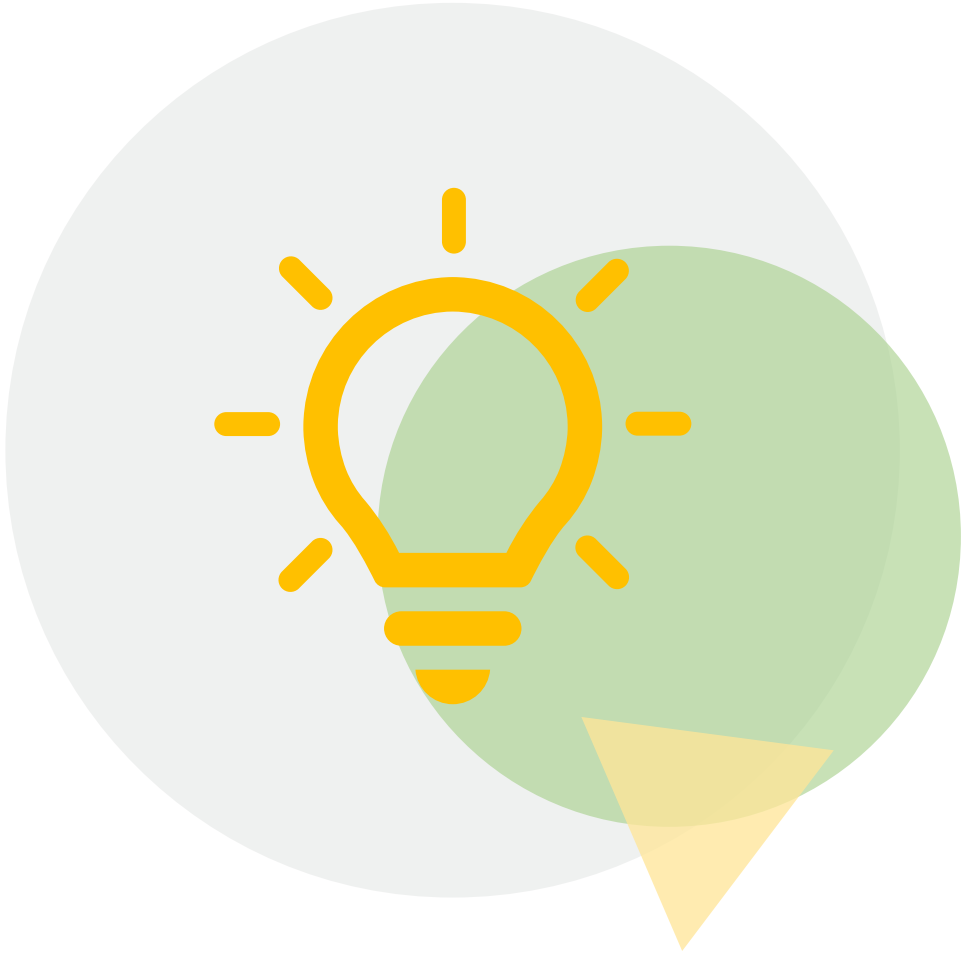
데이터 분석 비전문가도 활용 가능한
범용성 있는 AI 이상감지 모델 개발



응용 프로젝트 AI제조 4팀

고용노동부 KSQA KOREA TECH 직업능력심사평가원

목차

01 팀 구성 및 역할

02 프로젝트 개요

03 프로젝트 수행 및 과정

04 프로젝트 수행 결과

팀 구성 및 역할

팀 구성 및 역할

김정민 팀장

20학번 소프트웨어전공

역할 : CNN, PPT제작

김남훈

18학번 IT경영

역할 : XGBoost, 시각화

윤종헌

17학번 IT경영

역할 : LSTM, PPT제작

이상헌

18학번 게임공학

역할 : Random Forest, 시각화

이슬인

19학번 전자공학

역할 : LSTM, PPT제작

임규호

16학번 생명화학공학

역할 : XGBoost, PPT제작

정용규

19학번 컴퓨터공학

역할 : RNN, CNN, 시각화

프로젝트 개요

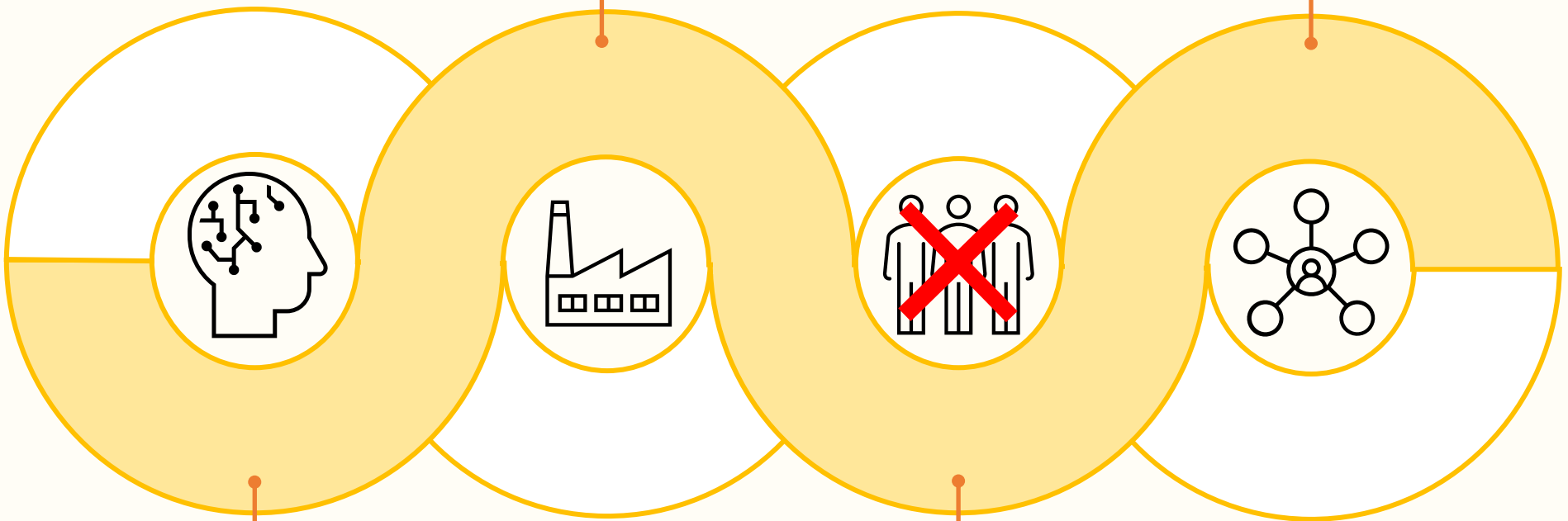
프로젝트 개요

산업 현장 속 수많은 문제들

최적화된 데이터 분석,
Feature Engineering 수행 한계

범용적으로 사용 가능한 모델

'시스템 정상 작동 여부와 센서 데이터'
기반 AI 예측 모델 개발



AI 예측 모델

현재 다양한 분야에서
좋은 성능

인적/물적 자원 부족

프로젝트 개요 - 목표

분석 목표

설비 내 여러 개의 계측 센서의 시계열 데이터(행 : 시간, 열 : 센서의 종류)에 대해,

설비의 정상 / 비정상을 판단하는 분류 문제를 해결

목표 수행을 위한 전략

도메인 지식에 기반한 **Feature Engineering 없이**,딥러닝 모델이 데이터에 내재된 **특징 직접 추출**

이를 통해 분류할 수 있는 분석 프레임워크 구축

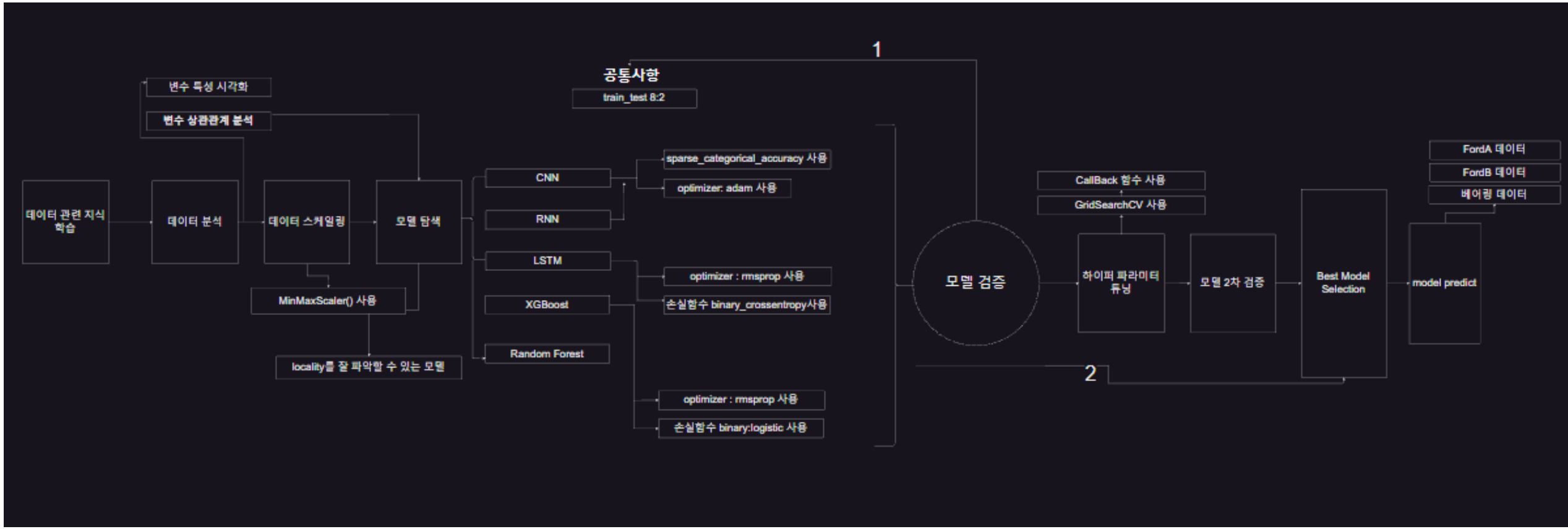
데이터 분석 기대효과

도메인 지식이 없는 분석 실무자가 기계학습 기반의 정상 / 비정상 **분류 과제 수행 가능**

프로젝트 수행 및 과정

프로젝트 수행 및 과정 – Overall process

< 수행과정 >



프로젝트 수행 및 과정 – 데이터 분석

A	B	C	D	E
id	att1	att2	att3	att4
1	-0.79717	-0.66439	-0.37301	0.040815
2	0.804855	0.634629	0.373474	0.038343
3	0.727985	0.111284	-0.49912	-1.06863
4	-0.23444	-0.50216	-0.73249	-0.94613
5	-0.17133	-0.06229	0.235829	0.710396
6	-0.5409	-1.01402	-1.29823	-1.32083
7	-0.33406	-1.00801	-1.55435	-1.92219
8	1.04589	0.611195	0.153108	-0.27967
9	0.825565	0.385282	-0.06242	-0.48098
10	-0.28418	-0.19261	-0.03229	0.172823
11	0.529562	0.695556	0.754557	0.688517
12	-1.07104	-1.10475	-1.0247	-0.81404
13	-0.10945	-0.226	-0.30023	-0.33921
14	-1.34854	-0.72549	-0.08562	0.464571
15	1.429452	1.079359	0.510714	-0.14623
16	2.252085	2.157468	1.848938	1.397509
17	0.453874	0.424807	0.399023	0.359174
18	-0.50796	-0.80718	-0.8914	-0.69334
19	1.66411	1.523809	1.318033	1.056137
20	-1.87791	-1.77482	-1.49303	-1.13564

총 4.921개의 시계열 데이터 구조
3,601 개의 Training 데이터와 1,320 개의 Test 데이터로 구성

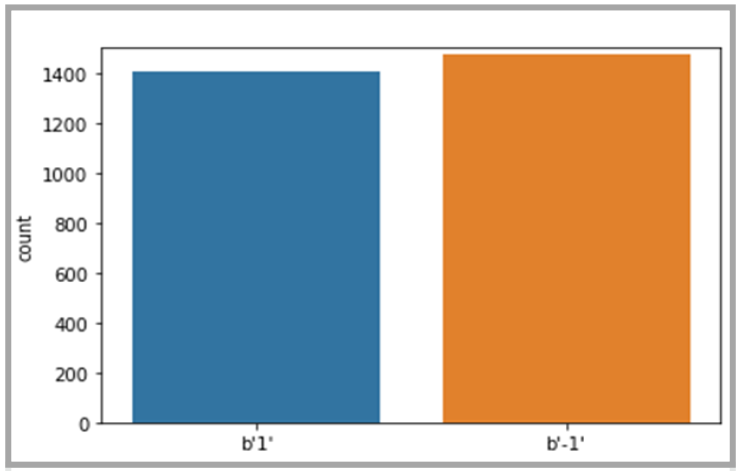
변수명	설명	데이터 타입
att(n)	특정 센서의 측정 값	숫자
Target	정상,비정상 상태 값	숫자

- 자동차 운영 체계 내 이상 여부를 판단하기 위해 관련된 500개의 센서로부터 수집된 계측 데이터로 구성
- 데이터 출처 : 미국 포드(Ford) 사에서 주최한 기계학습 대회내 오픈 데이터 셋
- Target 값이 1이면 정상, -1이면 비정상

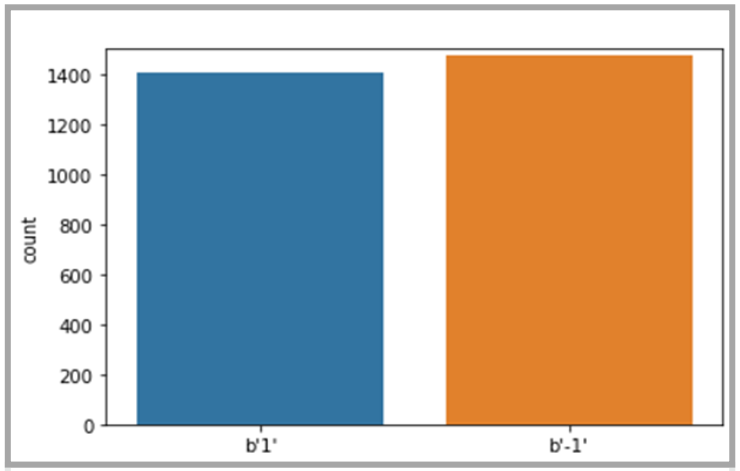
프로젝트 수행 및 과정 – 데이터 분석

3600 x 500개의 시계열 데이터에서

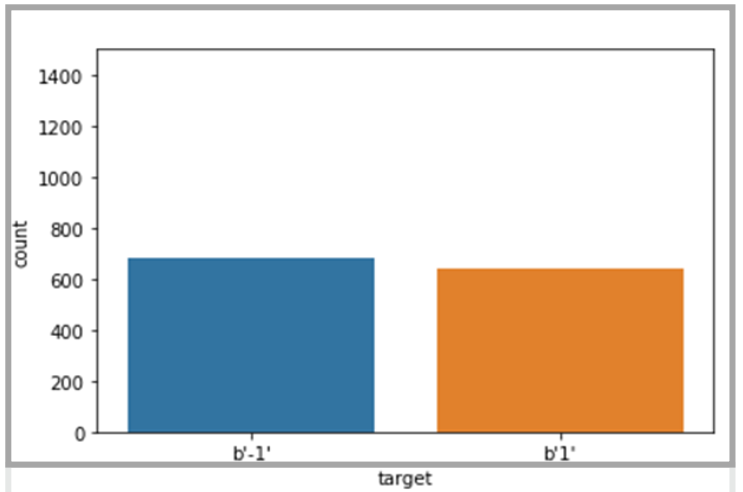
결측치와 데이터 불균형 문제가 있는지 확인



Train_FordA



Valid_FordA

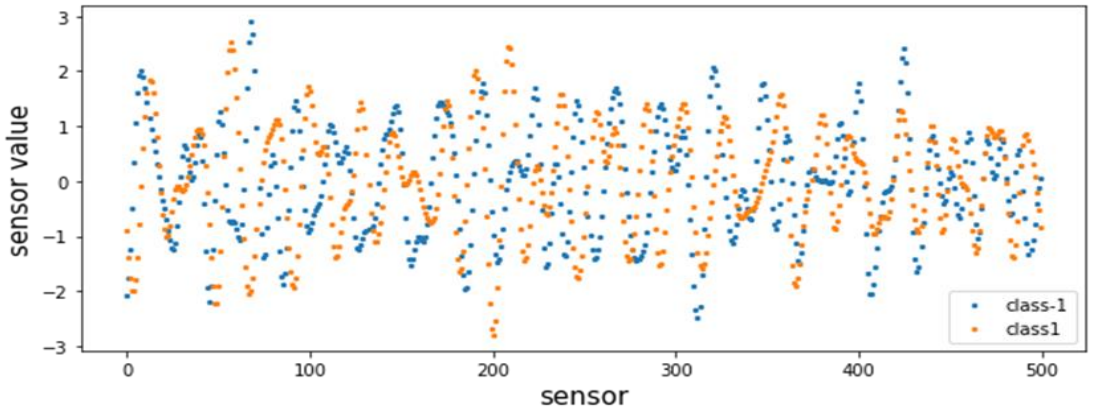


Test_FordA

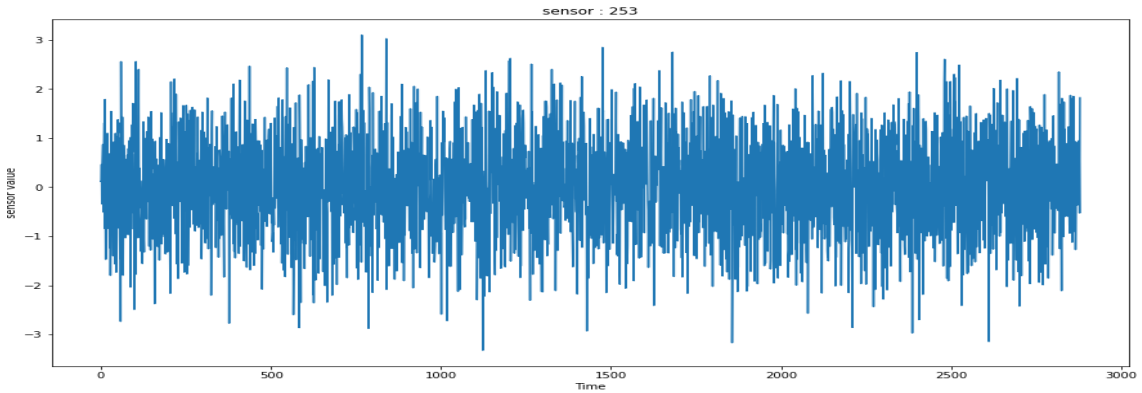
정상 / 비정상의 비율 & 분포도 비슷

➡ **데이터 불균형 아님 & 결측치 없음**

프로젝트 수행 및 과정 – 데이터 분석



Ford A 산점도
(class -1 : 비정상, class 0 : 정상)

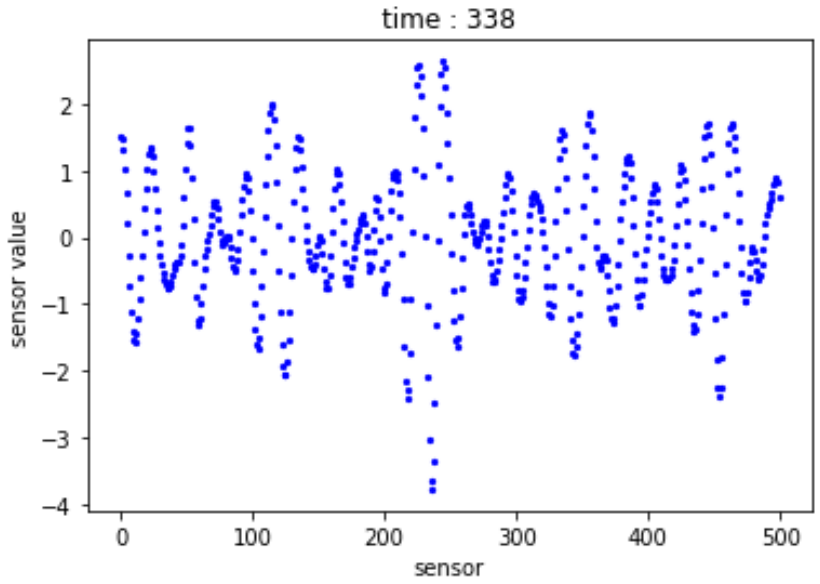


랜덤으로 추출한 시간에 따른
센서 하나의 측정값

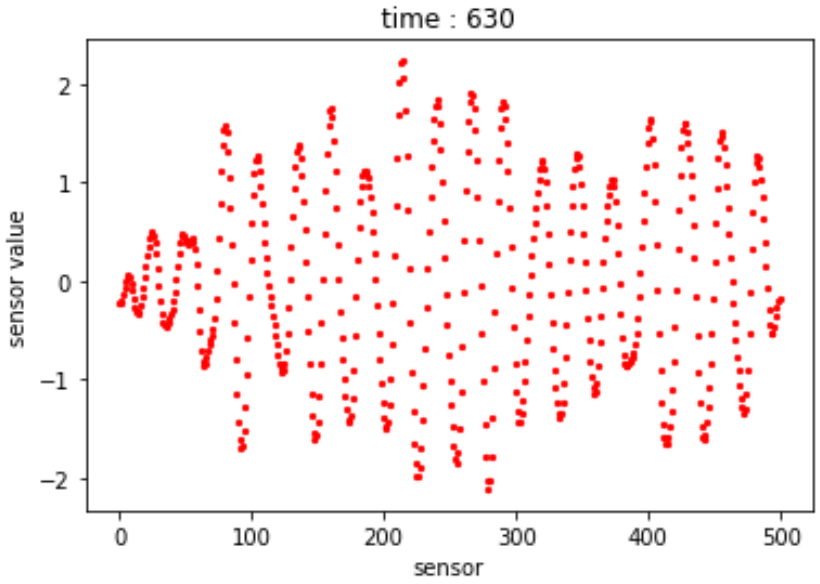
정상과 비정상의 비율이 비슷하며 양상이 거의 흡사

➡ 각 센서의 **양상이 거의 비슷함**

프로젝트 수행 및 과정 – 데이터 분석



Time 338번째
정상 데이터



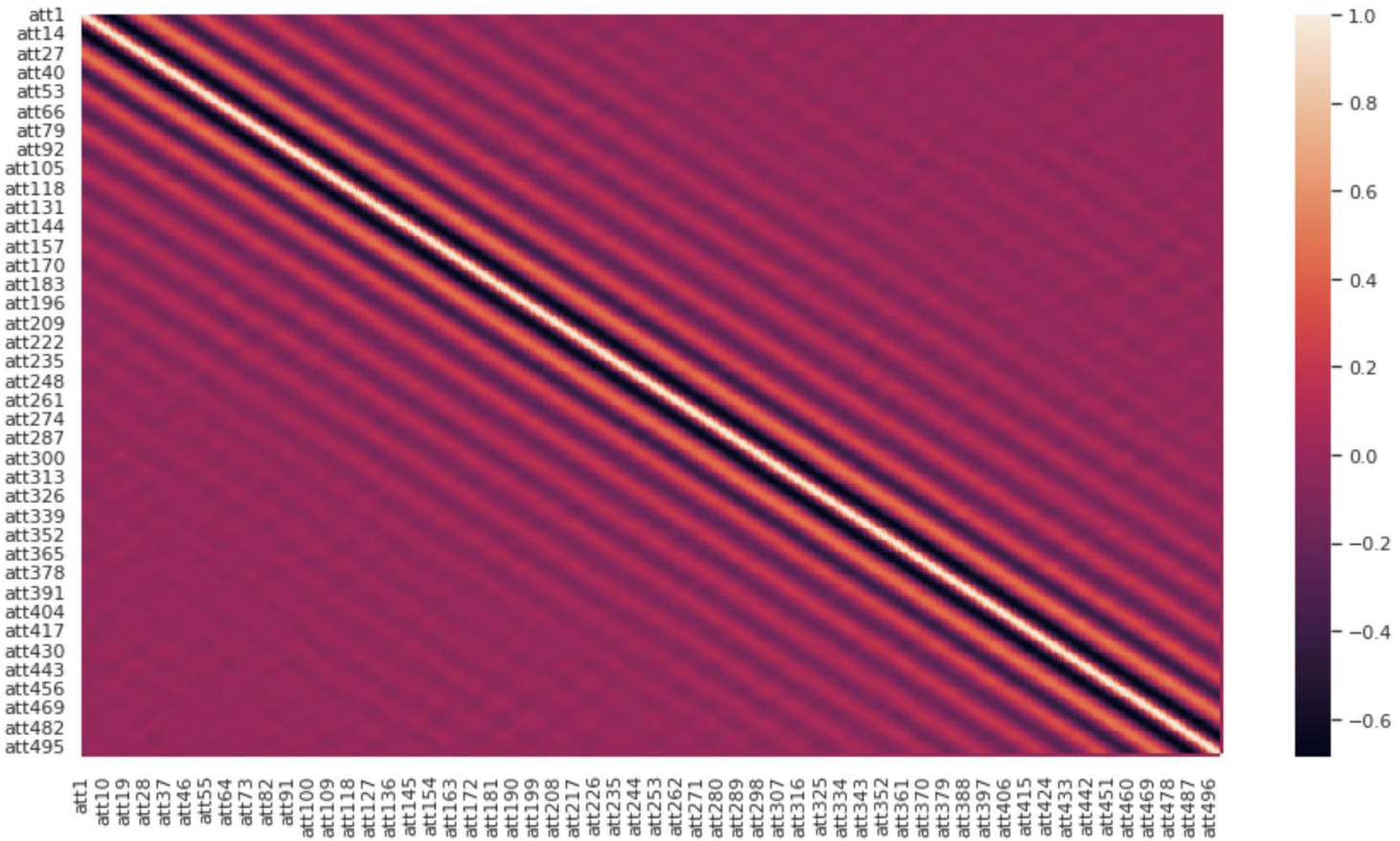
Time 630번째
비정상 데이터

정상 데이터에 비교했을 때

비정상 데이터들은 **-1 이하의 값들 비중이 많음**

프로젝트 수행 및 과정 – 데이터 분석

< 상관관계 분석 >



- 서로 **거리가 가까운** 센서끼리 유의한 관계가 있다는 것을 발견함

프로젝트 수행 및 과정 – 데이터 전처리

```
from sklearn.preprocessing import MinMaxScaler

col_list = train.columns.values.tolist()
train.sort_index(ascending=False).reset_index(drop=True)

scaler = MinMaxScaler()
scale_cols = col_list
df_scaled = scaler.fit_transform(train[scale_cols])
df_scaled = pd.DataFrame(df_scaled)
df_scaled.columns = scale_cols
```

< 데이터 스케일링>

- MinMaxScaler를 사용하여 전체 데이터 스케일링 (Ford A)

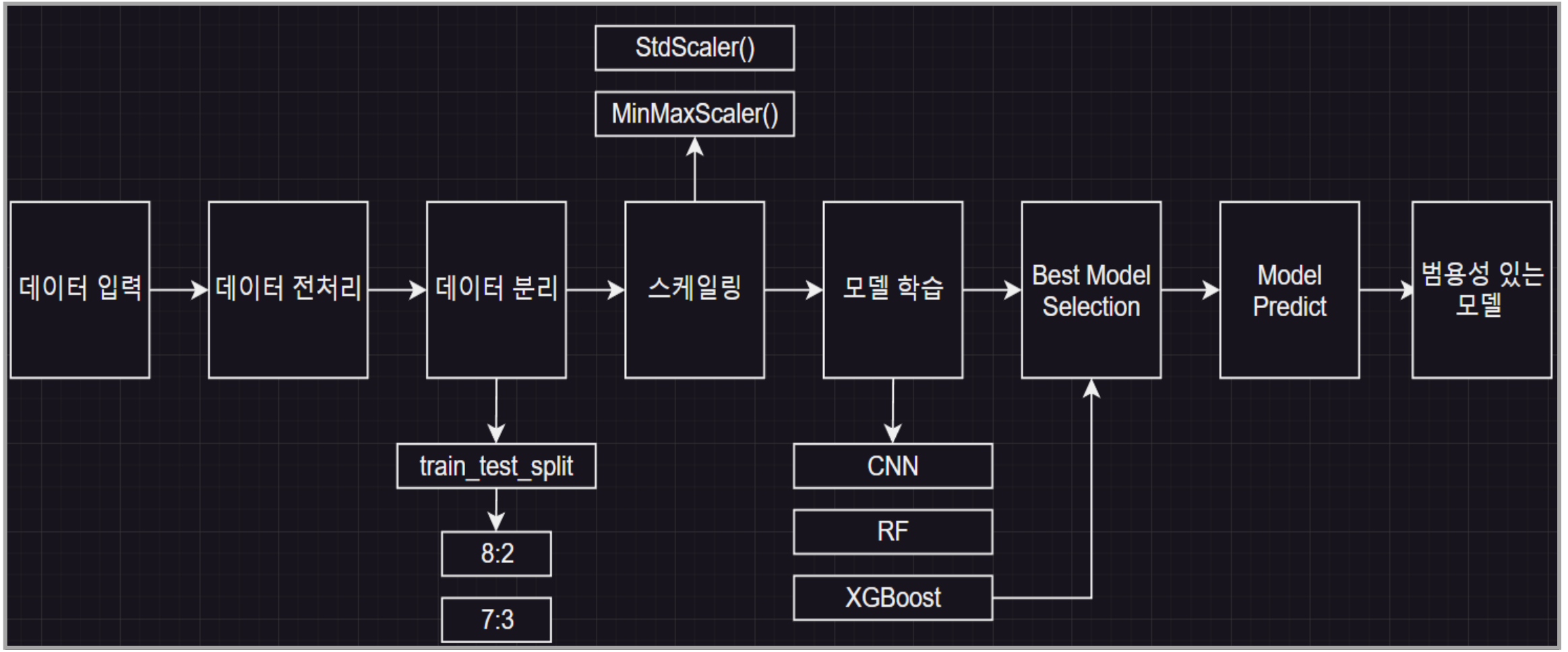
```
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.2)
x_train, x_valid, y_train, y_valid = train_test_split(x_train, y_train,
                                                      test_size=0.2, stratify=y_train,
                                                      random_state=1)
```

< 데이터 분리 >

- 데이터 불균형은 없었지만, 정상과 비정상 데이터의 비율을 맞추기 위해 진행함
- train_test_split 함수와 stratify를 이용해 **원본데이터의 Target비율에 맞춰** 데이터를 분리함

프로젝트 수행 및 과정 – 불량 예측 프레임 워크

< 시각자료 >



프로젝트 수행 및 과정 - 선정한 모델

<Extreme Gradient Boosting>

- Boosting 기법을 이용하는 라이브러리
- Regression, Classification 문제를 모두 지원함
- XGBoost는 자체에 과적합 규제 기능으로 강한 내구성 가짐
- 다양한 옵션을 제공, Customizing 용이함
- 병렬 처리로 학습, 분류 속도가 빠름

프로젝트 수행 및 과정 - 선정한 모델

<Convolutional Neural Network>

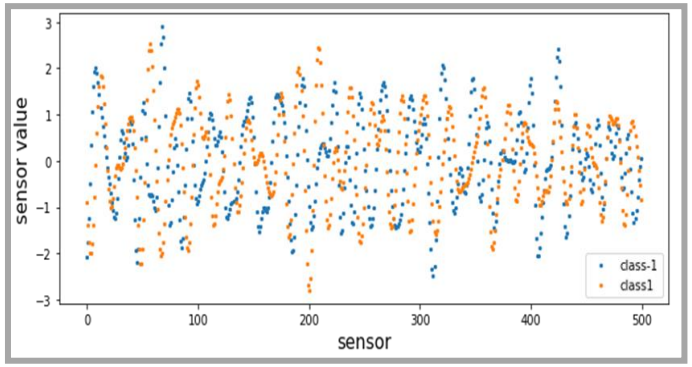
- 주로 이미지나 영상 데이터를 처리할 때 쓰임
- 데이터에서 직접 학습하고 패턴을 사용해 이미지 분류함
- 다양한 옵션을 제공, Customizing 용이함
- 병렬 처리로 학습, 분류 속도가 빠름

프로젝트 수행 및 과정 - 선정한 모델

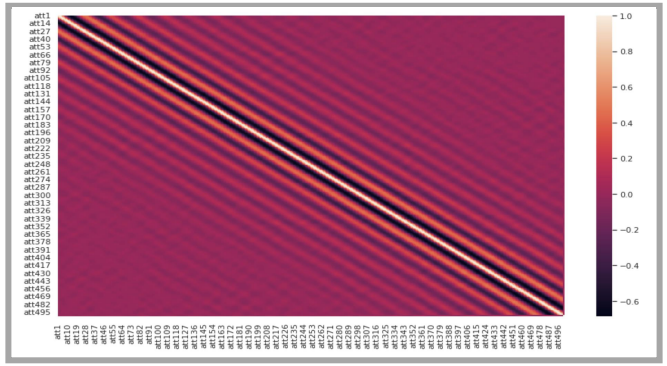
<Random Forest>

- 대용량 데이터 처리에 효과적
- Classification 및 Regression 문제에 모두 사용 가능
- Overfitting 문제를 회피하여 모델 정확도를 향상시킴
- 정확성, 단순성 및 유연성으로 인해 가장 많이 사용되는 알고리즘
- 분류 및 회귀에 사용할 수 있다는 점과 비선형 특성을 결합하면 다양한 데이터 및 상황에 매우 적합함

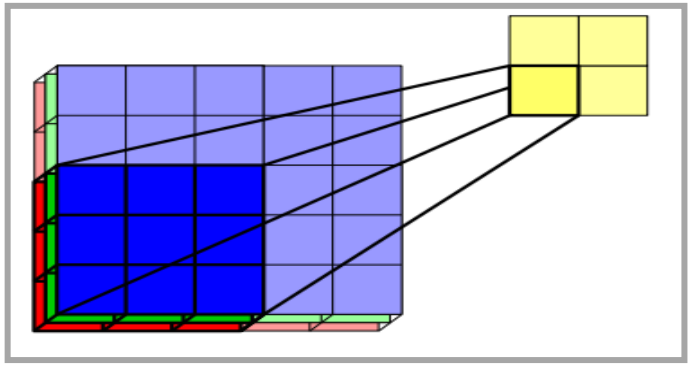
프로젝트 수행 및 과정 – 모델 선정 이유



→ 종속변수의 시각화를 통한 타겟값 확인



가까운 변수들끼리의 관계 유의 ←

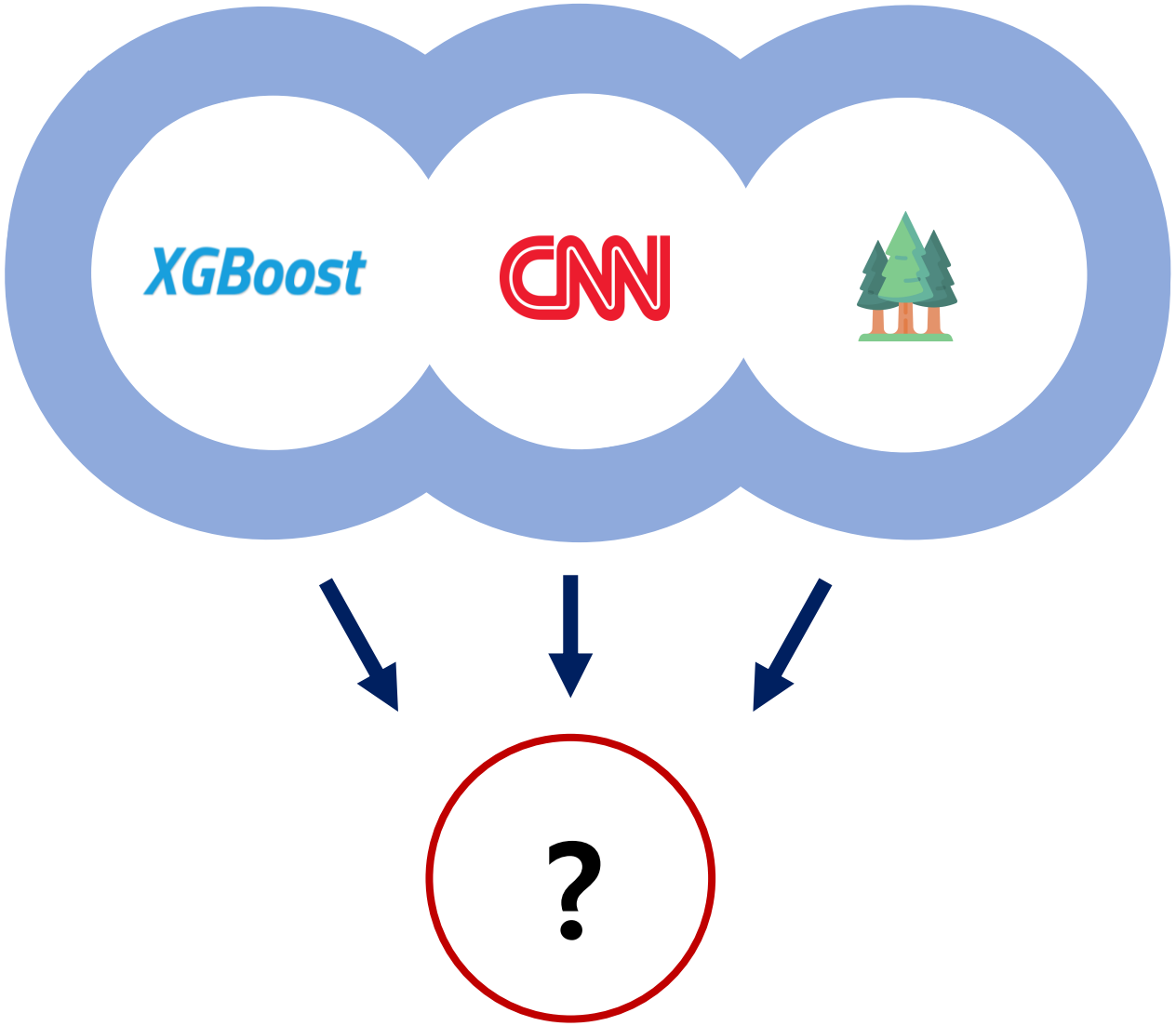


→ 따라서 non-local이 아닌 local한 관계를 잘 활용할 수 있는 모델들 선정

⇒ XGBoost, CNN, 

프로젝트 수행 결과

프로젝트 수행 결과 – Best model selection



- 새로운 데이터가 입력될 때마다
  여러 개의 서로 다른 모델을 학습시킨 후
  최적의 모델을 추론 & 선택
- Classification report, Roc curve,
  Confusion matrix를 통한 검증
- 다양한 데이터를 넣어서 **각 모델의 성능 검증**

프로젝트 수행 결과 – 모델 검증용 새 데이터 수집

Ford 엔진 상태 분류

센서 측정값에 따른
상태 유형을 분류한 데이터

데이터		설명	데이터 타입
구조	Sensor 1 ~ 500	엔진 관련 요소를 측정한 값	숫자
변수	독립변수	센서 데이터 1 ~ 500	숫자
	종속변수	타겟 값, -1, 1	숫자

프로젝트 수행 결과 – 모델 검증용 새 데이터 수집

Bearing 오작동 분류

센서 측정값에 따른
작동유형을 분류한 데이터

데이터		설명	데이터 타입
구조	Sensor 0 ~ 99	엔진 관련 요소를 측정한 값	숫자
변수	독립변수	센서 데이터 0 ~ 99	숫자
	종속변수	타겟 값, 0, 1, 2, 3	숫자

프로젝트 수행 결과 – 모델 검증 Ford-A

- **classification report**

Random Forest

	Precision	Recall	f1-score	Support
0	0.74	0.77	0.75	466
1	0.74	0.71	0.72	435
Accuracy			0.74	901
Macro avg	0.74	0.74	0.74	901
Weighted avg	0.74	0.74	0.74	901

CNN

	Precision	Recall	f1-score	Support
0	0.82	0.80	0.81	681
1	0.79	0.82	0.82	639
Accuracy			0.81	1320
Macro avg	0.81	0.81	0.81	1320
Weighted avg	0.81	0.81	0.81	1320

XGBoost

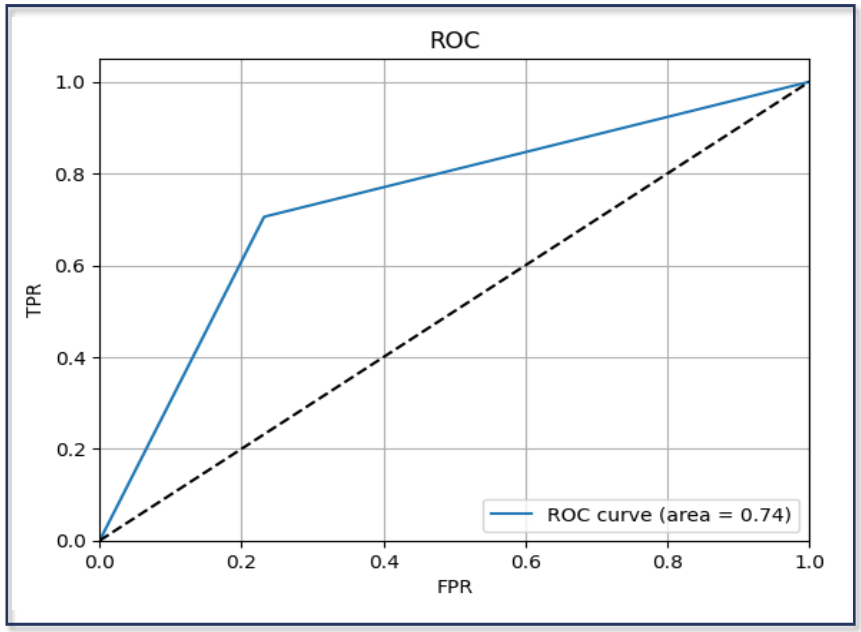
	Precision	Recall	f1-score	Support
0	0.82	0.80	0.81	368
1	0.79	0.82	0.81	353
Accuracy			0.81	721
Macro avg	0.81	0.81	0.81	721
Weighted avg	0.81	0.81	0.81	721

Random Forest < **CNN** = **XGBoost**

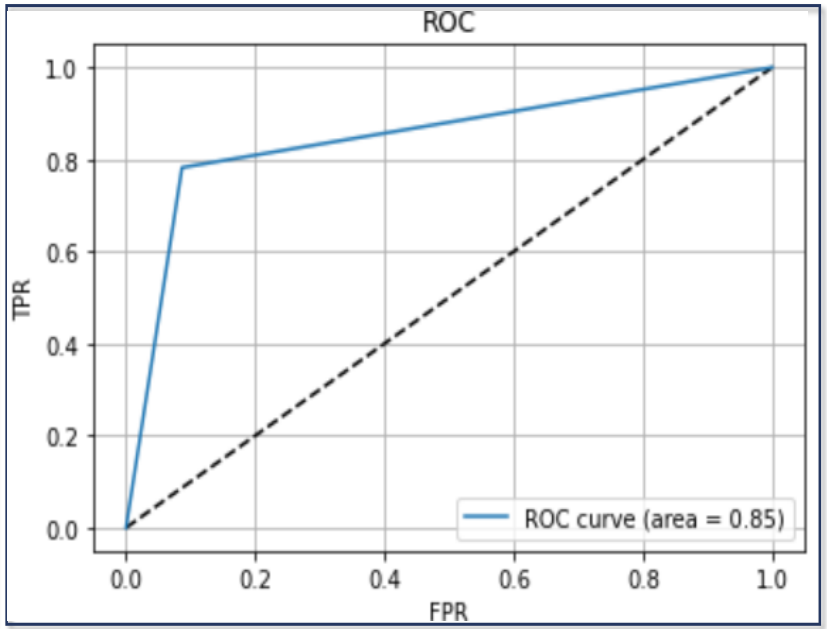
프로젝트 수행 결과 – 모델 검증 Ford-A

- **ROC curve**

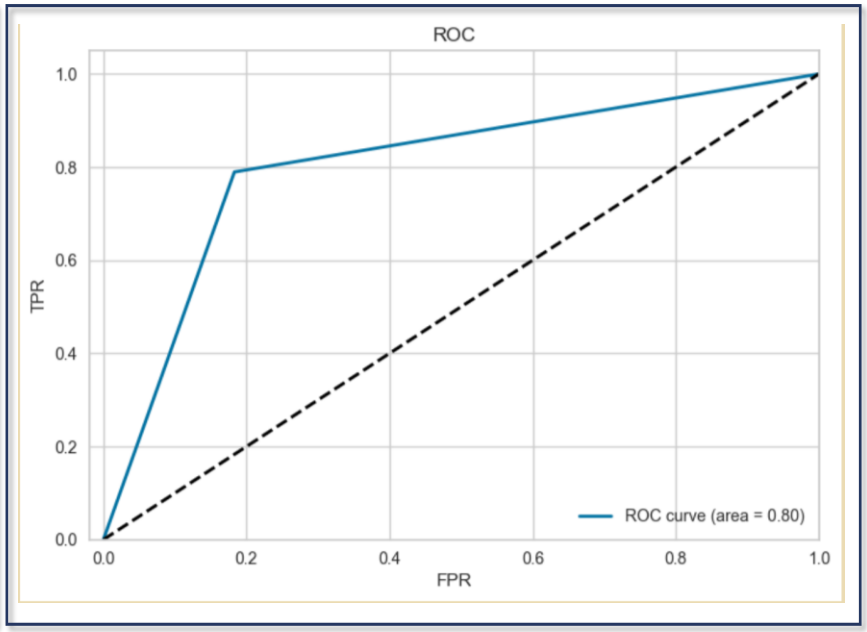
Random Forest



CNN



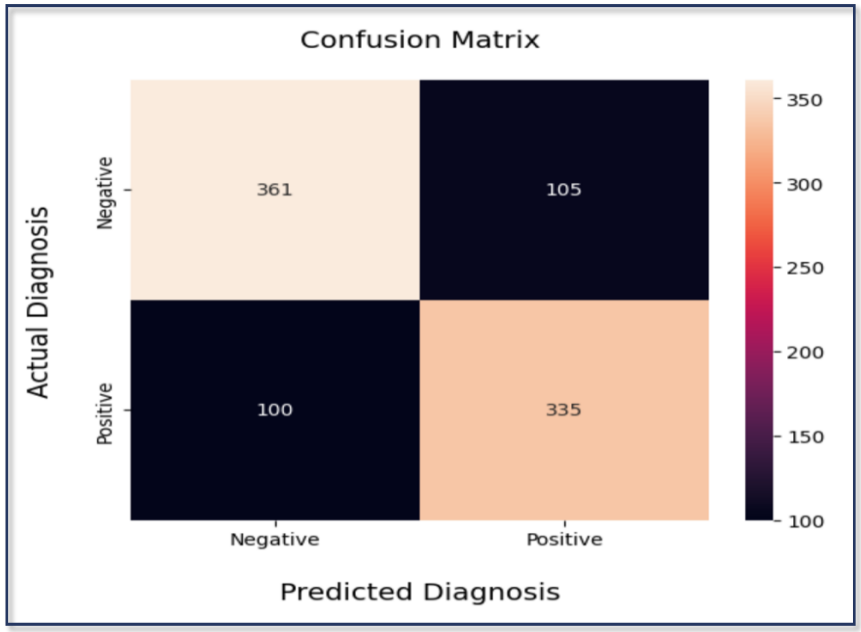
XGBoost



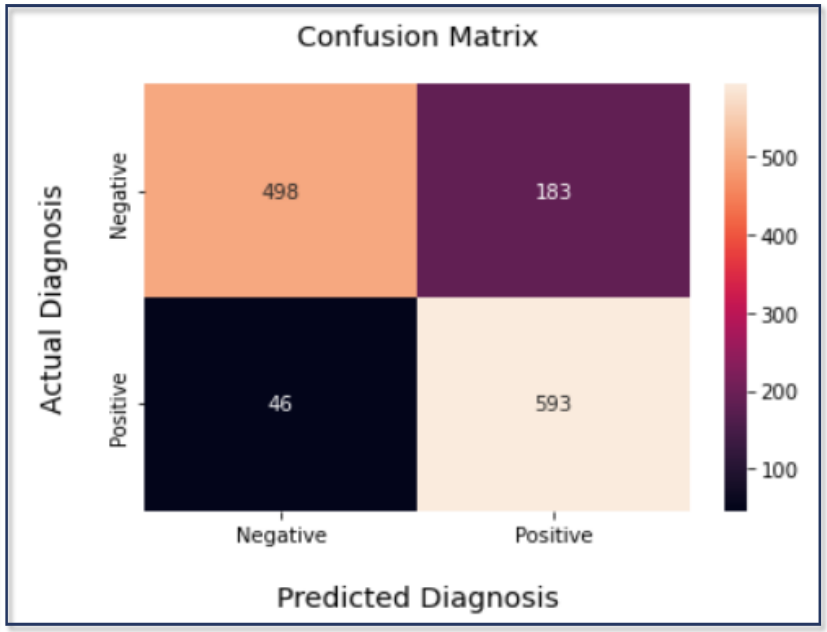
프로젝트 수행 결과 – 모델 검증 Ford-A

- **Confusion matrix**

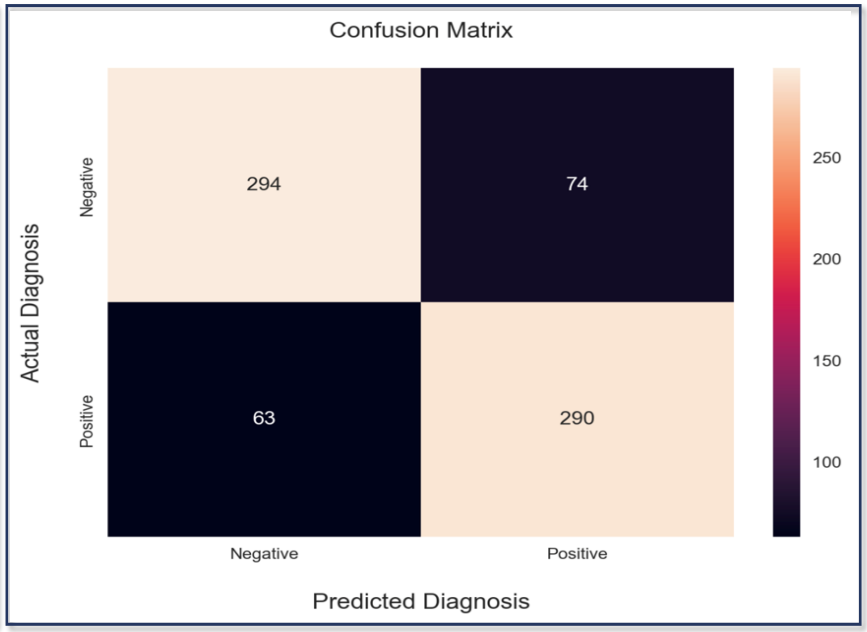
Random Forest



CNN



XGBoost



프로젝트 수행 결과 – 모델 검증 Ford-B

- **classification report**

Random Forest

	Precision	Recall	f1-score	Support
0	0.75	0.81	0.78	467
1	0.78	0.71	0.74	442
Accuracy			0.76	909
Macro avg	0.76	0.76	0.76	909
Weighted avg	0.76	0.76	0.76	909

CNN

	Precision	Recall	f1-score	Support
0	0.74	0.97	0.84	681
1	0.95	0.64	0.76	639
Accuracy			0.81	1320
Macro avg	0.84	0.80	0.80	1320
Weighted avg	0.84	0.81	0.80	1320

XGBoost

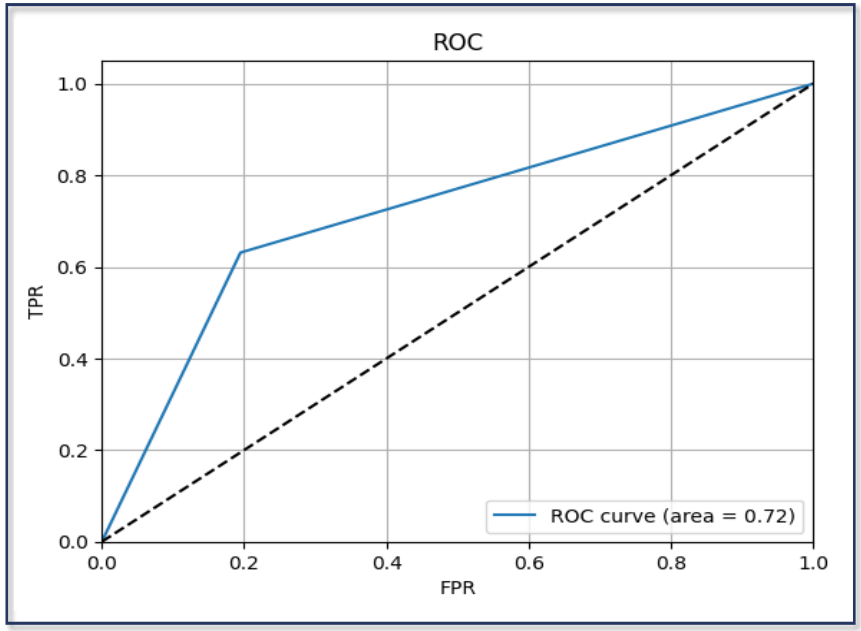
	Precision	Recall	f1-score	Support
0	0.85	0.82	0.83	393
1	0.80	0.83	0.81	335
Accuracy			0.82	728
Macro avg	0.82	0.82	0.82	728
Weighted avg	0.82	0.82	0.82	728

Random Forest < CNN < **XGBoost**

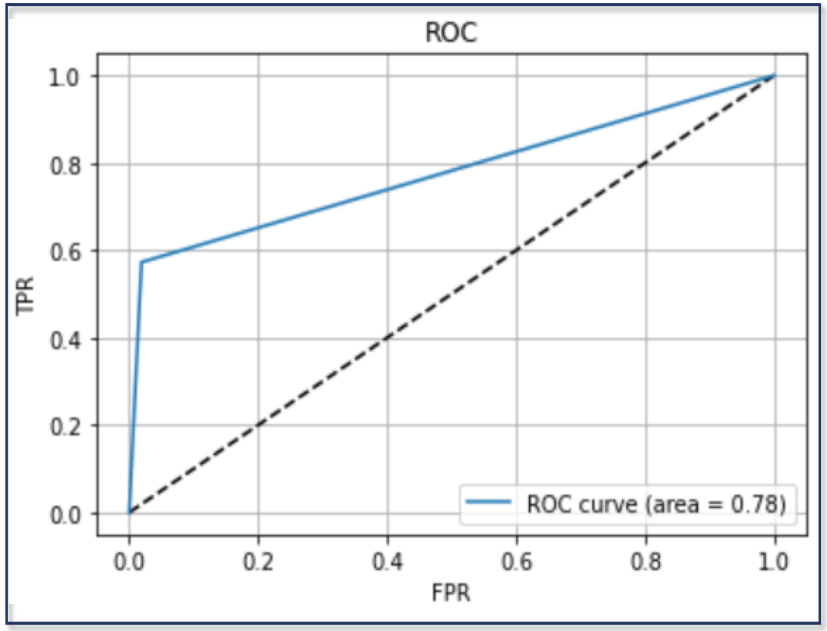
프로젝트 수행 결과 – 모델 검증 Ford-B

- **ROC curve**

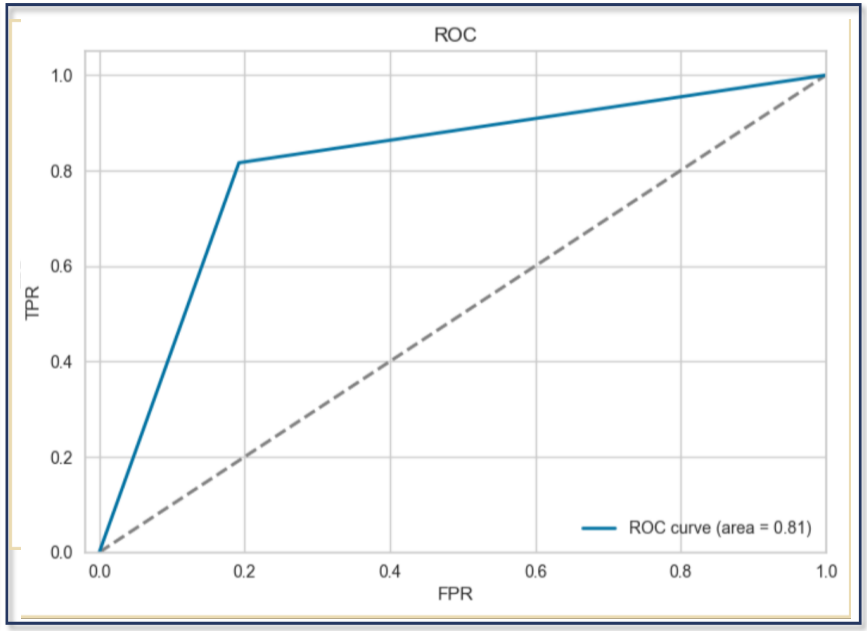
Random Forest



CNN



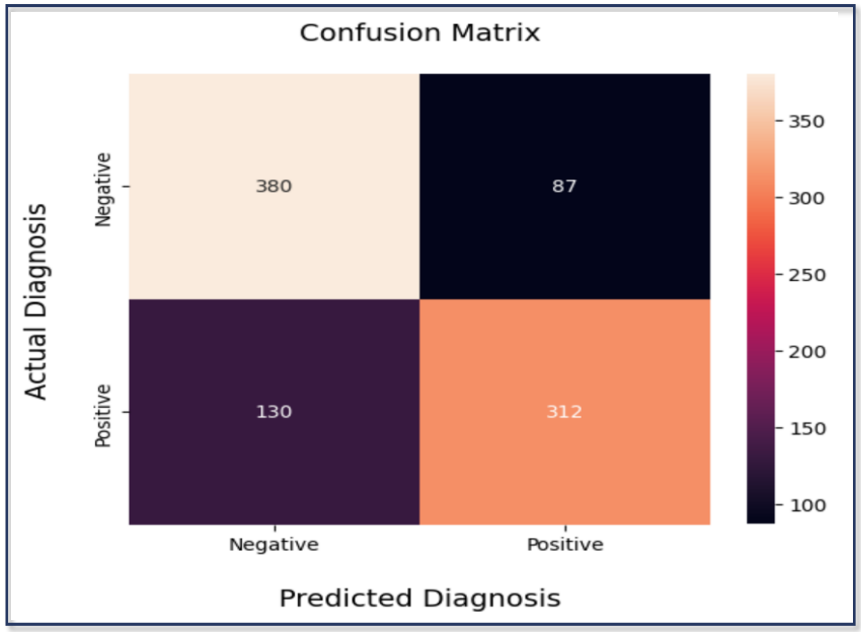
XGBoost



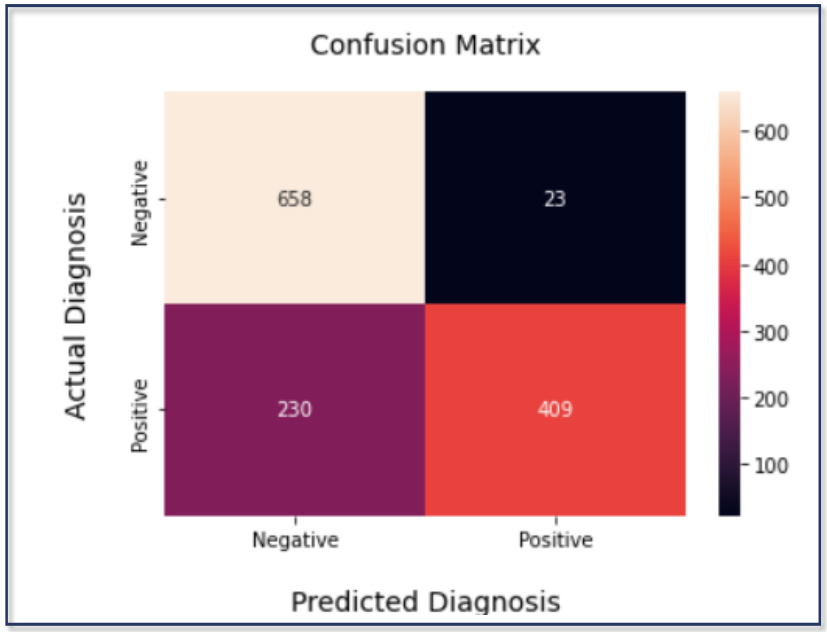
프로젝트 수행 결과 – 모델 검증 Ford-B

- **Confusion matrix**

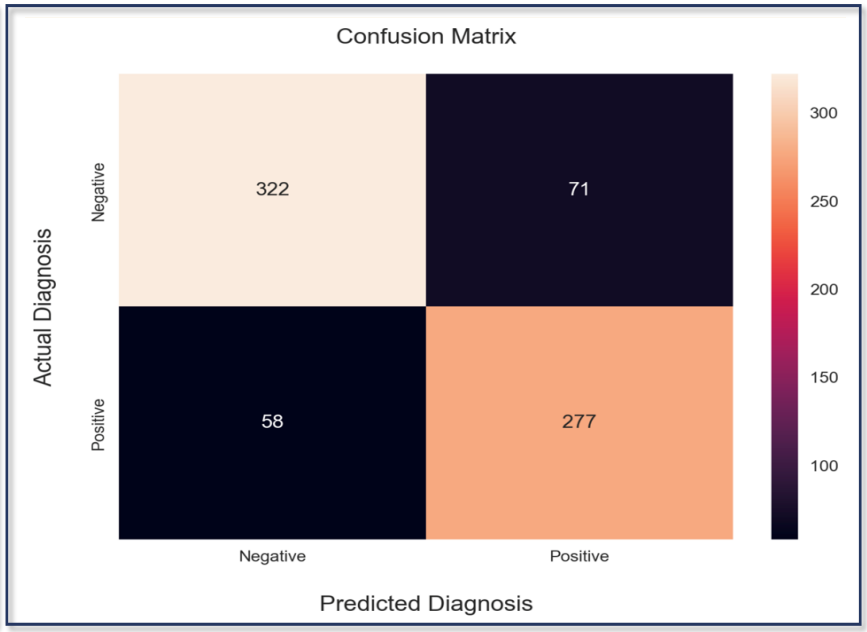
Random Forest



CNN



XGBoost



프로젝트 수행 결과 - 모델 검증 Bearing data

- **classification report**

Random Forest

	Precision	Recall	f1-score	Support
0	0.97	1.00	0.98	1269
1	0.89	0.75	0.81	699
2	0.89	0.65	0.75	515
3	0.74	0.91	0.82	1128
Accuracy			0.86	3750
Macro avg	0.87	0.83	0.84	3750
Weighted avg	0.87	0.86	0.86	3750

CNN

	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	978
1	0.91	0.90	0.91	561
2	0.89	0.90	0.89	538
3	0.92	0.91	0.92	923
Accuracy			0.94	3000
Macro avg	0.93	0.93	0.93	3000
Weighted avg	0.94	0.94	0.94	3000

XGBoost

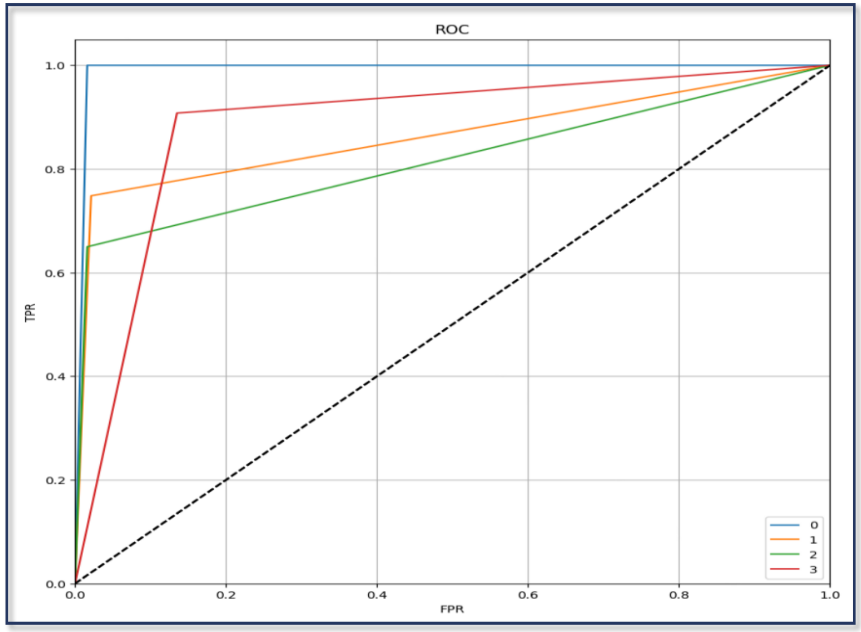
	Precision	Recall	f1-score	Support
0	0.98	1.00	0.99	1024
1	0.85	0.82	0.84	566
2	0.82	0.68	0.74	515
3	0.79	0.87	0.83	895
Accuracy			0.87	3000
Macro avg	0.86	0.84	0.85	3000
Weighted avg	0.87	0.87	0.87	3000

Random Forest < XGBoost < **CNN**

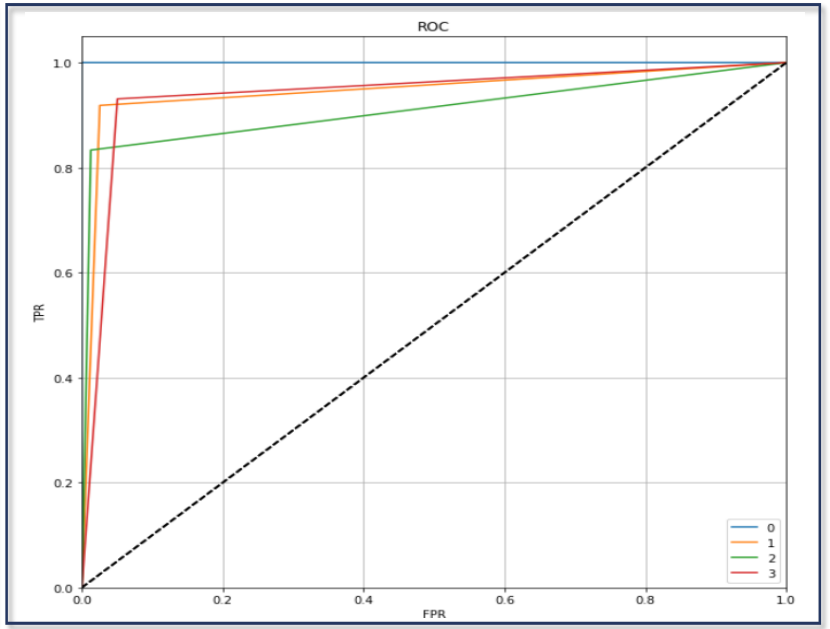
프로젝트 수행 결과 - 모델 검증 Bearing data

- ROC curve

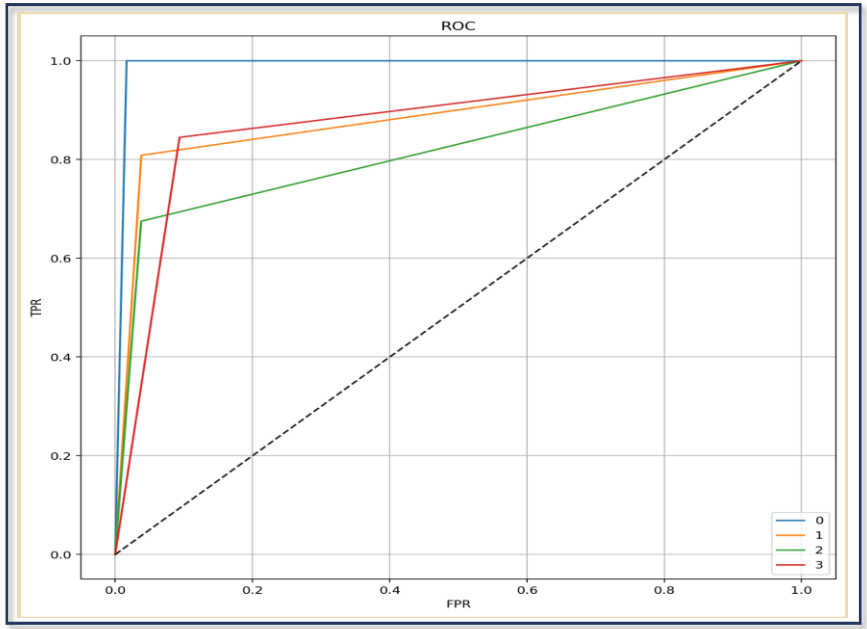
Random Forest



CNN



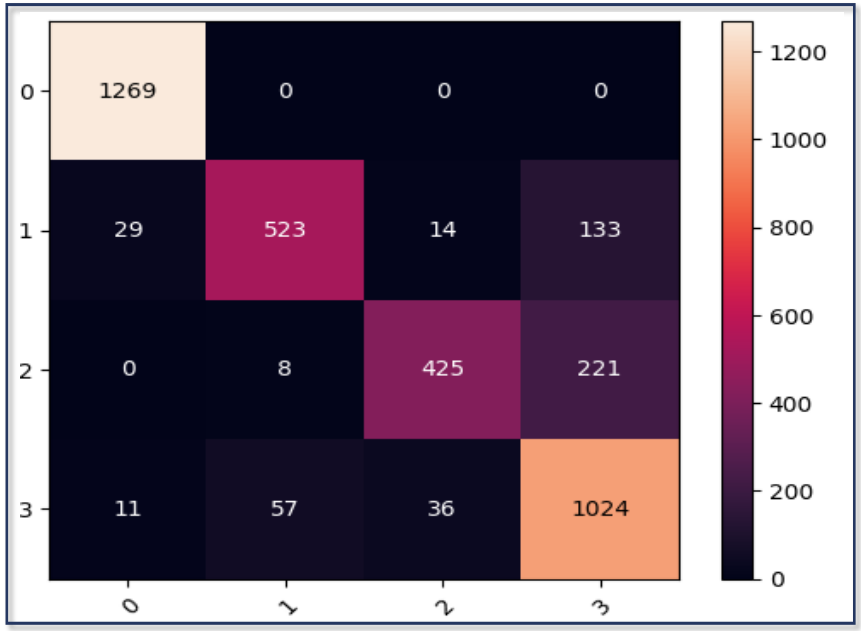
XGBoost



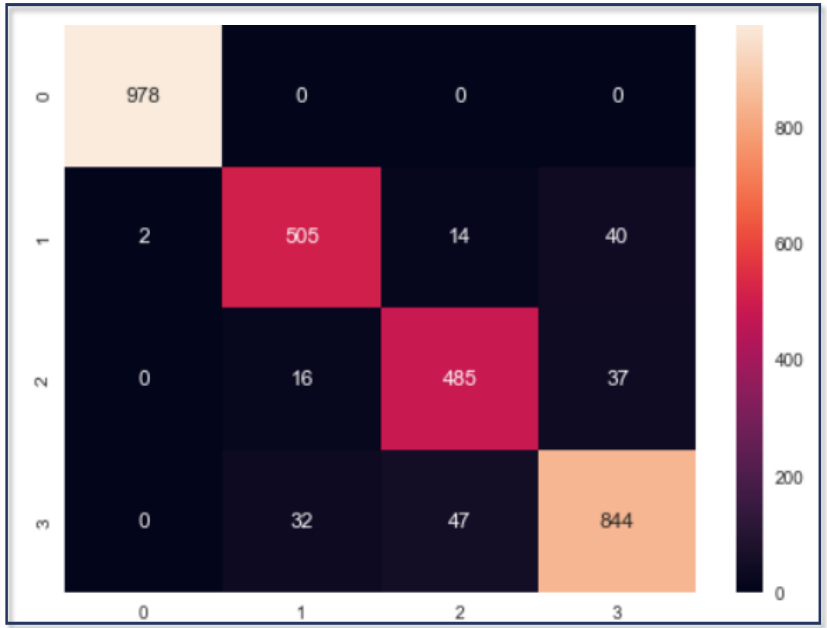
프로젝트 수행 결과 - 모델 검증 Bearing data

- Confusion matrix

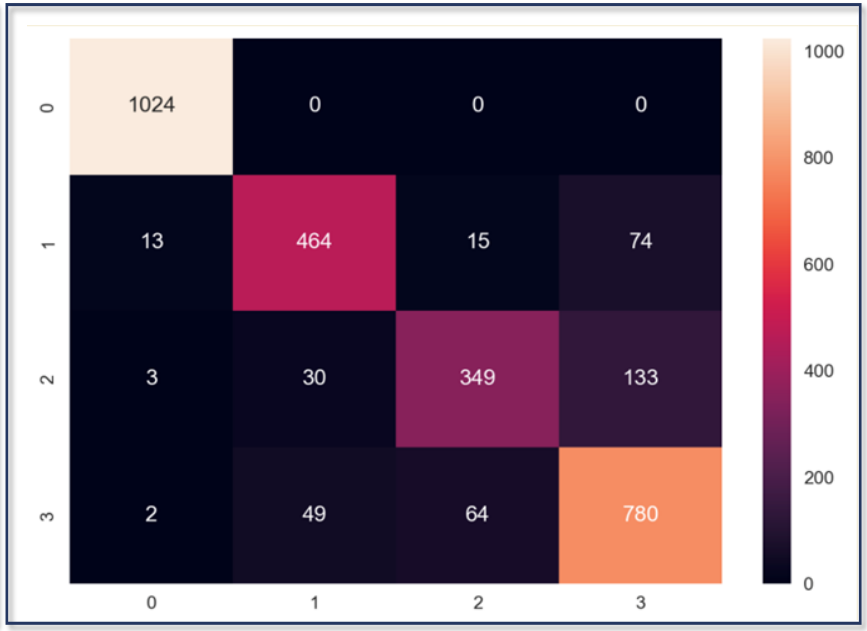
Random Forest



CNN

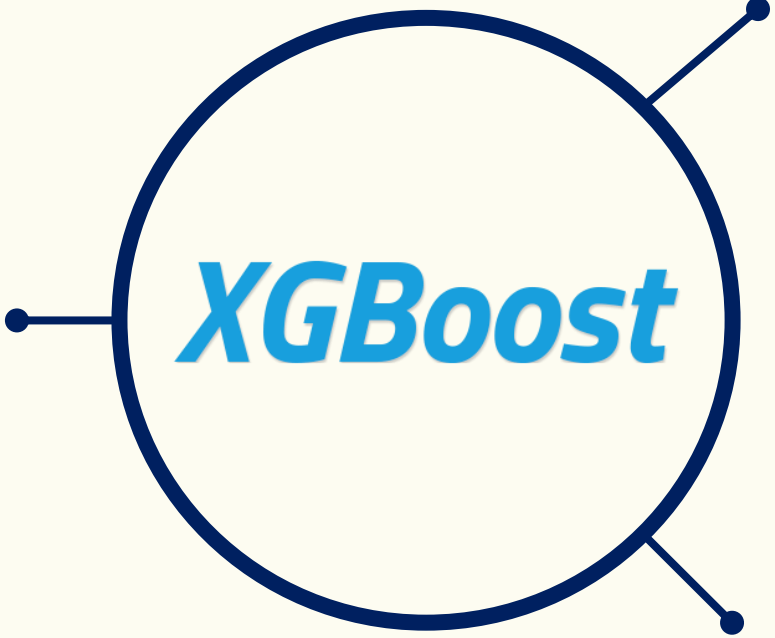


XGBoost



프로젝트 수행 결과 - 범용성 있는 모델 & 모델선택

1. Ford A, Ford B, Bearing data
이 세가지 데이터로 각 모델을 학습



2. Best Model Selection 과정에서,
평균 정확도가 가장 높은 모델
선택하기로 결정

3. 해당 데이터의 경우,
Best Model Selection을 통해 XGBoost 선택

느낀점

활용방안

- 스마트팩토리 환경에서 센서 데이터만 수집할 수 있다면 우리의 이상감지 모델을 활용할 수 있다

배운 점

- 딥러닝 모델의 종류의 다양함과 다양한 센서 데이터셋으로
  이러한 딥러닝 모델들을 활용하고 비교하는 과정을 거치며 모델에 맞는 데이터셋이 있다는 점
- 모델을 선정하기 위해서는, 충분한 데이터에 대한 이해가 필요함을 느낌

해보고 싶은 것

- 추후에 실제 스마트팩토리 설비에서 추출한 raw 데이터들을 가지고 직접 전처리하고 딥러닝 모델에 넣어서 학습하고 예측을 수행해 현장에 직접적인 솔루션을 제공할 수 있는 경험

감사합니다