

NASA Turbo Fan Engine RUL Prediction

럭키비키*

김선경
김영민
이유정
이준현
정민지



CONTENTS

01 개요

02 프로젝트 주제

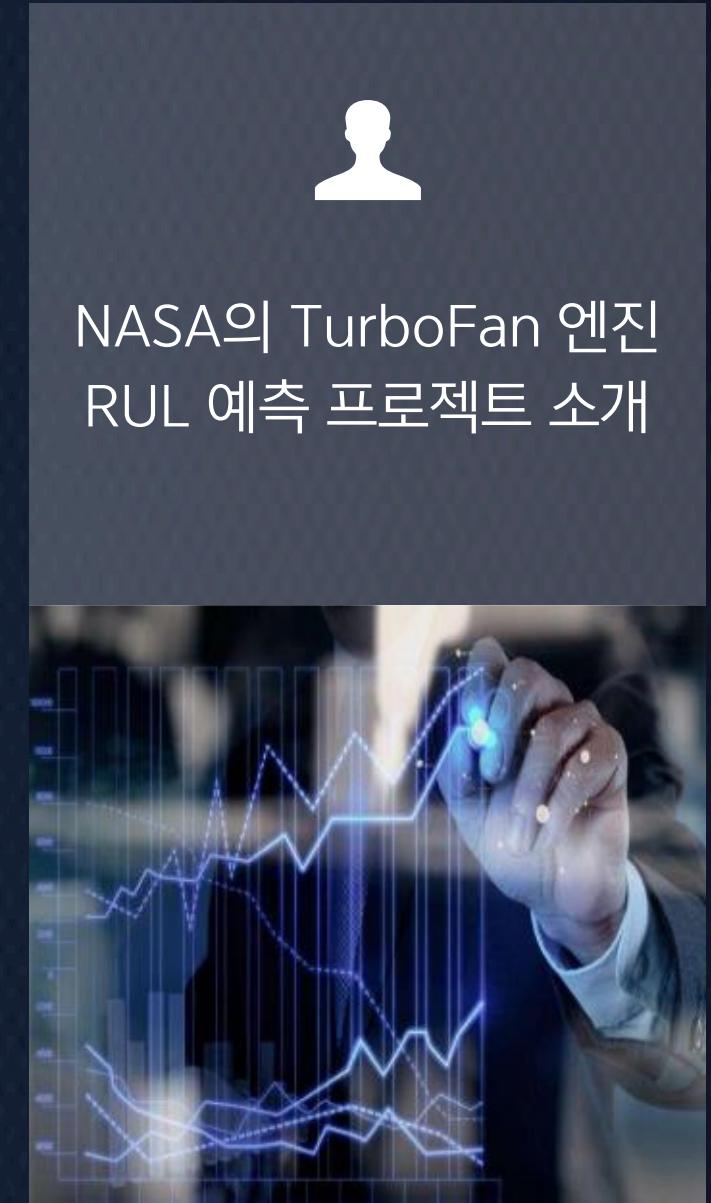
03 데이터 분석

04 모델 비교

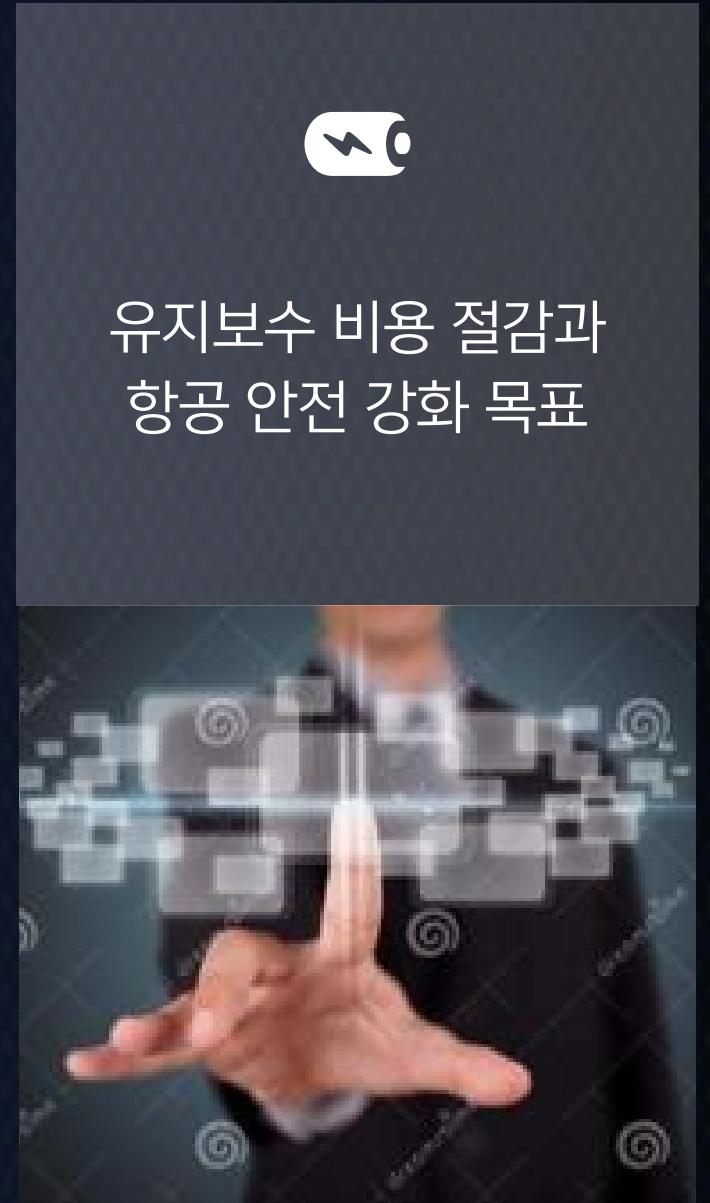
05 결과 및 결론

01 개요

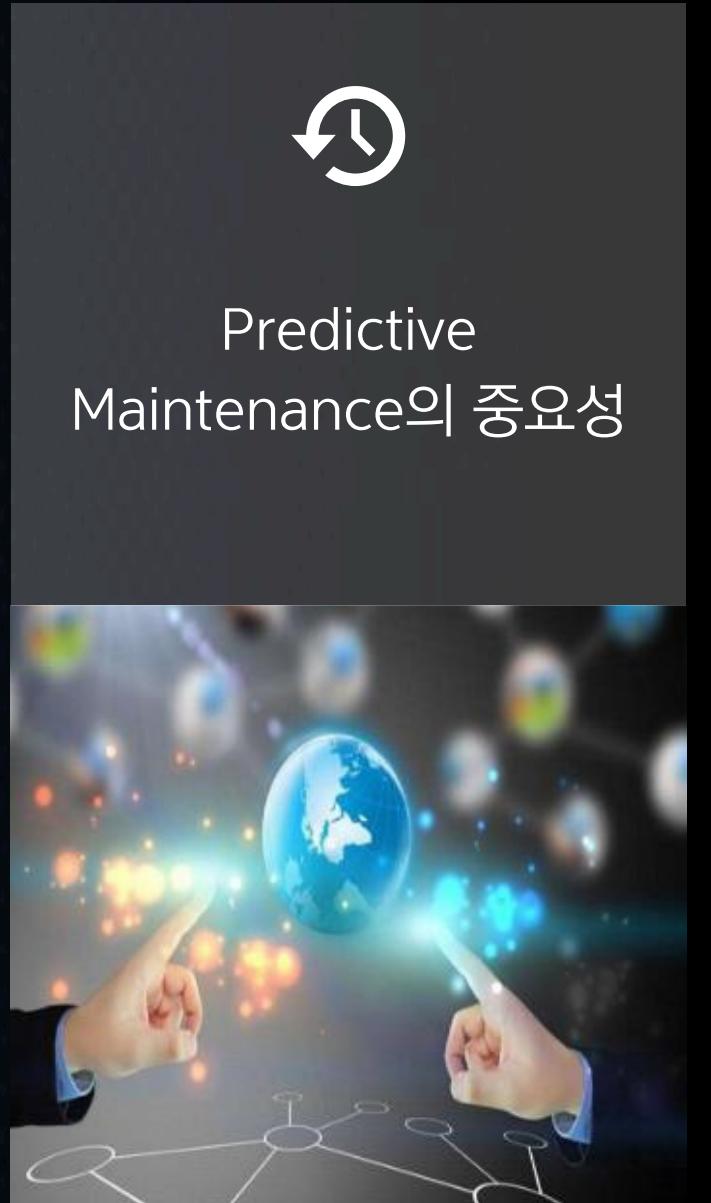
개요



NASA의 TurboFan 엔진
RUL 예측 프로젝트 소개



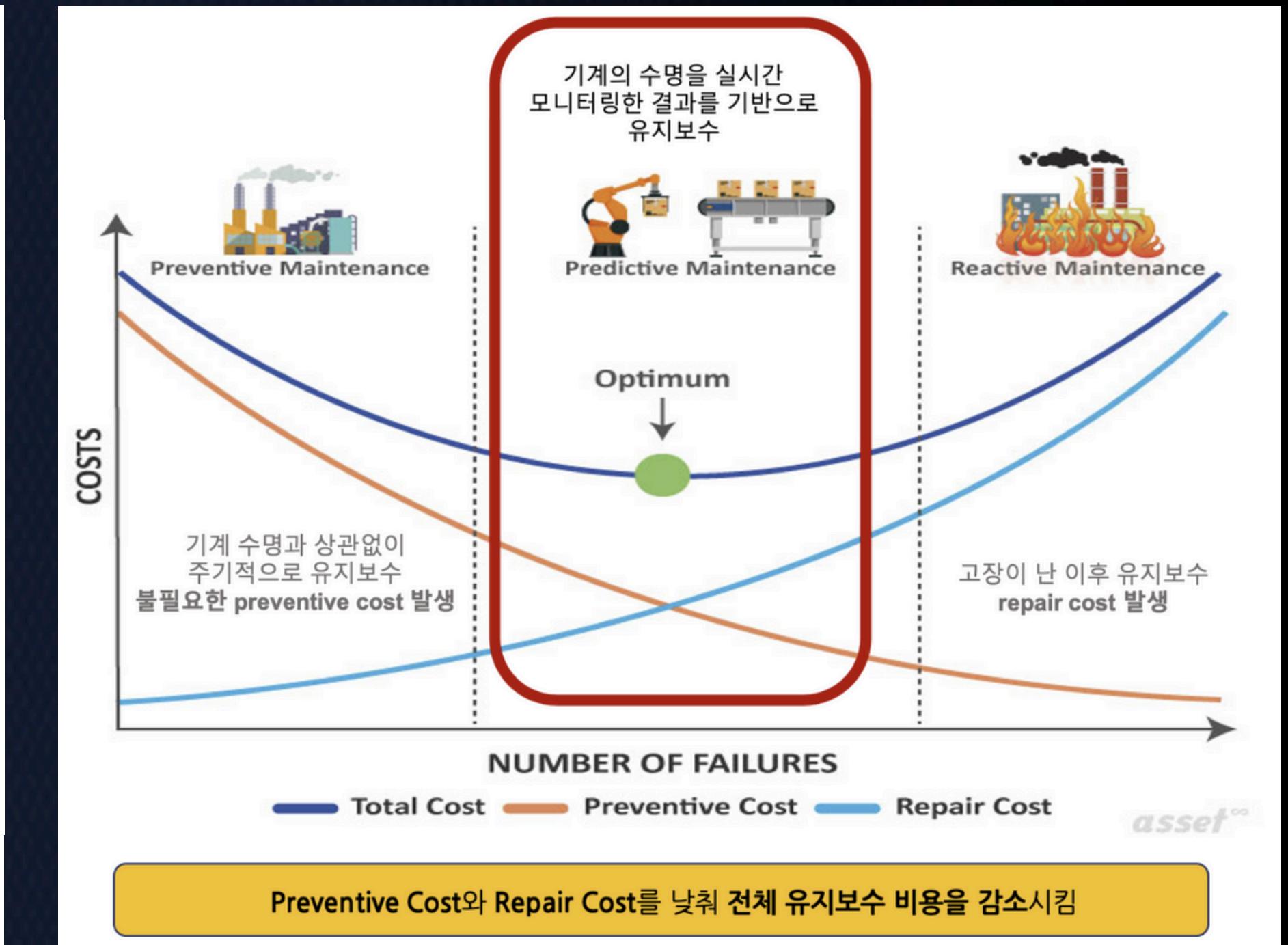
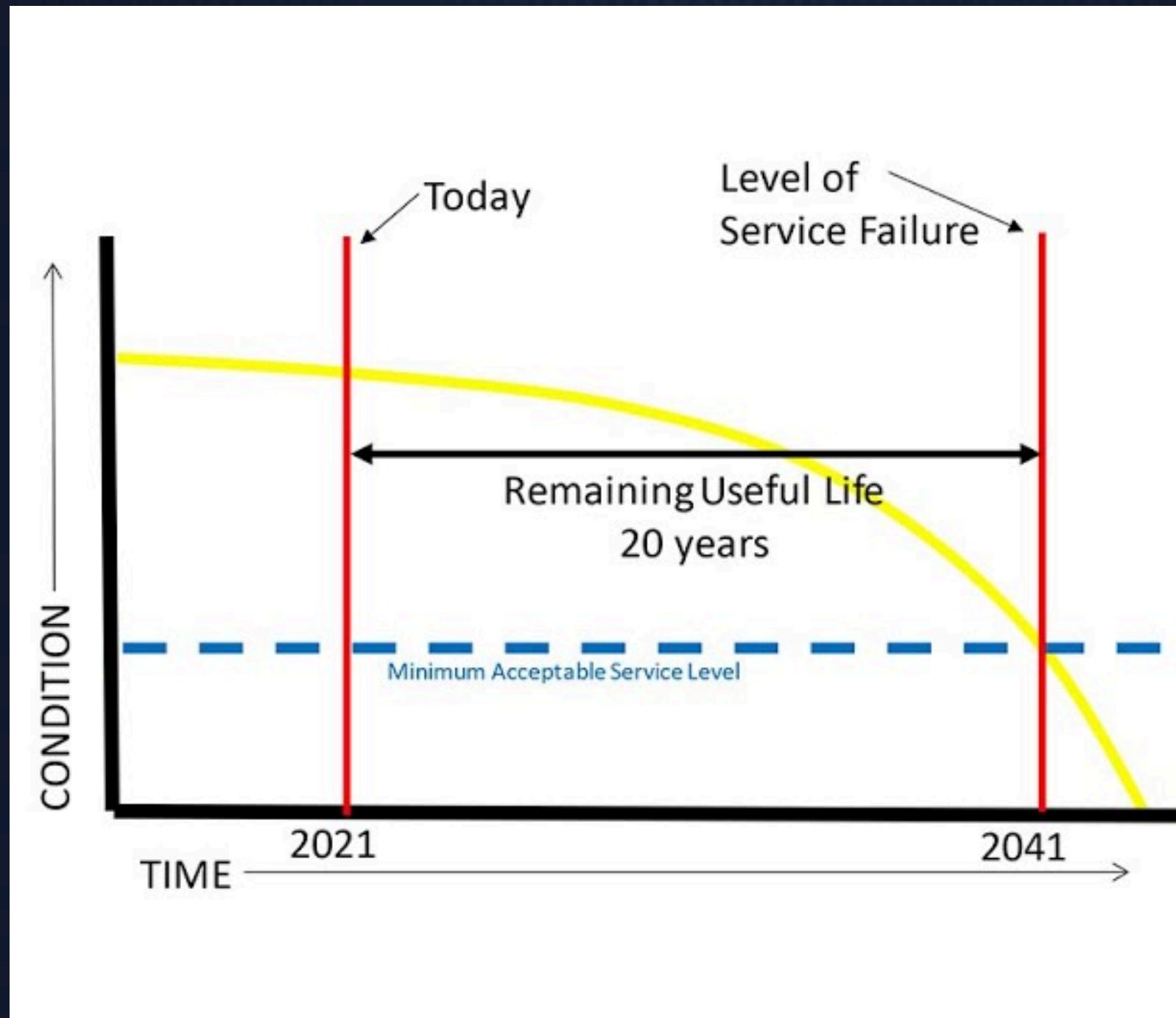
유지보수 비용 절감과
항공 안전 강화 목표



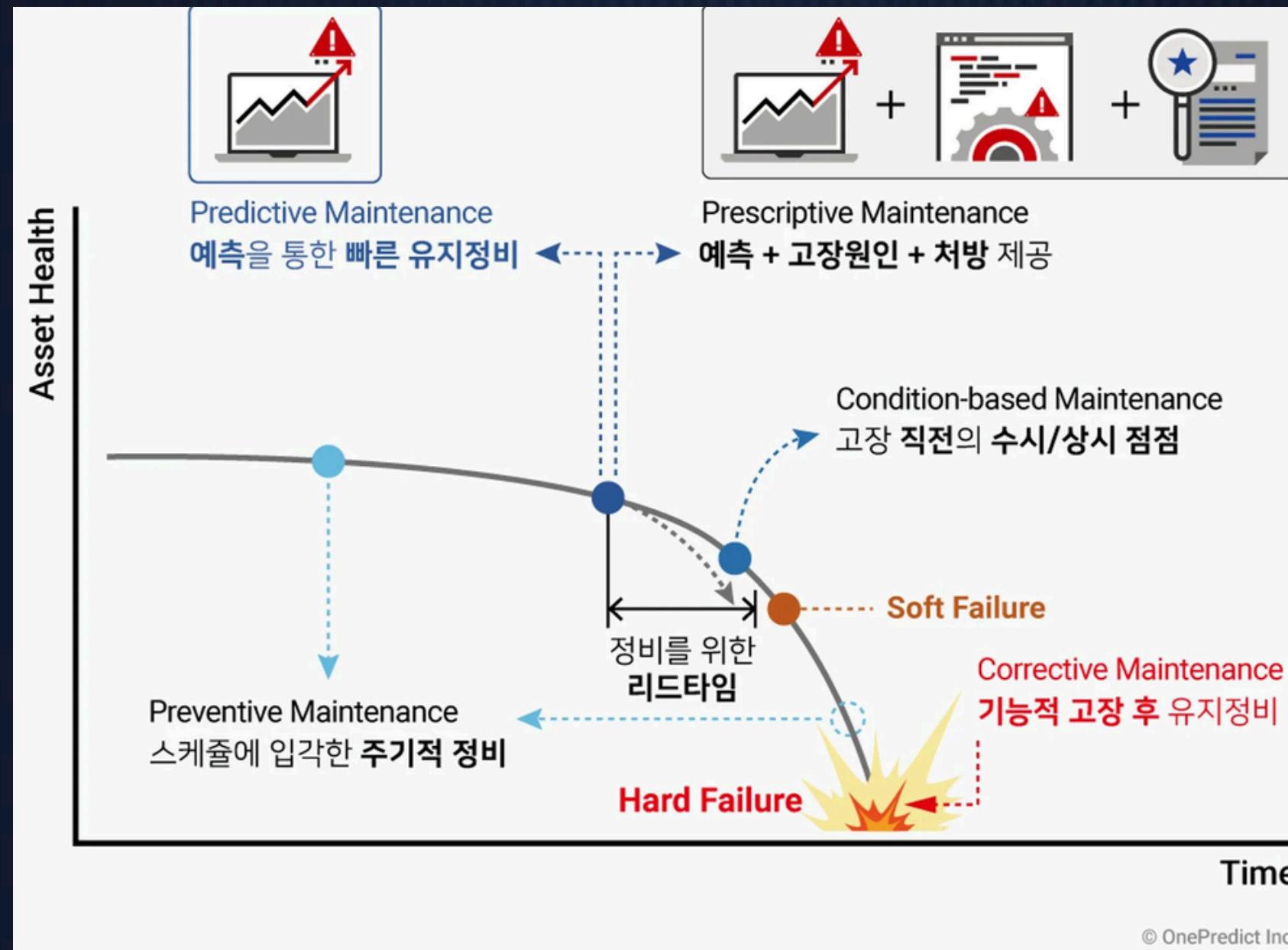
Predictive
Maintenance의 중요성

RUL (Remaining Useful Life)

기계의 수리, 교체가 필요하기 전까지 남아 있는 사용 시간 또는 예상 수명



터보팬 엔진 RUL 예측 유지 프로젝트

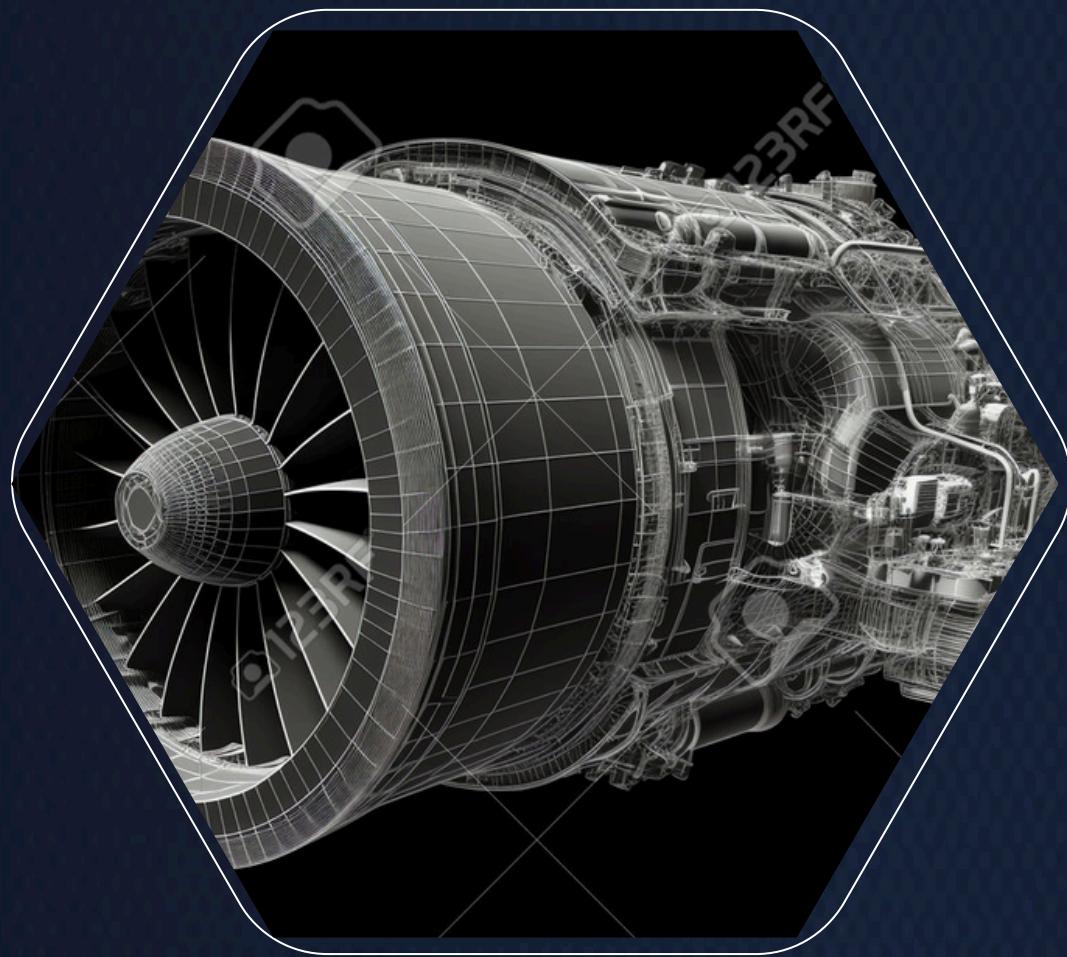


1 TurboFan 엔진의 남은 유용한 수명을 예측하기
위해 고안된 프로젝트

2 엔진 부품의 고장 시점(잔여 수명)을 정확하게 예측해
유지보수 **비용 절감 & 항공 안전 강화**

3 **Predictive Maintenance**
• 유지보수(maintenance)의 여러 단계 중 장치를
실시간으로 점검하여 이상징후 사전에 발견, 예방

배경



Turbo Fan Engine

- 1 TurboFan 엔진은 정기적인 유지보수와 정밀 검사가 필수적인 **고가의 장치**
- 2 엔진의 부품이 고장 나면 심각한 안전 문제와 **높은 수리 비용이 발생** → RUL을 예측하여, 고장 전에 사전 조치를 취할 수 있음
- 3 실시간 성능 데이터를 분석하여 정확한 RUL을 예측함으로써 막대한 비용 손실과 인력 낭비, 시간 지체 방지

03 프로젝트 주제

주제



센서 데이터를 활용한
RUL 예측



머신러닝 모델을 통한
엔진의 상태 평가 및 성능 저하 시점 예측

04 레이터 분석

데이터 전처리



결측치 확인

| Dataset | Has Null | Training Values | Test False | RUL False |
|---------|----------|-----------------|------------|-----------|
| | | | | |

모든 데이터에서 Null 값이 발견되지 않음
-> 결측치 처리 진행하지 않음



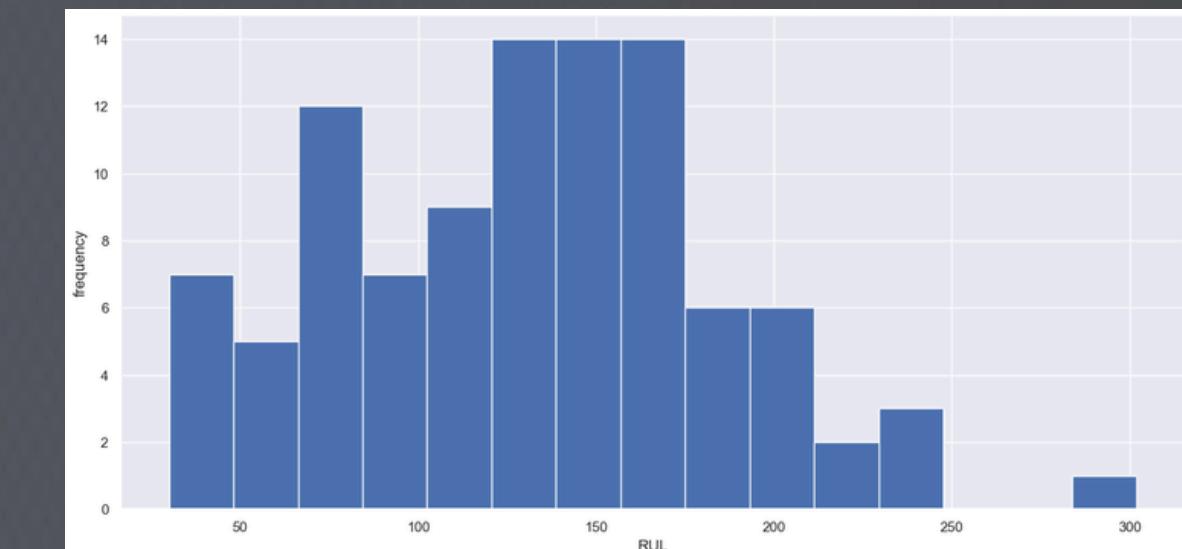
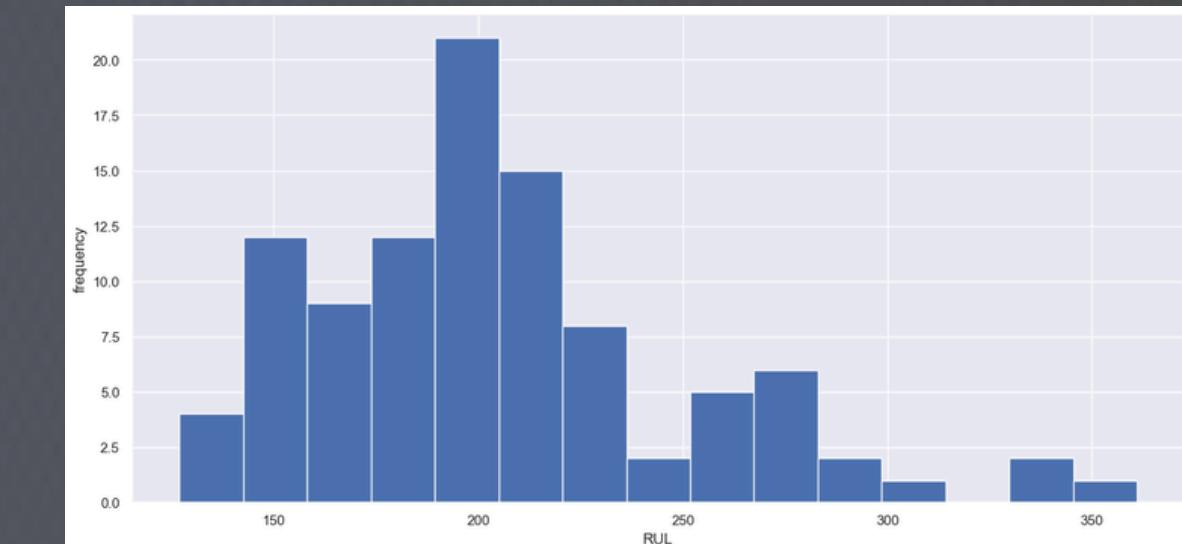
표준편차가 0인 경우

```
train_description = train.describe()  
  
zero_variance_columns = train_description.loc['std'] == 0  
zero_variance_columns_names = train_description.columns[zero_variance_columns].tolist()  
  
zero_variance_columns_names  
['setting_3', 's_18', 's_19']
```

1. 모든 데이터가 동일
 2. 데이터의 패턴을 인식할 수 없음
 3. 표준화 및 정규화 과정에서 계산이 불가능
- > 해당 열 제외

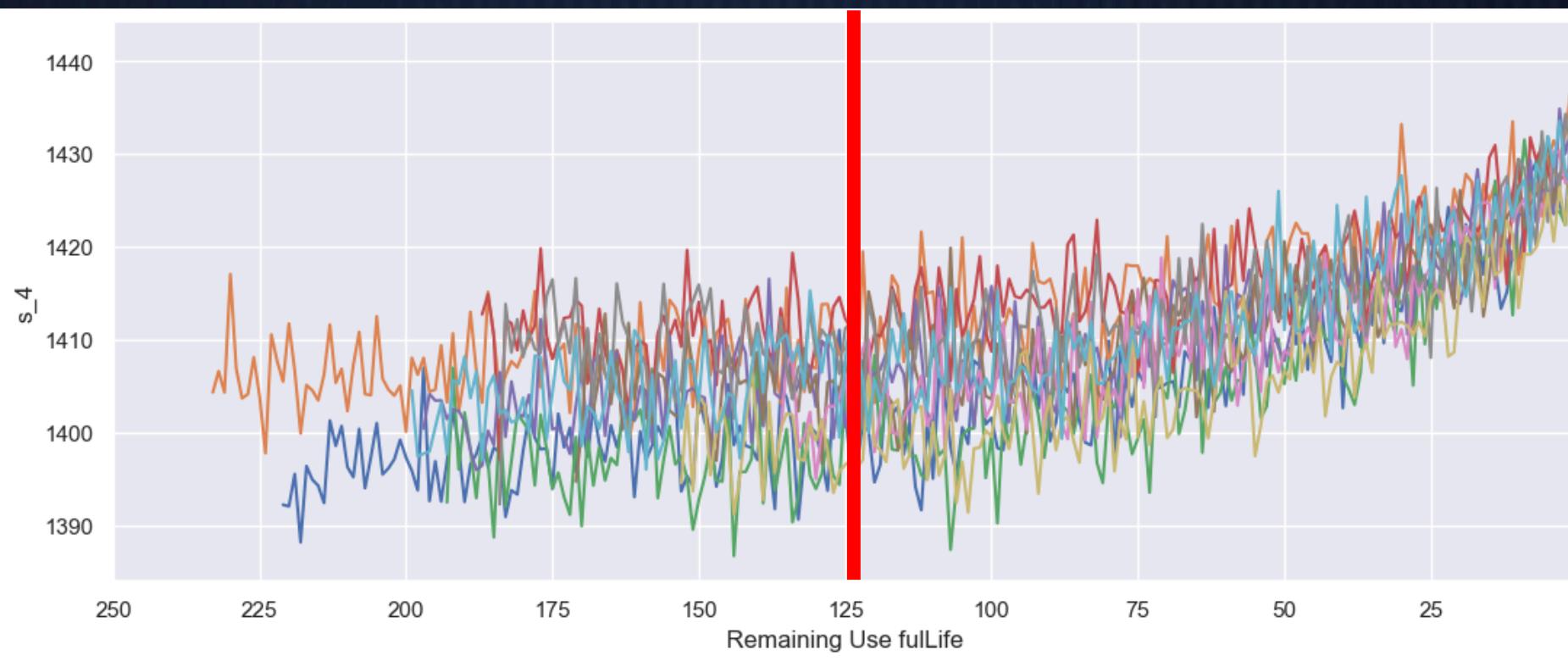
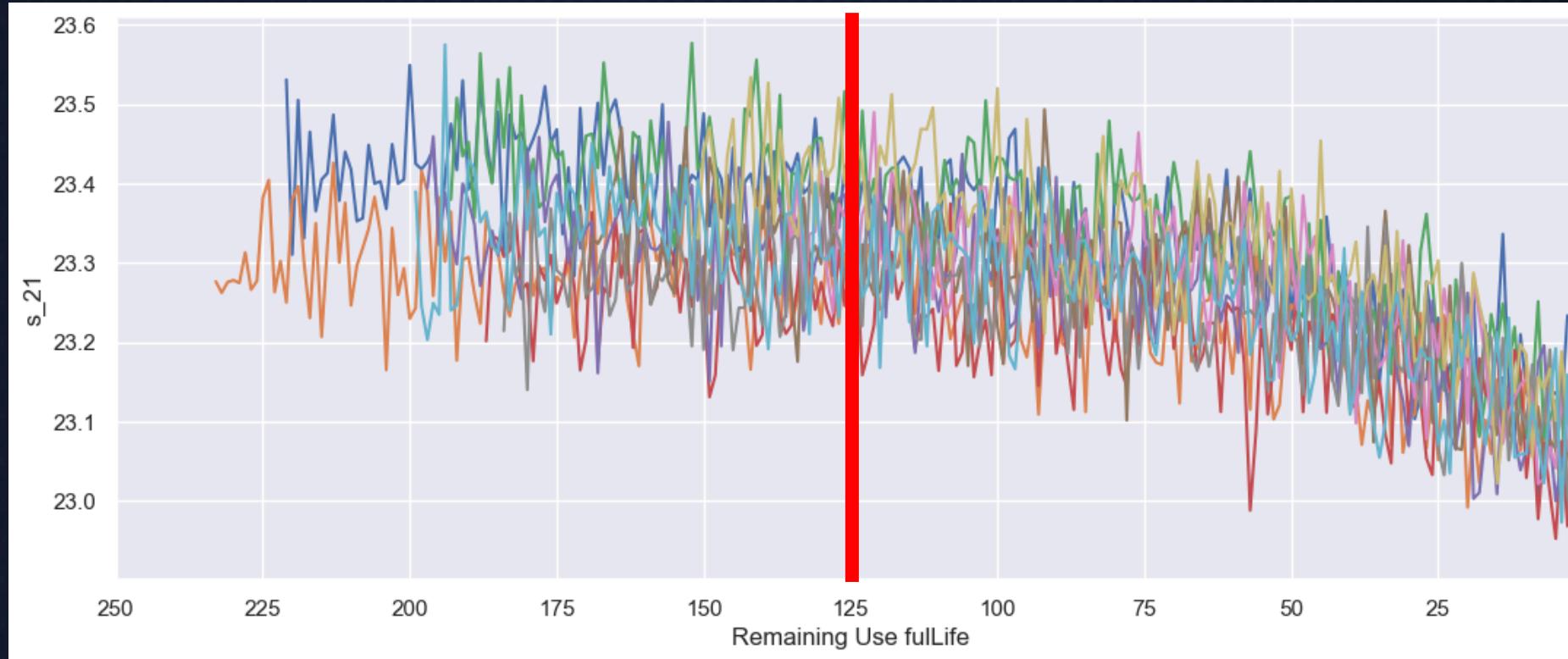


데이터 분포



train 데이터와 test 데이터 간의 분포는 비슷
-> 비교적 stratify 되어 있다는 것을 확인할 수 있음
-> 특별히 stratify 관련 전처리를 할 필요는 없어보임

데이터 전처리



Clipping

대부분의 센서 데이터에서 RUL값이
125인 부근에서 데이터의 변화가 급격
하게 나타남

-> 데이터 패턴이 단조로워져

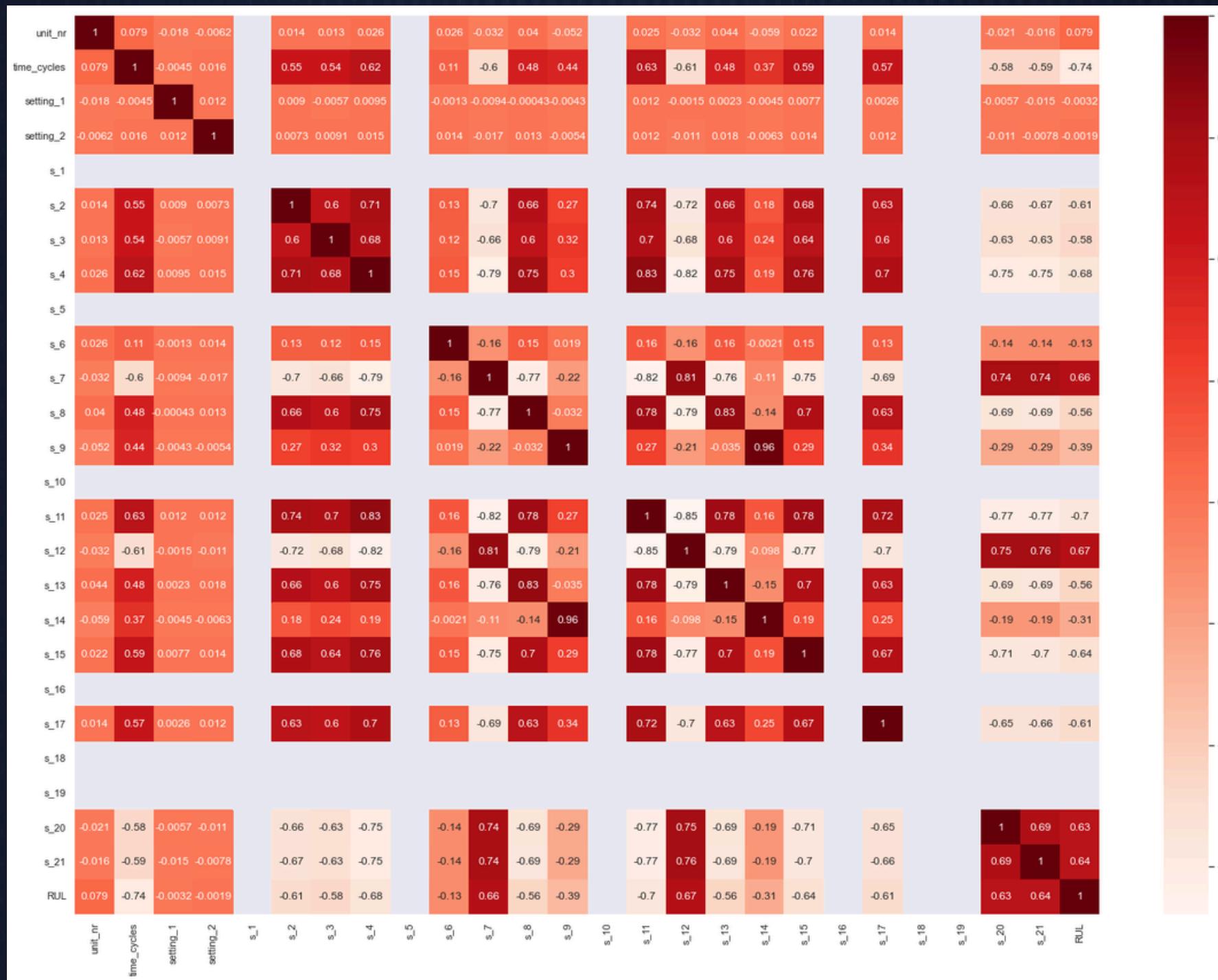
머신러닝의 예측이 어려움

-> 125를 기준으로 clipping 적용

데이터 전처리



상관관계



- 상관계수 계산이 안되는 컬럼 제외
- 약한 상관관계(절댓값 0.4)를 기준으로 나눈 데이터셋과 강한 상관관계(절댓값 0.6)를 기준으로 나눈 데이터셋으로 나눔

데이터 전처리



PCA(주성분분석)

장점

1. 계산 효율성 증가
2. 노이즈 제거
3. 과적합 방지

단점

1. 데이터 손실
2. Underfitting의 원인

두 가지 경우로 나누어서 테스트 진행

1. PCA 적용
2. PCA 미적용



StandardScaler

평균을 0, 표준편차를 1로 만들어줌

장점

분포를 정규화 시켜 비교를 쉽게 함
이상치에 민감하지 않음

단점

최대 최소 값을 정확히 알 수 없음

두 가지 경우로 나누어서 테스트 진행

1. StandardScaler 적용
2. StandardScaler 미적용

모델 비교

| Base | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|--------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regerssion | 398.697 | 458.826 | 0.770 | 0.714 | 0.48 | 0.45 |
| SVM Regression | 324.508 | 422.470 | 0.813 | 0.737 | 0.14 | 0.20 |
| Ridge Regerssion | 339.222 | 458.798 | 0.770 | 0.714 | 0.48 | 0.46 |
| Lasso Regerssion | 398.714 | 458.645 | 0.770 | 0.714 | 0.48 | 0.45 |
| Elasticnet | 400.269 | 456.870 | 0.770 | 0.715 | 0.48 | 0.45 |
| Random Forest Regression | 213.268 | 265.786 | 0.877 | 0.834 | 0.22 | 0.25 |
| XGboost | 238.424 | 260.508 | 0.863 | 0.838 | 0.22 | 0.23 |

clipping을 제외한 모든 전처리를 진행하지 않고 진행한 결과
Random Forest Regression, XGboost가 좋은 성능을 보임

모델 비교 (상관계수 0.6)

| PCA O 상관계수 0.6 | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|--------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regerssion | 437.002 | 511.763 | 0.748 | 0.681 | 0.44 | 0.48 |
| SVM Regression | 341.681 | 516.239 | 0.803 | 0.679 | 0.15 | 0.23 |
| Ridge Regerssion | 437.002 | 511.762 | 0.748 | 0.681 | 0.44 | 0.48 |
| Lasso Regerssion | 439.628 | 505.660 | 0.747 | 0.685 | 0.43 | 0.47 |
| Elasticnet | 445.190 | 502.390 | 0.744 | 0.687 | 0.43 | 0.48 |
| Random Forest Regression | 331.300 | 363.131 | 0.810 | 0.773 | 0.43 | 0.37 |
| XGboost | 196.838 | 443.188 | 0.887 | 0.724 | 0.21 | 0.30 |

| PCA X 상관계수 0.6 | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|--------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regerssion | 436.105 | 506.359 | 0.749 | 0.685 | 0.43 | 0.48 |
| SVM Regression | 324.508 | 422.475 | 0.813 | 0.737 | 0.14 | 0.20 |
| Ridge Regerssion | 436.105 | 506.359 | 0.745 | 0.685 | 0.43 | 0.48 |
| Lasso Regerssion | 436.105 | 506.357 | 0.749 | 0.685 | 0.43 | 0.48 |
| Elasticnet | 442.013 | 500.383 | 0.745 | 0.688 | 0.43 | 0.48 |
| Random Forest Regression | 306.692 | 365.040 | 0.823 | 0.772 | 0.30 | 0.32 |
| XGboost | 242.051 | 402.707 | 0.861 | 0.749 | 0.22 | 0.29 |

두 경우 모두 Random Forest Regerssion,
XGboost의 성능이 좋게 나왔고
PCA를 적용하지 않은 경우 SVM도 좋은 성능을 보임

모델 비교 (상관계수 0.4)

| PCA O 상관계수 0.4 | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|--------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regerssion | 430.091 | 504.721 | 0.752 | 0.686 | 0.43 | 0.47 |
| SVM Regression | 324.034 | 398.009 | 0.813 | 0.752 | 0.14 | 0.20 |
| Ridge Regerssion | 430.091 | 504.720 | 0.752 | 0.686 | 0.43 | 0.47 |
| Lasso Regerssion | 435.657 | 491.160 | 0.750 | 0.694 | 0.42 | 0.46 |
| Elasticnet | 441.968 | 487.210 | 0.746 | 0.697 | 0.42 | 0.47 |
| Random Forest Regerssion | 337.283 | 385.198 | 0.806 | 0.760 | 0.41 | 0.35 |
| XGboost | 302.013 | 401.557 | 0.83 | 0.75 | 0.27 | 0.30 |

| PCA X 상관계수 0.4 | Train MSE | Test MSE | train r2 | Test R ² | Train 오차 | Test 오차 |
|--------------------------------|-----------|----------|----------|---------------------|----------|---------|
| Linear Regerssion | 427.402 | 498.591 | 0.754 | 0.690 | 0.43 | 0.46 |
| SVM Regression | 281.710 | 330.550 | 0.838 | 0.794 | 0.13 | 0.17 |
| Ridge Regerssion | 427.402 | 498.590 | 0.754 | 0.690 | 0.43 | 0.46 |
| Lasso Regerssion | 427.439 | 498.543 | 0.754 | 0.690 | 0.43 | 0.46 |
| Elasticnet | 437.469 | 489.685 | 0.748 | 0.695 | 0.13 | 0.48 |
| Random Forest Regerssion | 292.807 | 356.009 | 0.831 | 0.778 | 0.28 | 0.30 |
| XGboost | 269.764 | 342.671 | 0.845 | 0.787 | 0.24 | 0.26 |

두 경우 모두 SVM, Random Forest Regerssion, XGboost 의 성능이 좋게 나옴

모델 비교 (상관계수 0.6)

| PCA O 상관관계 0.6 스무딩 O | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|---------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regeression | 507.990 | 1335.150 | 0.707 | 0.168 | 0.48 | 1.13 |
| SVM Regression | 407.633 | 1369.051 | 0.765 | 0.147 | 0.16 | 0.42 |
| Ridge Regeression | 507.990 | 1335.151 | 0.707 | 0.168 | 0.48 | 1.13 |
| Lasso Regeression | 507.990 | 1335.155 | 0.707 | 0.168 | 0.48 | 1.13 |
| Elasticnet | 507.990 | 1335.160 | 0.707 | 0.168 | 0.48 | 1.13 |
| Random Forest Regeression | 359.106 | 1310.714 | 0.793 | 0.183 | 0.42 | 1.01 |
| XGboost | 111.032 | 1265.155 | 0.936 | 0.212 | 0.17 | 0.96 |

| PCA X 상관관계 0.6 스무딩 O | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|---------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regeression | 505.066 | 1340.537 | 0.709 | 0.165 | 0.48 | 1.14 |
| SVM Regression | 378.290 | 1315.629 | 0.782 | 0.181 | 0.15 | 0.41 |
| Ridge Regeression | 505.066 | 1340.537 | 0.709 | 0.165 | 0.48 | 1.14 |
| Lasso Regeression | 506.124 | 1339.176 | 0.708 | 0.166 | 0.48 | 1.13 |
| Elasticnet | 506.432 | 1338.855 | 0.708 | 0.166 | 0.48 | 1.14 |
| Random Forest Regeression | 282.846 | 1374.486 | 0.837 | 0.144 | 0.25 | 1.02 |
| XGboost | 58.721 | 1491.978 | 0.966 | 0.070 | 0.11 | 1.03 |

두 경우 모든 모델이 Overfitting이 심함

모델 비교 (상관계수 0.4)

| PCA O 상관관계 0.4 스무딩 O | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|---------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regeression | 488.881 | 1316.663 | 0.718 | 0.180 | 0.47 | 1.14 |
| SVM Regression | 378.290 | 1315.630 | 0.782 | 0.181 | 0.15 | 0.41 |
| Ridge Regeression | 488.881 | 1316.663 | 0.718 | 0.180 | 0.47 | 1.14 |
| Lasso Regeression | 488.881 | 1316.663 | 0.718 | 0.180 | 0.47 | 1.14 |
| Elasticnet | 488.881 | 1316.665 | 0.718 | 0.180 | 0.47 | 1.14 |
| Random Forest Regeression | 343.747 | 1284.097 | 0.802 | 0.200 | 0.40 | 1.05 |
| XGboost | 126.178 | 1385.408 | 0.927 | 0.137 | 0.18 | 1.07 |

| PCA X 상관관계 0.4 스무딩 O | Train MSE | Test MSE | Train R ² | Test R ² | Train 오차 | Test 오차 |
|---------------------------------|-----------|----------|----------------------|---------------------|----------|---------|
| Linear Regeression | 486.630 | 1341.011 | 0.719 | 0.164 | 0.47 | 1.15 |
| SVM Regression | 378.290 | 1315.630 | 0.782 | 0.180 | 0.15 | 0.41 |
| Ridge Regeression | 486.630 | 1341.009 | 0.719 | 0.164 | 0.47 | 1.15 |
| Lasso Regeression | 495.970 | 1327.997 | 0.714 | 0.173 | 0.48 | 1.14 |
| Elasticnet | 495.368 | 1332.623 | 0.714 | 0.170 | 0.48 | 1.14 |
| Random Forest Regeression | 215.676 | 1332.409 | 0.875 | 0.170 | 0.21 | 1.07 |
| XGboost | 90.069 | 1392.484 | 0.948 | 0.132 | 0.14 | 1.07 |

두 경우 모든 모델이 Overfitting이 심함

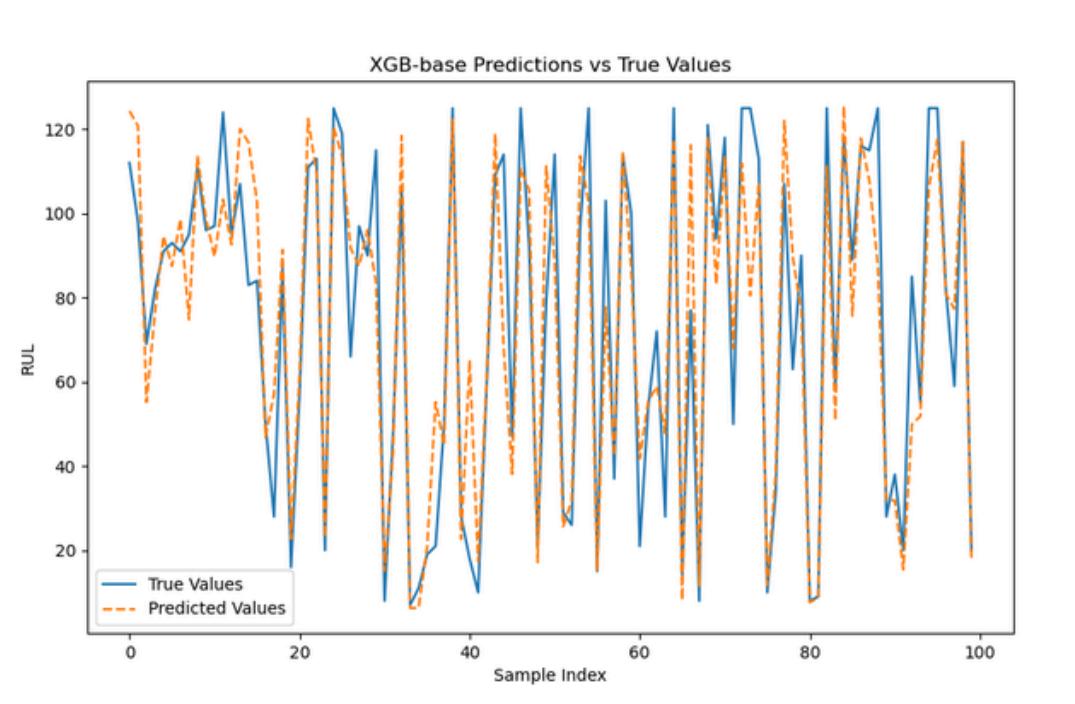
| 딥러닝 모델 | MSE | R ² | 평균오차율 |
|-----------------|---------|----------------|-------|
| LSTM | 257.797 | 0.839 | 0.208 |
| CNN-LSTM | 285.70 | 0.822 | 0.239 |
| Trasformer-LSTM | 282.180 | 0.824 | 0.219 |
| Attention-LSTM | 431.717 | 0.731 | 0.260 |

LSTM,CNN-LSTM, Transformer-LSTM이 좋은 결과를 보임

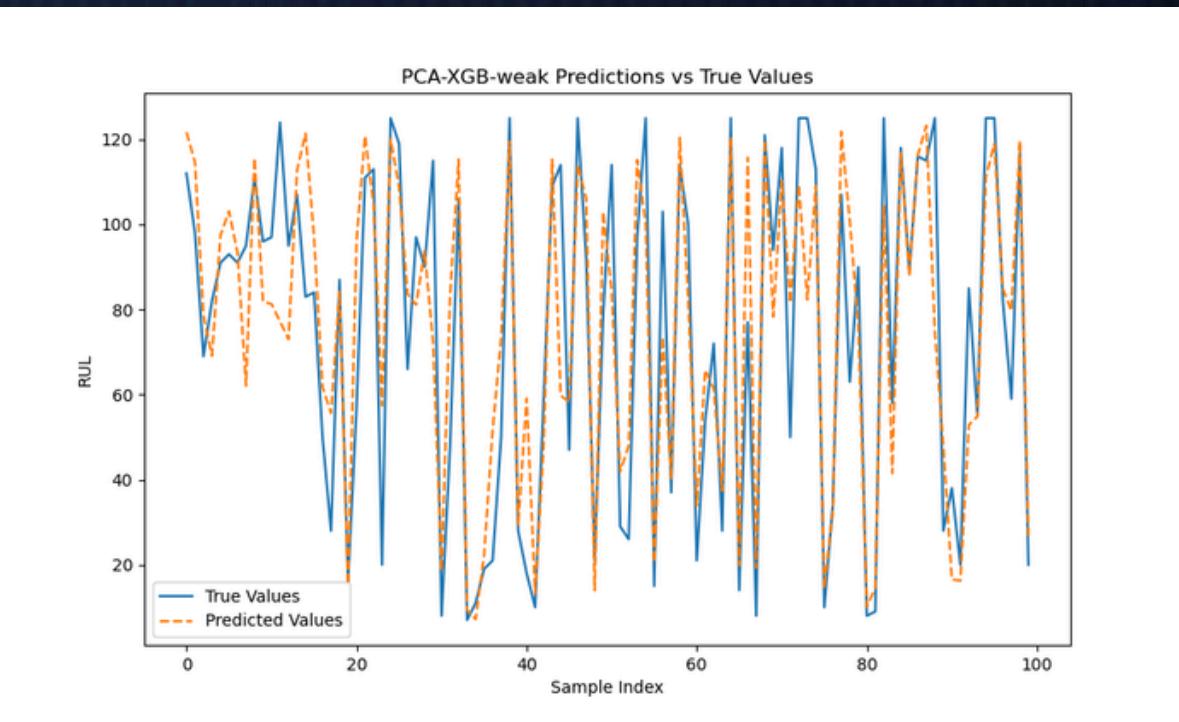
06 결과 및 결론

XGboost 결과 모음

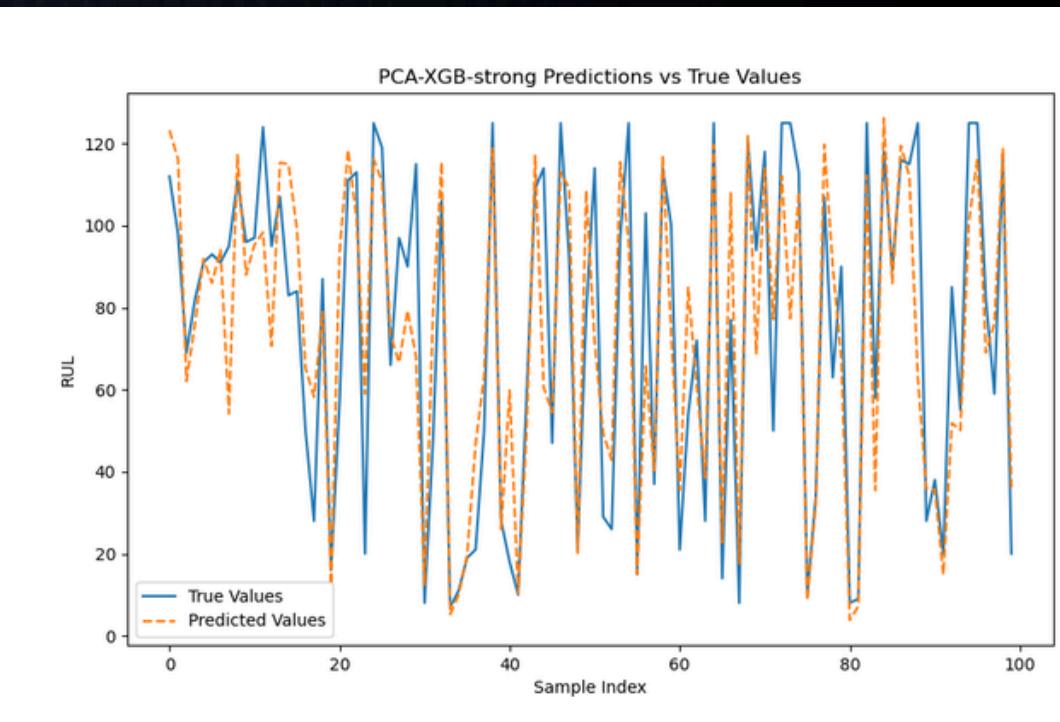
Base



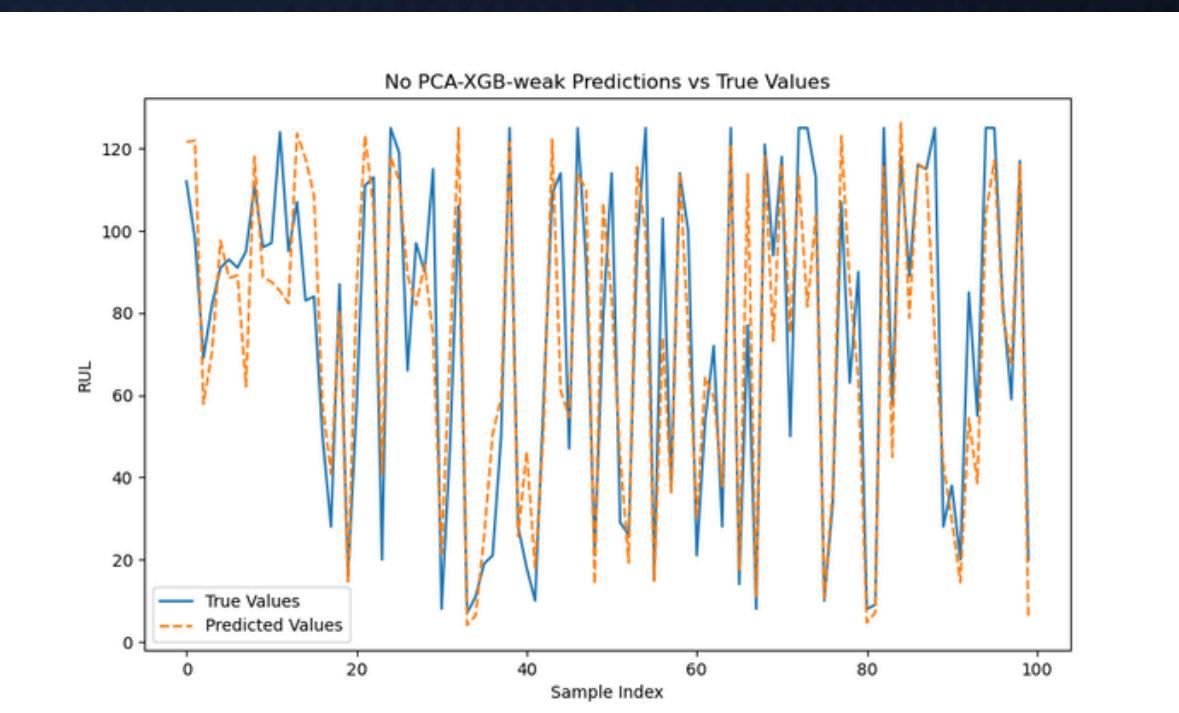
PCA O 상관계수 0.4



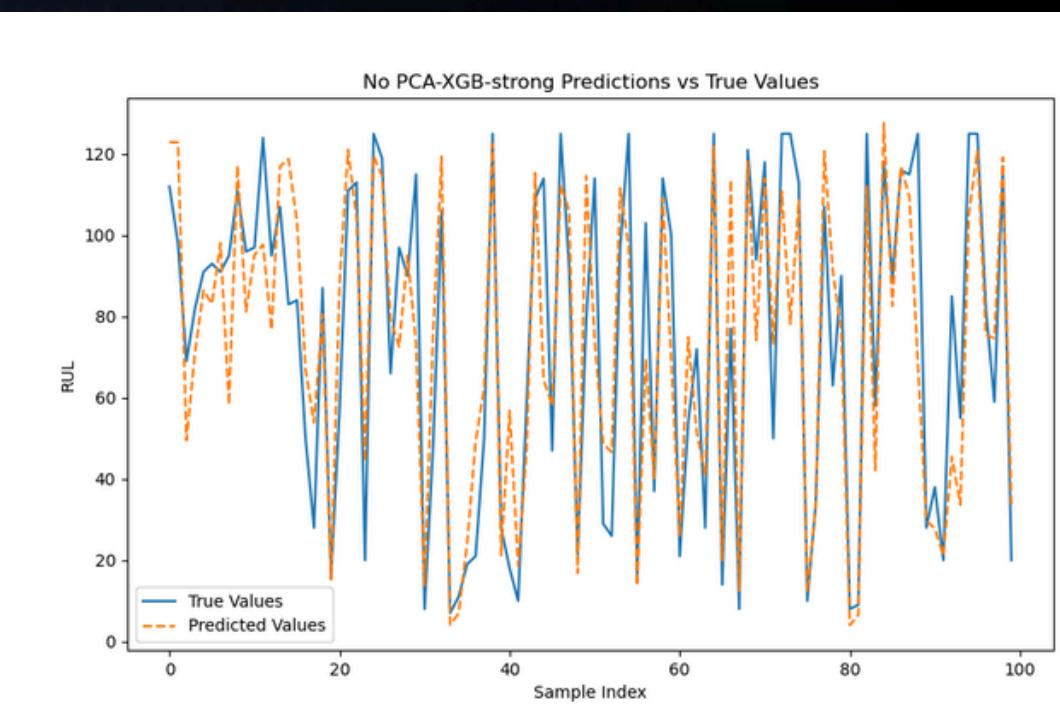
PCA O 상관계수 0.6



PCA X 상관계수 0.4

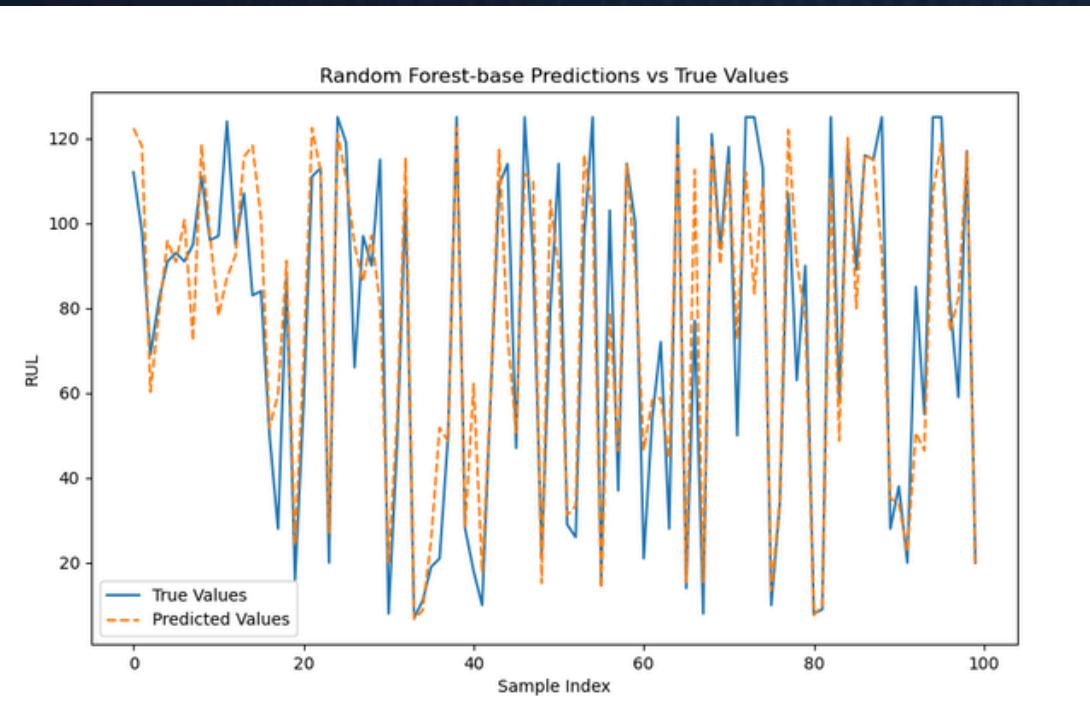


PCA X 상관계수 0.6

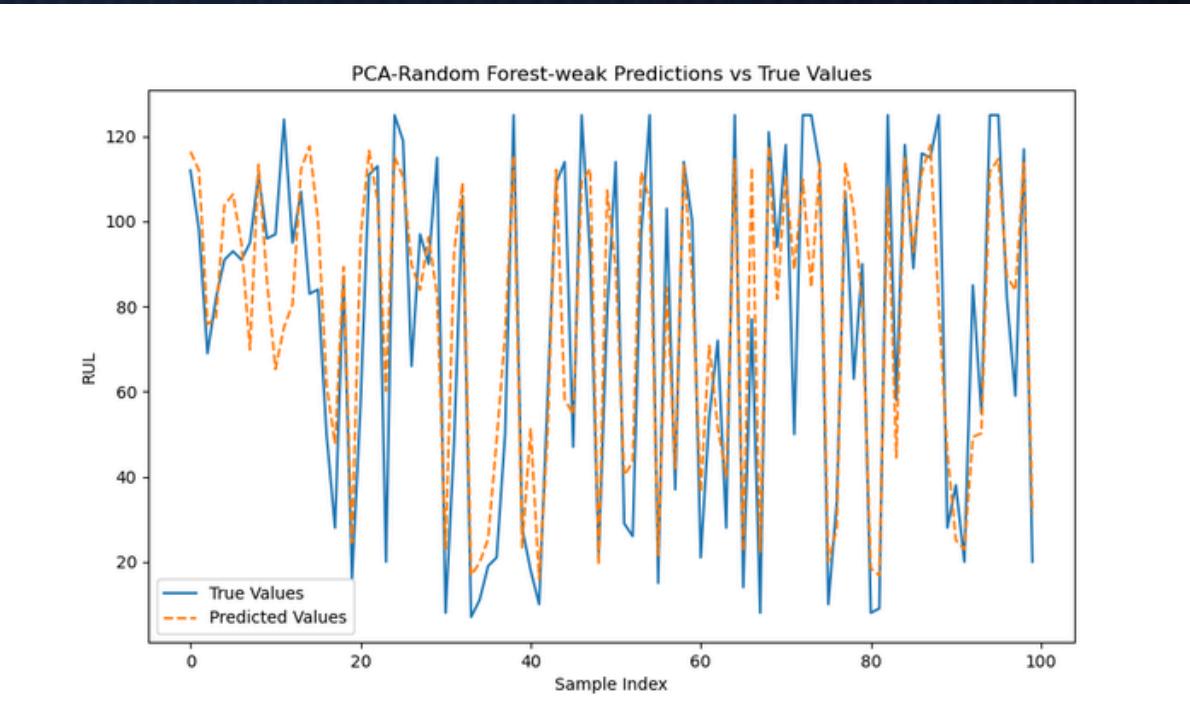


Random Forest 결과 모음

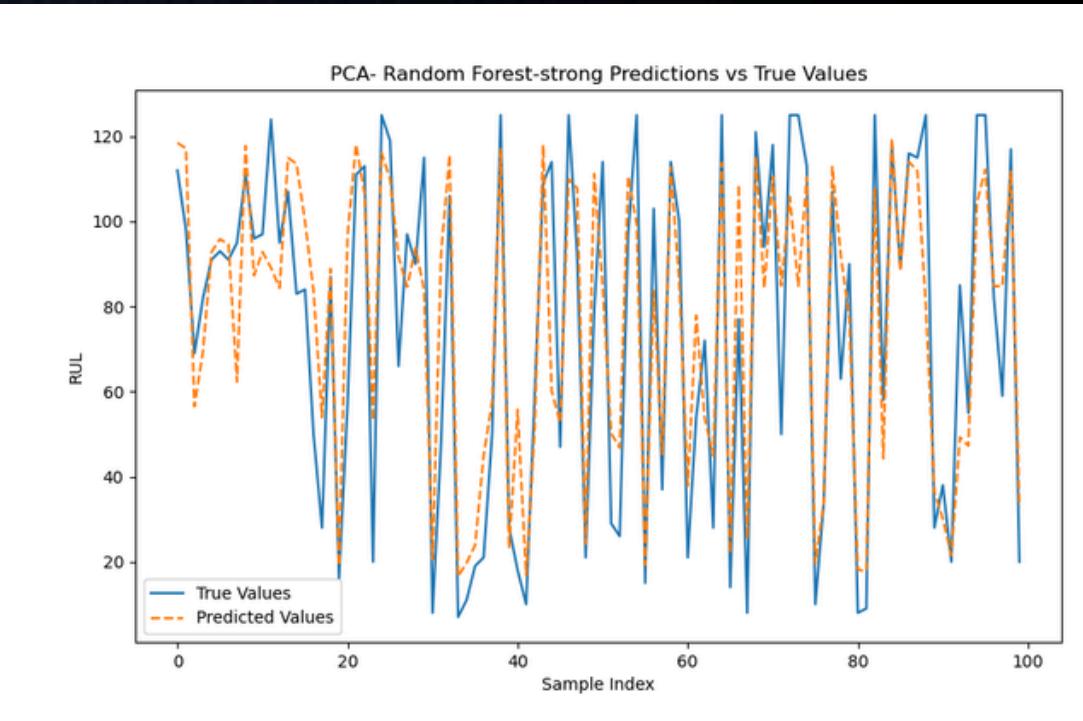
Base



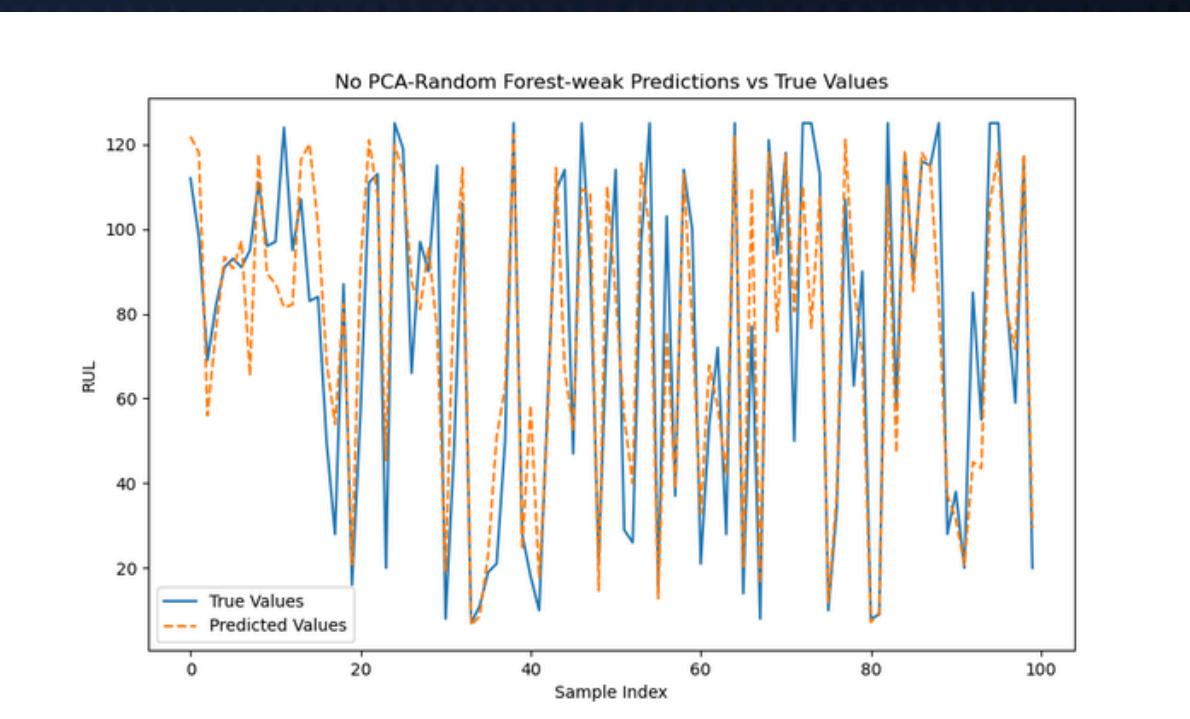
PCA O 상관계수 0.4



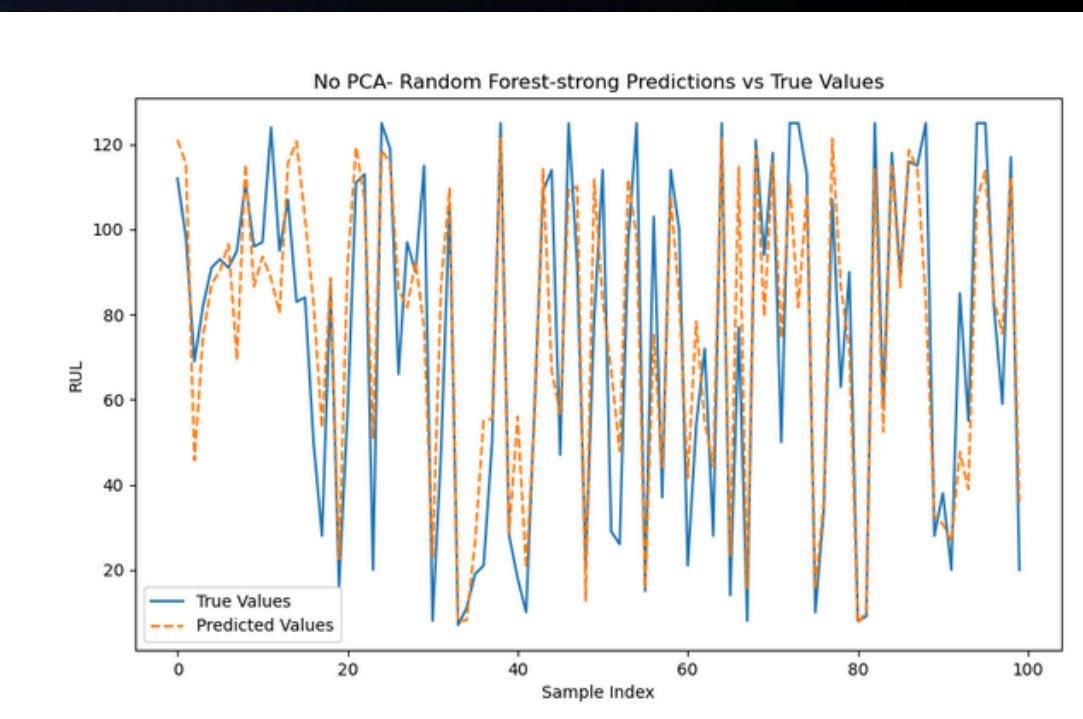
PCA O 상관계수 0.6



PCA X 상관계수 0.4

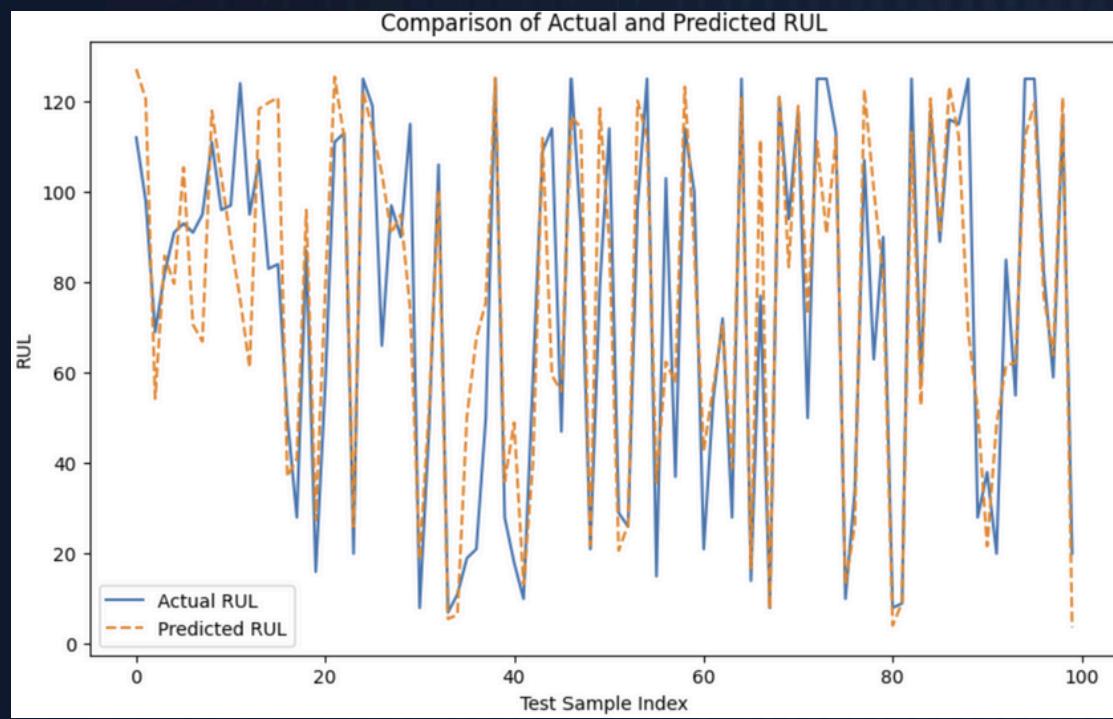


PCA X 상관계수 0.6

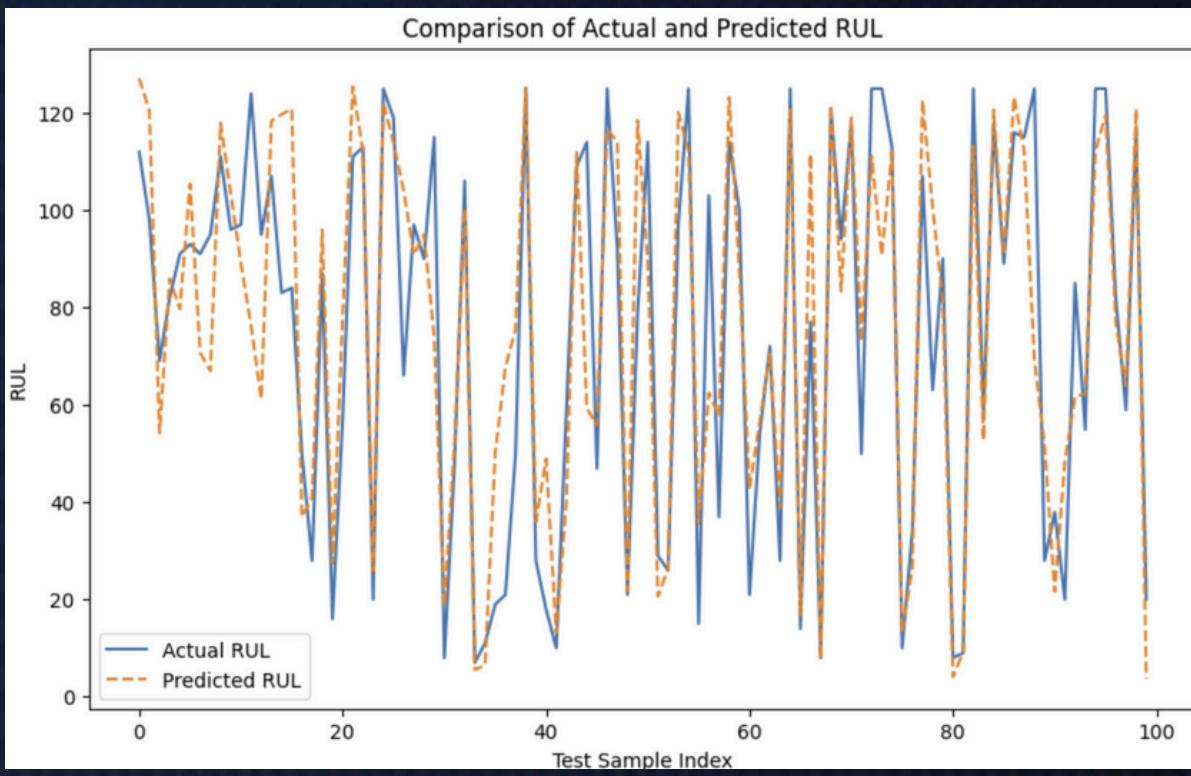


SVM 결과 모음

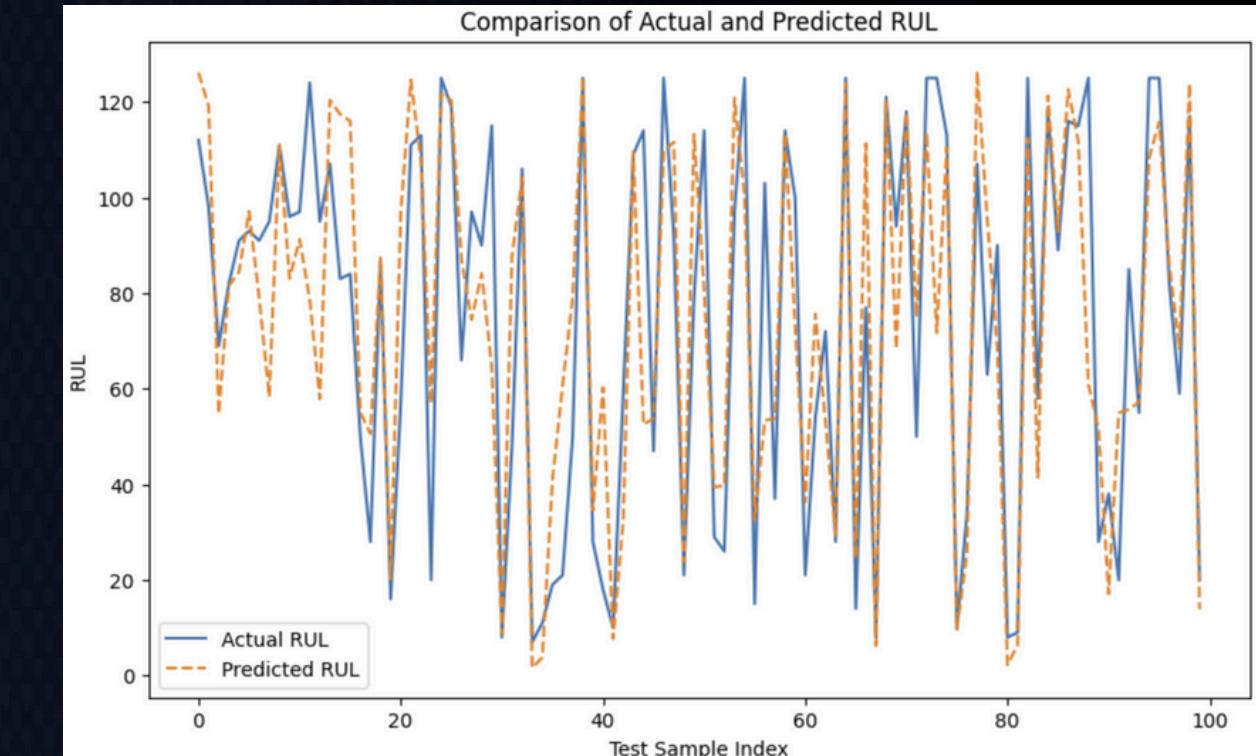
Base



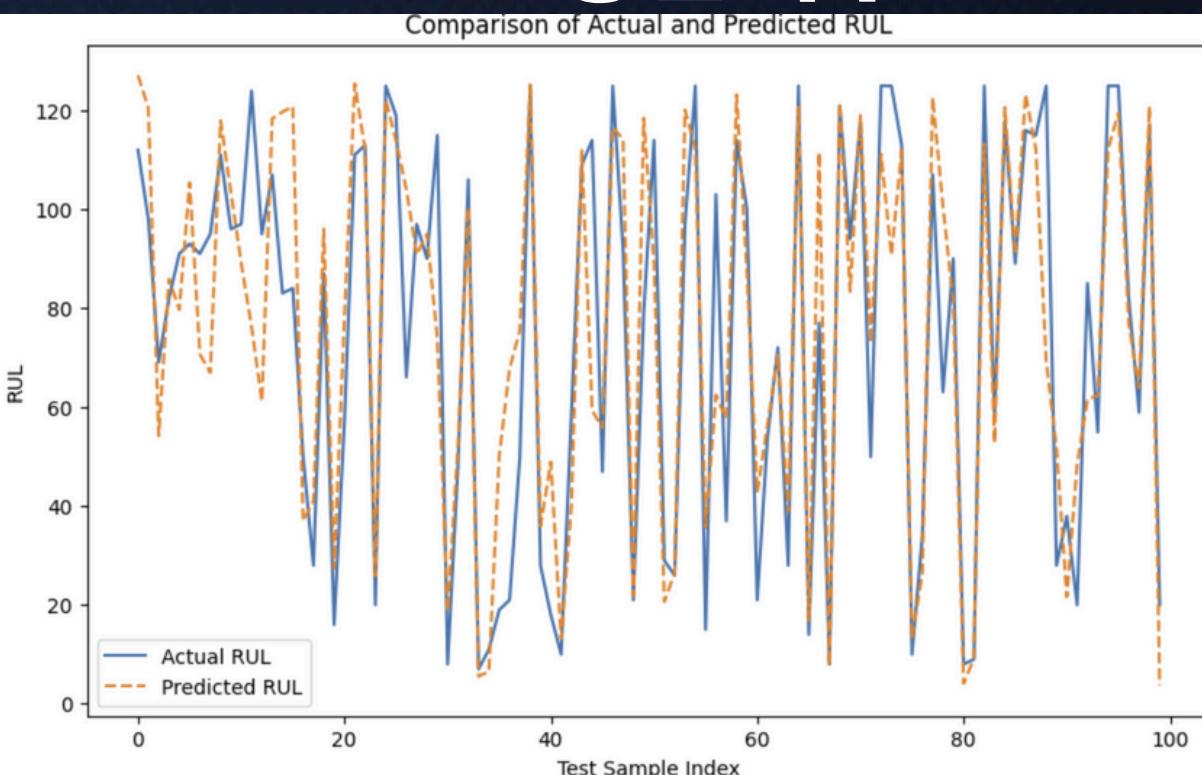
PCA O 상관계수 0.4



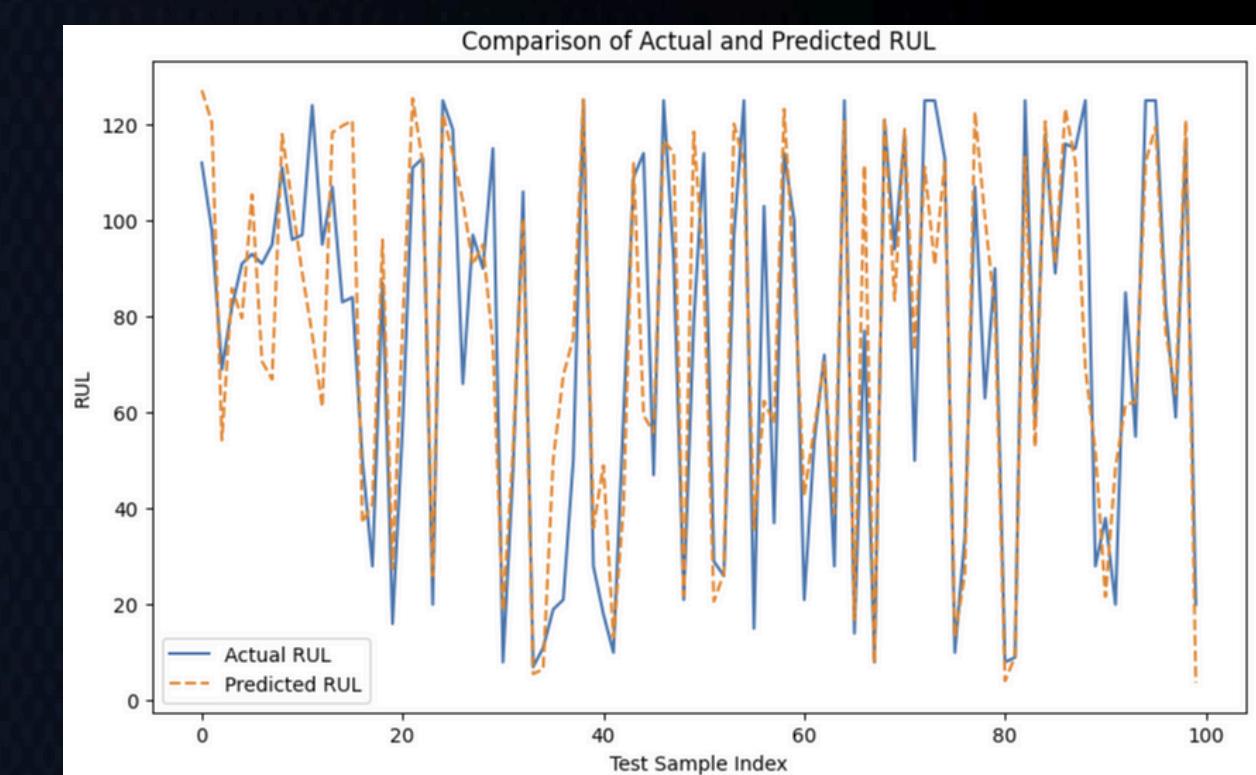
PCA O 상관계수 0.6



PCA X 상관계수 0.4

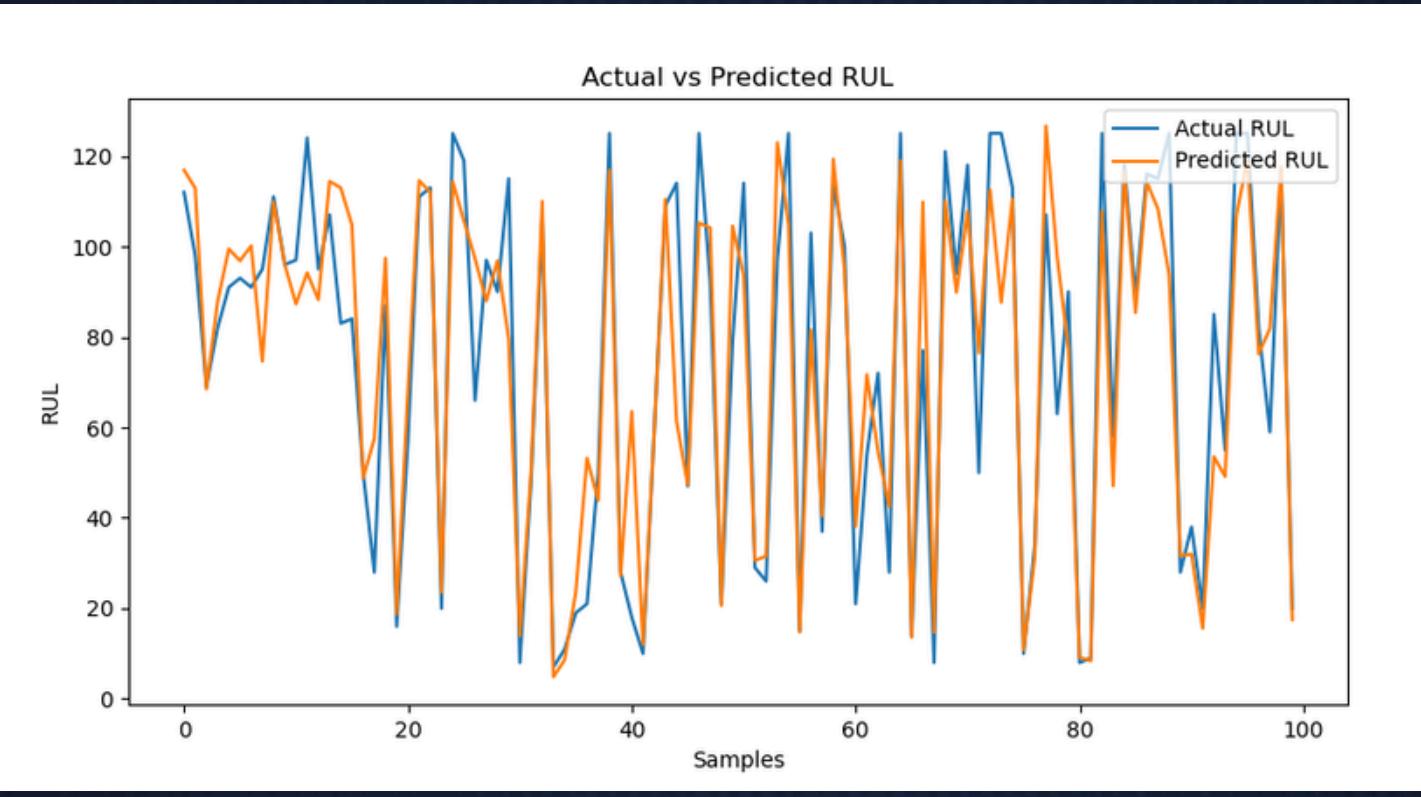


PCA X 상관계수 0.6

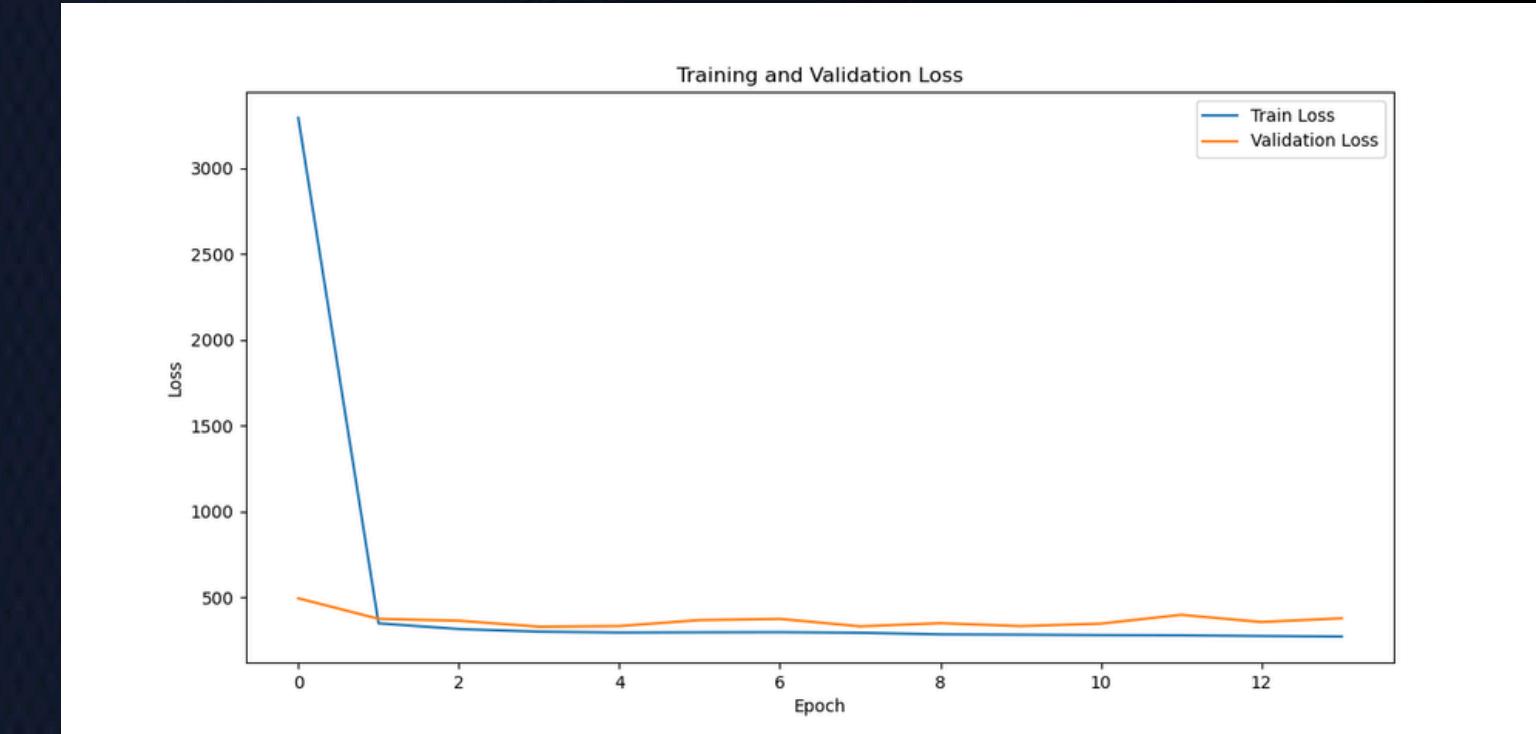
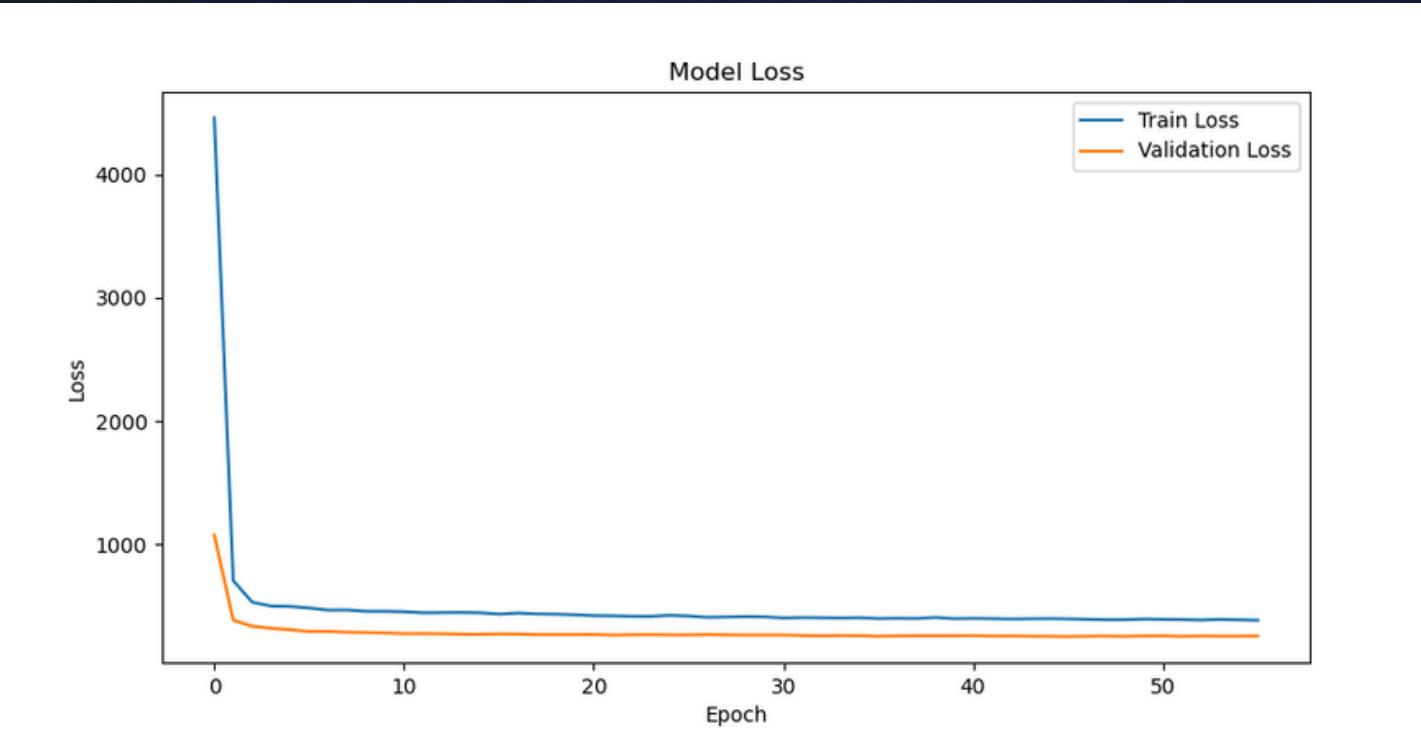
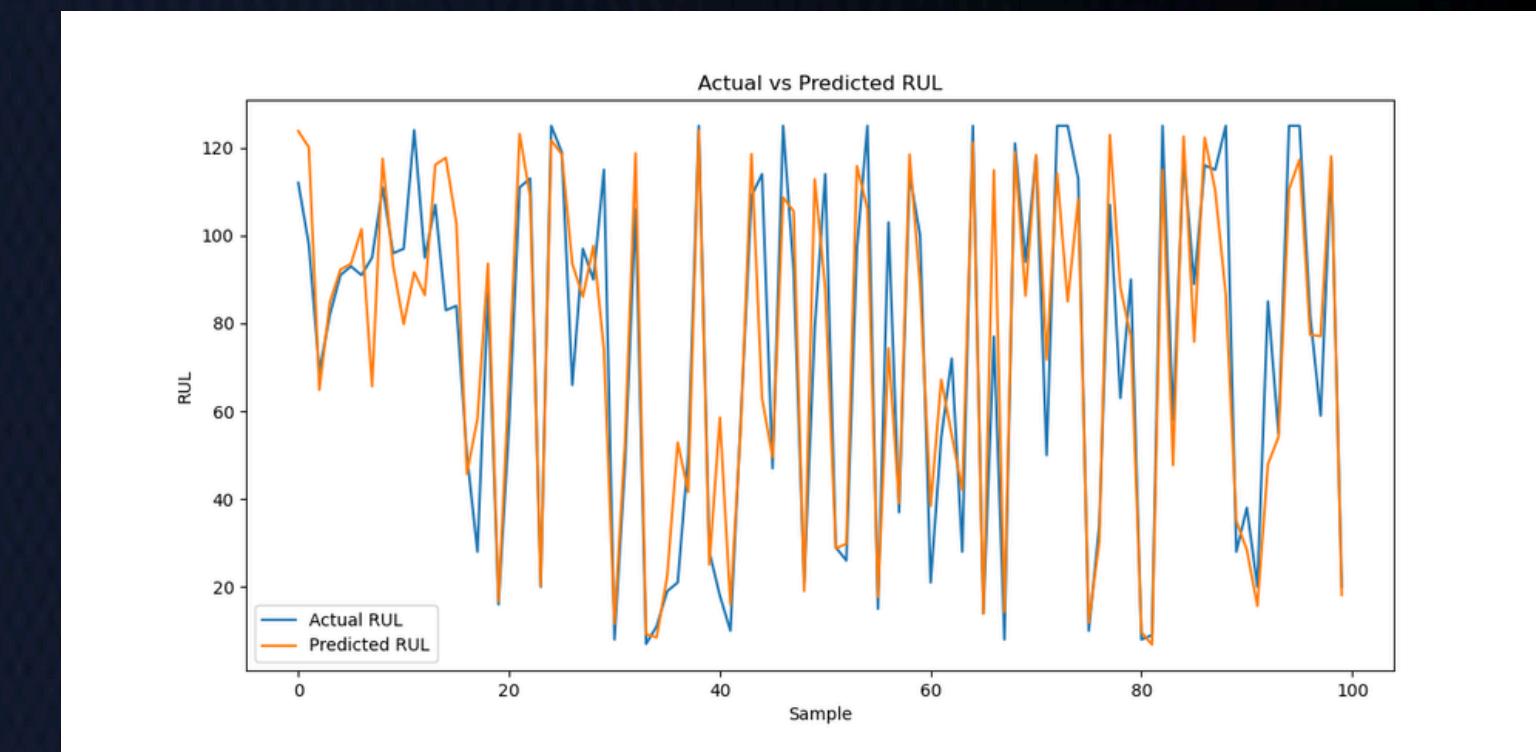


딥러닝 LSTM

LSTM RUL 예측과 실제 RUL



Transformer LSTM RUL 예측과 실제 RUL



과적합 문제 없음

과적합 문제 있음

가장 좋은 성능을 보여주는 모델



LSTM 선정 이유

딥러닝

LSTM

- 평가 지표에서 다른 모델보다 점수가 좋음
- Transformer LSTM에는 과적합 문제
가 발생하나 LSTM에서는 발생하지 않음

머신러닝

XGB

모든 경우에서 가장 좋은 지표를 보임

Random
Forest

모든 경우에서 가장 좋은 지표를 보임

SVM

모든 경우에서 가장 작은 오차율을 보임



THANKS