상추의 생육 환경 생성

:상추세요 팀

likeeeeeee, 범범범즈,
dhop, hesoo, 쥬혁이

*이미지 출처: 미리캔버스

목차

상추의 생육 환경 생성

01. 환경변수 설정

01. 환경변수 설정

변수 타입	변수명
독립 변수 (16 개)	DAT, obs_time, 내부온도관측치, 내부습도관측치, co2관측치, ec관측치, 시간당분무량, 일간누적분무량, 시간당백색광량, 일간누적백색광량, 시간당적색광량, 일간누적적색광량, 시간당청색광량, 일간누적청색광량, 시간당총광량, 일간누적총광량
파생 변수 (229 개)	DAT, 누적값, ec × 분무량, 수분량, 하루평균 + 스무딩 기법인 로우패스 필터, 칼만 필터, 이동 중앙값, 이동 평균값 적용 (*뒷장에서 상세 설명)

*독립변수로 만든 파생 변수만 사용
*현실에는 측정오차가 존재하니 분산을 줄이기 위해 스무딩 기법을 사용함

01. 환경변수 설정

누적값

0~5시, 6~19시, 20~23시에 대한 온도, 습도, co2, ec, 분무량, 적색광을 누적함
변수명) 05시내부온도관측치, 09시내부온도관측치, 23시내부온도관측치, ...

ec × 분무량

0~5시, 6~19시, 20~23시, 전체 평균에 대한 ec관측치와 분무량을 곱함
변수명) ec_x_분무05, ec_x_분무19, ec_x_분무23, ec_x_분무평균

수분량

수분량 공식
변수명) 수분량합, 수분량합12, 수분량합13, 수분량합23

하루 평균

온도, 습도, co2, ec, 분무량, 적색광에 대한 하루 평균값
변수명) 하루평균온도, 하루평균습도, 하루평균co2, 하루평균ec, 하루평균분무량, 하루평균적색광

로우패스 필터(Low-pass filter; LPF)

로우패스 필터란, 저주파 통과 필터라고도 하며 특정한 차단 주파수 이상 주파수의 신호를 감쇠시켜 차단 주파수 이하의 주파수 신호만 통과시키는 필터로 노이즈 캔슬링 등에 사용됨
누적값, ec × 분무량, 수분량, 하루평균에 적용

변수명

1_LPF, 2_LPF, 3_LPF, ..., 31_LPF, 32_LPF

칼만 필터(Kalman filter)

칼만 필터란, 과거에 수행한 측정값을 바탕으로 현재의 상태 변수의 결합분포를 추정하며 예측 단계와 업데이트 단계로 이루어져있으며 컴퓨터 비전, 레이다에 활용됨
예측 단계에서 현재 상태 변수의 값과 정확도를 예측하고 업데이트 단계에서는 예측한 측정치와 실제 측청치의 차이를 반영해 다음 상태 변수를 업데이트함
누적값, ec × 분무량, 수분량, 하루평균에 적용

변수명

kf_X_2, kf_X_3, kf_X_4, kf_X_5, ..., kf_X_33

이동 중앙값(Moving median)

이동 중앙값이란, 지정된 갯수의 최근 측정값 중에서 중앙값을 구한 것이며 중앙값은 순서대로 나열하여 가운데 있는 값을 대표값으로 선정하기 때문에 이상치의 영향을 크게 받지 않는다는 장점이 있음
window 값을 7, 14로 하여 DAT, 누적값, ec × 분무량, 수분량, 하루평균에 적용

변수명

DAT_median_7, DAT_median_14, 05시내부온도관측치누적_median_7, 05시내부온도관측치누적_median_14, ..., 수분량합23_median_7, 수분량합23_median_14

이동 평균값(Moving Average)

이동 평균값이란, 모든 측정 데이터가 아닌 지정된 갯수의 최근 측정값만 가지고 계산한 평균이며 새로운 데이터가 들어오면 가장 오래된 데이터는 버리는 방식으로 데이터 갯수를 일정하게 유지하면서 평균을 계산함
단순 평균은 전체 데이터에 대한 평균을 구하기 때문에 최근 데이터의 변화량을 감지할 수 없지만, 이동 평균은 시간에 따른 데이터에 대한 변화를 반영함
window 값을 7, 14로 하여 DAT, 누적값, ec × 분무량, 수분량, 하루평균에 적용

변수명

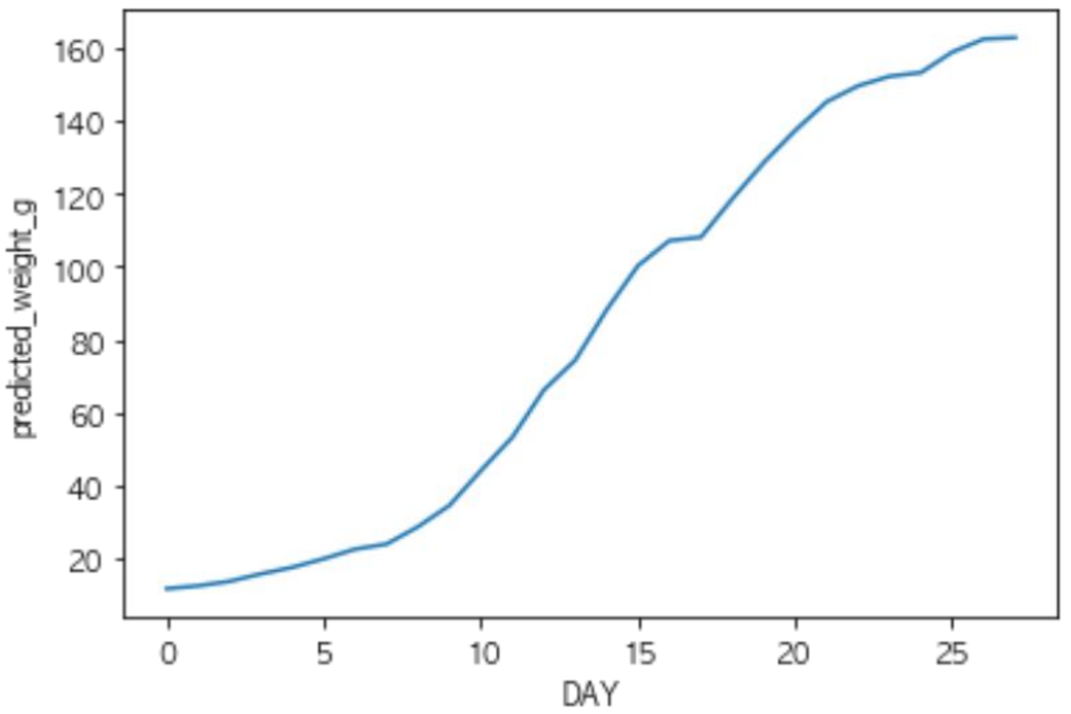
DAT_mean_7, DAT_mean_14, 05시내부온도관측치누적_mean_7, 05시내부온도관측치누적_mean_14, ..., 수분량합23_mean_7, 수분량합23_mean_14

01. 환경변수 설정

환경변수와 최대 잎 중량과의 관계 분석

best case - 마지막 target 값이 가장 큰 case

DAT	predicted_weight_g
1	11.7641629206507
2	12.5232646566145
3	13.7609128649562
4	15.8139043767443
5	17.5964697535799
6	19.9662753476084
7	22.5874871211339
8	23.9870997517786
9	28.7505807566479
10	34.5824651579793
11	44.1286075355256
12	53.2839511847254
13	66.1978360135162
14	74.4114716455442
15	88.2183960847072
16	100.348686633551
17	107.197092872452
18	108.12025351127
19	118.670513538443
20	128.598181425359
21	137.372643471174
22	145.274517466308
23	149.634797475652
24	152.2444216243
25	153.284016738578
26	158.894610733963
27	162.479376875913
28	164.150299846821



01. 환경변수 설정

환경변수와 최대 잎 중량과의 관계 분석

best case - 마지막 target 값이 가장 큰 case

15	88.2183960847072
16	100.348686633551
17	107.197092872452
18	108.12025351127
19	118.670513538443
20	128.598181425359
21	137.372643471174
22	145.274517466308
23	149.634797475652
24	152.2444216243
25	153.284016738578
26	158.894610733963
27	162.479376875913
28	164.150299846821

train dataset은 선행 연구 기반으로 다양환 환경에서 실험을 해봤을 것이라고 정의함
따라서, 대회에서 논문 기반으로 생육 환경에 대한 insight를 제공하는 것보다 논문에서 언급되지 않은 현실적이지만 unique한 환경 조성을 만드는 것에 초점을 맞춤

왼쪽 base case처럼 최대 잎 중량은 약 160임
train dataset이 real world data라는 것을 근거로 최대 잎 중량을 예측했을 때 160이 넘어가면 이상치로 판단

*전체 팀 중 제일 많은 제출로 어떤 환경 변수가 노이즈인지 파악하는데 기여하며, 가장 중요한 환경 변수를 파악함

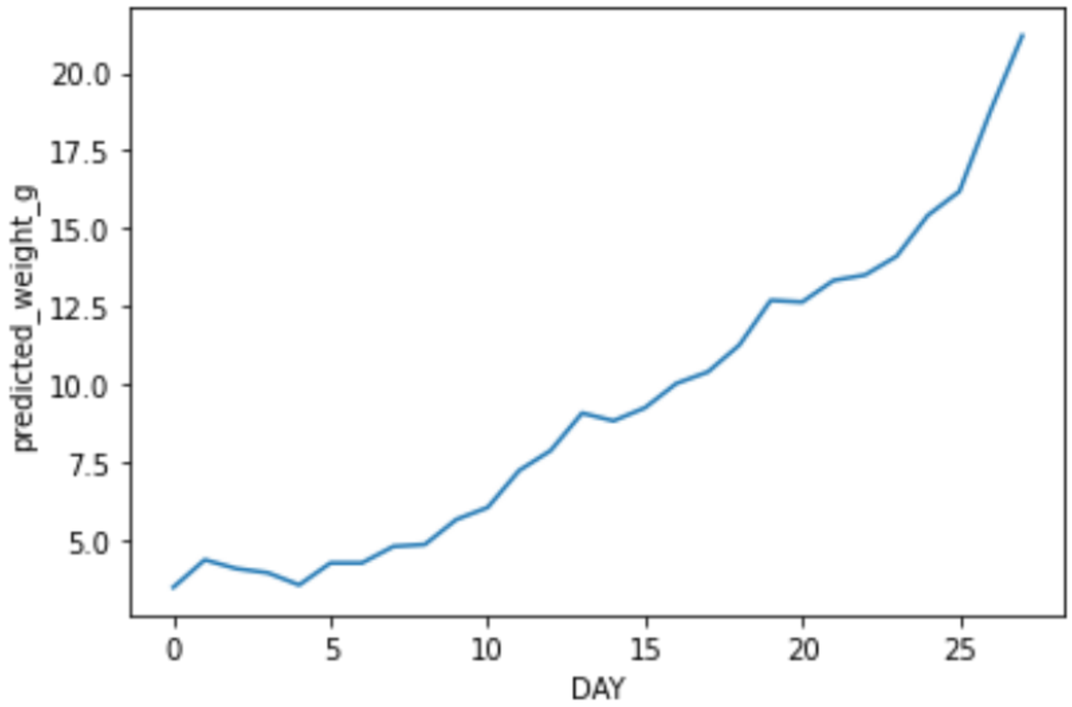
01. 환경변수 설정

환경변수와 최대 잎 중량과의 관계 분석

분무량을 최소로 했을 때

target_분무량최소

DAT	predicted_weight_g
0	3.4515975
1	4.3360786
2	4.0524273
3	3.9198053
4	3.5266352
5	4.2356
6	4.2356
7	4.7667193
8	4.828097
9	5.6325107
10	6.020572
11	7.208989
12	7.848515
13	9.053897
14	8.80445
15	9.225063
16	10.007685
17	10.376978
18	11.240925
19	12.679903
20	12.623694
21	13.321308
22	13.494307
23	14.095034
24	15.415711
25	16.182491
26	18.797394
27	21.19059



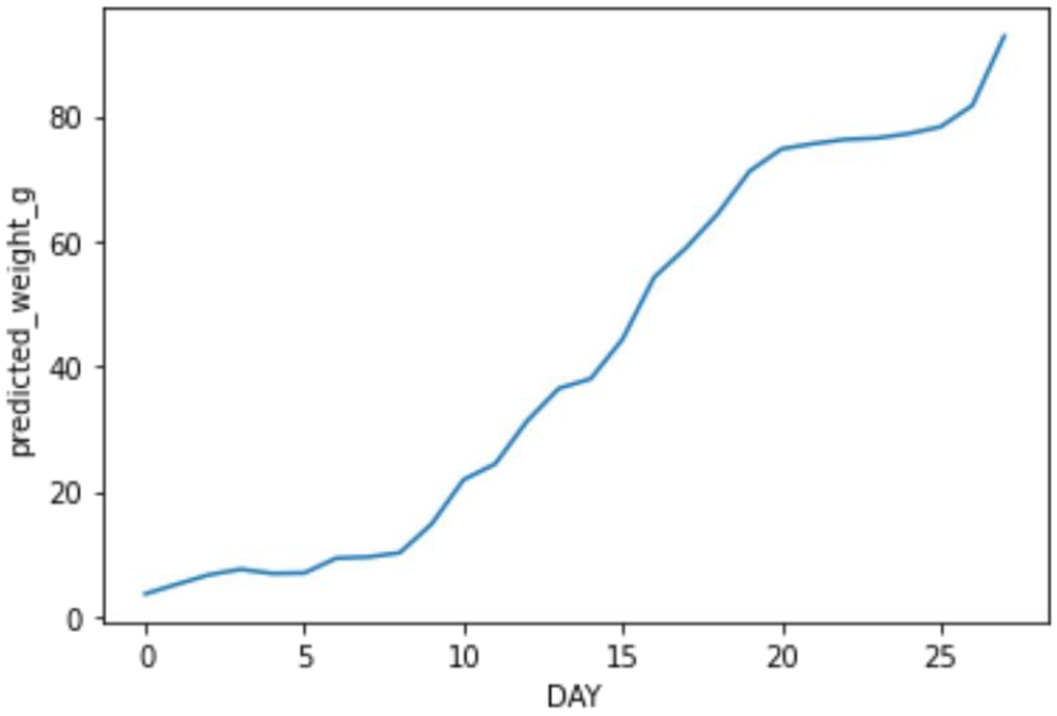
매우 변화 큼, 영향력 ↑

01. 환경변수 설정

환경변수와 최대 잎 중량과의 관계 분석
적색광량을 최소로 했을 때

target_적색광최소

DAT	predicted_weight_g
0	3.60328
1	5.150257
2	6.661715
3	7.538571
4	6.8605685
5	6.957528
6	9.307611
7	9.495847
8	10.177613
9	14.778733
10	21.84606
11	24.391579
12	31.23454
13	36.455875
14	38.007313
15	44.31647
16	54.25822
17	58.938683
18	64.43626
19	71.21745
20	74.74738
21	75.591034
22	76.26664
23	76.4861
24	77.20001
25	78.29274
26	81.71188
27	92.79037



변화 있음, 영향력 ↑

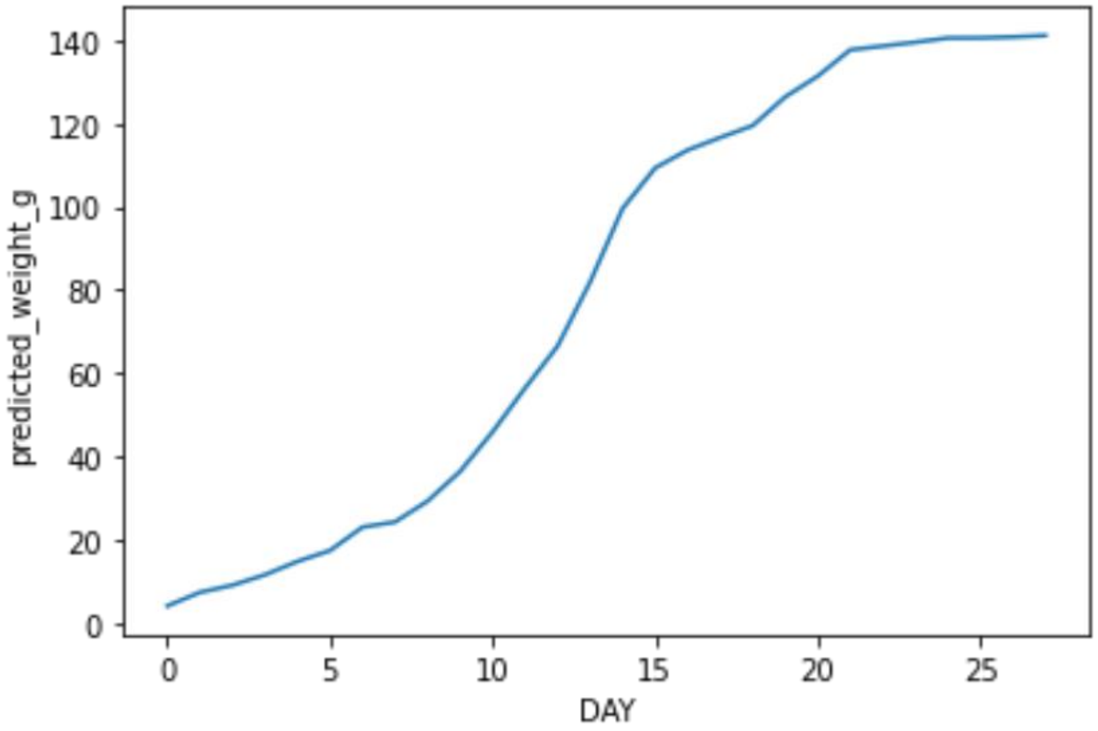
01. 환경변수 설정

환경변수와 최대 잎 중량과의 관계 분석

co2을 최소로 했을 때

target_co2최소

DAT	predicted_weight_g
0	4.0585814
1	7.321054
2	9.03376
3	11.54036
4	14.758236
5	17.364277
6	23.003403
7	24.240118
8	29.251808
9	36.379997
10	45.89212
11	56.445446
12	66.5182
13	81.931526
14	99.62398
15	109.36211
16	113.66837
17	116.66571
18	119.54024
19	126.44517
20	131.43607
21	137.76402
22	138.66518
23	139.58492
24	140.61081
25	140.64262
26	140.85811
27	141.2068



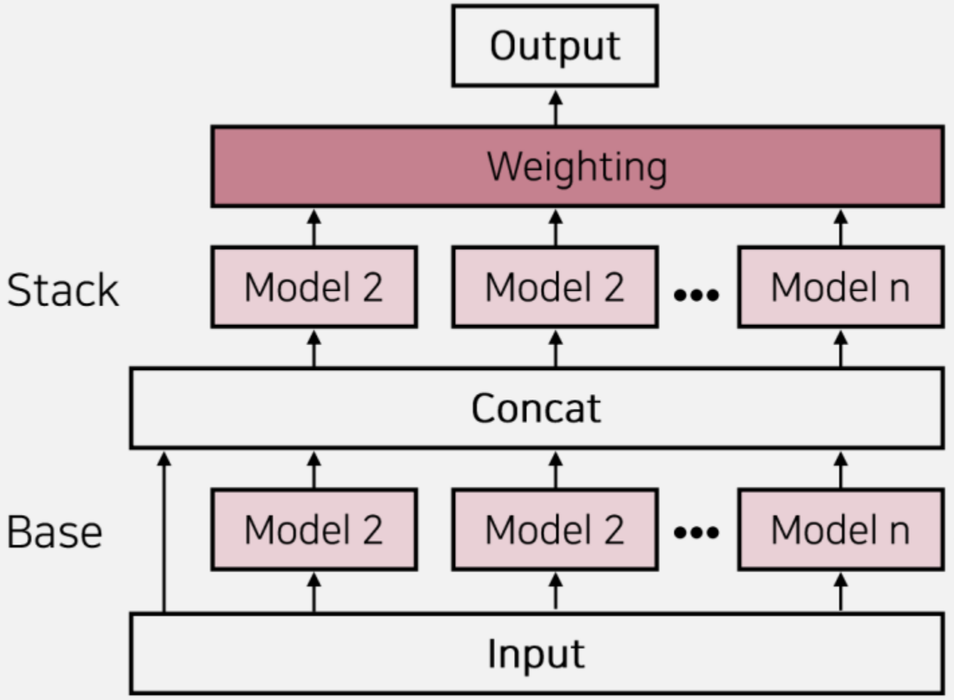
변화 작음, 영향력 ↓

상추의 생육 환경 생성

02. 예측 모델 알고리즘

02. 예측 모델 알고리즘

AutoGluon



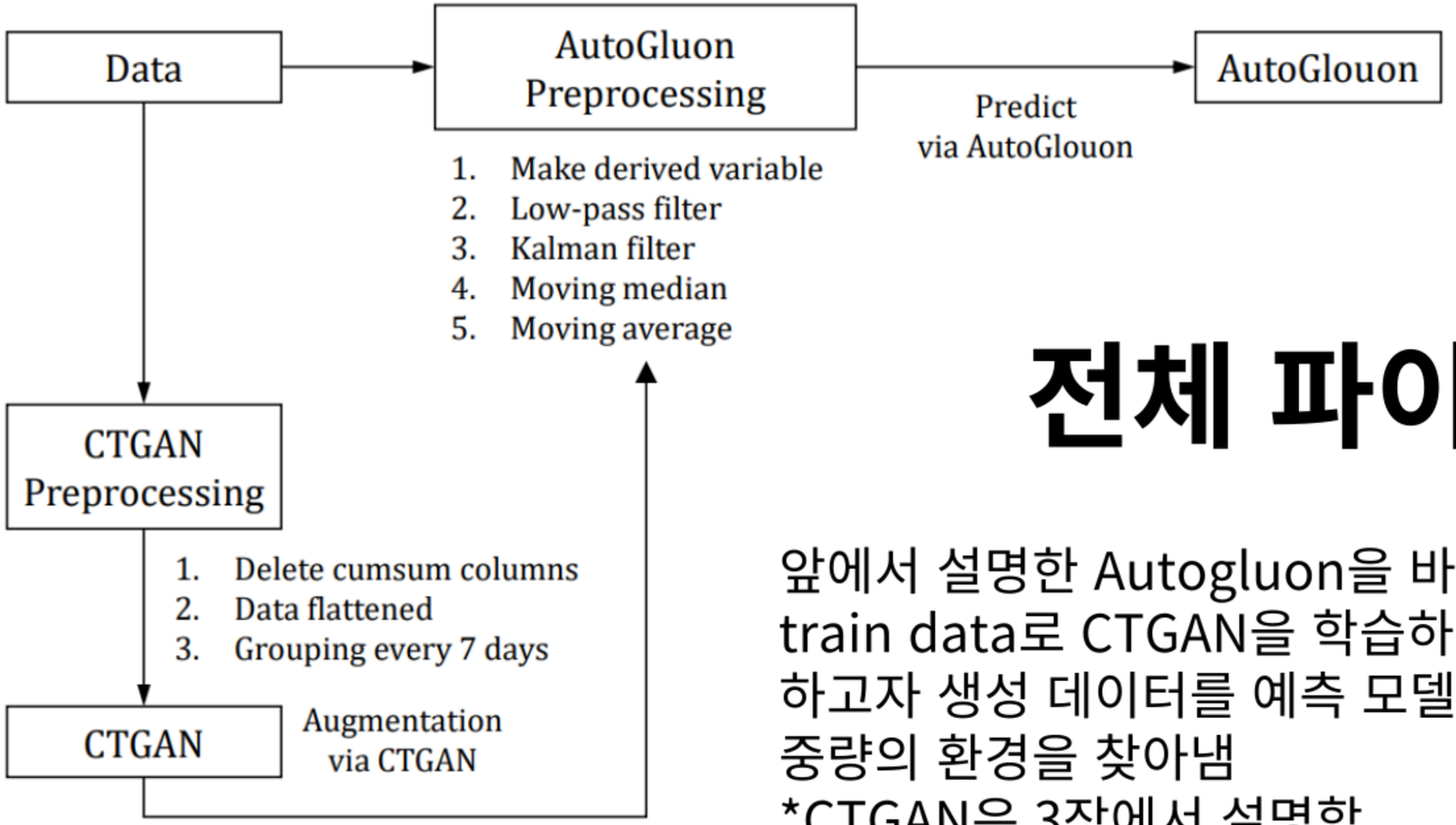
1. AutoML 프레임워크

2. 보편적인 AutoML이 주로 사용하는 CASH(Combined Algorithm Selection and Hyper-parameter optimization)이 아닌, **ensembling**과 **stacking**을 활용

3. 각 모델 학습시 Repeated k-fold ensemble bagging 사용 -> 과적합 방지

https://github.com/autogluon/autogluon

02. 예측 모델 알고리즘



전체 파이프라인

앞에서 설명한 Autogluon을 바탕으로 예측 모델을 만들고, train data로 CTGAN을 학습하여 최적의 생육 환경을 생성하고자 생성 데이터를 예측 모델에 input으로 넣어 최대 잎 중량의 환경을 찾아냄
*CTGAN은 3장에서 설명함

상추의 생육 환경 생성

03. 생성 AI 모델 알고리즘

03. 생성 AI 모델 알고리즘

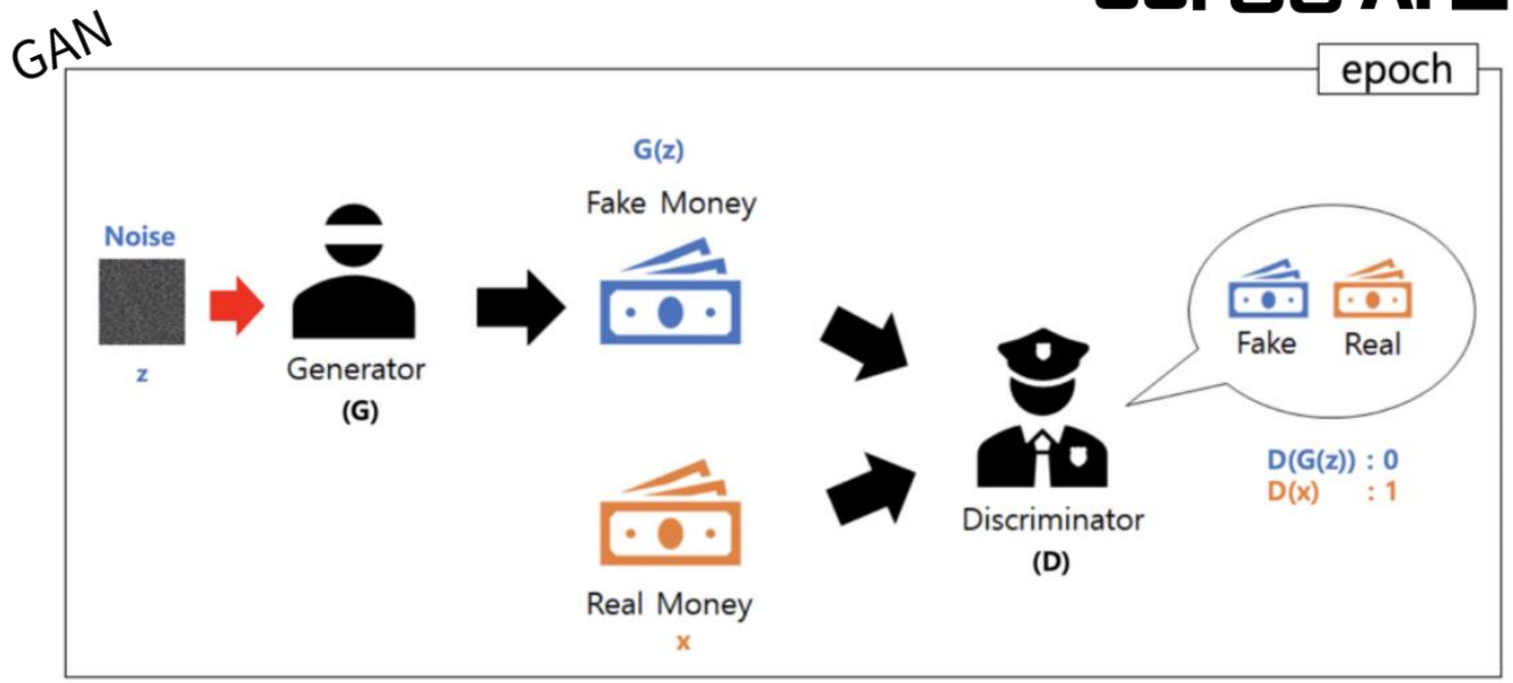


그림 . GAN 알고리즘의 대표적인 예시인 경찰(D)과 도둑(G) 이미지

GAN 알고리즘은 경찰과 도둑을 예시로 많이 설명되는데 도둑(G)이 가짜 화폐(G(z))를 만들고 경찰(D)이 진짜/가짜 여부를 판별(0/1)해 도둑이 가짜 화폐를 점점 진짜처럼 만들도록 학습시키고 경찰은 점점 가짜 화폐를 잘 판별하도록 학습시킴

https://baechu-story.tistory.com/12

03. 생성 AI 모델 알고리즘

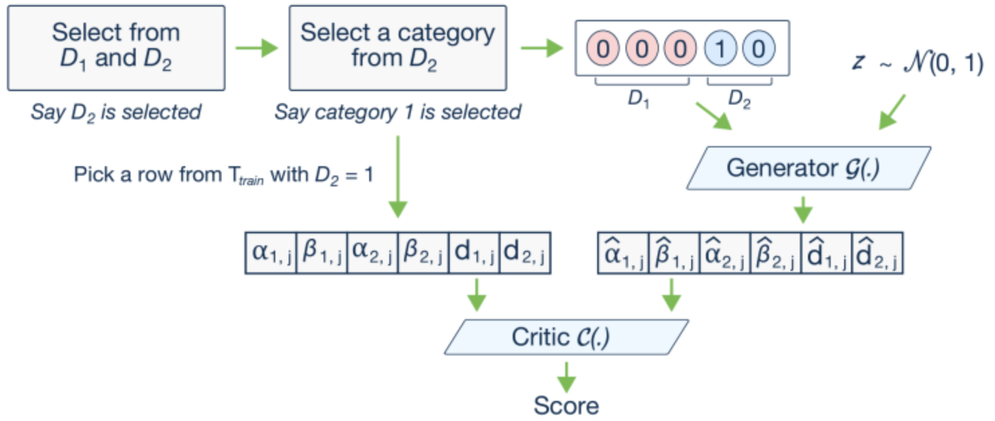
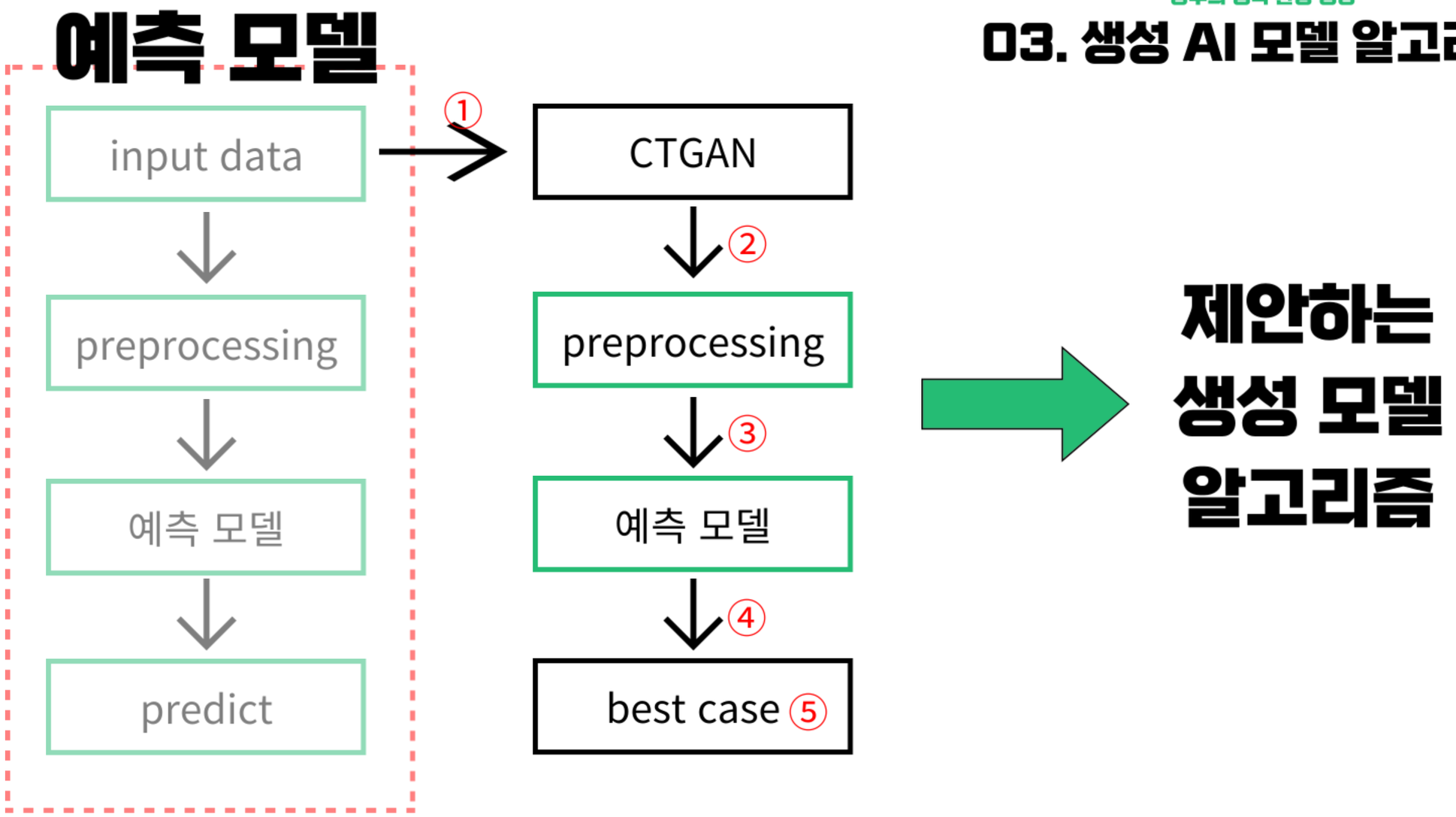


Figure 2: CTGAN model. The conditional generator can generate synthetic rows conditioned on one of the discrete columns. With training-by-sampling, the *cond* and training data are sampled according to the log-frequency of each category, thus CTGAN can evenly explore all possible discrete values.

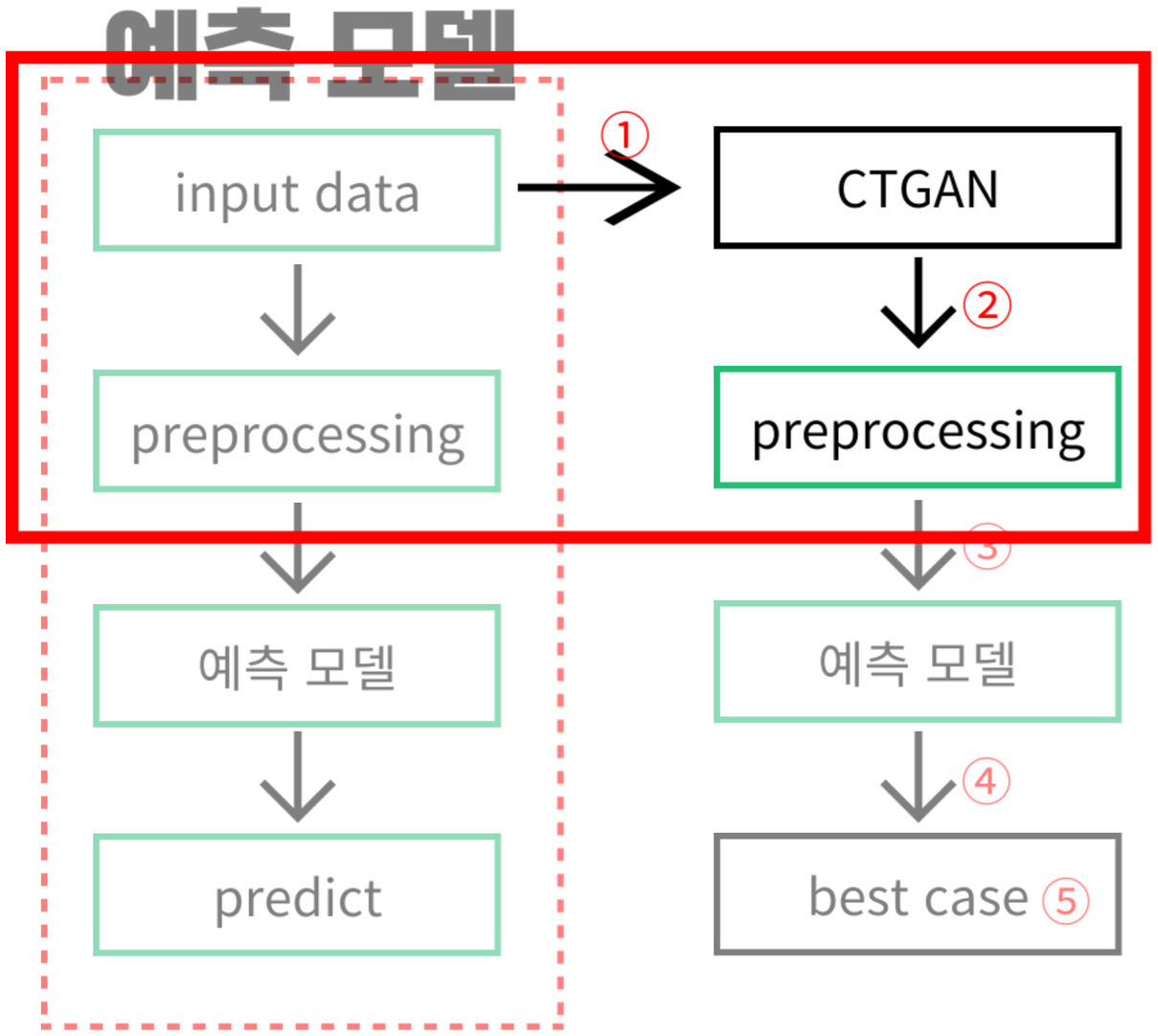
tabular data에는 discrete data와 continuous data가 동시에 존재하는데, continuous data는 multi modal distribution 문제가 있고 discrete data는 카테고리별로 빈도수가 모두 다르다는 성질이 존재함
CTGAN은 Mode-specific Normalization을 적용해 문제를 해결함
discrete data는 단순히 전체 category 개수만큼의 비트로 one-hot encoding을 진행하고 continuous data는 VGM 통해 분포를 독립적으로 나누고 확률밀도함수로 가장 확률이 높게 나오는 분포를 1, 그렇지 않은 것은 전부 0으로 one-hot encooding을 진행함

https://arxiv.org/pdf/1907.00503

03. 생성 AI 모델 알고리즘



03. 생성 AI 모델 알고리즘



① n번째 일의 input data를 생성 모델에 넣어 n번째 일에 대한 가짜 데이터 생성

컬럼명>>

내부온도관측치, 내부습도관측치, co2관측치, ec관측치, 시간당분무량, 시간당백색광량, 시간당적색광량, 시간당청색광량

② 예측 모델의 전처리 방식 적용

03. 생성 AI 모델 알고리즘

① 예시

내부온도관측치_0	내부습도관측치_0	co2관측치_0	...	시간당청색광량_23
25.3432	74.1231	470.3432	...	0
25.9882	70.5832	464.9882		0
...	...			
28.4298	69.8293			

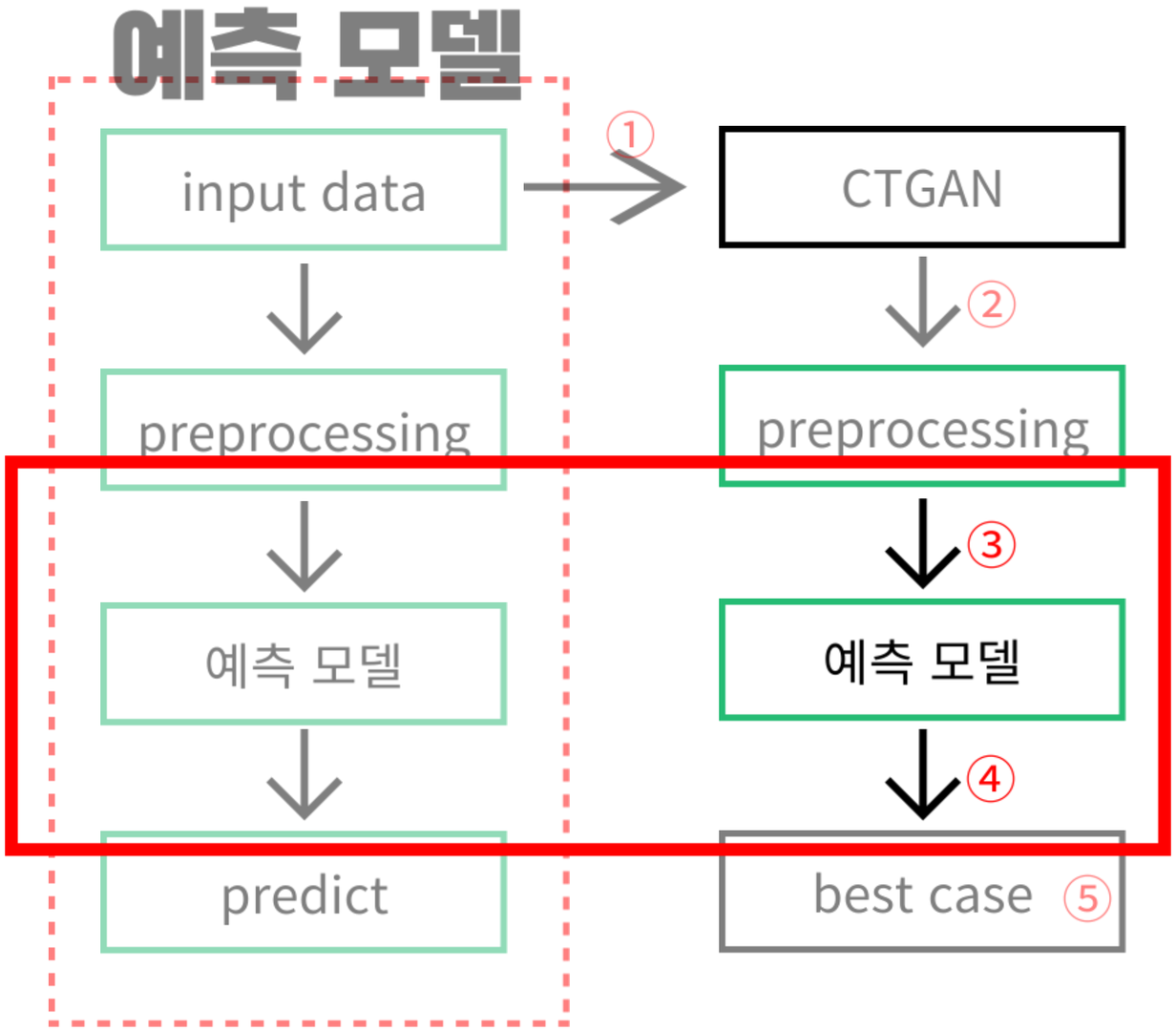
train case 28개의 0일 데이터
input.shape=(28, 192)
*24시간 × 컬럼 8개 = 192

CTGAN

0일에 대한 생성 데이터 k개
output.shape=(k, 192)

내부온도관측치_0	내부습도관측치_0	co2관측치_0	...	시간당청색광량_23
24.3432	74.1231	470.3432	...	0
24.9882	72.5834	464.9882	...	0
...	...	...	...	...
24.9882	71.5832	463.9882	...	0
...	...	...	...	...
25.4298	68.8293	450.4298	...	0

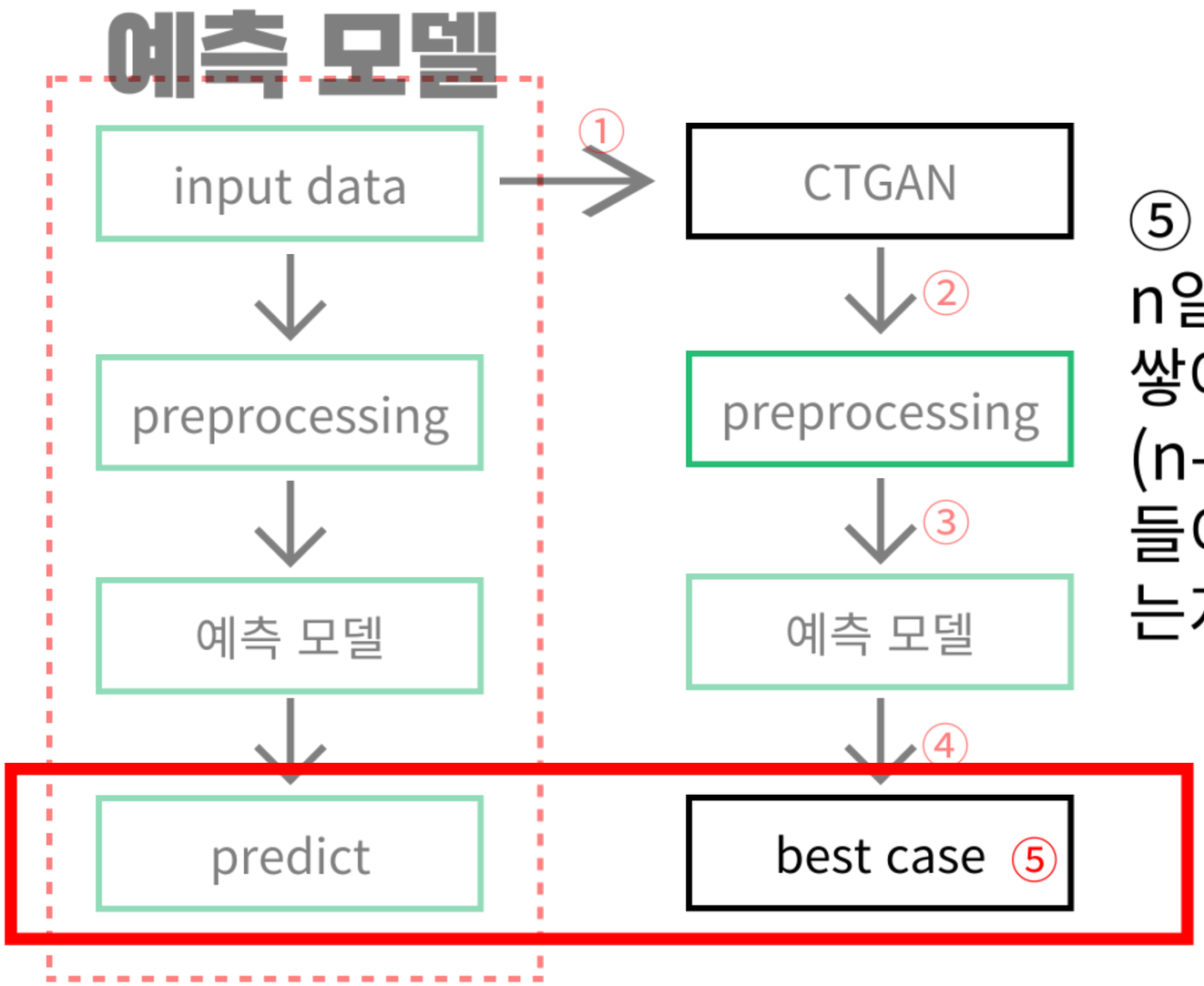
03. 생성 AI 모델 알고리즘



③ 전처리된 가짜 데이터를 예측 모델에 넣어 잎 중량을 예측

④ 예측한 잎 중량 중 최대 잎 중량값을 가진 케이스를 best case로 선별

03. 생성 AI 모델 알고리즘



⑤

n일 best data를 n-1일까지 쌓여온 best data와 함께 저장 (n-1일까지 누적된 생육 환경들이 n일에 얼마나 영향을 주는지 반영된 생성 모델임)

개발 환경

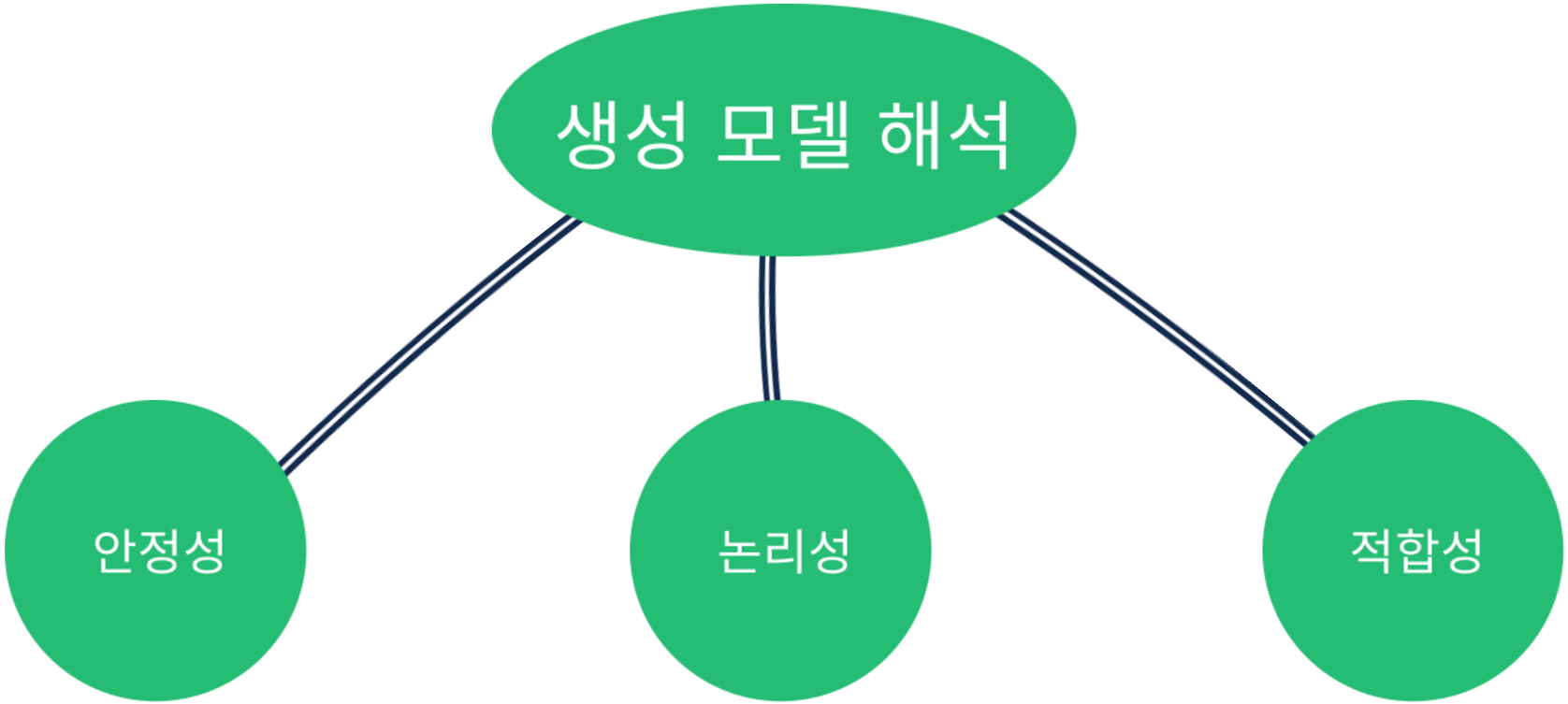
개발 플랫폼	Colab
epoch	100
generator_lr	0.0002
discriminator_lr	0.0002
seed	torch.manual_seed(0) np.random.seed(0)
batch_size	500

*그 외 파라미터들은 기본값을 사용

상추의 생육 환경 생성

04. 생성 AI 모델 결과 해석

04. 생성 AI 모델 결과 해석



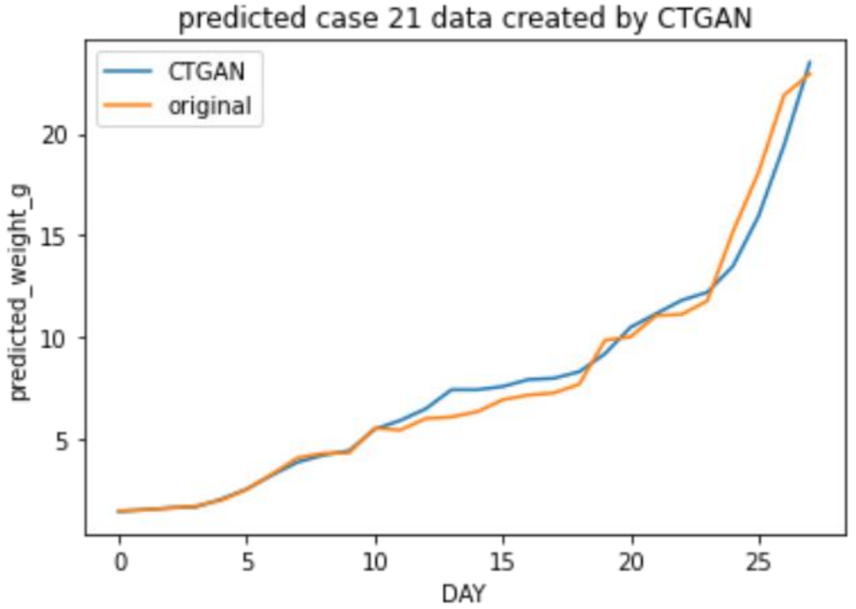
일별 최대 잎 중량을 도출할 수 있는 최적의 생육 환경 조성을 위해 160 이상 값을 생성하기 보단, 최소 잎 중량의 환경을 최대 잎 중량의 환경으로 변화시키는 것을 목표로 하여 데이터를 생성한 후에 예측 모델을 통해 최대 잎 중량을 도출해내는지 검증함

04. 생성 AI 모델 결과 해석

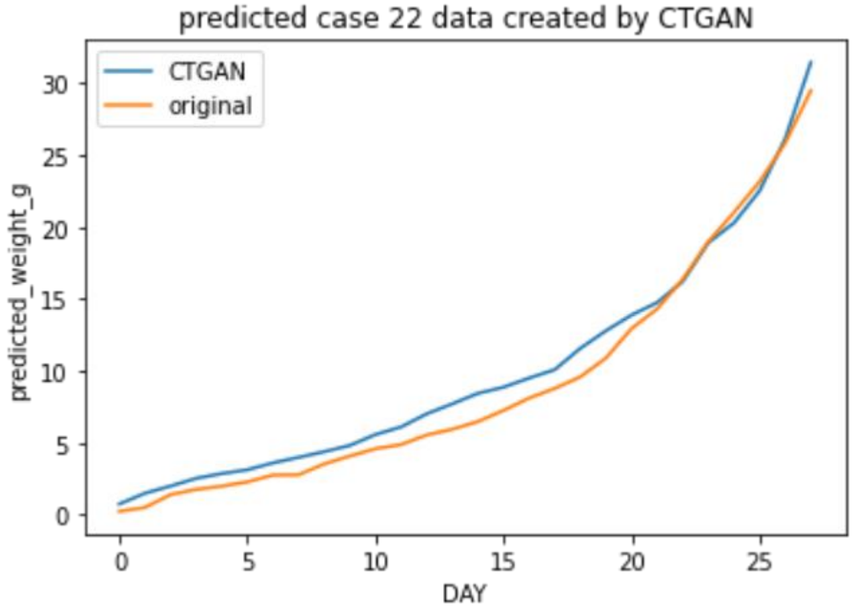
안정성

생성 AI 모델 결과에 대한 검증 방법

생성 모델에 train case을 input으로 넣어 얻은 가짜 데이터를
예측 모델에 넣었을 때 똑같은 최대 잎 중량이 나오는지 검증



train case 21에 대한 안정성



train case 22에 대한 안정성

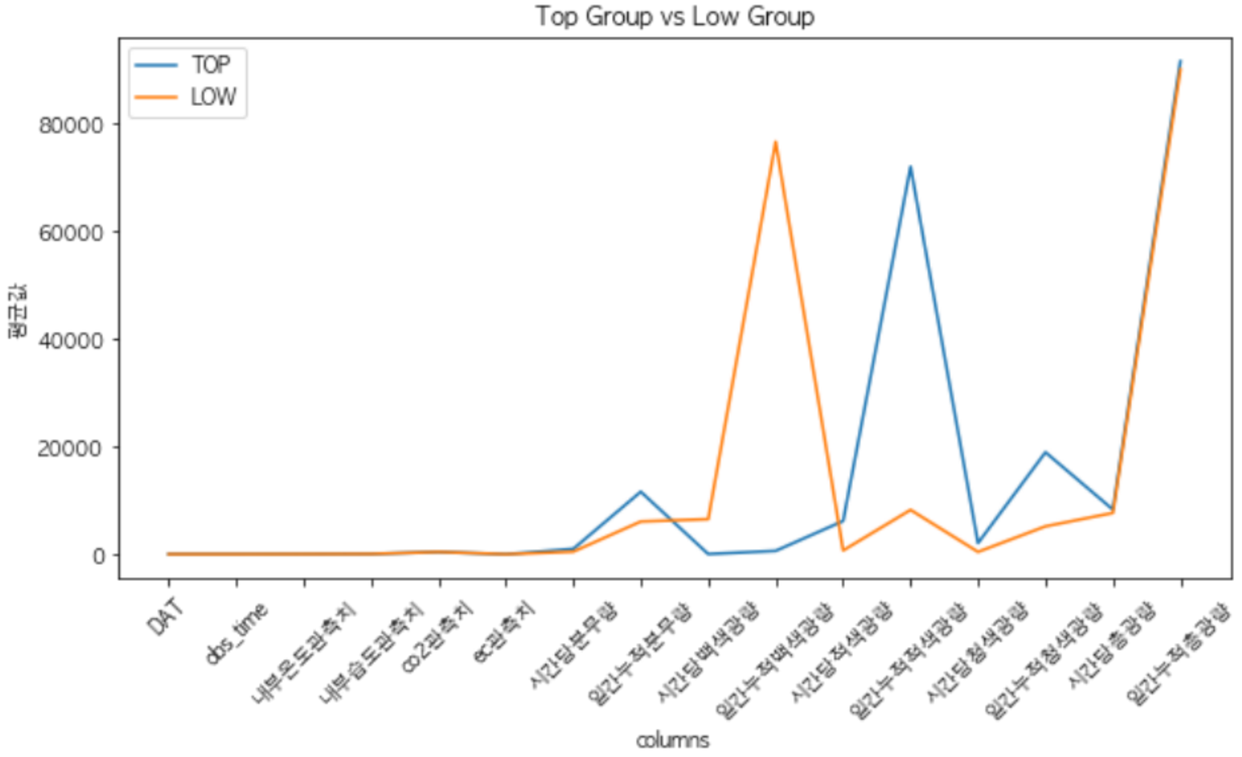
생성된 가짜 데이터가 원본 데이터의 분포를 잘 따라가며 만들어짐

04. 생성 AI 모델 결과 해석

논리성

조성된 생육 환경에 대한 해석 자료

target 값에 따라 상•중•하 그룹을 나눠 CTGAN을 학습하고,
그룹별로 생성된 데이터들끼리 특징들을 비교하여 그룹 환경 비교



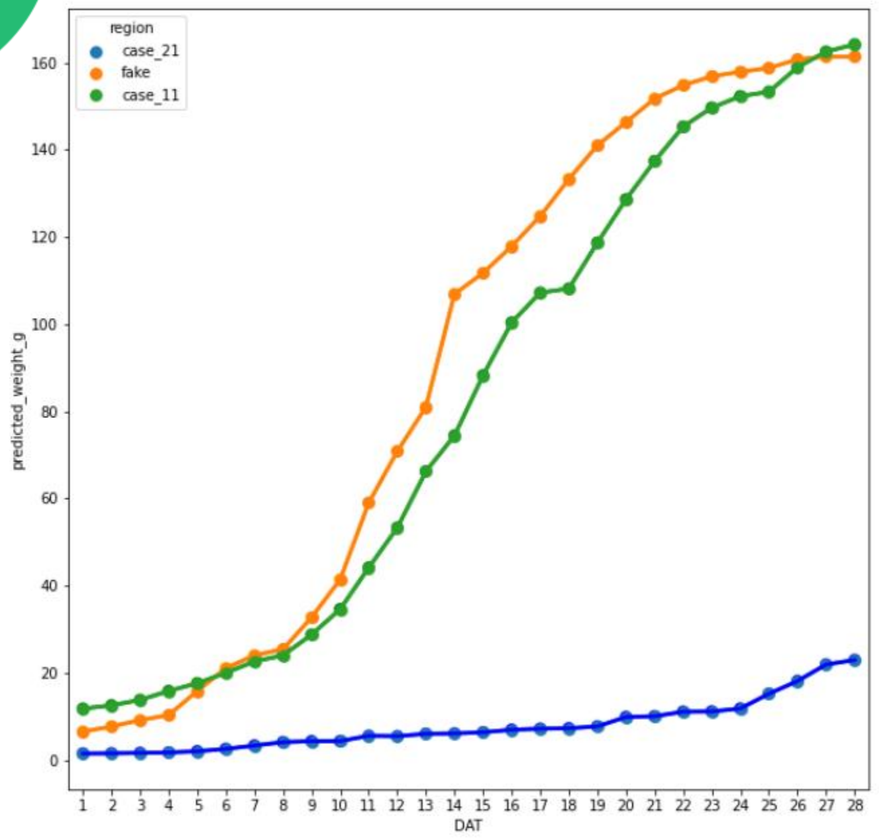
생성된 데이터로 얻은 target 값이 높은 상위 그룹은 하위 그룹보다 일간누적분무량, 일간누적적색광량, 일간누적청색광량이 높게 나왔음
반면에, 하위 그룹은 전체적으로 상위 그룹보다 낮은 값을 가지고 있으며 일간누적백색광량만은 압도적으로 큰 값을 가지고 있음

04. 생성 AI 모델 결과 해석

적합성

조성된 생육 환경의 적합성

target 값 2차 곡선 환경 비교



결론 : 예측모델로 마지막날 target이 약 20이 나오는 환경을 생성모델을 통해 마지막날 target 값을 약 160까지 예측을 할 수 있었음

train case 11이 최대 잎 중량을 가지는 best case인데 생성 데이터 fake가 case 11과 비슷한 곡선을 보여주는 것을 근거로 생성 모델을 통해 조성된 생육 환경이 적합하다는 것을 보여줌

상추의 생육 환경 생성

05. OUTRO

insight

1. 상추의 생육에는 분무량이 중요한것으로 파악

2. train 데이터에서 마지막날 상추의 무게 약 20을 가진 case 21의 환경을 생성 모델을 통해 비슷한 환경의 데이터를 생성 후 예측 모델의 타겟값 예측으로 비슷한 환경이 조성됨을 확인하였다.

3. 분무량을 자주, 보통(약1200)의 양으로 분무를 해주는 환경 조성만으로도 일별 상추 잎 중량을 최대화 시킬 수 있었다.

상추의 생육 환경 생성

05. OUTRO

곽명빈

Hypothesis
EDA
Preprocessing
Predictive model
-Catboost
Generative model
- GAN, CTGAN

최다희

Hypothesis
Preprocessing
Predictive model
-Catboost
Generative model
- GAN, CTGAN

전주혁

Hypothesis
Feature engineering
- LPF, Kalman ···
Predictive model
-Autogluon
Generative model
- CTGAN

반소희

Hypothesis
Preprocessing
Generative model
- CTGAN

김기범

EDA
Predictive model
-XGBoost
PPT

감사합니다.

상추세요 팀