

기계 설비 데이터 토이 프로젝트



목차



- 분류 문제
 - 데이터 전처리
 - 데이터 EDA
 - 모델 성능 평가
 - 한계점/보완점

- 시계열 예측 문제
 - 데이터 전처리
 - 모델 성능 평가
 - 한계점/보완점

분류 문제

분류 문제

데이터 탐색 및 전처리

- 데이터 발전소에서는 터빈과 보일러에 부착된 센서를 통해 시간 단위로 주요 지표를 수집
- 결측치 Temperature 70, Vibration 60, GasFlow 70 총 200개 결측치 발견 → 단순 평균 대체로 대처함

	Timestamp	UnitID	Temperature	Pressure	Vibration	GasFlow	SteamOutput	Failure
0	2024-01-01 0:00	Boiler_1	565.417235	147.464665	0.033765	59.297265	193.546471	0
1	2024-01-01 1:00	Boiler_2	555.387453	176.458303	0.032691	80.411795	202.970620	0
2	2024-01-01 2:00	Turbine_A	560.518903	173.007811	0.007250	85.136752	176.899403	0
3	2024-01-01 3:00	Boiler_1	564.675614	152.539684	0.024761	73.385121	215.198257	0
4	2024-01-01 4:00	Boiler_1	569.041644	168.646770	0.054248	80.746280	180.062658	0
...
4363	2024-06-30 19:00	Turbine_A	571.673410	141.246760	0.030002	94.776929	215.427889	0
4364	2024-06-30 20:00	Boiler_1	539.041574	153.662562	0.003924	77.142353	181.032918	0
4365	2024-06-30 21:00	Turbine_A	621.639031	175.415781	0.024623	68.837098	183.814482	0
4366	2024-06-30 22:00	Turbine_A	555.402131	164.732844	0.027001	82.749308	164.403537	0
4367	2024-06-30 23:00	Turbine_A	541.812000	159.282116	0.041144	82.044452	204.287044	0

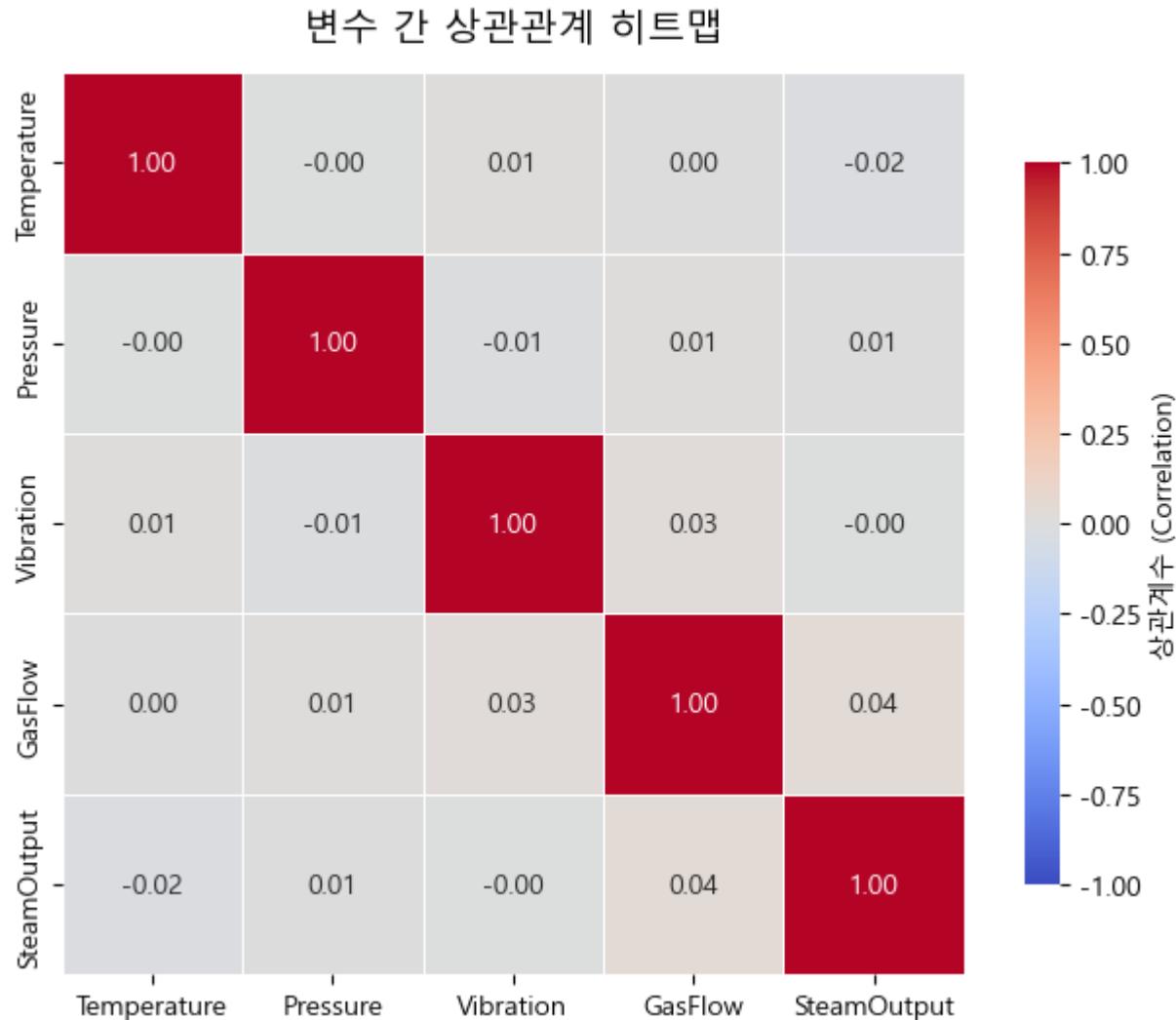
컬럼명	설명
Timestamp	시간 (1시간 단위)
UnitID	설비 ID (Turbine_A, Boiler_1 등)
Temperature	설비 내부 온도 (°C)
Pressure	내부 압력 (bar)
Vibration	진동 수치 (mm/s)
GasFlow	연료 유량 (m³/h)
SteamOutput	증기 생산량 (ton/h)
Failure	고장 여부 (1=고장, 0=정상)

분류 문제

데이터 탐색 및 전처리

- 독립 변수들 간의 상관관계 (Temperature, Pressure, Vibration, GasFlow, SteamOutput)

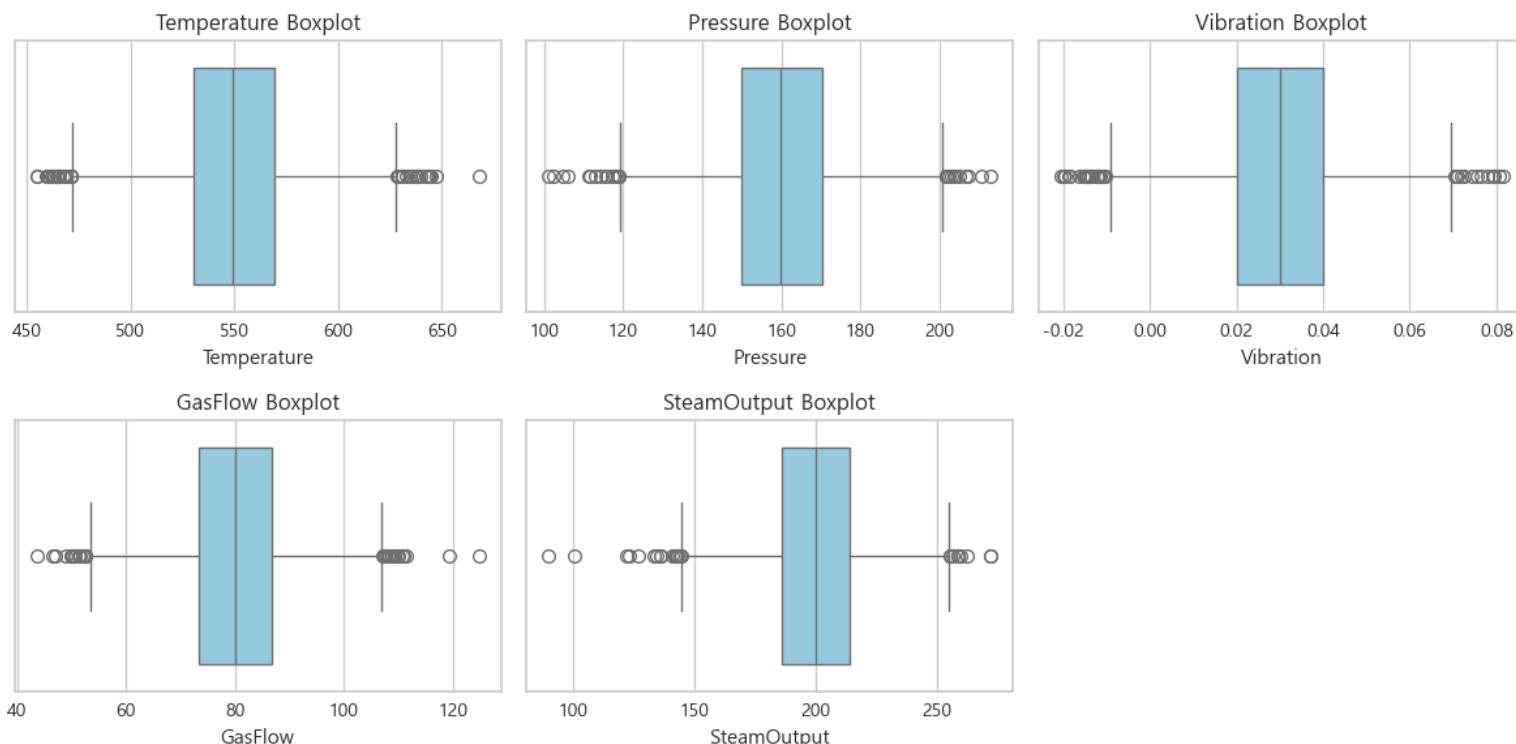
- 변수들 상관 관계 X



분류 문제

데이터 탐색 및 전처리

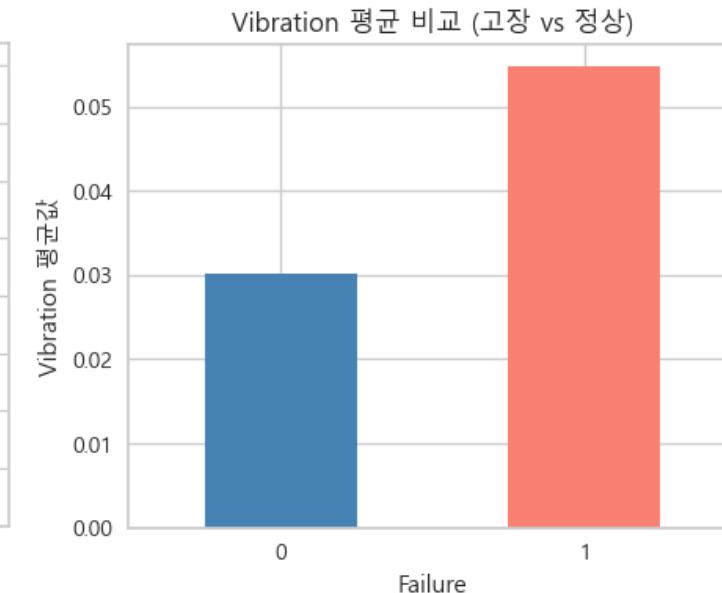
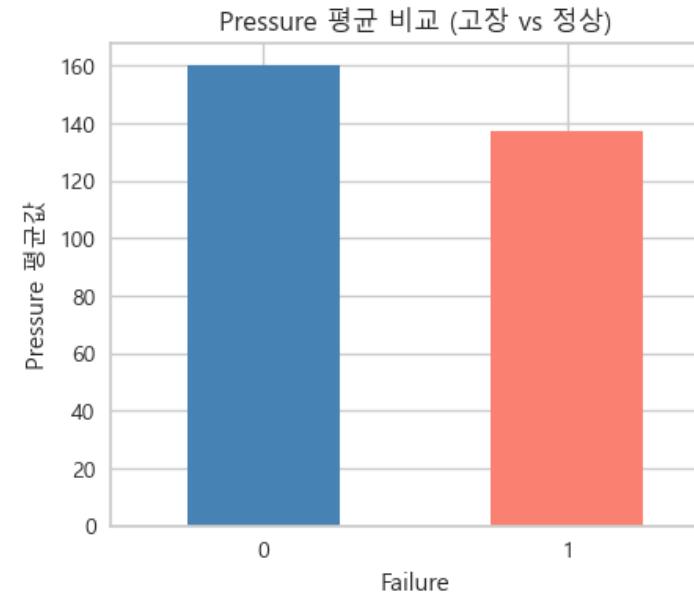
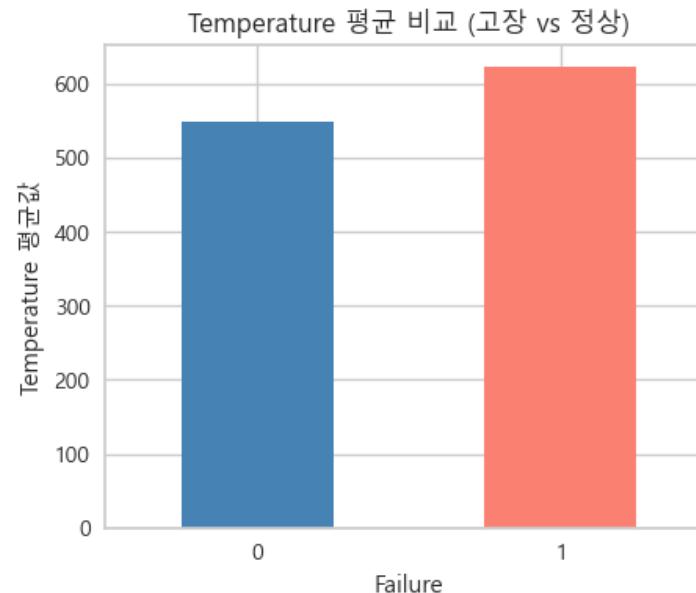
- 데이터 탐색 했을 때 데이터 불균형이 있다는 것을 확인함
- Failure 고장 여부 (1=고장, 0=정상) 전체 4368 개 중에서 고장으로 라벨링 된 데이터가 3개 인 것을 확인
- 고장이 발생한 장치에서 데이터를 살펴 봤을 때 **온도와 진동** 수치는 정상 보다 높게 나오고 **증기**가 적게 나오는 것으로 판단 (고장 데이터가 적어서 불확실)



분류 문제

데이터 탐색 및 전처리

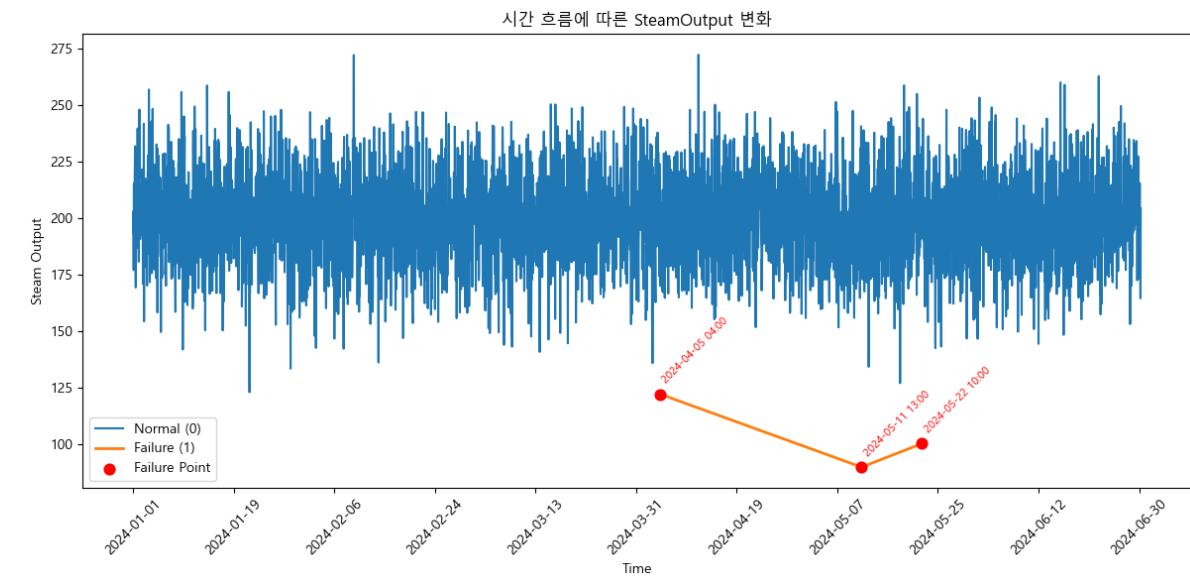
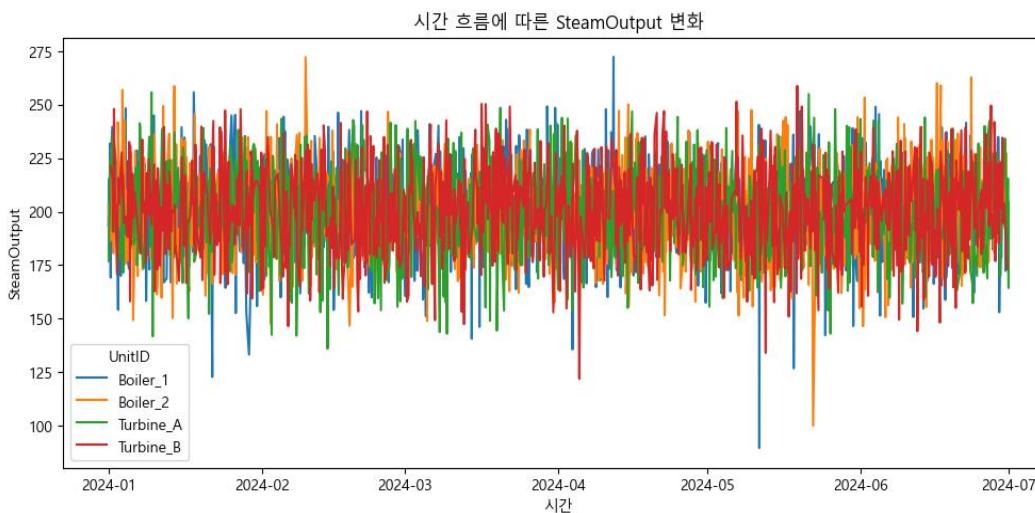
- 고장(1) vs 정상(0) 평균 비교 (고장 데이터가 적어서 불확실)
- 고장이 나면 온도와 진동이 증가함



분류 문제

데이터 탐색 및 전처리

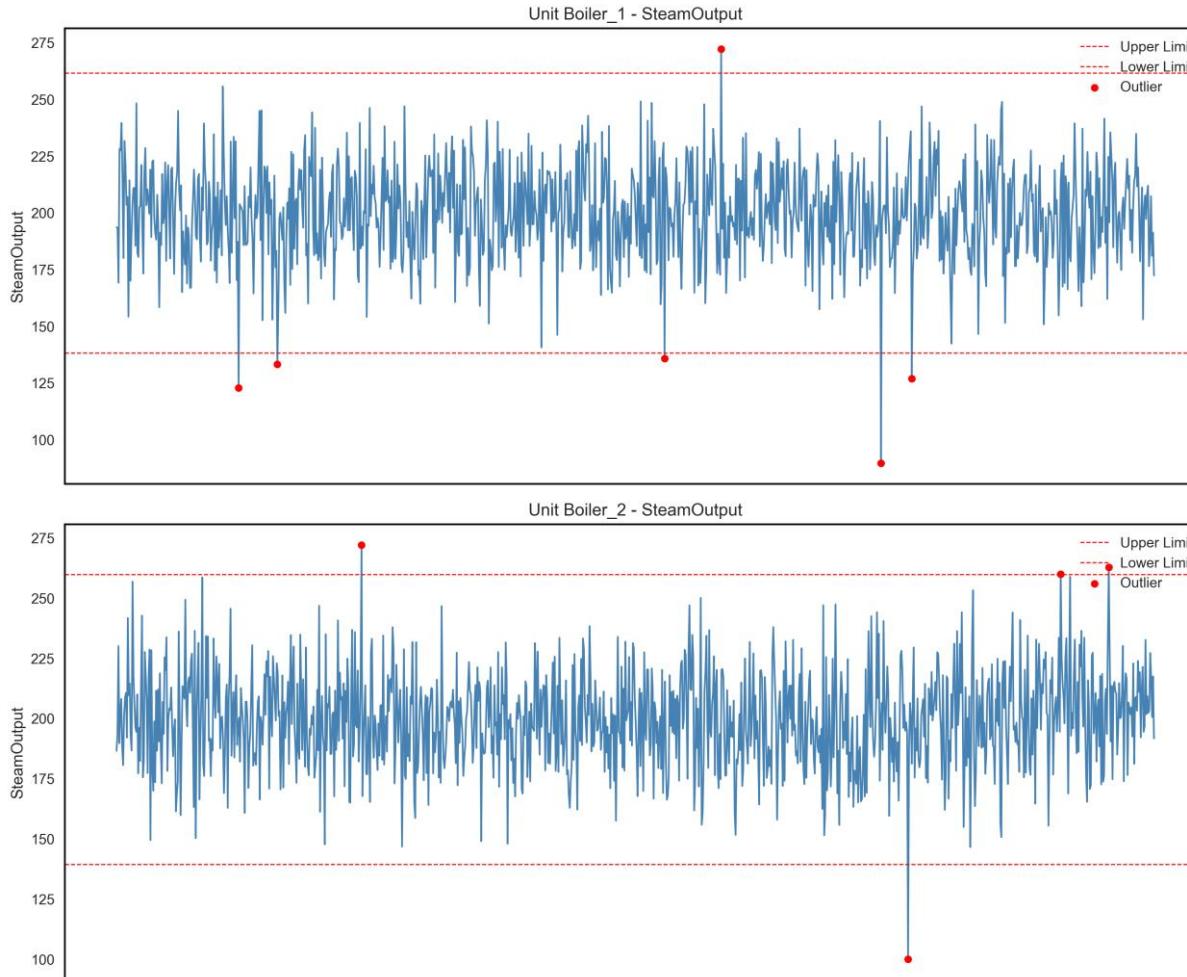
- SteamOutput 설비 ID 별 변화 추이
- 통계적 기준을 초과한 지점이 다수 발생 → Boiler 장비가 Turbine 장비보다 통계적인 기준 범위를 벗어난 값들이 많음
- 이는 센서 노이즈, 운영 조건의 변화, 또는 장비의 이상 징후 가능성을 시사
- 단순 통계 기준 이상치이므로, 실제 고장과의 직접적 연관성은 추가 분석이 필요



분류 문제

데이터 탐색 및 전처리

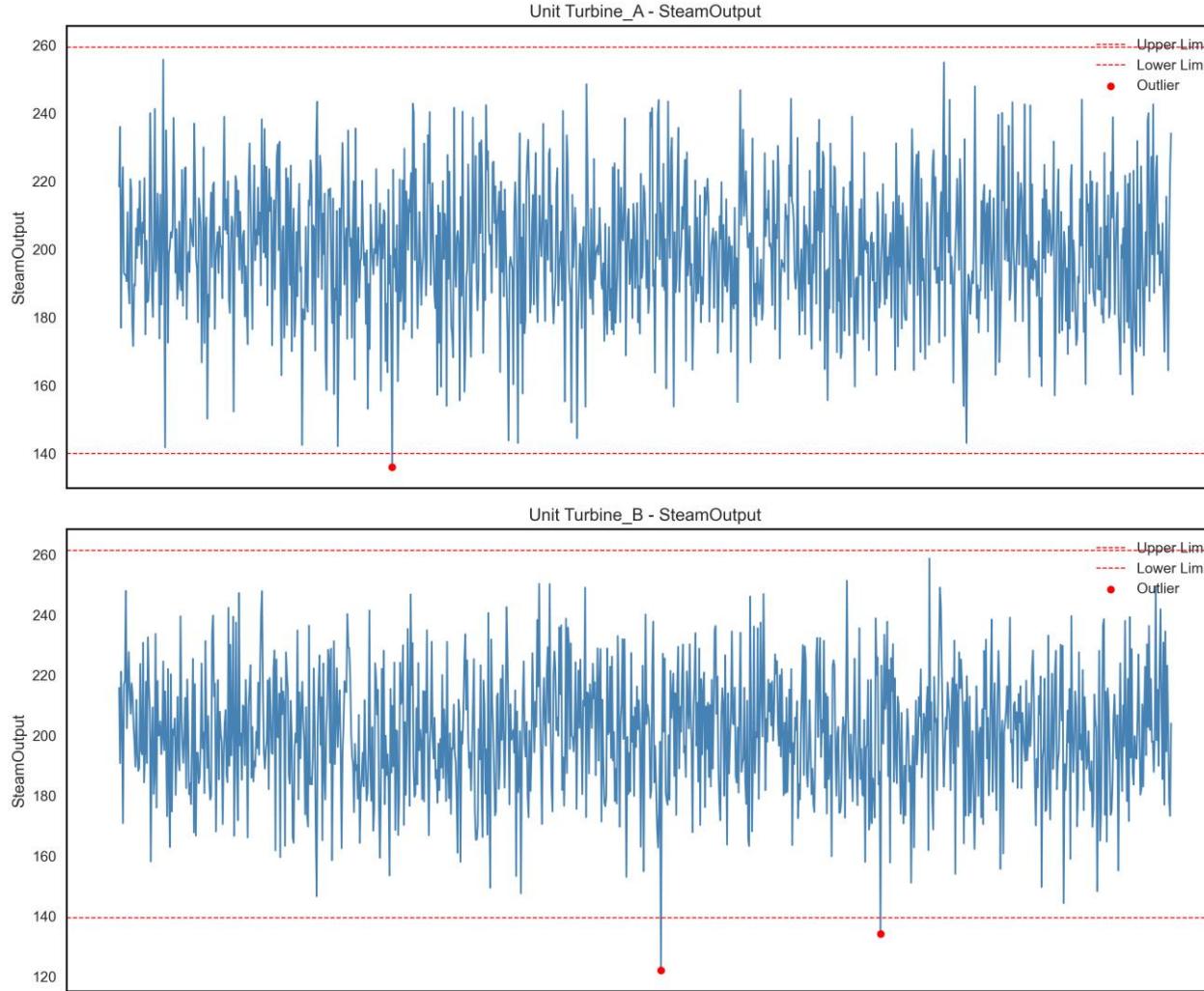
- Boiler 장비 – 평균에서 $\pm 3\sigma$ 준편차 범위 안에 전체 데이터의 약 99.7% 벗어난 데이터가 3개 이상



분류 문제

데이터 탐색 및 전처리

- Turbine 장비 – 평균에서 $\pm 3\sigma$ 준편차 범위 안에 전체 데이터의 약 99.7% 벗어난 데이터가 3개 이하



분류 문제

분류 모델 적용

- 데이터가 불균형이 있기 때문에 일반적인 분류 모델을 적용하면 정상 데이터만 정확도 좋을 것으로 예상
- 독립 변수 : 온도, 압력, 진동, 연료 유량, 증기 생산량 종속 변수 : 고장 여부
- 간단한 딥러닝 모델을 구성해서 훈련 진행
- 훈련 데이터와 테스트 데이터 8:2 batch 16 epochs 50

```
FailurePredictor(  
    model): Sequential(  
        (0): Linear(in_features=5, out_features=64, bias=True)  
        (1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (2): ReLU()  
        (3): Dropout(p=0.3, inplace=False)  
        (4): Linear(in_features=64, out_features=128, bias=True)  
        (5): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (6): ReLU()  
        (7): Dropout(p=0.3, inplace=False)  
        (8): Linear(in_features=128, out_features=64, bias=True)  
        (9): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (10): ReLU()  
        (11): Dropout(p=0.2, inplace=False)  
        (12): Linear(in_features=64, out_features=1, bias=True)  
        (13): Sigmoid()  
    )  
)
```

분류 문제

분류 모델 적용

- Test 데이터에 성능 평가
- 정상 데이터는 완벽하게 맞춤 하지만 고장 데이터는 1 건 있는데 놓침
- 정상 데이터에 대해서는 성능이 좋음 → 고장데이터를 분류를 못하기 때문에 일반적인 분류 모델을 적용하면 X

```
혼동행렬:
```

```
[[873  0]
 [  1  0]]
```

```
분류 리포트:
```

	precision	recall	f1-score	support
0.0	0.999	1.000	0.999	873.000
1.0	0.000	0.000	0.000	1.000
accuracy	0.999	0.999	0.999	0.999
macro avg	0.499	0.500	0.500	874.000
weighted avg	0.998	0.999	0.998	874.000

인코더-디코더 모델 적용

- 문제상황 : 전체 4368개 데이터 중 고장 데이터가 단 3 개 → 일반적인 분류 모델로 성능을 평가 어려움
- 정상 데이터만 학습하는 인코더 기반 이상 탐지 모델을 적용하여 재구성 오차(Reconstruction Error)가 큰 데이터를 고장이라고 판단

```
AE(  
    encoder: Sequential(  
        (0): Linear(in_features=5, out_features=64, bias=True)  
        (1): ReLU()  
        (2): Linear(in_features=64, out_features=128, bias=True)  
        (3): ReLU()  
        (4): Linear(in_features=128, out_features=3, bias=True)  
        (5): ReLU()  
    )  
    decoder: Sequential(  
        (0): Linear(in_features=3, out_features=128, bias=True)  
        (1): ReLU()  
        (2): Linear(in_features=128, out_features=64, bias=True)  
        (3): ReLU()  
        (4): Linear(in_features=64, out_features=5, bias=True)  
    )  
)
```

분류 문제

인코더-디코더 모델 적용

- 고장 탐지 기준
- Reconstruction Error = $(\text{입력} - \text{복원값})^2$ 평균
- Threshold: 정상데이터의 평균 + $3 \times \text{표준편차}$
 - 오차 > 기준선 → 고장(1)
 - 오차 ≤ 기준선 → 정상(0)
- 고장 데이터를 잘 분류 했지만 정상 데이터 93개를 고장이라고 판단함 → 정상 데이터를 잘못 분류 하는것 도 문제

혼동행렬:

```
[[4272  93]
 [  0   3]]
```

분류 리포트:

	precision	recall	f1-score	support
0	1.000	0.979	0.989	4365.000
1	0.031	1.000	0.061	3.000
accuracy	0.979	0.979	0.979	0.979
macro avg	0.516	0.989	0.525	4368.000
weighted avg	0.999	0.979	0.989	4368.000

분류 문제

인코더-디코더 모델 적용

- Threshold(임계값) ROC/Youden Index 기준
- Threshold: Youden Index = TPR-FPR 최대값 정상과 고장을 잘 구분할 수 있는 기준점
 - 오차 > 기준선 → 고장(1)
 - 오차 ≤ 기준선 → 정상(0)
- 고장 데이터를 하나 분류를 못했지만 정상 데이터는 잘 분류함

```
ROC 기반 Threshold 혼동행렬:
```

```
[[4365    0]
 [   1    2]]
```

```
ROC 기반 Threshold 분류 리포트:
```

	precision	recall	f1-score	support
0	1.0	1.000	1.0	4365.0
1	1.0	0.667	0.8	3.0
accuracy	1.0	1.000	1.0	1.0
macro avg	1.0	0.833	0.9	4368.0
weighted avg	1.0	1.000	1.0	4368.0

한계점/보완점

극단적 데이터 불균형

- 고장 데이터가 거의 없어서 모델이 정상 패턴에만 학습함
- 데이터 증강으로 데이터 생성

Threshold 민감성

- 통계적 기준, ROC 기반 기준을 간단하게 정했지만 많은 기준이 있고 기준을 선택한 이유가 명확하지 않음

시계열 예측 문제

데이터 탐색 및 전처리

- 발전소 센서 데이터를 이용해 SteamOutput 예측
- LSTM 레이어를 사용해서 모델 설계함 → 시간에 따른 의존성문제를 잡을 수 있는 LSTM 레이어를 사용(간단하게 레이어만 쌓아서 모델 만듬)
- 일반적인 Full Connected 레이어만 사용하면 시간 순서 정보를 반영할 수 없음
- 두 번째 LSTM: 첫 LSTM이 추출한 패턴을 학습, 장기 의존성 강화

```
DeepLSTM(  
    (lstm1): LSTM(5, 128, num_layers=2, batch_first=True, dropout=0.3)  
    (lstm2): LSTM(128, 128, batch_first=True, dropout=0.3)  
    (fc1): Linear(in_features=128, out_features=64, bias=True)  
    (fc2): Linear(in_features=64, out_features=1, bias=True)  
    (relu): ReLU()  
)
```

시계열 예측 문제

데이터 탐색 및 전처리

- 독립 변수: Temperature, Pressure, Vibration, GasFlow, SteamOutput
- 종속 변수: T+1 ~ T+6 시점 SteamOutput
- 스케일링 StandardScaler 적용
- 시퀀스 생성 seq_len = 6
- 훈련 데이터와 테스트 데이터 8:2 batch 16 epochs 100

시계열 예측 문제

LSTM 모델 적용

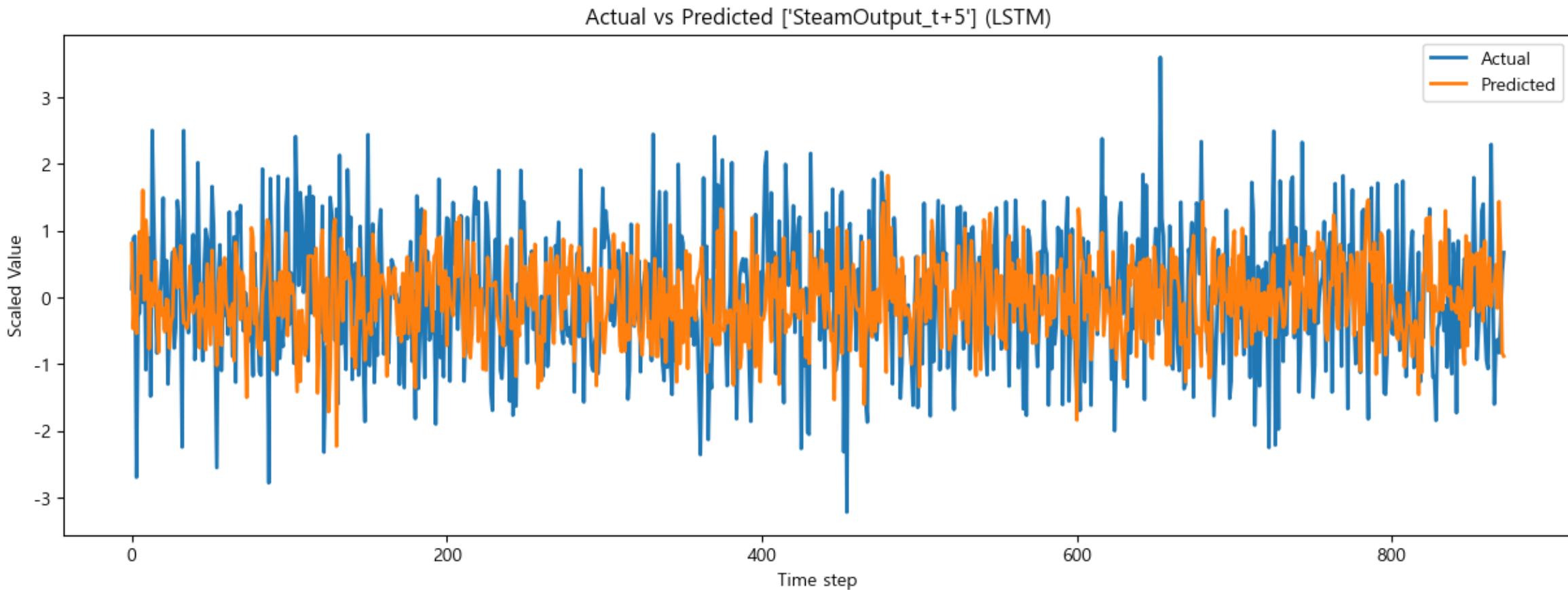
- SteamOutput_t+1 ~ SteamOutput_t+6 모델 6가지를 만들어서 성능 비교
- MSE 지표를 사용해서 모델을 각 평가함
- 모델 T+5 시점을 종속변수로 둔 모델이 성능이 제일 우수함

종속 변수에 따른 모델	MSE / 정규화 이전 값 MSE
모델 T+1	1.397 / 562.31
모델 T+2	1.43 / 575.33
모델 T+3	1.385 / 557.36
모델 T+4	1.418 / 569.88
모델 T+5	1.308 / 525.15
모델 T+6	1.454 / 583.93

시계열 예측 문제

LSTM 모델 적용

□ SteamOutput_t+5



시계열 예측 문제

한계점/보완점

□ 시점 간 상관관계 미반영

- 시계열 데이터는 일반적으로 시간 순서에 따라 값이 상관관계를 갖음 → 개별 시점으로 예측했기 때문에 데이터의 패턴을 반영 못할 수도 있음

□ 모델 효율성 저하

- 각 시점을 독립적으로 예측하려면 모델을 여러 번 학습하거나, 각 시점마다 별도의 모델을 구성해야 함 이는 계산 비용과 학습 시간이 불필요하게 증가하게 만들고 또 한 동일한 데이터에서 반복 학습을 하므로 데이터 활용 효율성이 떨어짐