

---

## 머신 러닝 기반의 연소 설비 모니터링 시스템 구축

---

- 탄소중립 산업현장 문제해결형 디지털 산업혁신 빅데이터 데이터 챌린지-

2022.11.11

- I 서론
- II 제안 사항
- III 분석 결과
- IV 예상 화면
- V 결론
- A 별첨

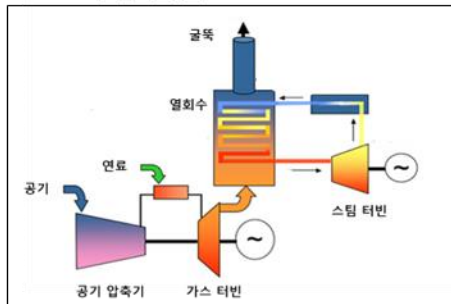
# I. 서론

## 01. 산업 현장의 연소 설비

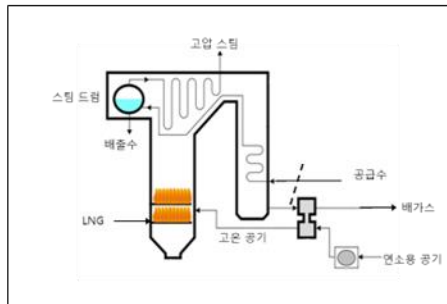
- 보일러, 가열로, 소각로, 고로 등의 연소설비는 다량의 에너지를 사용하고 많은 환경오염물질을 배출함.<sup>1</sup>
- 연소 설비에서 발생하는 에너지 손실을 줄이고 환경 오염을 줄이기 위한 다양한 기술이 개발되고 있음.

<sup>1</sup> 산업부문 39.5%와 발전부문 13.4%로 총 52.9%배출, 미세먼지 관리 종합대책, 2017

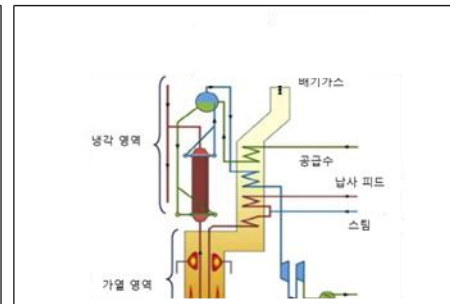
### • 복합발전



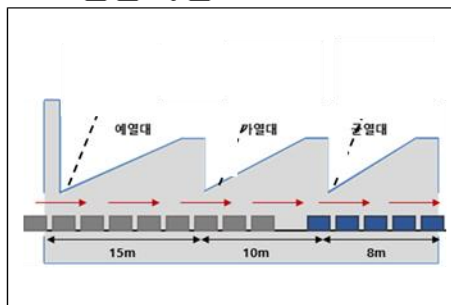
### • 가스 보일러



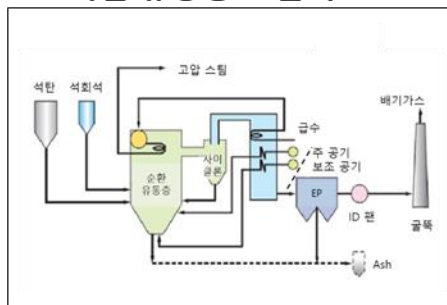
### • NCC 분해로



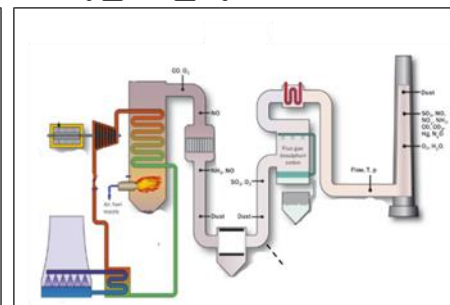
### • 열연 가열로



### • 석탄 유동층 보일러



### • 석탄 보일러

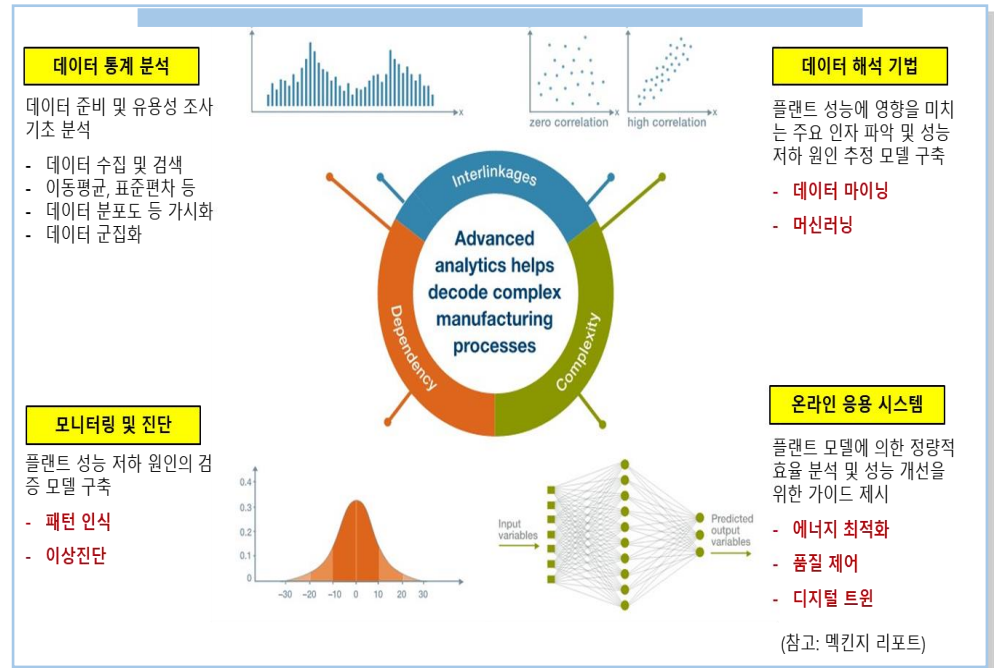
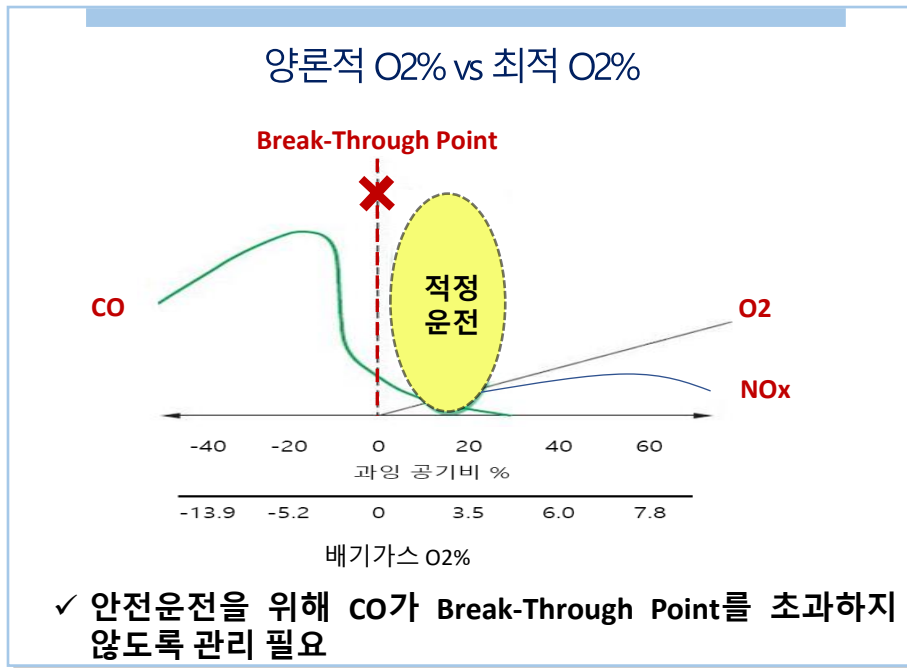


발생한 대기오염 물질은 집진 설비 및 환경정화설비를 통해 배출하는데 이에 대한 감시와 변동에 대한 관리가 필요

# I. 서론

## 02. 온실가스 감축 목표 - 탄소 중립

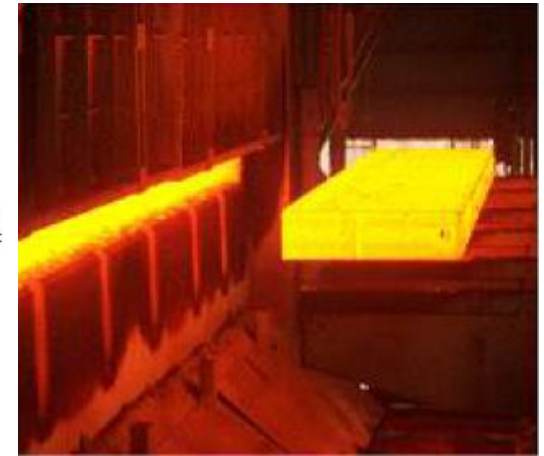
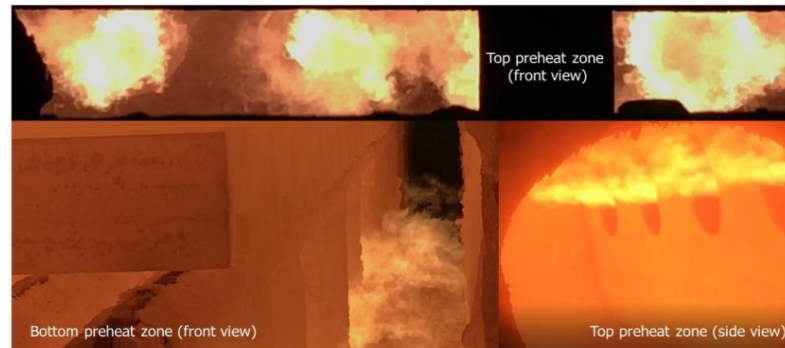
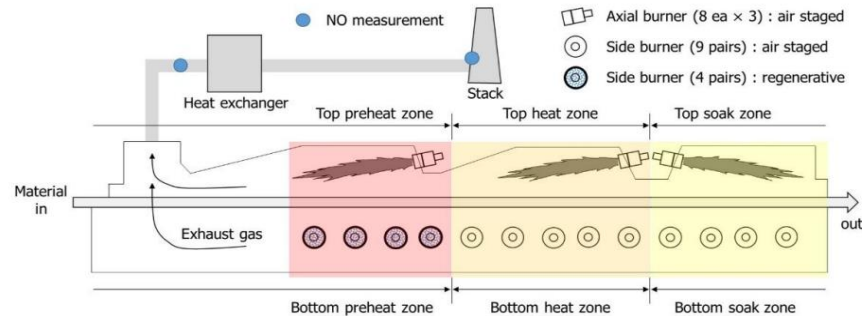
- 온실가스 감축 목표
  - 정부는 기후위기 대응을 위해 2030년까지 40% 감축 및 2050년 까지 탄소중립 달성 계획 수립
  - 혁신적인 생산기술 개발 및 온실가스 배출에 대한 정밀한 관리체제 구축이 필요



에너지 설비의 효율 파악, 절감 잠재력 분석 및 온실가스의 정밀한 배출현황 파악 필요  
조업 데이터와 경험, 정보의 디지털화를 기반으로 에너지 설비의 실시간 효율 관리 필요

## 03. 가열로 공정 분석(1/2)

- 열연 공정은 철강 제조 공장의 연주 공정에서 생산된 슬라브를 일정한 형태로 가공하는 공정 중 하나임.
- 열연 공정의 가열로는 슬라브를 가공하기 쉽도록 1200~ 1500°C까지 가열함.
- 가열에 필요한 연료는 각 대의 상부와 하부의 버너를 통해 공급되며 연소용 공기는 연료의 주입량에 따라 일정한 비율로 조절함.



가열로를 운전하는 동안 온실가스와 질소산화물( $\text{NO}_x$ ) 등의 대기오염 물질이 발생함.

가열로에서 발생한 대기오염 물질은 집진 설비 및 환경정화 설비를 통해 배출하는데 이에 대한 감시와 변동에 대한 관리가 필요

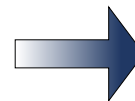
## 03. 가열로 공정 분석(2/2)

- 로 내 온도를 유지할 수 있도록 연료(COG, LNG, LPG 등)와 공기의 유량을 제어함.
- 효율적인 연소를 위하여는 연료량에 맞는 적정량의 공기가 공급
- 연소용 공기량: 구해진 이론 공연비에 과잉 공기비를 곱한 수치를 사용

| 공기비에 따른 공정 변화                                                                                                                                                |                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 공기비 1.0 이하                                                                                                                                                   | 공기비 1.0 이상                                                                                                                                                           |
| <ul style="list-style-type: none"> <li>• 불완전 연소로 실열 증가</li> <li>• 불완전 연소로 미연 발생 가스 폭발 사고 위험</li> <li>• 소재 스케일 박리성 불량</li> <li>• 미연소에 의한 연료 소비량 증가</li> </ul> | <ul style="list-style-type: none"> <li>• 연소 온도 저하</li> <li>• 피가열물의 전열 성능 저하</li> <li>• 연소 가스 증가에 의한 폐손실열 증가</li> <li>• 저온 부식 발생</li> <li>• 탈탄, 스케일 생성량 증가</li> </ul> |

| 로압 제어                                                                                  |                                                                                           |
|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| 로압 클 경우                                                                                | 로압 작을 경우                                                                                  |
| <ul style="list-style-type: none"> <li>• 폐가스가 로 틈 사이로 새어나와 구조물 손상 및 열 손실 발생</li> </ul> | <ul style="list-style-type: none"> <li>• 외부 공기 침입 → 소재 산화로 인한 스케일 생성량 증대 및 열손실</li> </ul> |

공정 분석을 통한 변수의 이해  
(공기비, 로압, 온도, 연료량, 공기량)



관리 필요 대상 식별 및 중요 변수 인자 추정

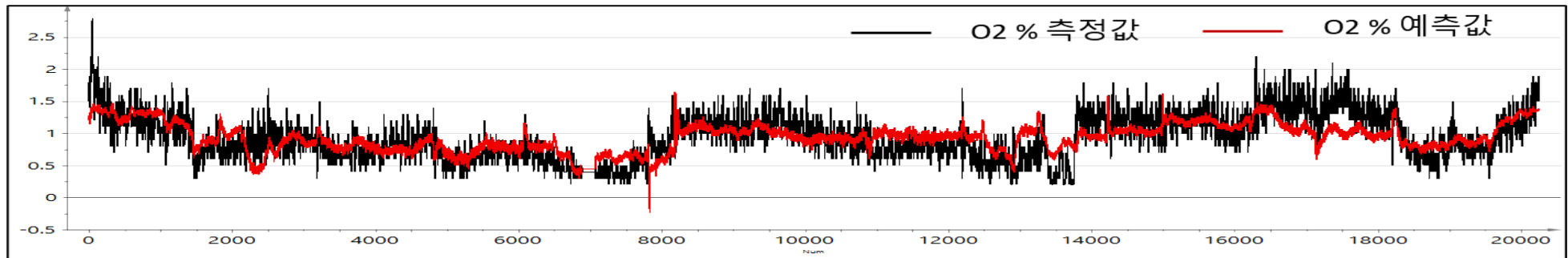
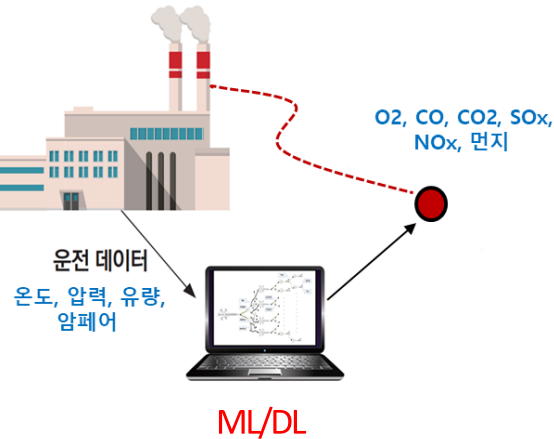
- I 서론
- II 제안 사항
- III 분석 결과
- IV 예상 화면
- V 결론
- A 별첨

## II. 제안 사항

### 01. 배기가스 농도 예측 모델

- 데이터 기반의 연소 설비 모니터링 시스템을 구축하고자 함.
- 연소 설비 관리를 위하여 연소 상태(배기가스 농도)를 타깃으로 하는 예측 모델을 개발하고자 함.
- 예측 모델을 이용하여 배기가스 배출 현황을 파악하고 연소 설비의 이상 유무를 파악할 수 있음.

제조 공장 내 온도, 유량, 압력과 같은 물리적인 센서가 만들어낸 데이터를 머신러닝/딥러닝 기법으로 배기가스 농도를 예측



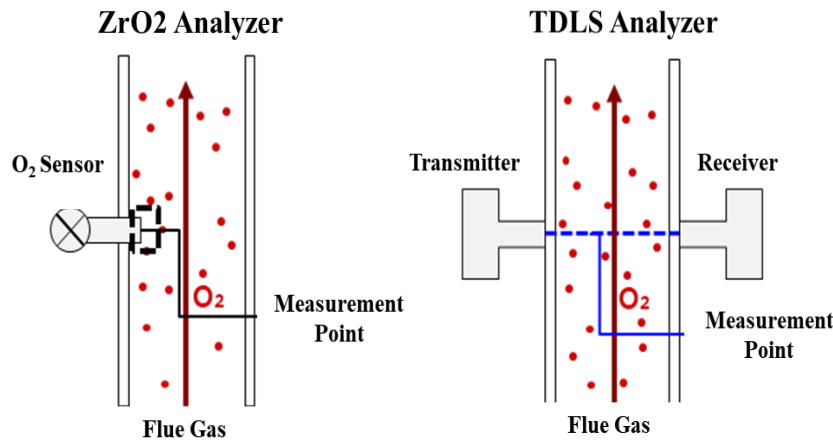


## II. 제안 사항

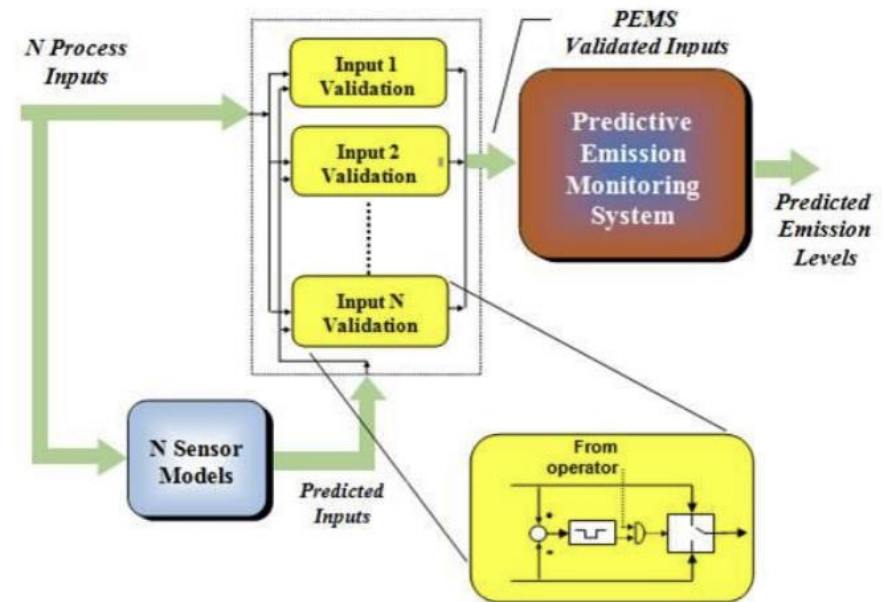
### 02 예측 모델의 필요성

- 공장의 비상가동정지 시스템의 입력 변수이거나, 제품의 품질, 또는 안전과 관련된 측정값은 운전에서 매우 중요함.
- 센서를 이중 또는 삼중으로 설치하여 센서의 고장으로 인한 오작동이나 오판을 방지해야 함.

배기가스 농도 계측 센서의 종류 예시



|       |                                            |                                 |
|-------|--------------------------------------------|---------------------------------|
| 분석 방법 | 대기와 배관 내 O <sub>2</sub> 농도 차에 따른 기전력 발생 활용 | 적외선 영역의 파장형 레이저를 이용한 흡수 스펙트럼 분석 |
| 장점    | 낮은 가격                                      | 시간 응답성과 정밀도가 높음<br>설치 및 유지보수 용이 |
| 단점    | 국소값 측정, 느린 응답속도                            | 높은 가격                           |



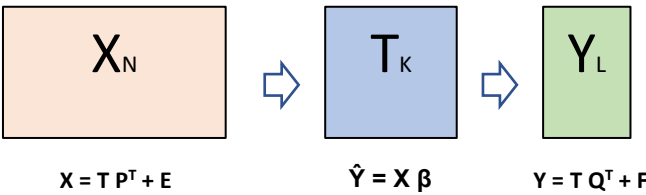
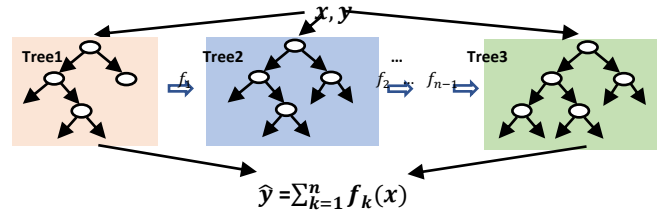
G. Ciarloa, E. Bonicab, B. Bosio, and N. Bonavita, "Assessment and Testing of Sensor Validation Algorithms for Environmental Monitoring Applications," Chemical Engineering Transactions, vol. 57, pp.331-336, March 2017  
Fig. 1

# II. 제안 사항

## 03. 예측 모델 수립

- 다양한 머신러닝/딥러닝 기법 중에서 배기가스 농도 예측에 적합한 회귀 모델 개발
- 부분최소제곱법(PLS), XGBoost, LightGBM, RandomForest 등의 다양한 예측 기법을 적용해보고, 성능이 우수한 기법 선택/ 조합하여 예측 모델을 고도화 하고자 함.

정형 데이터에서 강력한 효과를 보이는 예측 모델 종류 예시

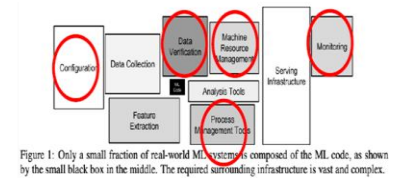
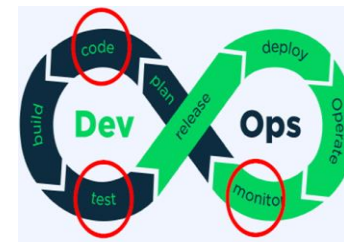
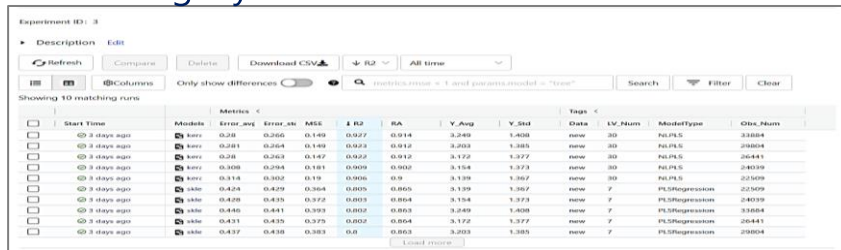
| 기법      | 도식화                                                                                            | 설명                                                              |
|---------|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| PLS     | <div></div>  | 잠재변수를 이용하여 예측식을 구하는 기법<br>선형식이지만 노이즈가 포함된 데이터를 처리하는 장<br>점이 있음. |
| XGBoost | <div></div> | 앙상블을 이용하는 예측식을 구하는 기법                                           |

## II. 제안 사항

### 04. 재학습 시스템 구축

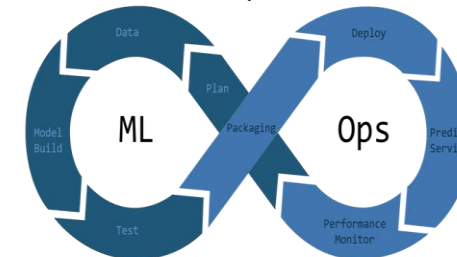
- 제조 공장에서는 시간에 따른 변화(설비 노후화, Fouling, 환경 등)로 인해 머신러닝 모델의 성능이 저하됨
- 재학습 시스템을 도입하여 모델 성능 저하를 꾸준히 관찰하고, 모델의 자동 업데이트 가능하게 하고자 함.

#### Model Registry

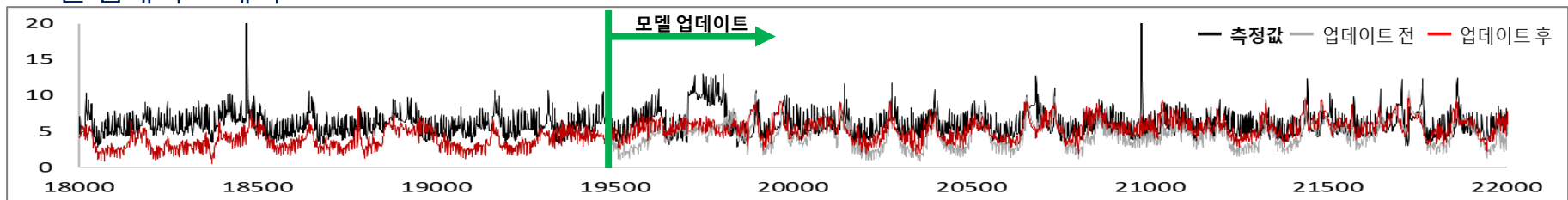


#### 업데이트 전 후 모델의 성능 비교

| 구분        | 업데이트 전 | 업데이트 후 |
|-----------|--------|--------|
| 예측력(결정계수) | 62.08% | 79.75% |
| 오차 절대값 평균 | 1.01   | 0.64   |
| 실측 평균     | 4.72   | 5.14   |
| 상대 정확도    | 78.56% | 87.59% |



#### 모델 업데이트 예시



- I 서론
- II 제안 사항
- III 분석 결과
- IV 예상 화면
- V 결론
- A 별첨

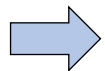
# III. 분석 결과

## 01. 데이터 수집 및 결측치 처리

- 수집된 데이터는 07월24일~12월22일까지의 01분간격의 총 218,478 개의 데이터 있음.
  - 학습 데이터: 07월 24일 ~ 10월 31일 144,000개
  - 시험 데이터: 11월 01일 ~ 12월 22일 74,478개
- 데이터 중 주요 변수를 설정하여 예측(회귀) 모델을 개발하고자 함.

### 연소가스 배기가스 분석기

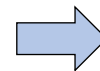
| Description |
|-------------|
| CO 측정값 1    |
| O2 측정값 1    |
| O2 측정값 2    |
| O2 측정값 3    |
| O2 (배기구)    |



주요 예측 타겟  
변수 설정

### 가열로 내 계측 센서

| Description | Description |
|-------------|-------------|
| 공정변수 1      | 공정변수 6      |
| 공정변수 2      | 공정변수 7      |
| 공정변수 3      | 공정변수 8      |
| 공정변수 4      | 공정변수 9      |
| 공정변수 5      | 공정변수 10     |



총 10개 그룹의  
가열로 내 주요  
센서 변수

## 02. 데이터 결측치 처리

- 목표하고자 하는 예측 모델은 회귀 모델이므로 수집된 데이터에서 결측치가 있는 행을 모두 제거함.  
(데이터는 모두 float64형의 연속형 데이터임.)

### 결측치 개수

| 학습데이터    | nan | 학습데이터    | nan | 학습데이터  | nan | 시험데이터    | nan |
|----------|-----|----------|-----|--------|-----|----------|-----|
| 공정변수 2-1 | 45  | 공정변수 4-1 | 108 | 공정변수 6 | 257 | O2 측정값 2 | 42  |
| 공정변수 2-2 | 45  | 공정변수 4-2 | 108 | 공정변수 7 | 37  | CO 측정값   | 210 |
| 공정변수 2-3 | 45  | 공정변수 4-3 | 108 | 공정변수 8 | 505 |          |     |
| 공정변수 2-4 | 45  | 공정변수 4-4 | 108 |        |     |          |     |
| 공정변수 2-5 | 108 | 공정변수 4-6 | 108 |        |     |          |     |
| 공정변수 2-6 | 45  | 공정변수 4-8 | 108 |        |     |          |     |
| 공정변수 2-7 | 108 |          |     |        |     |          |     |
| 공정변수 2-8 | 45  |          |     |        |     |          |     |

학습 데이터  
제거 전/(144000, 41)  
제거 후 (143495, 41)

시험 데이터  
제거 전 (74478, 41)  
제거 후 (74231, 41)

# III. 분석 결과

## 03. 데이터 전처리 : Bandwidth 필터링

- 가상센서에 사용되는 변수에 대하여 이상값을 초기에 제거하기 위하여 Bandwidth 기법 사용.
- 학습 데이터를 기준으로 High/ Low 기준을 결정함.

Bandwidth Filtering Table

| Tag      | Bandwidth | High  | Low  |
|----------|-----------|-------|------|
| 공정변수 1-1 | O         | 35000 | 4500 |
| 공정변수 1-2 | O         | 35000 | 3000 |
| 공정변수 1-3 | O         | 18000 | 1800 |
| 공정변수 1-4 | O         | 18000 | 3000 |
| 공정변수 1-5 | O         | 2800  | 500  |
| 공정변수 1-7 | O         | 3600  | 600  |
| 공정변수 2-1 | O         | 2.2   | 0.8  |
| 공정변수 2-2 | O         | 2.3   | 0.8  |
| 공정변수 2-3 | O         | 2     | 0.8  |
| 공정변수 2-4 | O         | 2.4   | 0.8  |
| 공정변수 2-8 | O         | 2     | 0.8  |
| 공정변수 4-1 | O         | 7300  | 600  |
| 공정변수 4-2 | O         | 7300  | 550  |
| 공정변수 4-3 | O         | 3600  | 400  |
| 공정변수 4-4 | O         | 3800  | 300  |
| 공정변수 4-5 | O         | 550   | 70   |
| 공정변수 4-8 | O         | 2200  | 200  |
| 공정변수 4-9 | O         | 20000 | 3000 |

### 수식

If  $X \geq \text{High}$ :

$X = \text{High}$

If  $X \leq \text{Low}$ :

$X = \text{Low}$

### 코드

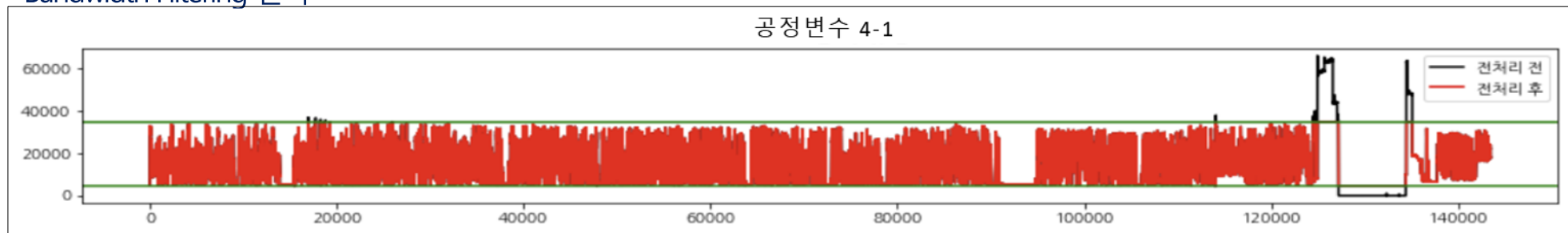
```
import copy

filtered = copy.deepcopy(train_data)
for tag in list(tag_desc['Tag'].values):

    tag_index = list(tag_desc['Tag'].values).index(tag)
    high = tag_desc['High'][tag_index]
    low = tag_desc['Low'][tag_index]

    filtered.loc[train_data[tag]<=low, tag]=low
    filtered.loc[train_data[tag]>=high, tag]=high
```

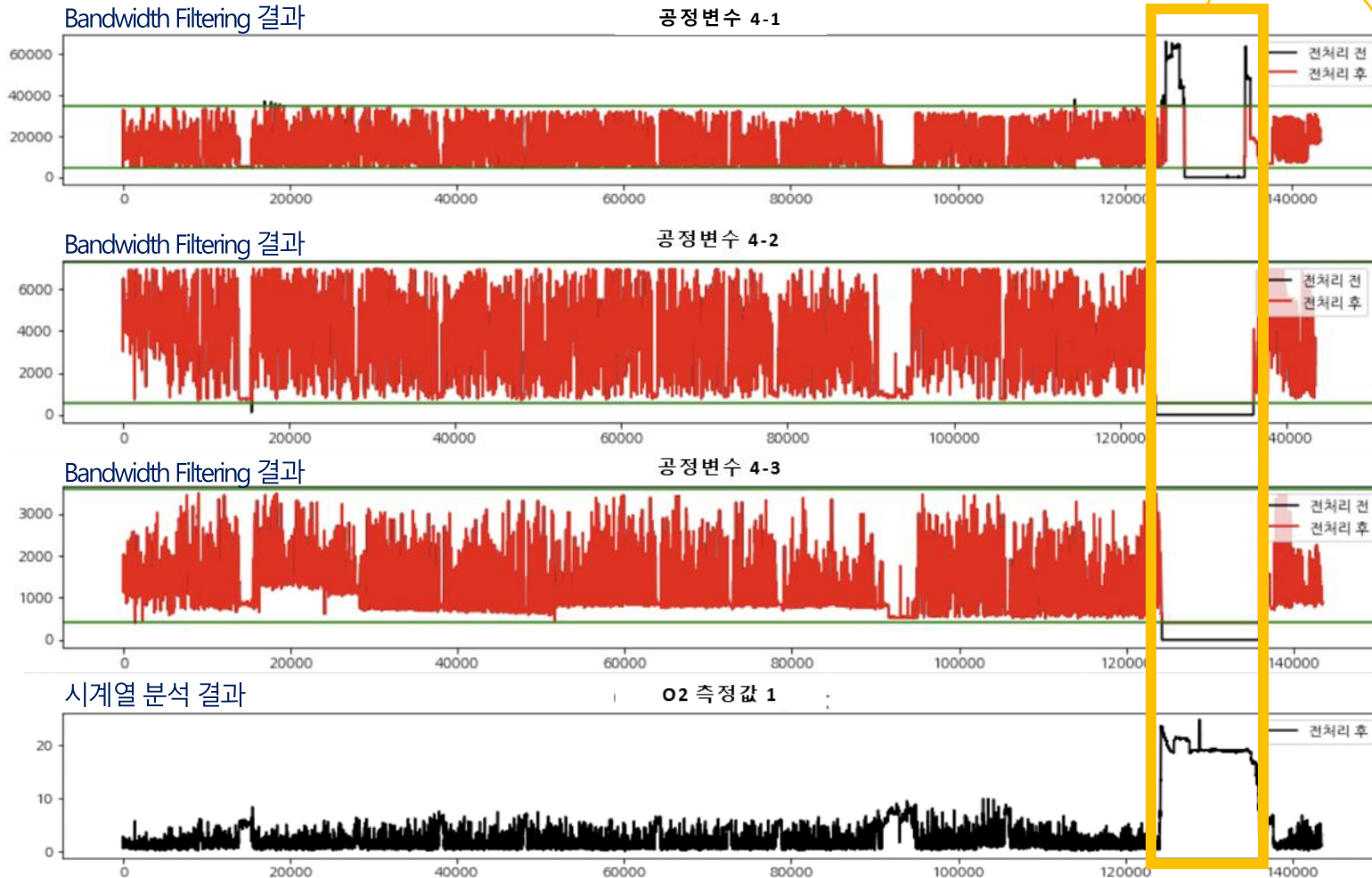
Bandwidth Filtering 결과



# III. 분석 결과

## 04. 데이터 전처리 : 이상 구간 제거

- 공정 변수 중 그룹 4의 변수가 이상치로 기록된 구간을 시계열 분석을 통하여 구분함.(타 변수도 동일한 구간 동안 이상치를 기록함)
- 해당 구간 동안 O2 측정값 1도 이상치임을 확인하고 이상 구간으로 규정/전체 데이터에서 제외함.



### 산업 현장 내 시스템의 적용 사례

데이터 이상 구간 발견

이상 구간  
인덱스 확인

10/11 ~ 10/26  
대략 2주 정도 **모든**  
공정 변수 4가 이상  
구간을 보임

현장 근무자 확인  
및  
공정 이슈 확인  
**필요**

데이터 분석을 통한  
공정 해석

# III. 분석 결과

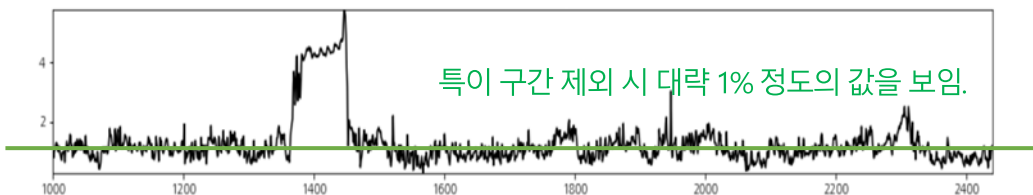
## 05. 데이터 분석(1/2)

- 예측 변수인 O2, CO 농도에 대한 기초 통계량 분석 및 시계열 분석을 실시하였음.
  - O2: O2 측정값 1가 가장 높은 평균 농도 값을 보이고, O2 측정값 3, O2 측정값 1 순으로 높았음.
  - CO: 데이터의 분포가 5~12000 ppm 정도로 매우 넓었음(표준편차:395)

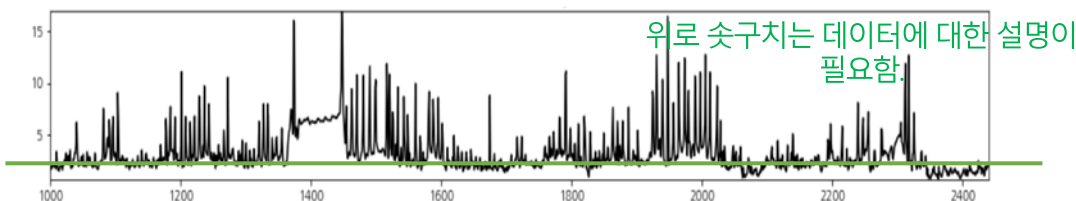
학습 데이터 기초 통계량

|       | O2 측정값 1  | O2 측정값 2  | O2 측정값 3  | CO 측정값    |
|-------|-----------|-----------|-----------|-----------|
| count | 131508.00 | 131508.00 | 131508.00 | 131508.00 |
| mean  | 1.83      | 5.49      | 3.69      | 32.78     |
| std   | 1.56      | 4.45      | 1.53      | 107.43    |
| min   | 0.05      | 0.12      | 0.13      | 5.79      |
| 25%   | 0.96      | 2.46      | 2.77      | 11.57     |
| 50%   | 1.26      | 3.57      | 3.49      | 12.98     |
| 75%   | 1.84      | 7.33      | 4.34      | 13.51     |
| max   | 14.69     | 23.76     | 21.87     | 4592.12   |

O2 측정값 1



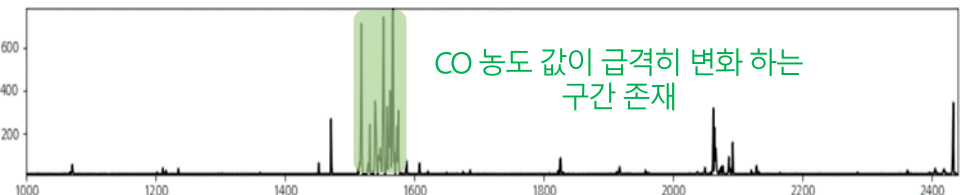
O2 측정값 2



시험 데이터 기초 통계량

|       | O2 측정값 1 | O2 측정값 2 | O2 측정값 3 | CO 측정값   |
|-------|----------|----------|----------|----------|
| count | 74231.00 | 74231.00 | 74231.00 | 74231.00 |
| mean  | 1.88     | 5.68     | 5.17     | 64.78    |
| std   | 1.76     | 3.66     | 1.67     | 187.65   |
| min   | 0.05     | 0.36     | 0.48     | 8.18     |
| 25%   | 0.94     | 3.26     | 4.17     | 11.57    |
| 50%   | 1.29     | 4.63     | 5.06     | 12.92    |
| 75%   | 2.06     | 7.04     | 5.92     | 26.04    |
| max   | 16.72    | 23.76    | 22.32    | 4561.02  |

CO 측정값



1. 공정 데이터를 통한 타겟 변수에 대한 이해

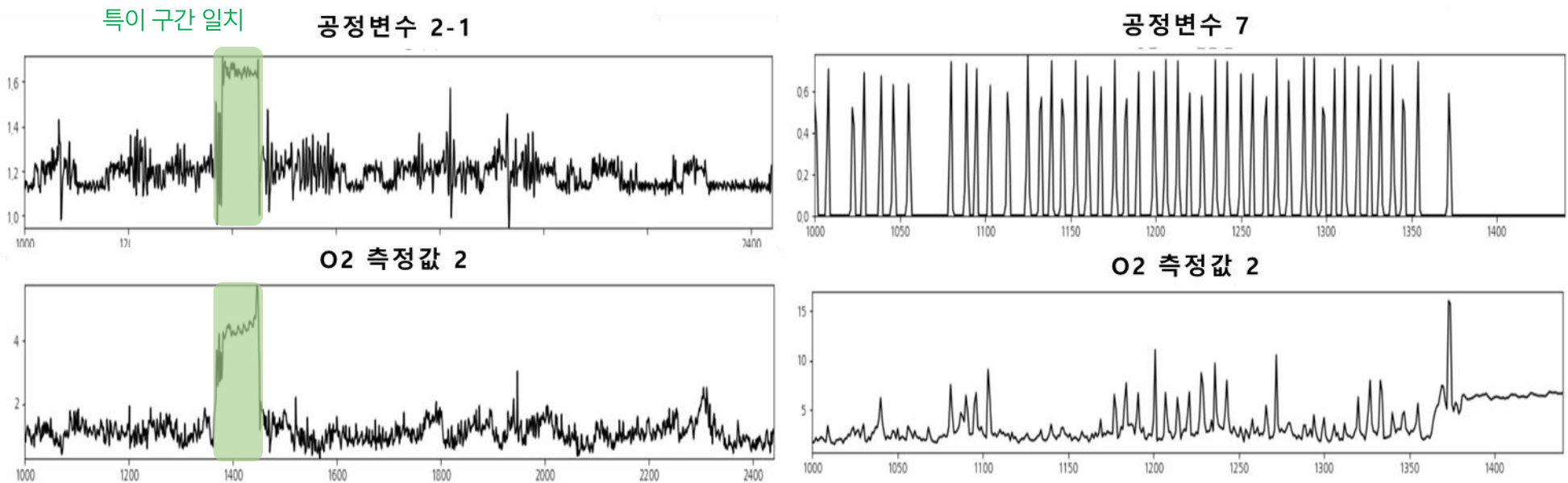
2. 예측 변수 별 변수 선택 필요



# III. 분석 결과

## 05. 데이터 분석(2/2)

- O2 측정값 1, O2 측정값 2의 특이구간은 공정 변수 그룹 2와 그 양상을 같이하여 중요 변수로 삼고자 함.
- O2 측정값 2의 Peak에 대하여 공정 계측 센서 데이터에서 관련 변수를 찾았고 이를 중요 변수로 삼고자 함.
  - 공정변수 7와 O2 측정값 2 측정값의 피크가 일치함.



공장 데이터 분석을 통하여 중요 변수를 추정할 수 있었음.

해당 변수를 예측 모델에 입력 변수로 활용하고자 함.

### III. 분석 결과

#### 06. 모델 수립(1/3)

- 4개의 O<sub>2</sub>, CO 분석기에 대해 예측 모델을 개발하였고 시험 데이터를 이용하여 아래의 성능 지표를 계산하였음.
- 시험 데이터 기준으로 R<sup>2</sup> 70% 이상의 우수한 결과를 보였으므로 **현장 적용 가능한 수치로 판단함.**

예측 모델 성능 지표) 시험 데이터 이용

| 모델 결과                | 결정계수 <sup>주1)</sup><br>(R <sup>2</sup> Y) | 실측 평균, A  | 잔차 절대값<br>평균, B | 상대 정확도<br>(1-B/A)*100 | 비고                 |
|----------------------|-------------------------------------------|-----------|-----------------|-----------------------|--------------------|
| O <sub>2</sub> 측정값 1 | 93.4%                                     | 1.88      | 0.375           | 80.1%                 | 양상블 <sup>주2)</sup> |
| O <sub>2</sub> 측정값 2 | 73.4%                                     | 5.68      | 1.29            | 77.1%                 | 양상블 <sup>주3)</sup> |
| O <sub>2</sub> 측정값 3 | 70.0%                                     | 5.17 %    | 0.77            | 85.2%                 | 양상블 <sup>주4)</sup> |
| 예열대 상부 CO            | 76.4%                                     | 64.78 ppm | 24.17 ppm       | 64.1%                 | 양상블 <sup>주5)</sup> |

주1) 결정계수 80% 이상 매우 우수, 60~80% 우수, 40~60% 보통, 40~20% 미흡, 20% 이하 매우 미흡

양상블 조합 과정: 별첨 참조

주2) Extra Trees Regressor + Light Gradient Boosting Machine + Extreme Gradient Boosting

주3) Extra Trees Regressor + Light Gradient Boosting Machine + Random Forest Regressor

주4) Extra Trees Regressor + Light Gradient Boosting Machine + Extreme Gradient Boosting

주5) Gradient Boosting Regressor + Light Gradient Boosting Machine + Random Forest Regressor

## 06. 모델 수립(2/3)

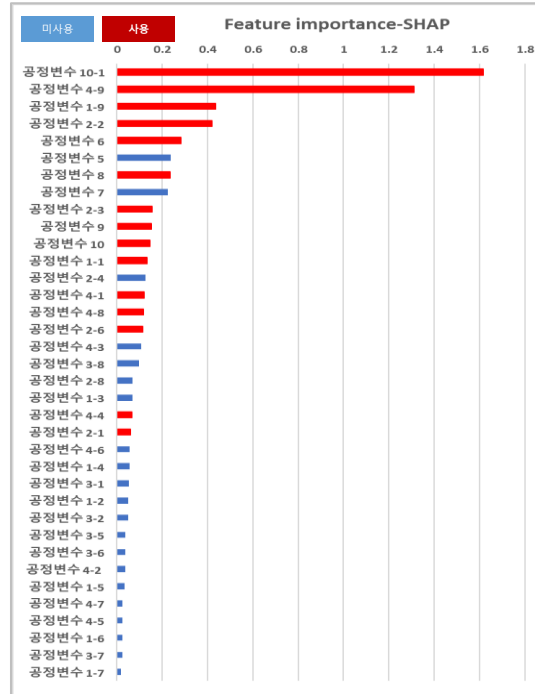
- O2, CO 예측에 있어 변수 별 입력 변수를 선택하고자 함.
- 앞서 데이터 분석을 이용하여 설정한 변수를 필수적으로 포함한 채 전진 선택법(Forward Selection)을 이용하여 모델의 변수를 선택함.
- 변수 중요도는 Light GBM의 SHAP\*을 이용하여 산출하였음.
- CO는 초기 모델의 예측력이 낮아 전체 변수와 O2 측정값 1를 입력변수로 이용함.

SHAP(Shapley Additive exPlanations): 머신러닝 모델의 결과를 설명하고자 하는 지표 중 하나  
입력 변수는 별도로 지정 후 지정된 변수를 이용하여 앙상블 진행

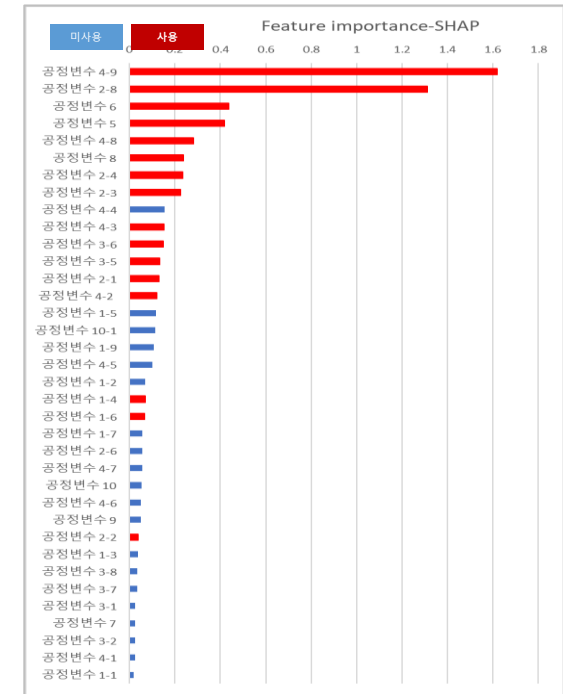
Feature Importance-SHAP : O2 측정값 1



Feature Importance-SHAP : O2 측정값 2



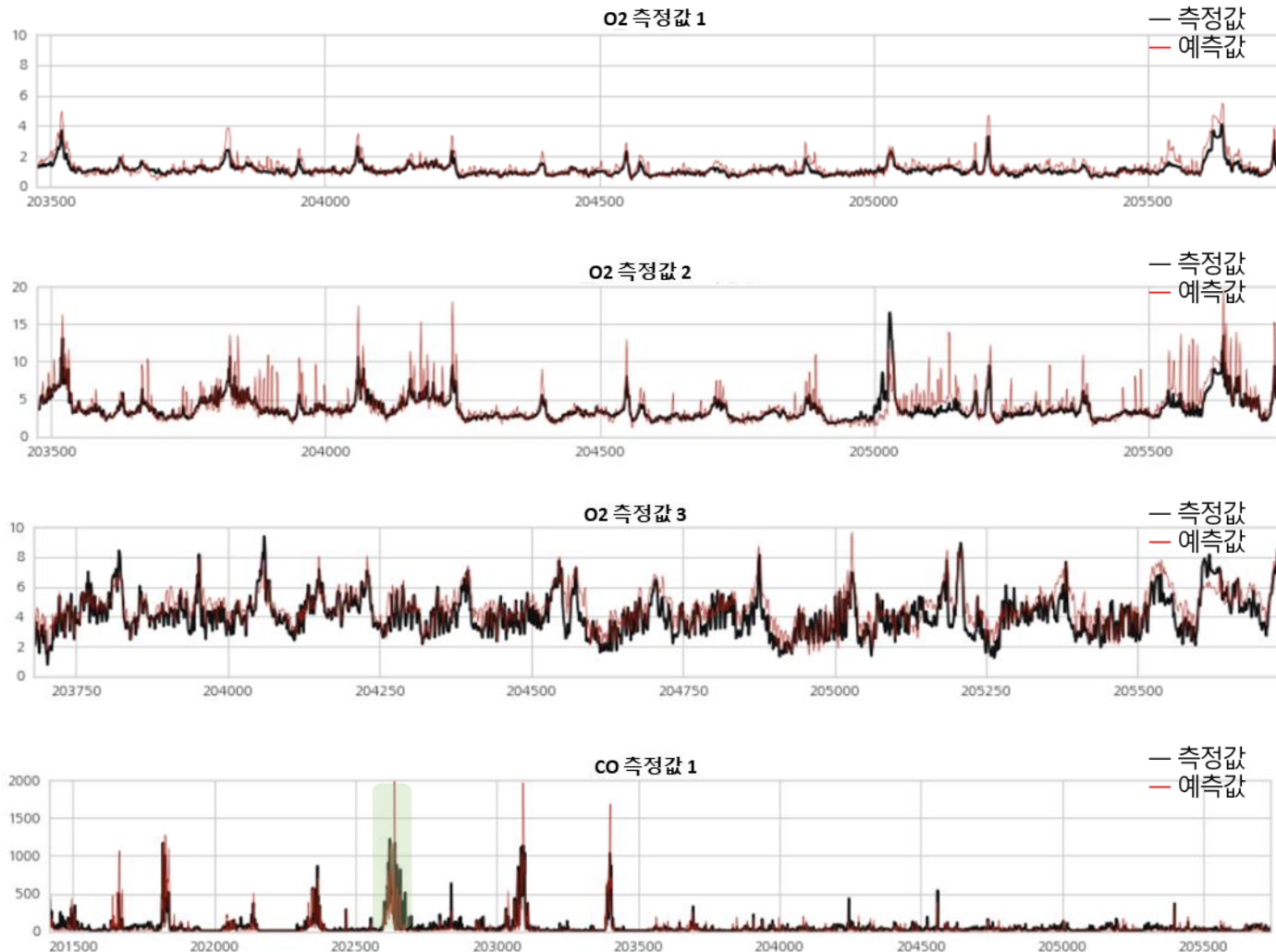
Feature Importance-SHAP : O2 측정값 3



# III. 분석 결과

## 06. 모델 수립(3/3)

- 아래는 4개의 예측 변수의 시험 데이터 예측 결과로, 측정값과 예측 값의 추이가 잘 맞는 것을 확인할 수 있음.



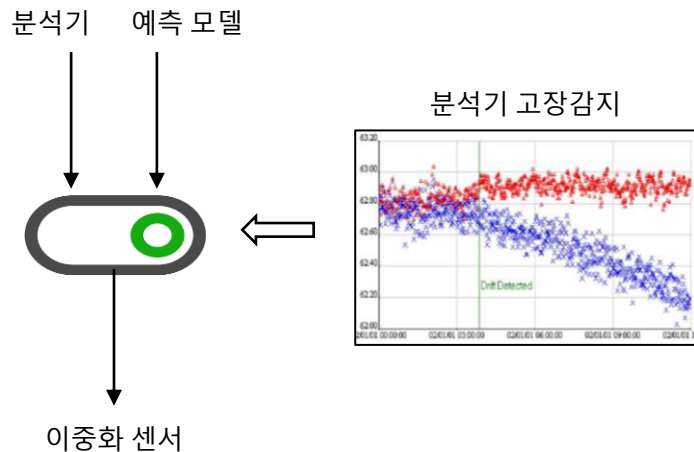
CO 농도가 급격하게  
증가하는 부분도  
추이를 잘 예측함.

# III. 분석 결과

## 07. 예측 모델 활용 방안

- 센서를 이중, 삼중으로 설치하여 센서의 고장으로 인한 오작동이나 오판을 방지해야 함.
- 이를 위하여 예측모델을 아래와 같이 이용하고자 함.
  - 분석기 고장 진단: 잔차 분석법을 이용하여 분석기를 이중화하여 끊임없이 모니터링하고자 함.
  - 공정 이상 진단: 예측모델의 강건성 유지와 공정 상태 변화를 모니터링하기 위하여 도입하고자 함.

예측 모델을 활용한 분석기 고장 진단 방안

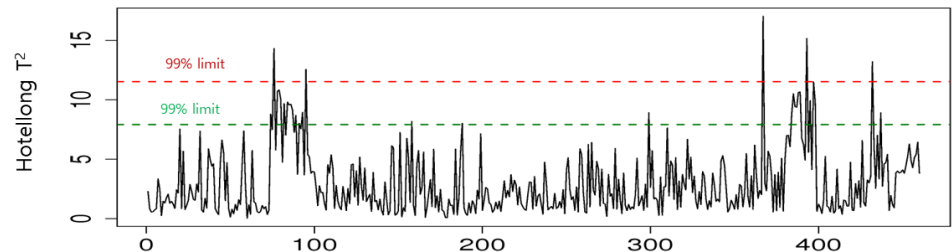
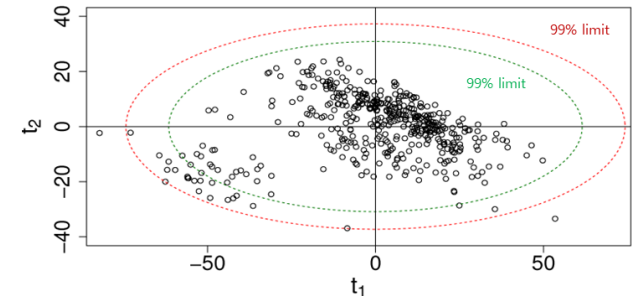


Anomaly Score( $T^2$ )를 통한 공정 이상 감지 방안

$$T^2 = \sum_{k=1}^K \left( \frac{t_k}{s_k} \right)^2$$

$K$ : 주성분 개수

$s_k$ :  $k$ 번째 주성분의 표준편차

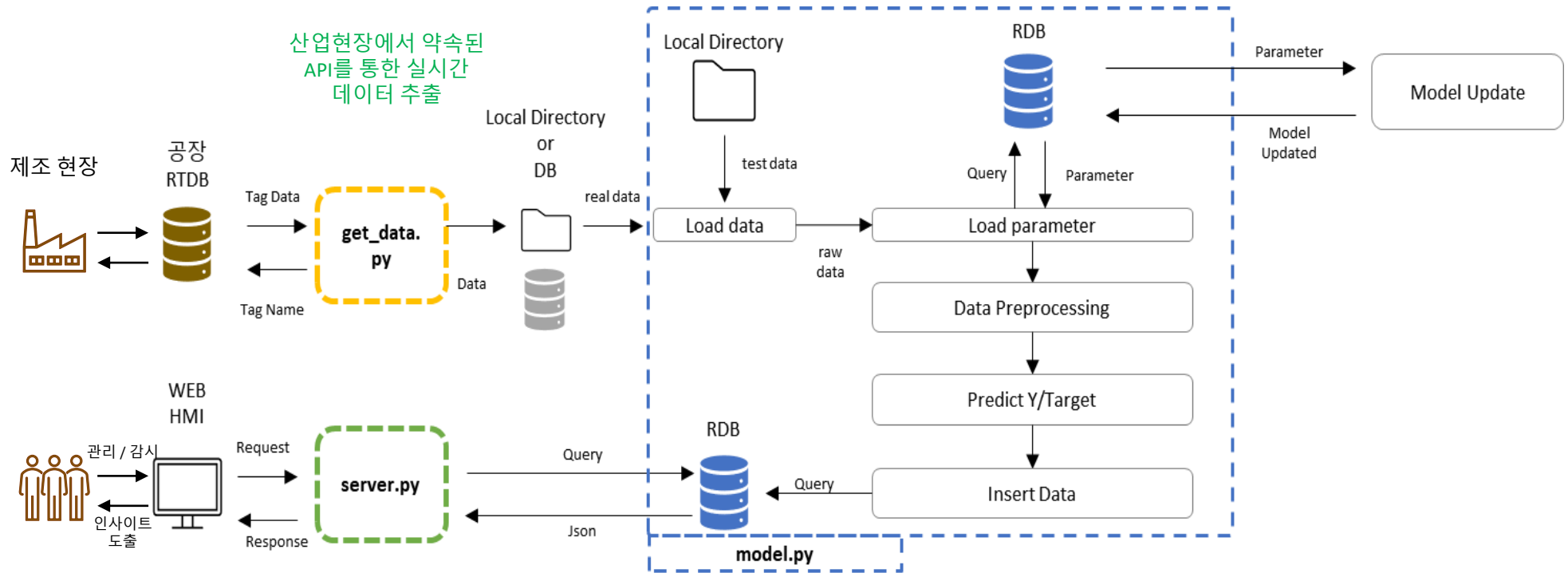


- I 서론
- II 제안 사항
- III 분석 결과
- IV 예상 화면
- V 결론
- A 별첨

# IV. 예상 화면

## 01. 예상 시스템 구성도

- 산업 현장 내 많은 이들이 관리할 수 있도록 웹 기반의 시스템을 아래와 같이 구성하고자 하였음.
  - 제조 산업 현장의 RTDB에서 데이터를 추출(현재: Test Data이용)
  - 데이터 전처리 후 Target 예측
  - 웹 서버 구현

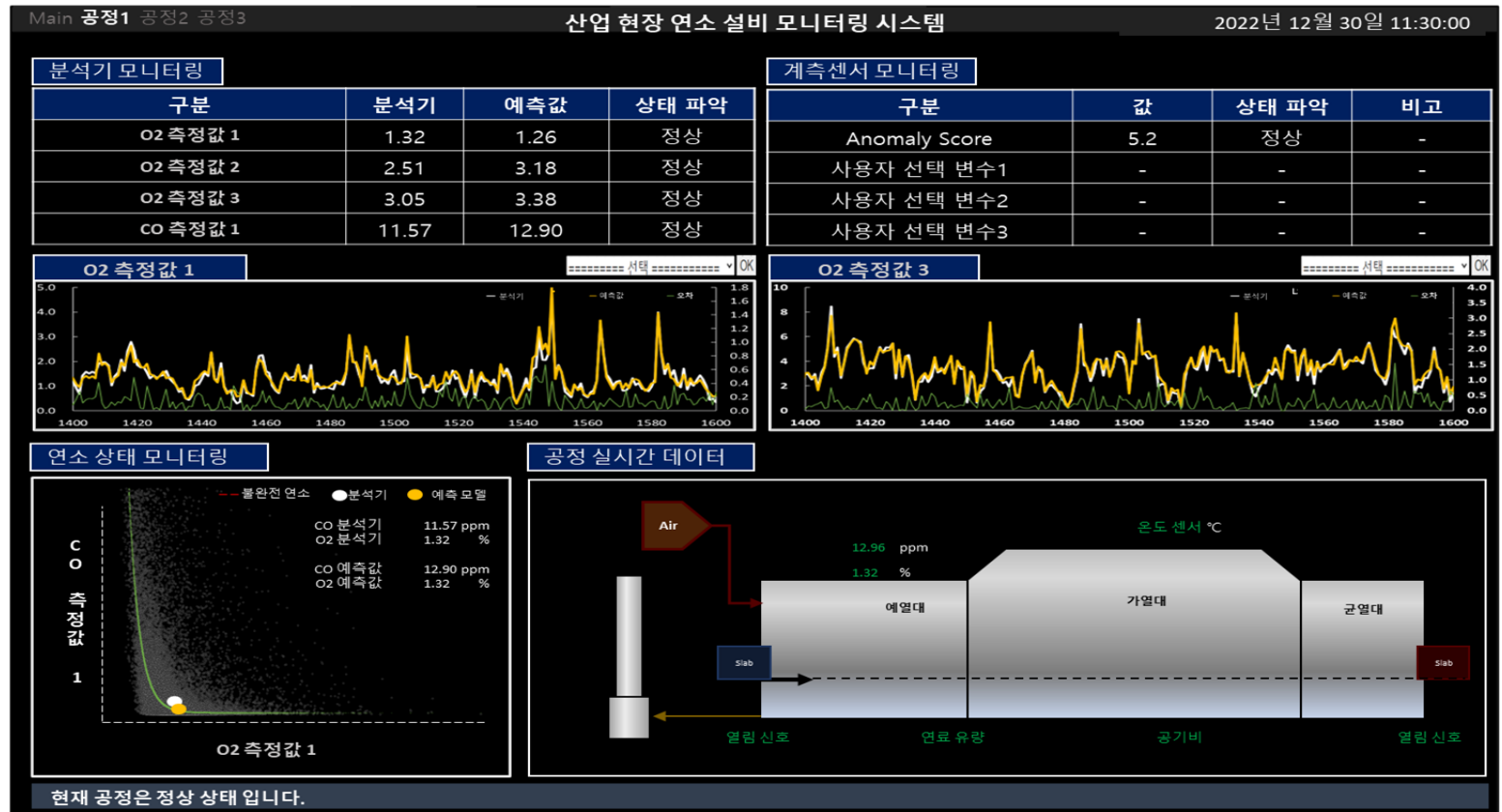


# IV. 예상 화면

## 02 예상 시스템 화면

- 단위 공정 별 화면으로 가열로의 현 상황에 대한 정보를 표현하고자 함.
- 공정 내 개별 센서/설비에 대한 이상 유무를 확인할 수 있음.

화면 주요 타겟)  
현장 운전자



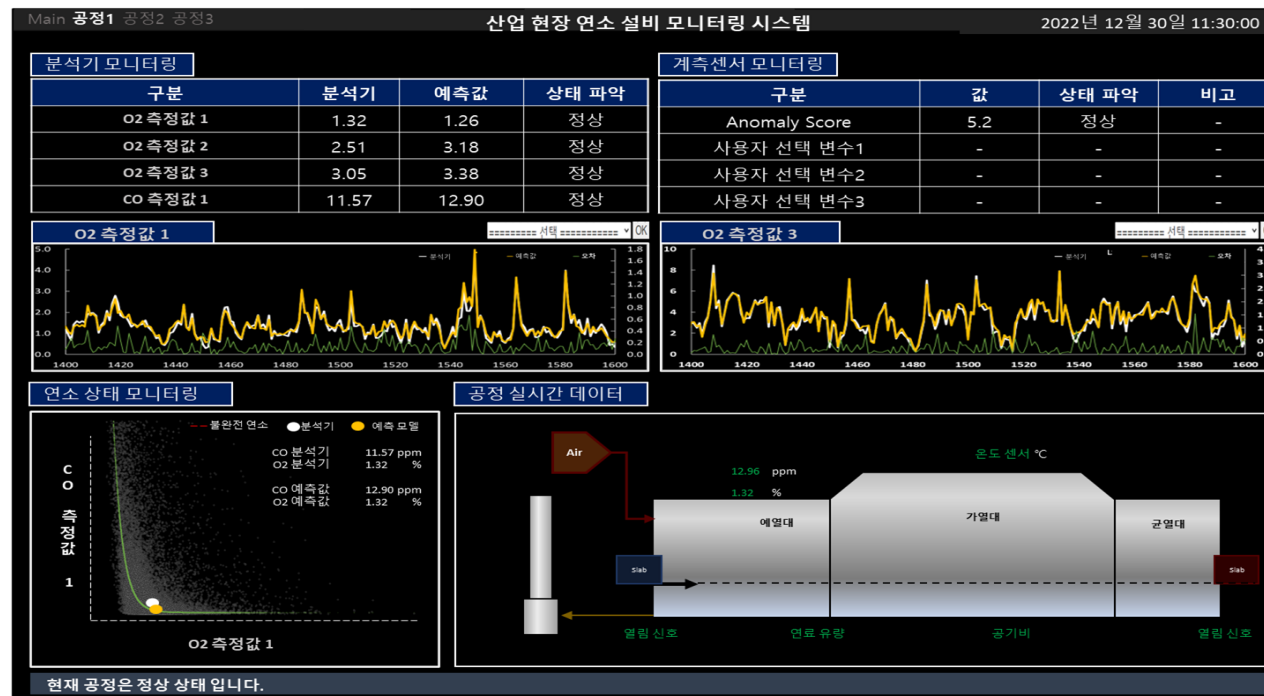
현장의 근무자/엔지니어의 리뷰를 통하여 현장에 최적화된 화면으로 구축 필요



- I 서론
- II 제안 사항
- III 분석 결과
- IV 예상 화면
- V 결론
- A 별첨

## 결론

- 탄소중립/온실가스 감축을 위하여 산업 현장의 설비를 감시/관리할 수 있는 방안을 제안하였음.
  - 설비 제조사의 조업 방식과 경험 기반의 조업 방식 외 데이터를 기반으로 한 조업 관리 기준이 되었음에 의의가 있음.
  - 제안 방안을 이용하여 설비 모니터링 자동화를 통한 현장의 생산 고도화를 기대할 수 있음.
  - 제안 방안을 통하여 산업 전반의 탄소 중립 등의 효과를 기대할 수 있음.



- I 서론
- II 제안 사항
- III 분석 결과
- IV 예상 화면
- V 결론
- A 별첨

## 앙상블을 이용한 예측 모델 조합 (1/2)

O2 측정값 1 – 예측 모델 조합 결과

각종 통계지표: 별첨 참조

|          | Model                           | MAE    | MSE    | RMSE   | R2     | RMSLE  | MAPE   |
|----------|---------------------------------|--------|--------|--------|--------|--------|--------|
| et       | Extra Trees Regressor           | 0.2130 | 0.0905 | 0.3008 | 0.9627 | 0.1130 | 0.1727 |
| lightgbm | Light Gradient Boosting Machine | 0.2143 | 0.0948 | 0.3069 | 0.9610 | 0.1105 | 0.1669 |
| xgboost  | Extreme Gradient Boosting       | 0.2195 | 0.0951 | 0.3082 | 0.9609 | 0.1140 | 0.1695 |
| rf       | Random Forest Regressor         | 0.2187 | 0.0958 | 0.3092 | 0.9606 | 0.1152 | 0.1755 |
| gbr      | Gradient Boosting Regressor     | 0.2361 | 0.1050 | 0.3236 | 0.9568 | 0.1190 | 0.1851 |
| dt       | Decision Tree Regressor         | 0.3165 | 0.2018 | 0.4491 | 0.9169 | 0.1677 | 0.2442 |
| lr       | Linear Regression               | 0.3346 | 0.2138 | 0.4621 | 0.9121 | 0.1613 | 0.2417 |
| ridge    | Ridge Regression                | 0.3349 | 0.2138 | 0.4621 | 0.9122 | 0.1616 | 0.2421 |
| br       | Bayesian Ridge                  | 0.3348 | 0.2138 | 0.4621 | 0.9122 | 0.1616 | 0.2419 |
| lar      | Least Angle Regression          | 0.3428 | 0.2198 | 0.4684 | 0.9097 | 0.1669 | 0.2496 |
| knn      | K Neighbors Regressor           | 0.3303 | 0.2267 | 0.4759 | 0.9067 | 0.1761 | 0.2634 |

Top 3 Model Extract

Ensemble

| # 가능성 있는 머신러닝 기법을 조합하여 새로운 모델을 개발                                               |        |        |        |        |        |        |  |
|---------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--|
| blended_model = blend_models(estimator_list = top_3, optimize = 'RMSE', fold=3) |        |        |        |        |        |        |  |
|                                                                                 | MAE    | MSE    | RMSE   | R2     | RMSLE  | MAPE   |  |
| Fold                                                                            |        |        |        |        |        |        |  |
| 0                                                                               | 0.2030 | 0.0809 | 0.2845 | 0.9653 | 0.1061 | 0.1603 |  |
| 1                                                                               | 0.2095 | 0.0870 | 0.2950 | 0.9644 | 0.1085 | 0.1633 |  |
| 2                                                                               | 0.2097 | 0.0868 | 0.2946 | 0.9656 | 0.1120 | 0.1713 |  |
| Mean                                                                            | 0.2074 | 0.0849 | 0.2914 | 0.9651 | 0.1089 | 0.1650 |  |
| Std                                                                             | 0.0031 | 0.0028 | 0.0049 | 0.0005 | 0.0024 | 0.0046 |  |

O2 측정값 2 – 예측 모델 조합 결과

|          | Model                           | MAE    | MSE     | RMSE   | R2     | RMSLE  | MAPE   |
|----------|---------------------------------|--------|---------|--------|--------|--------|--------|
| lightgbm | Light Gradient Boosting Machine | 1.1579 | 3.4556  | 1.8587 | 0.8248 | 0.2572 | 0.2596 |
| et       | Extra Trees Regressor           | 1.1681 | 3.4736  | 1.8636 | 0.8240 | 0.2600 | 0.2698 |
| rf       | Random Forest Regressor         | 1.1680 | 3.5757  | 1.8906 | 0.8188 | 0.2617 | 0.2682 |
| xgboost  | Extreme Gradient Boosting       | 1.1976 | 3.6629  | 1.9136 | 0.8144 | 0.2656 | 0.2678 |
| gbr      | Gradient Boosting Regressor     | 1.2474 | 3.8039  | 1.9502 | 0.8071 | 0.2672 | 0.2788 |
| lr       | Linear Regression               | 1.6063 | 5.4680  | 2.3375 | 0.7232 | 0.3482 | 0.3780 |
| ridge    | Ridge Regression                | 1.6064 | 5.4680  | 2.3375 | 0.7232 | 0.3482 | 0.3780 |
| br       | Bayesian Ridge                  | 1.6076 | 5.4686  | 2.3376 | 0.7232 | 0.3480 | 0.3781 |
| en       | Elastic Net                     | 1.6831 | 5.7608  | 2.3989 | 0.7085 | 0.3522 | 0.3951 |
| lasso    | Lasso Regression                | 1.6876 | 5.7737  | 2.4016 | 0.7078 | 0.3515 | 0.3958 |
| dt       | Decision Tree Regressor         | 1.5748 | 7.1871  | 2.6805 | 0.6356 | 0.3591 | 0.3573 |
| knn      | K Neighbors Regressor           | 2.1889 | 12.1989 | 3.4920 | 0.3818 | 0.4609 | 0.4908 |

Top 3 Model Extract

Ensemble

| # 가능성 있는 머신러닝 기법을 조합하여 새로운 모델을 개발                                               |        |        |        |        |        |        |  |
|---------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--|
| blended_model = blend_models(estimator_list = top_3, optimize = 'RMSE', fold=3) |        |        |        |        |        |        |  |
|                                                                                 | MAE    | MSE    | RMSE   | R2     | RMSLE  | MAPE   |  |
| Fold                                                                            |        |        |        |        |        |        |  |
| 0                                                                               | 1.1389 | 3.3065 | 1.8184 | 0.8374 | 0.2480 | 0.2578 |  |
| 1                                                                               | 1.1401 | 3.4441 | 1.8558 | 0.8274 | 0.2557 | 0.2609 |  |
| 2                                                                               | 1.1615 | 3.4347 | 1.8533 | 0.8188 | 0.2594 | 0.2664 |  |
| Mean                                                                            | 1.1468 | 3.3951 | 1.8425 | 0.8278 | 0.2544 | 0.2617 |  |
| Std                                                                             | 0.0104 | 0.0628 | 0.0171 | 0.0076 | 0.0047 | 0.0035 |  |

## 앙상블을 이용한 예측 모델 조합 (2/2)

O2 측정값 1 – 예측 모델 조합 결과

|  | Model                                           | MAE    | MSE    | RMSE   | R2     | RMSLE  | MAPE   |
|--|-------------------------------------------------|--------|--------|--------|--------|--------|--------|
|  | <b>lightgbm</b> Light Gradient Boosting Machine | 0.5158 | 0.5178 | 0.7194 | 0.7727 | 0.1592 | 0.1722 |
|  | <b>et</b> Extra Trees Regressor                 | 0.5219 | 0.5376 | 0.7331 | 0.7640 | 0.1664 | 0.1839 |
|  | <b>xgboost</b> Extreme Gradient Boosting        | 0.5349 | 0.5616 | 0.7493 | 0.7535 | 0.1661 | 0.1777 |
|  | <b>rf</b> Random Forest Regressor               | 0.5309 | 0.5674 | 0.7531 | 0.7510 | 0.1683 | 0.1849 |
|  | <b>gbr</b> Gradient Boosting Regressor          | 0.5593 | 0.6043 | 0.7771 | 0.7349 | 0.1728 | 0.1917 |
|  | <b>lr</b> Linear Regression                     | 0.6755 | 0.8546 | 0.9243 | 0.6247 | 0.2014 | 0.2268 |
|  | <b>ridge</b> Ridge Regression                   | 0.6756 | 0.8546 | 0.9243 | 0.6247 | 0.2013 | 0.2268 |
|  | <b>br</b> Bayesian Ridge                        | 0.6757 | 0.8547 | 0.9244 | 0.6247 | 0.2013 | 0.2269 |
|  | <b>en</b> Elastic Net                           | 0.7217 | 0.9904 | 0.9951 | 0.5652 | 0.2144 | 0.2453 |
|  | <b>ada</b> AdaBoost Regressor                   | 0.7848 | 1.0093 | 1.0043 | 0.5569 | 0.2356 | 0.3064 |
|  | <b>lasso</b> Lasso Regression                   | 0.7457 | 1.0727 | 1.0354 | 0.5294 | 0.2239 | 0.2596 |

Top 3 Model Extract

Ensemble

```
# 가능성 있는 머신러닝 기법을 조합하여 새로운 모델을 개발
blended_model = blend_models(estimator_list = top_3, optimize = 'RMSE', fold=3)
```

|             | MAE    | MSE    | RMSE   | R2     | RMSLE  | MAPE   |
|-------------|--------|--------|--------|--------|--------|--------|
| <b>Fold</b> |        |        |        |        |        |        |
| <b>0</b>    | 0.5159 | 0.5079 | 0.7127 | 0.7842 | 0.1629 | 0.1805 |
| <b>1</b>    | 0.5057 | 0.5292 | 0.7275 | 0.7713 | 0.1597 | 0.1736 |
| <b>2</b>    | 0.5028 | 0.4903 | 0.7002 | 0.7741 | 0.1560 | 0.1660 |
| <b>Mean</b> | 0.5081 | 0.5091 | 0.7135 | 0.7766 | 0.1595 | 0.1733 |
| <b>Std</b>  | 0.0056 | 0.0159 | 0.0111 | 0.0056 | 0.0028 | 0.0059 |

O2 측정값 2 – 예측 모델 조합 결과

|  | Model                                           | MAE     | MSE        | RMSE    | R2      | RMSLE  | MAPE   |
|--|-------------------------------------------------|---------|------------|---------|---------|--------|--------|
|  | <b>gbr</b> Gradient Boosting Regressor          | 16.2306 | 3930.4453  | 62.5039 | 0.6372  | 0.5174 | 0.4855 |
|  | <b>lightgbm</b> Light Gradient Boosting Machine | 15.5984 | 3951.1983  | 62.6978 | 0.6374  | 0.5125 | 0.4233 |
|  | <b>rf</b> Random Forest Regressor               | 15.1284 | 4165.4755  | 64.4113 | 0.6159  | 0.4449 | 0.3668 |
|  | <b>et</b> Extra Trees Regressor                 | 15.3004 | 4220.9644  | 64.6767 | 0.6174  | 0.4625 | 0.4337 |
|  | <b>xgboost</b> Extreme Gradient Boosting        | 15.6517 | 4275.2856  | 65.0641 | 0.6110  | 0.5104 | 0.4187 |
|  | <b>dt</b> Decision Tree Regressor               | 19.6833 | 8181.3245  | 90.3075 | 0.2307  | 0.5336 | 0.4264 |
|  | <b>lr</b> Linear Regression                     | 39.1400 | 9502.5267  | 96.7920 | 0.1480  | 1.0617 | 1.9945 |
|  | <b>ridge</b> Ridge Regression                   | 39.1016 | 9502.4042  | 96.7921 | 0.1480  | 1.0607 | 1.9914 |
|  | <b>en</b> Elastic Net                           | 39.3637 | 9641.1366  | 97.5328 | 0.1345  | 1.0708 | 2.0117 |
|  | <b>lasso</b> Lasso Regression                   | 39.3758 | 9641.6085  | 97.5347 | 0.1344  | 1.0703 | 2.0130 |
|  | <b>br</b> Bayesian Ridge                        | 39.1423 | 9663.7649  | 97.6523 | 0.1322  | 1.0712 | 1.9935 |
|  | <b>ada</b> AdaBoost Regressor                   | 67.4763 | 11270.0067 | 99.9233 | -0.2418 | 1.3878 | 4.2215 |

Top 3 Model Extract

Ensemble

```
# 가능성 있는 머신러닝 기법을 조합하여 새로운 모델을 개발
blended_model = blend_models(estimator_list = top_3, optimize = 'RMSE', fold=3)

# 머신러닝 모델 최종 상태 수립
blended_model = finalize_model(blended_model)
```

|             | MAE     | MSE       | RMSE    | R2     | RMSLE  | MAPE   |
|-------------|---------|-----------|---------|--------|--------|--------|
| <b>Fold</b> |         |           |         |        |        |        |
| <b>0</b>    | 14.2859 | 2992.7911 | 54.7064 | 0.5892 | 0.4469 | 0.3985 |
| <b>1</b>    | 15.5842 | 4160.4209 | 64.5013 | 0.6742 | 0.4581 | 0.3963 |
| <b>2</b>    | 14.3659 | 3888.9820 | 62.3617 | 0.7073 | 0.4351 | 0.3870 |
| <b>Mean</b> | 14.7453 | 3680.7313 | 60.5231 | 0.6569 | 0.4467 | 0.3939 |
| <b>Std</b>  | 0.5941  | 498.9095  | 4.2048  | 0.0497 | 0.0094 | 0.0050 |

## 관련 통계 지표

- 성능 지표를 통해 모델을 통한 예측 값이 측정 값에 가까운 정도를 수치적으로 나타냄
- 여러 예측 모델 중 최적의 모델 선정 기준이 되기도 함.
  - 예측 정확도: 학습에 사용하지 않던 새로운 데이터에서 모델이 얼마나 잘 맞추는 가를 표현한 값  
수치가 작을 수록 예측값과 실제값의 차이가 없으며 좋은 예측력을 갖는다고 판단할 수 있음.
  - 예측 정확도의 절대적 기준은 없으며 분석 목적과 예측 모델의 특성에 맞는 지표, 기준을 선정할 필요가 있음.  
(데이터 분석가와 현장 실무자가 비즈니스적 활용도를 고려하여 선정해야 함.)
- 결정계수 R2: 0과 1사이의 값으로 1에 가까워질수록 예측 모델의 정확도를 높게 판단할 수 도 있음.  
모델의 예측하고자 하는 값에 대한 입력 변수들의 설명력을 알고자 할 때 사용

### • 평균 절대 오차(MAE)

$$- \frac{\sum | \text{실제값} - \text{예측값} |}{\text{데이터 수}}$$

### • 평균 제곱 오차(MSE)

$$- \frac{\sum (\text{실제값} - \text{예측값})^2}{\text{데이터 수}}$$

### • 평균 절대비율 오차(MAPE)

$$- \frac{\sum \left| \frac{\text{실제값} - \text{예측값}}{\text{실제값}} \right|}{\text{데이터 수}} \times 100\%$$

### • 평균 제곱근 오차(RMSE)

$$- \sqrt{\frac{\sum (\text{실제값} - \text{예측값})^2}{\text{데이터 수}}}$$

### <평균 오차에 관한 파이썬 코드>

```
def calc_mean_errors(self, y_true, y_pred):
    from sklearn.metrics import mean_absolute_error
    from sklearn.metrics import mean_squared_error
    from sklearn.metrics import mean_squared_log_error
    """
    1. MSE (Mean Squared Error):
    2. RMSE (Root Mean Squared Error)
    3. MSLE (Mean Squared Log Error)
    4. MAE (Mean Absolute Error)
    5. MAPE (Mean Absolute Percentage Error)
    6. MPE (Mean Percentage Error)
    """
    labels = ["MSE", "RMSE", "MSLE", "MAE", "MAPE", "MPE"]

    y_true = np.array(y_true)
    y_test = np.array(y_pred)

    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    msle = mean_squared_log_error(y_true, y_pred)
    mae = mean_absolute_error(y_true, y_pred)
    mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100
    mpe = np.mean((y_true - y_pred) / y_true) * 100

    results = {"MSE": mse, "RMSE": rmse, "MSLE": msle, "MAE": mae, "MAPE": mape, "MPE": mpe}
    return results
```

감사합니다.