mean()
)

# 결과 확인 (상위 50개만 시각화용 데이터로 추출)
subset = df_filled[['Process temperature [K]', 'Process
temperature_smooth']].head(50)
print(subset.tail())
```

Filtering Result

● Original (Noisy) ● Smoothed (MA)



상관관계(Correlation) 분석

Understanding Sensor Relationships



통계적 의미 (Data)

Relationship between variables

두 변수가 서로 얼마나 밀접하게 관련되어 있는지 -1에서 1 사이의 수치로 표현.

- ↗ 양의 상관관계 (+1): 온도가 오르면 압력도 오름
- ↘ 음의 상관관계 (-1): 속도가 빠르면 힘은 줄어듦
- = 관계 없음 (0): 서로 아무런 영향이 없음



물리적 의미 (Physics)

Domain Knowledge Check

데이터 분석 결과가 실제 물리 법칙과 일치하는지 검증하는 도구.

- ✓ 상식 검증: RPM(속도)과 Torque(힘)는 반비례
- ✓ 물리 공식: $T = \tau \times \omega$ (トル = 토크 × 각속도)
- ✓ 만약 둘 다 같이 오르면? 센서 오류일 가능성!

주의사항

CORRELATION

상관관계

"A와 B가 같이 움직인다"

≠

CAUSATION

인과관계

"A 때문에 B가 변했다"

"상관관계가 높다고 해서 반드시 원인과 결과는 아닙니다."

히트맵(Heatmap) 시각화

"숫자 뭉치보다는 색상(Color)으로 관계를 직관적으로 파악합니다."

1. 숫자형 컬럼 선택



상관계수(.corr)는 수치형 데이터끼리만 계산 가능.
문자열 컬럼 제외 필수.

2. 색상 스케일 고정



vmin=-1, vmax=1 설정을 통해
색상 왜곡 없이 정확한 강도 비교.

3. 수치 표기 (Annotation)



annot=True, fmt='.{2f}'
색상 위에 실제 상관계수 값을 소수점 2자리로 표기.

4. 해석 (Interpretation)



- 1에 가까움: 양의 상관 (같이 증가)
- 1에 가까움: 음의 상관 (반대로 움직임)

Sensor Correlation Matrix

Method: Pearson

Red: Positive Correlation
Target: Multi-collinearity
Blue: Negative Correlation

	Air Temp	Proc Temp	RPM	Torque	Tool Wear
Air Temp	1.00	0.87	-0.02	-0.01	0.01
Proc Temp	0.87	1.00	-0.03	-0.02	0.02
RPM	-0.02	-0.03	1.00	-0.87	0.00
Torque	-0.01	-0.02	-0.87	1.00	0.02
Tool Wear	0.01	0.02	0.00	0.02	1.00

🐍 Correlation Heatmap Practice

seaborn을 활용하여 변수 간의 상관관계를 시각적으로 확인합니다.



heatmap_viz.py

```
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Select Numerical Columns
cols = ['Air temperature [K]', 'Process temperature [K]',
        'Rotational speed [rpm]', 'Torque [Nm]',
        'Tool wear [min]']

# 2. Calculate Correlation Matrix
corr = df[cols].corr()

# 3. Plot Heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr,
            annot=True, # Show numbers
            cmap='coolwarm', # Red-Blue color map
            vmin=-1, vmax=1, # Fix scale range
            fmt='.2f') # 2 decimal places

plt.title('Sensor Correlation Heatmap')
plt.show()
```

Output Plot

● Positive ● Negative

	Air Temp	Proc Temp	RPM	Torque	Wear
Air Temp	1.00	0.87	0.02	-0.01	0.01
Proc Temp	0.87	1.00	0.01	-0.02	0.01
RPM	0.02	0.01	1.00	-0.87	0.00
Torque	-0.01	-0.02	-0.87	1.00	0.00
Wear	0.01	0.01	0.00	0.00	1.00

Key Findings

- 1. Air Temp ↔ Proc Temp (0.87): 강한 양의 상관관계
- 2. RPM ↔ Torque (-0.87): 강한 음의 상관관계 (물리 법칙 일치)

NEXT STEP PREVIEW

LLM으로 여는 제조 AI의 새 지평

오늘 닦은 데이터 (Data) → 내일의 AI 연료 (Fuel)



설비 매뉴얼 QA 시스템

복잡한 매뉴얼을 LLM이 학습.
"에러코드 E04가 뭐야?"라고 물으면
즉시 해결책을 답변하는 챗봇.



자연어 리포트 생성

오늘 실습한 센서 데이터를 분석하여
"오후 2시부터 공정 온도가 상승세임"
같은 요약 보고서를 자동 작성.



이상 원인 추론 & 제안

단순한 고장 알림을 넘어,
"베어링 마모 가능성 80%,
교체 필요"와 같이 조치를 제안.

KEYWORDS:

#LangChain

#Vector_DB

#Prompt_Engineering

#RAG

Tomorrow

