LG AI Research 주최 데이콘 주관

자율주행 센서의 안테나 성능 예측 AI

스태킹 앙상블 모델을 통한 성능 향상

KOPS팀 김민석 박정기 장시연 오윤주

2022

KOPS 팀 소개

김민석 팀장

프로젝트 개발 총괄
스태킹 앙상블 코드 개발

박정기

전처리 실험
데이터 시각화 및 분석

오윤주

칼럼 K Best 실험
모델 자원 사용량 분석

장시연

전처리 실험
PCA 적용 및 분석

목차

01

분석 개요

정형 데이터셋을 통한
다중 출력 회귀 예측

02

데이터 분석

PCA를 통한
예측 정확도 향상

03

모델링

하이퍼파라미터 튜닝
스태킹 앙상블을 통한
성능 향상

04

적용 가능성

소요 시간 &
자원 소모

01

분석 개요

정형 데이터셋을 통한 다중 출력 회귀 예측

문제 정의

공정 데이터를 활용하여 Radar 센서의 안테나 성능 예측을 위한 AI 모델 개발

⇩

정형 데이터 X Feature 56개를 통해 Y Feature 14개를 예측하는 AI 모델 개발

가중치가 부여된 Normalized RMSE의 총합으로 AI 모델의 성능을 평가

딥러닝 모델 vs Tree기반 앙상블 모델

딥러닝 모델

정형 데이터에서
over-parameterized 되어
overfitting을 유발함

단점을 보완한 TabNet이
있지만 최적화 난이도가 높음

Tree기반 앙상블 모델

학습이 빠르고 쉽게 개발 가능

높은 해석력을 가져 딥러닝 모델에
비해 상대적으로 해석이 용이

다중 출력 회귀 모델

Sklearn MultiOutputRegressor

다중 출력을 지원하지 않는 모델을 Y Feature 마다 회귀 모델을 생성하여 다중 출력을 지원

단점으로 모든 모델에 동일 파라미터 적용

Sklearn RegressorChain

예측값을 다음 예측에 사용해 타깃들간의 상관관계를 고려하는 다중 출력 모델을 지원

Output의 낮은 정확도로 인해 성능 하락

각 모델에 다른 파라미터를 적용가능한 MultiOutput Regressor 모델을 직접 개발

02

데이터 분석

PCA를 통한 예측 정확도 향상

데이터 전처리 방법 탐색

○ 기준 데이터

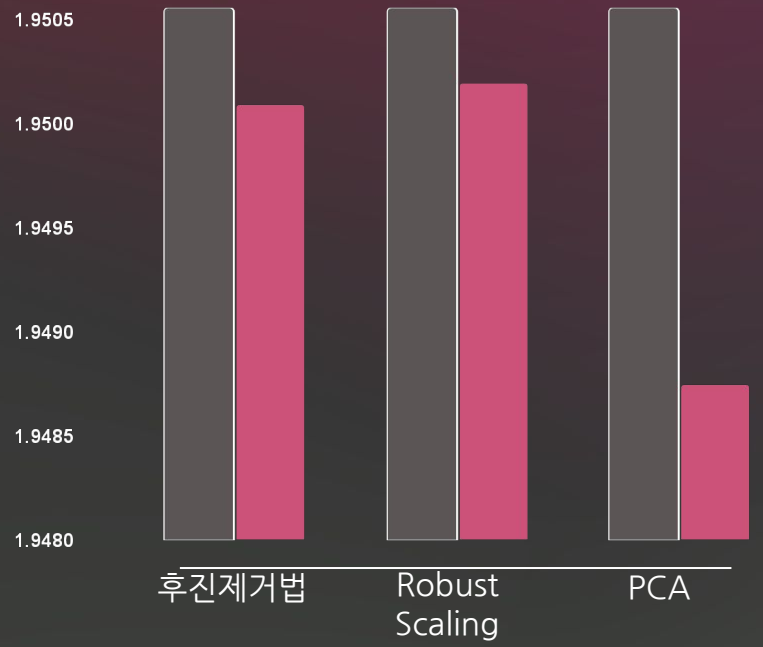
10-Fold CV LGBMRegressor score

● 전처리 데이터

후진제거법, Robust Scaling, PCA

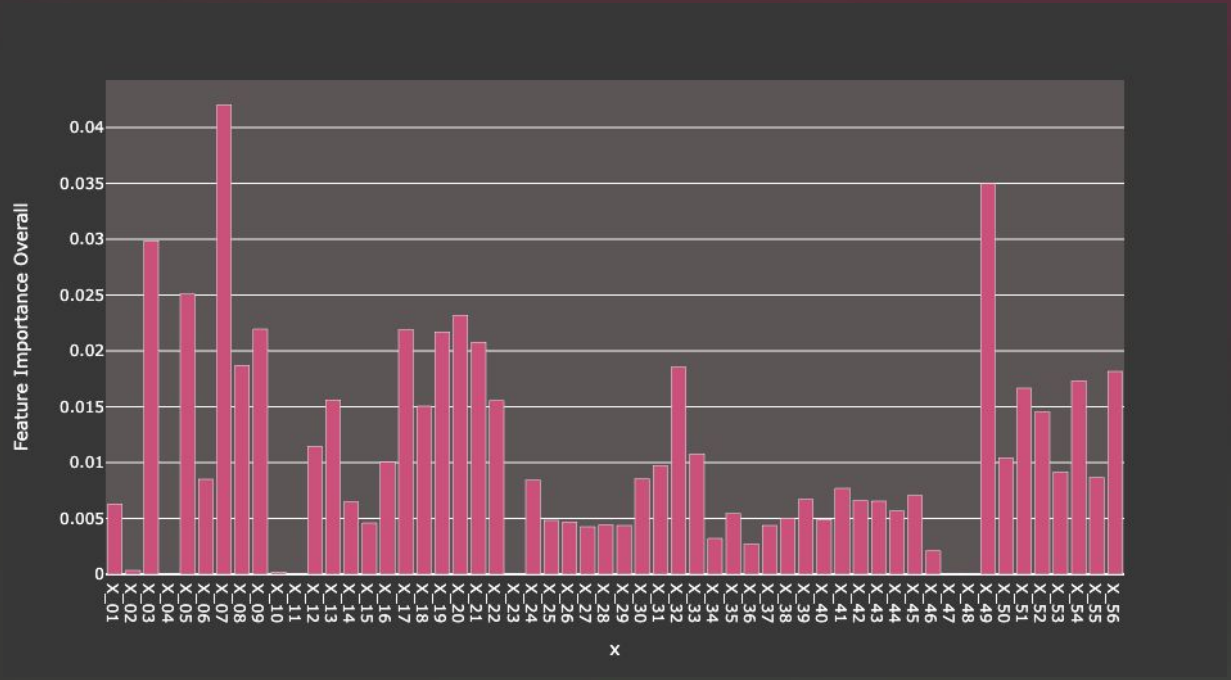
성능이 가장 좋은 방법으로 데이터 전처리

PCA 선정



이외의 변수선택법, preprocessing의 경우 유의미한 성능 향상이 없음. PCA의 경우 최적화가 완료된 수치임.

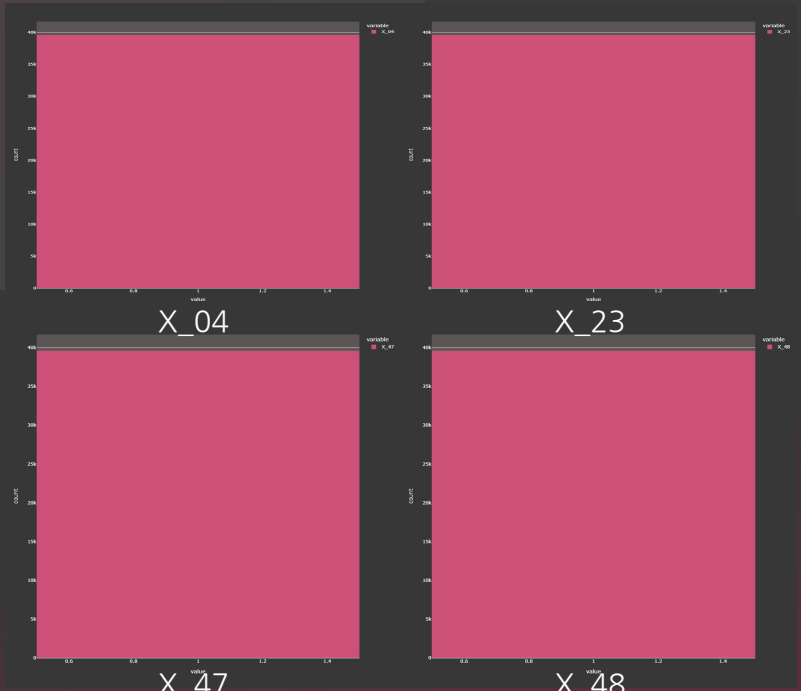
SHAP(XAI)을 통한 Feature Selection



X_02, X_04,
X_10, X_11,
X_23, X_47, X_48

위 7개 Feature는
Y Feature 14개에
미치는 평균
Feature
Importance가
매우 낮음

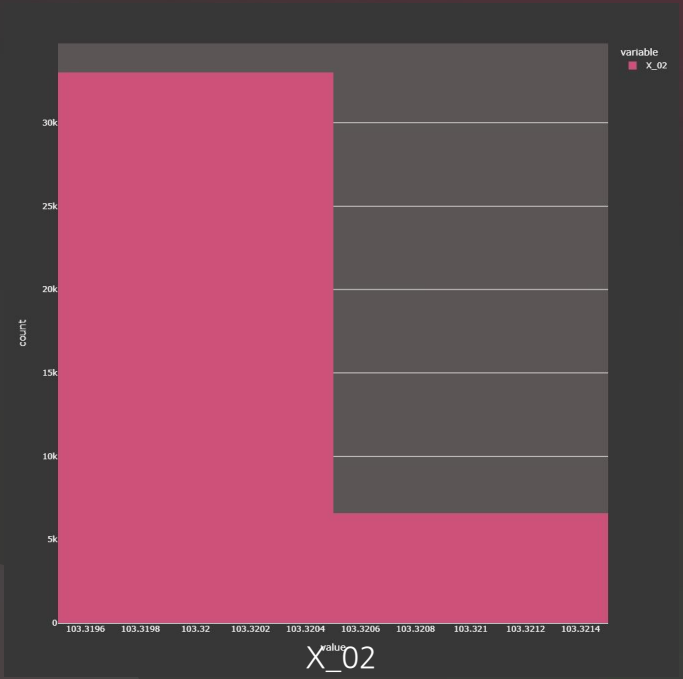
제거한 Feature



X_04, X_23, X_47, X_48은

모든 Feature가 1인 합격이므로 제거

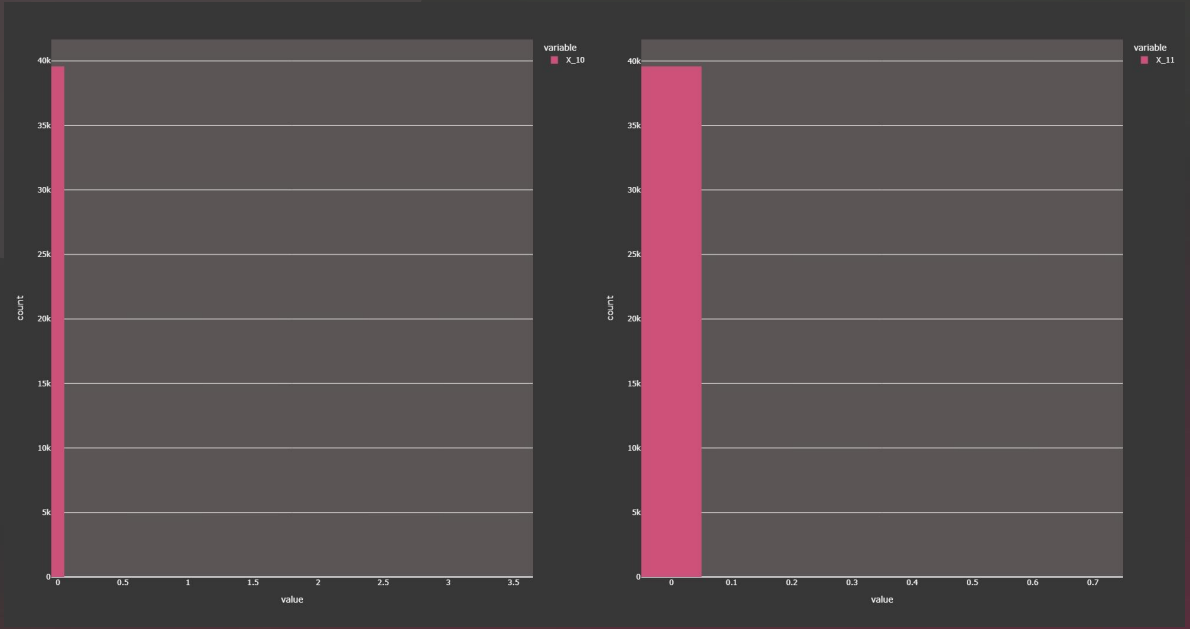
제거한 Feature



실제 측정값이 0.001 차이(0.0009%) 이며

SHAP의 Feature Importance가 매우 낮아 제거

제거한 Feature

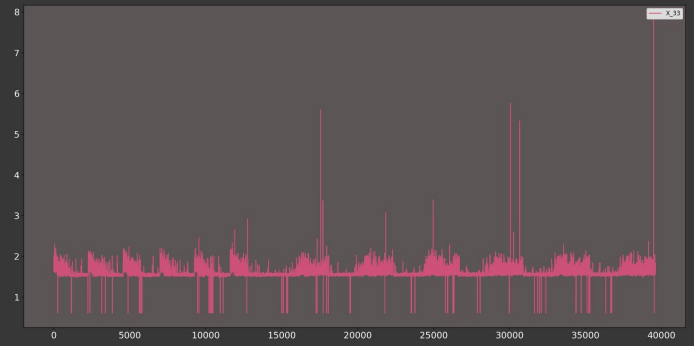


X_10 X_11

X_10, X_11의
0으로 표현된 값은
결측치임
(방열 재료 무게)

이러한 결측치가
Feature의 대부분을
차지하므로 Feature
제거

제거한 이상치

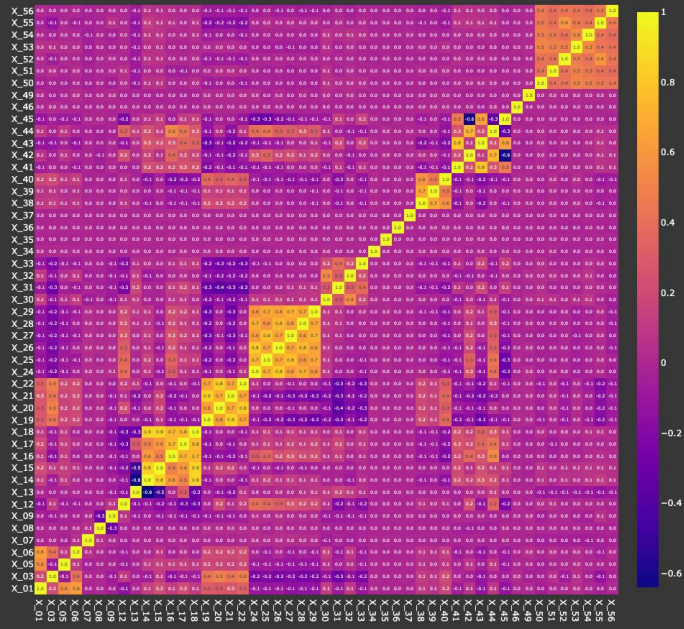


X_33

X_33(스크류 삽입 깊이)에서

6이 넘어가는 이상치 제거

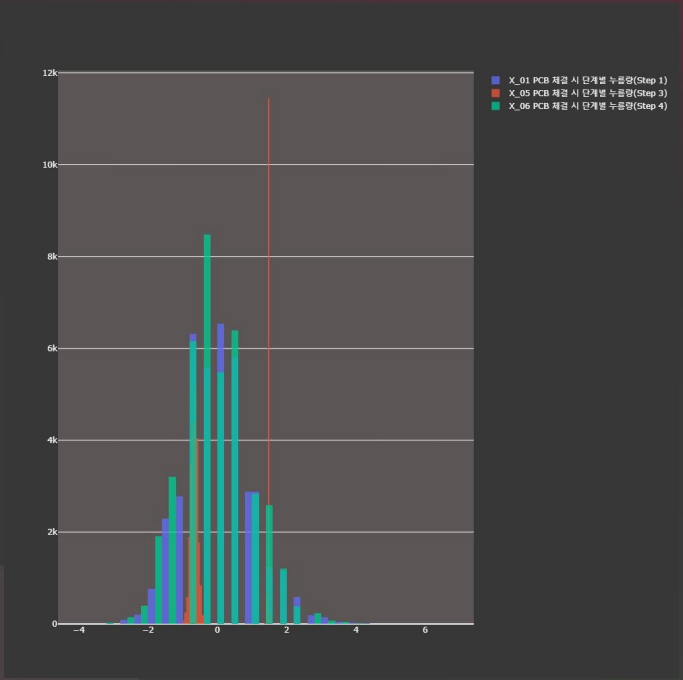
X Feature 상관계수 행렬



일부 인자들 사이에 강한 상관관계가 있음

다중공선성 제거를 위해 PCA를 통한 차원축소 시도

PCA 차원 축소

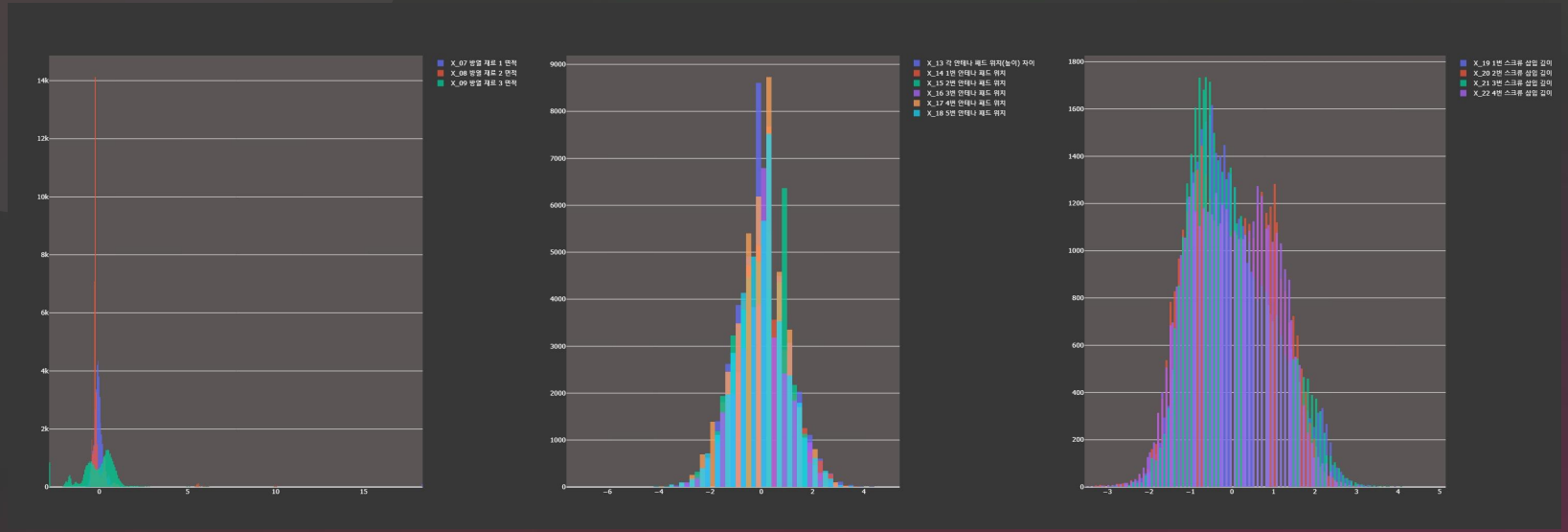


Feature 간의 상관계수가 높거나 측정 항목 사이에 연관이 있는 Feature의 경우

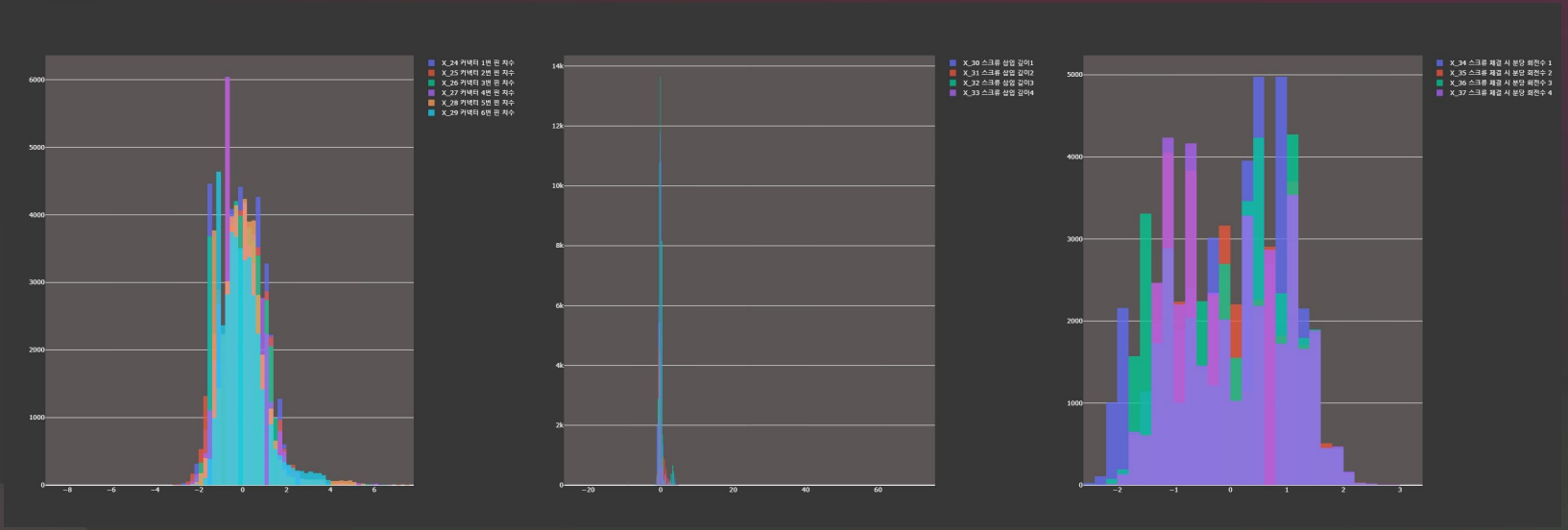
MinMaxScaling 후
Multi Histogram 분석

상관계수가 높은 인자의 경우 Histogram의 모양이 유사함

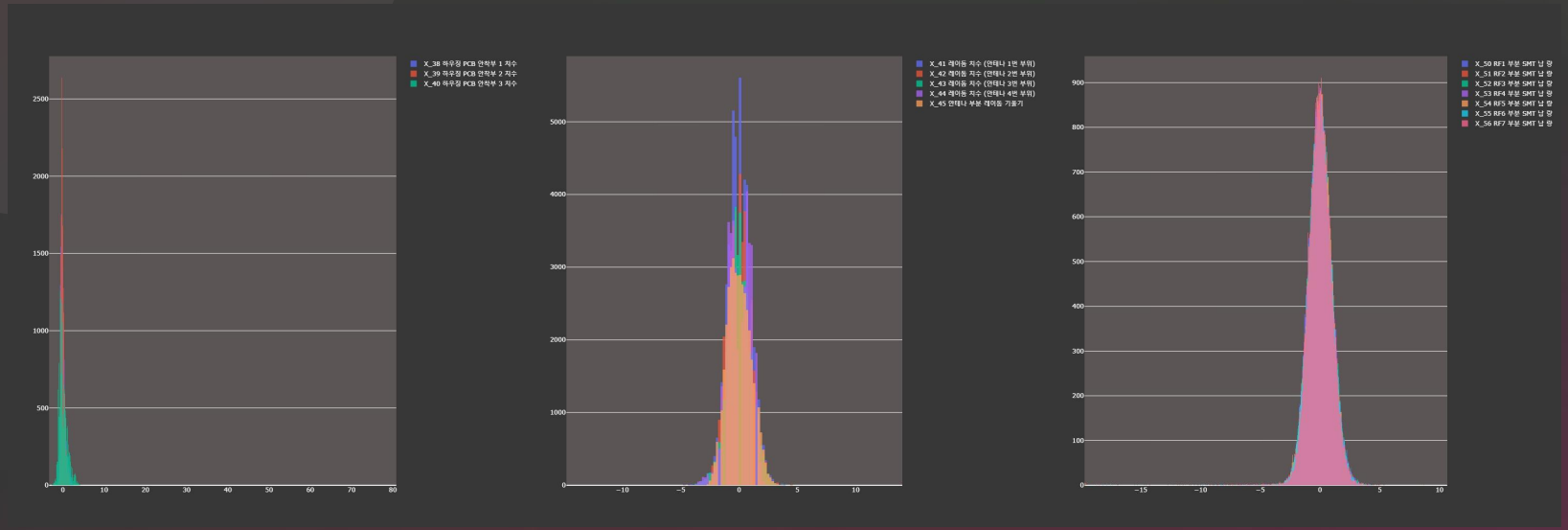
PCA 차원 축소


X_07 방열 재료 1 면적
X_08 방열 재료 2 면적
X_09 방열 재료 3 면적
X_13 각 안테나 패드 위치(높이) 차이
X_14 1번 안테나 패드 위치
X_15 2번 안테나 패드 위치
X_16 3번 안테나 패드 위치
X_17 4번 안테나 패드 위치
X_18 5번 안테나 패드 위치
X_19 1번 스크류 삽입 깊이
X_20 2번 스크류 삽입 깊이
X_21 3번 스크류 삽입 깊이
X_22 4번 스크류 삽입 깊이

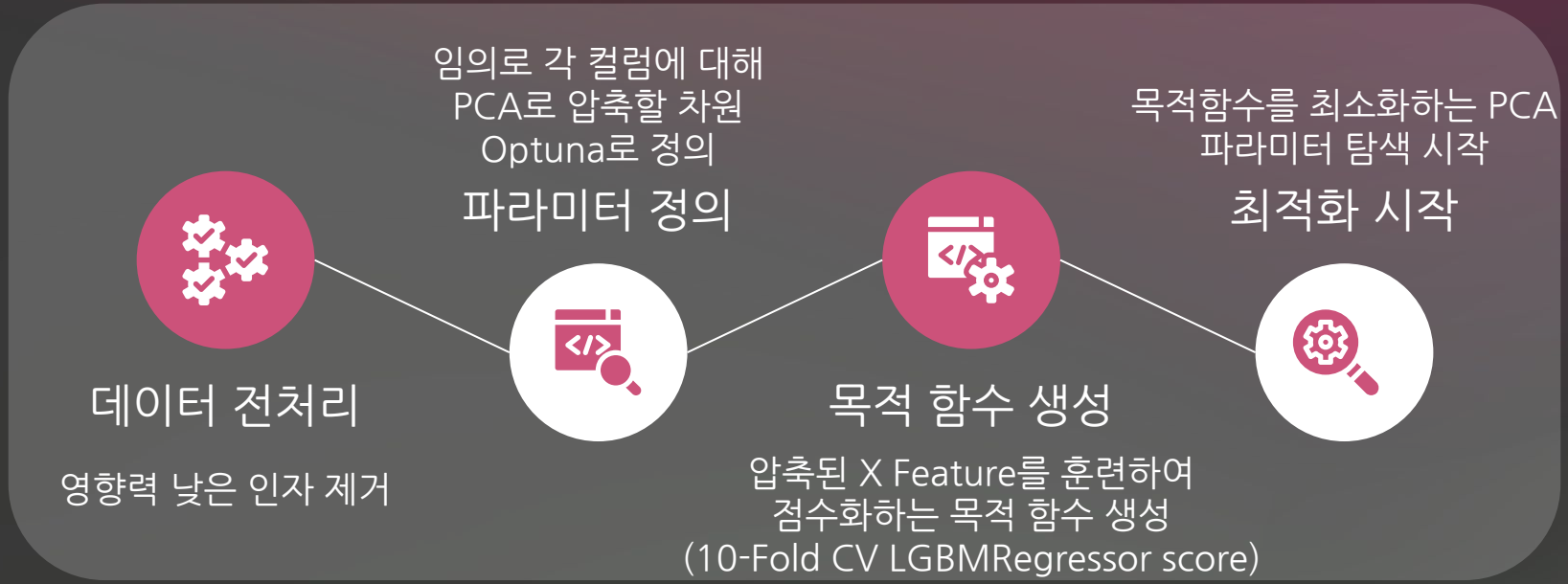
PCA 차원 축소


X_24 커넥터 1번 핀 저수
X_25 커넥터 2번 핀 저수
X_26 커넥터 3번 핀 저수
X_27 커넥터 4번 핀 저수
X_28 커넥터 5번 핀 저수
X_29 커넥터 6번 핀 저수
X_30 스크류 삽입 깊이1
X_31 스크류 삽입 깊이2
X_32 스크류 삽입 깊이3
X_33 스크류 삽입 깊이4
X_34 스크류 체결 시 분당 회전수 1
X_35 스크류 체결 시 분당 회전수 2
X_36 스크류 체결 시 분당 회전수 3
X_37 스크류 체결 시 분당 회전수 4

PCA 차원 축소

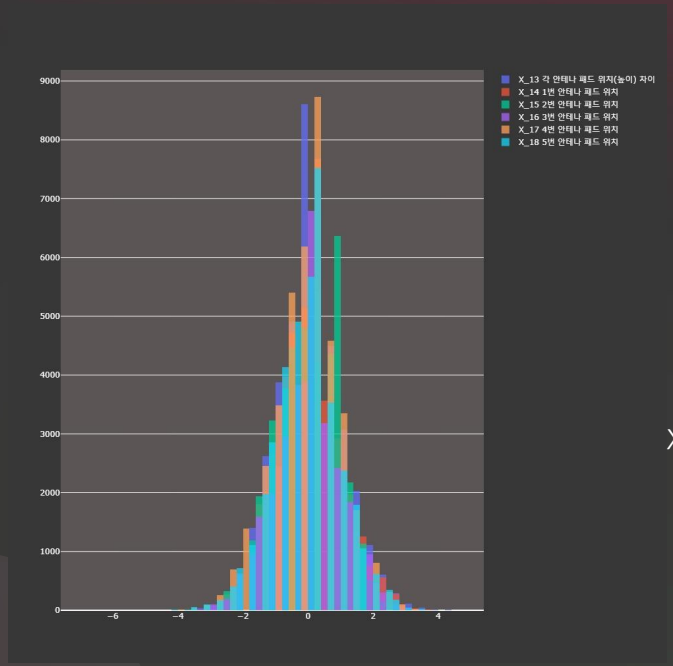

X_38 좌우장 PCB 안착부 1 치수
X_39 좌우장 PCB 안착부 2 치수
X_40 좌우장 PCB 안착부 3 치수
X_41 레이돔 치수 (안테나 1번 부위)
X_42 레이돔 치수 (안테나 2번 부위)
X_43 레이돔 치수 (안테나 3번 부위)
X_44 레이돔 치수 (안테나 4번 부위)
X_45 안테나 부분 레이돔 기울기
X_50 RF1 부분 SMT 납 량
X_51 RF2 부분 SMT 납 량
X_52 RF3 부분 SMT 납 량
X_53 RF4 부분 SMT 납 량
X_54 RF5 부분 SMT 납 량
X_55 RF6 부분 SMT 납 량
X_56 RF7 부분 SMT 납 량

Optuna를 통한 PCA 최적화



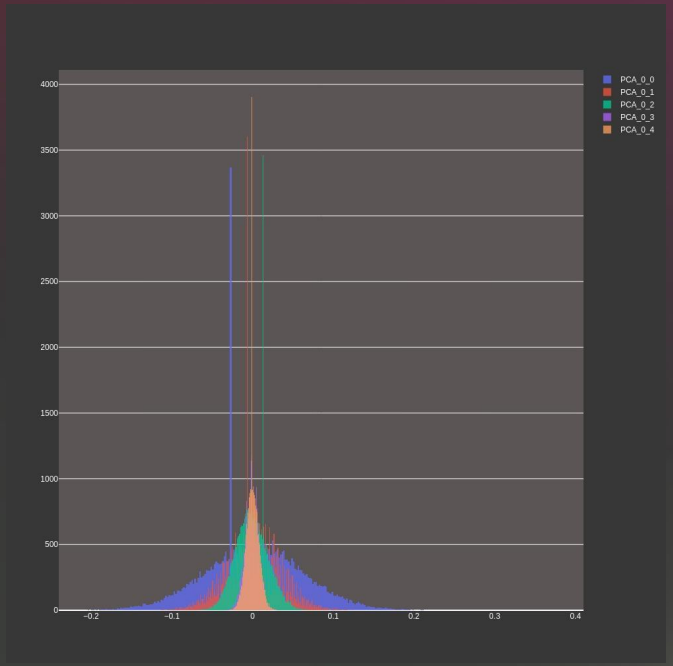
이후 하이퍼파라미터 튜닝 시 Optuna에 대한 자세히 설명

PCA 결과

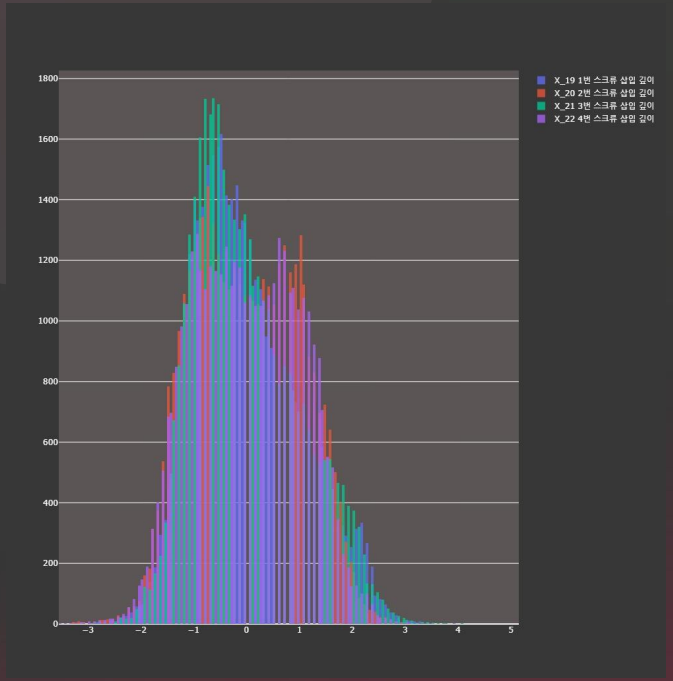


Optuna
Optimization

X_13, 14, 15, 16, 17, 18
5차원으로 축소

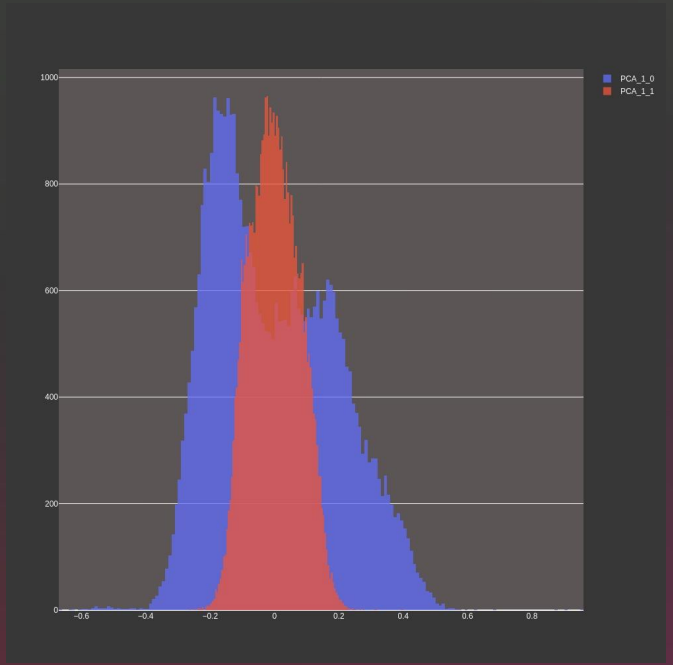


PCA 결과



Optuna
Optimization

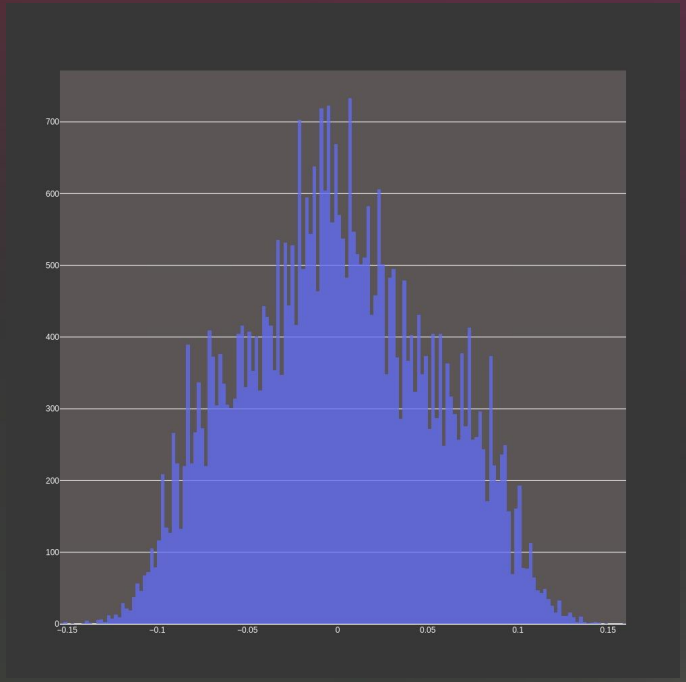
X_19,20,21,22
2차원으로 축소



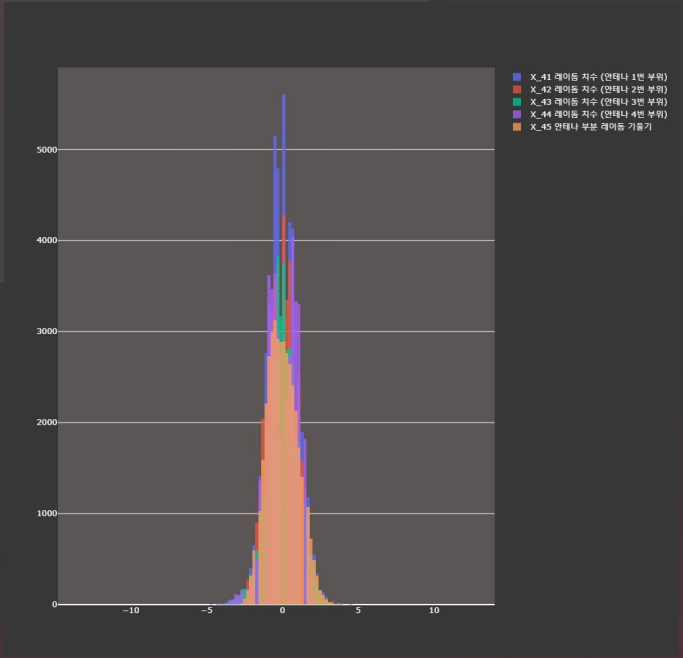
PCA 결과

Optuna
Optimization

X_34,35,36,37
1차원으로 축소

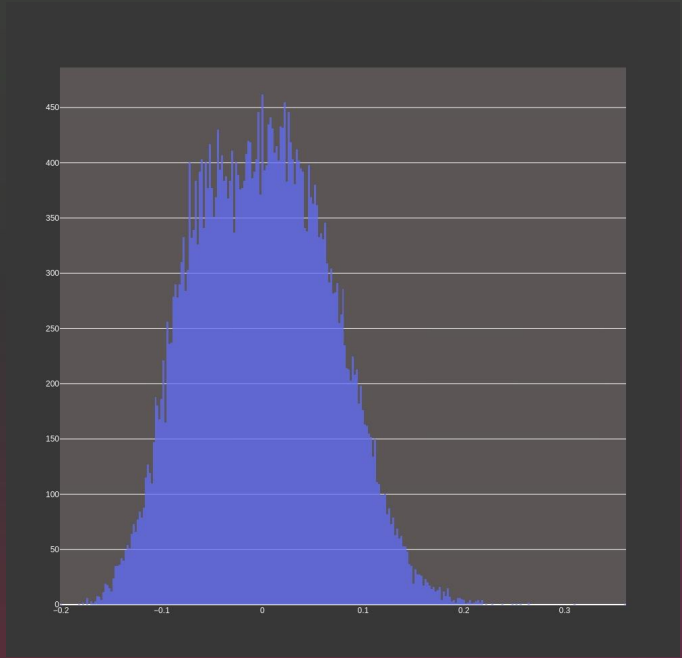


PCA 결과

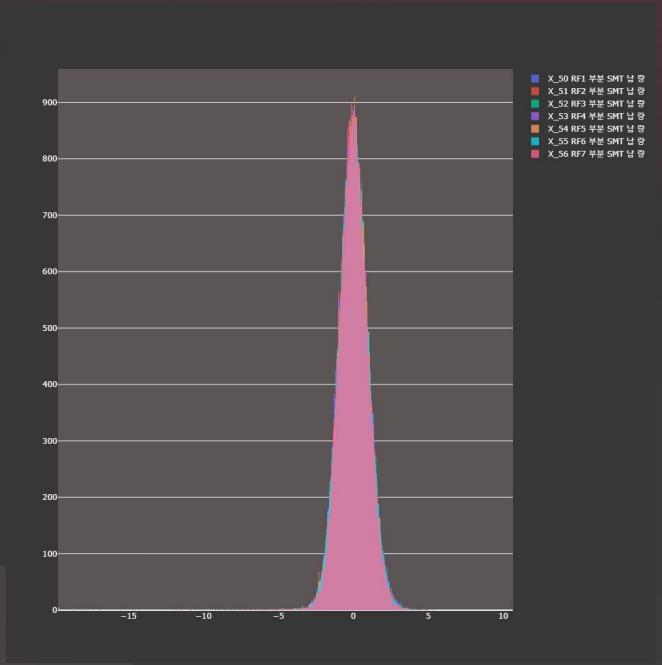


Optuna
Optimization

X_41,42,43,44,45
1차원으로 축소

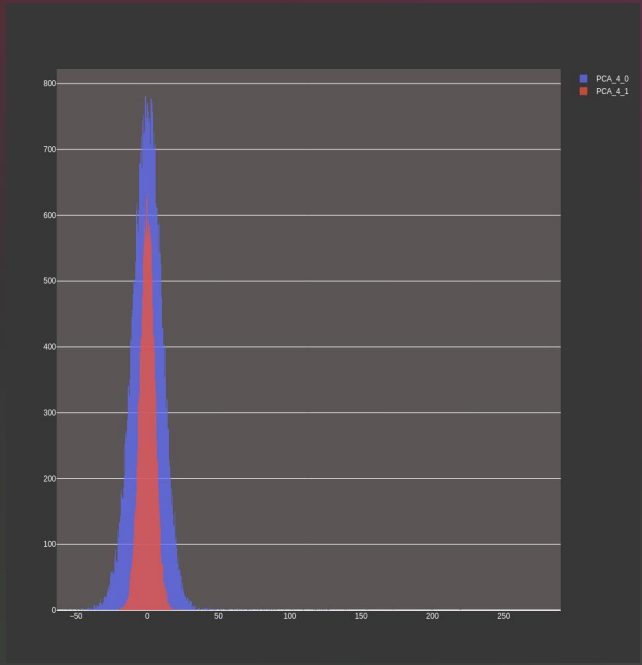


PCA 결과



Optuna
Optimization

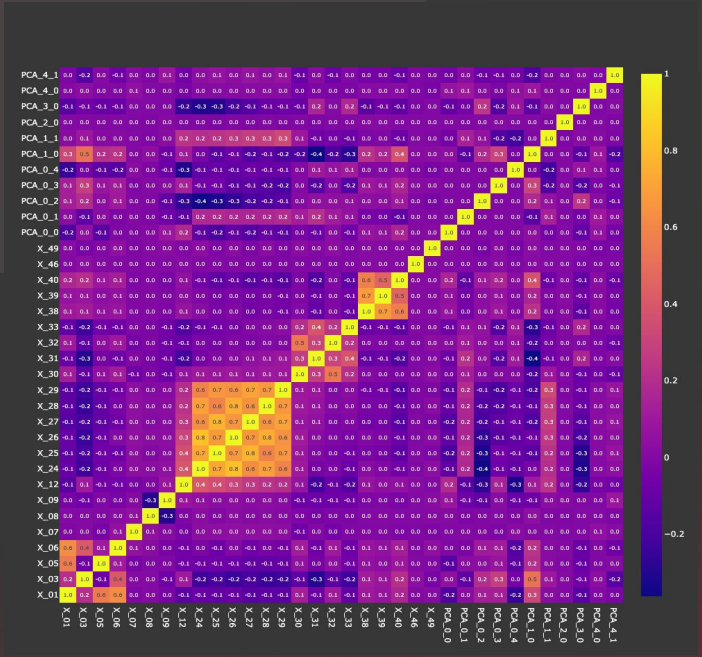
X_50,51,52,53,54,55,56
2차원으로 축소



PCA Columns


X_01 … X_12 X_13 X_14 X_15 X_16 X_17 X_18 X_19 X_20 X_21 X_22 X_24 … X_33
X_01 … X_12 PCA_0_0 PCA_0_1 PCA_0_2 PCA_0_3 PCA0_4 PCA_1_0 PCA_1_1 X_24 … X_33
X_34 X_35 X_36 X_37 … X_41 X_42 X_43 X_44 X_45 … X_50 X_51 X_52 X_53 X_54 X_55 X_56
PCA_2_0 … PCA_3_0 … PCA_4_0 PCA_4_1

PCA 후 X Feature 상관계수 행렬



PCA를 통해 다중공선성이 상당부분 제거됨

Feature가 56 개에서 34개로 축소됨

10-Fold CV LGBMRegressor Score가
1.9506에서 1.9488로 개선됨

03

모델링

하이퍼파라미터 튜닝과 스태킹 앙상블을 통한 성능 향상

데이터 분리 방법 선정

Holdout

데이터셋의 분할에 민감한 성능 추정
과대적합의 위험성이 있음
소요 시간이 상대적으로 적음

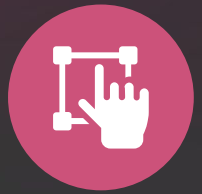
100회 이상 훈련하는 하이퍼파라미터
튜닝을 위해 속도 빠른 Holdout을 사용

K-Fold

데이터셋의 분할에 둔감한 성능 추정
성능 추정이 일반화되어 과적합 방지
소요 시간이 K배로 늘어남

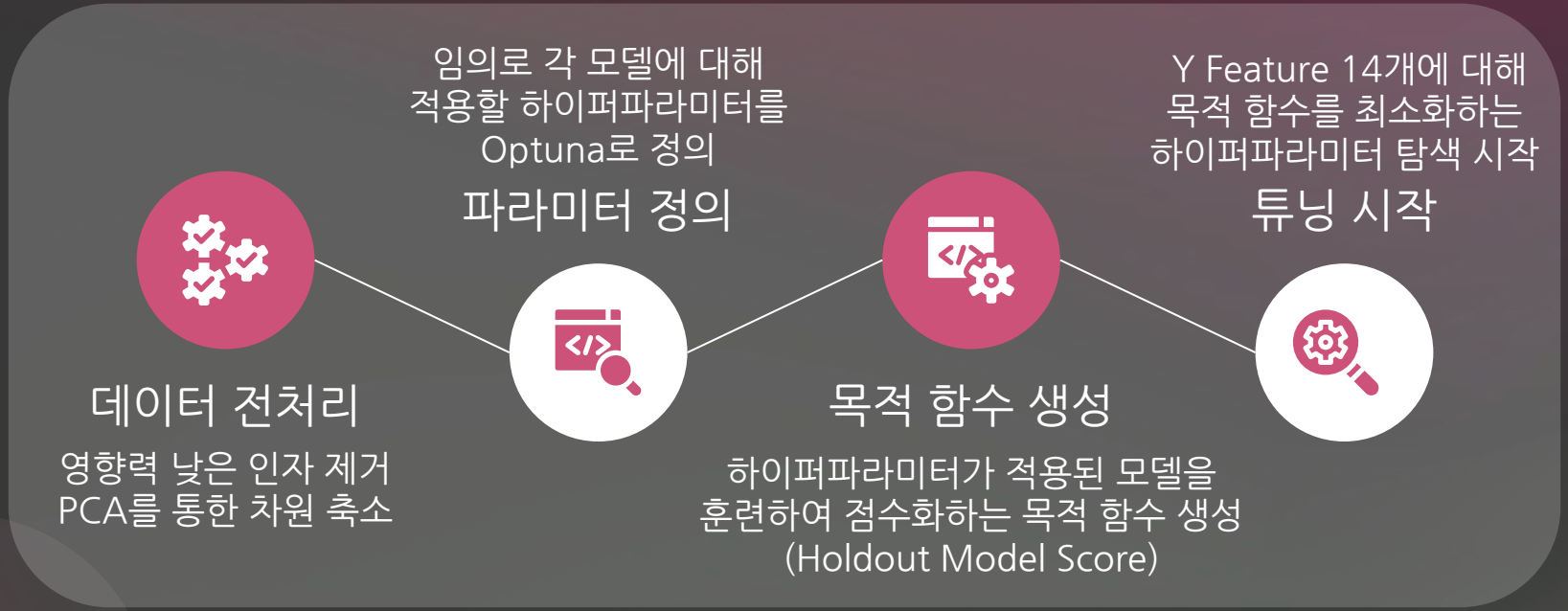
스태킹 앙상블의 메타 데이터셋 생성시
과적합 방지를 위해 K-Fold를 사용

하이퍼파라미터 튜닝 방법




Grid Search	Random Search	Bayesian Optimization	Optuna
가능한 모든 조합의 하이퍼파라미터로 훈련 시간이 매우 오래 걸림	임의의 조합을 추출하여 최적의 조합을 찾는 방법 넓은 범위를 탐색하여 비효율적	목적함수를 최대 또는 최소로 하는 최적해를 찾는 방법 효율적으로 탐색함	앞선 알고리즘을 포함한 최신 동향의 다양한 최적화 알고리즘을 갖춤 Optuna로 선정

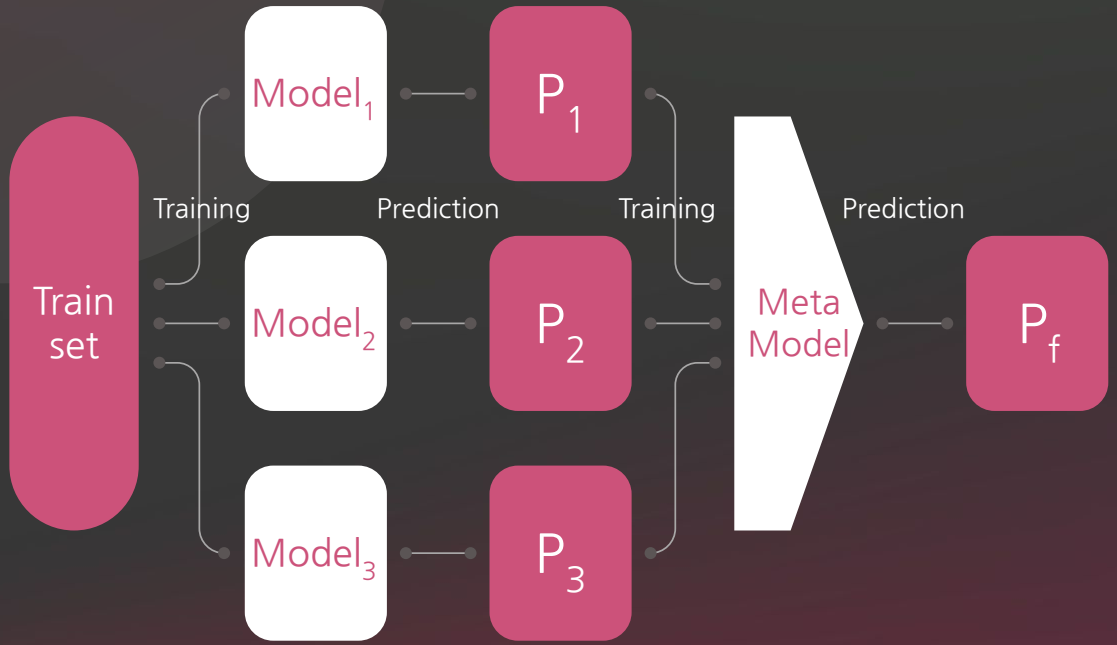
하이퍼파라미터 튜닝 과정



하이퍼파라미터 튜닝 유무 비교

10-Fold CV score	튜닝 전	튜닝 후	성능 향상
HistGradientBoosting	1.95030	1.94523	0.260%
XGBRegressor	1.99406	1.93597	2.913%
LGBMRegressor	1.94875	1.93851	0.525%
CatBoostRegressor	1.94680	1.93860	0.4212%

스태킹 앙상블



훈련 데이터 셋을 통해 개별 알고리즘이 결과를 예측

⇩

예측한 결과를 최종 메타 데이터 세트로 제작

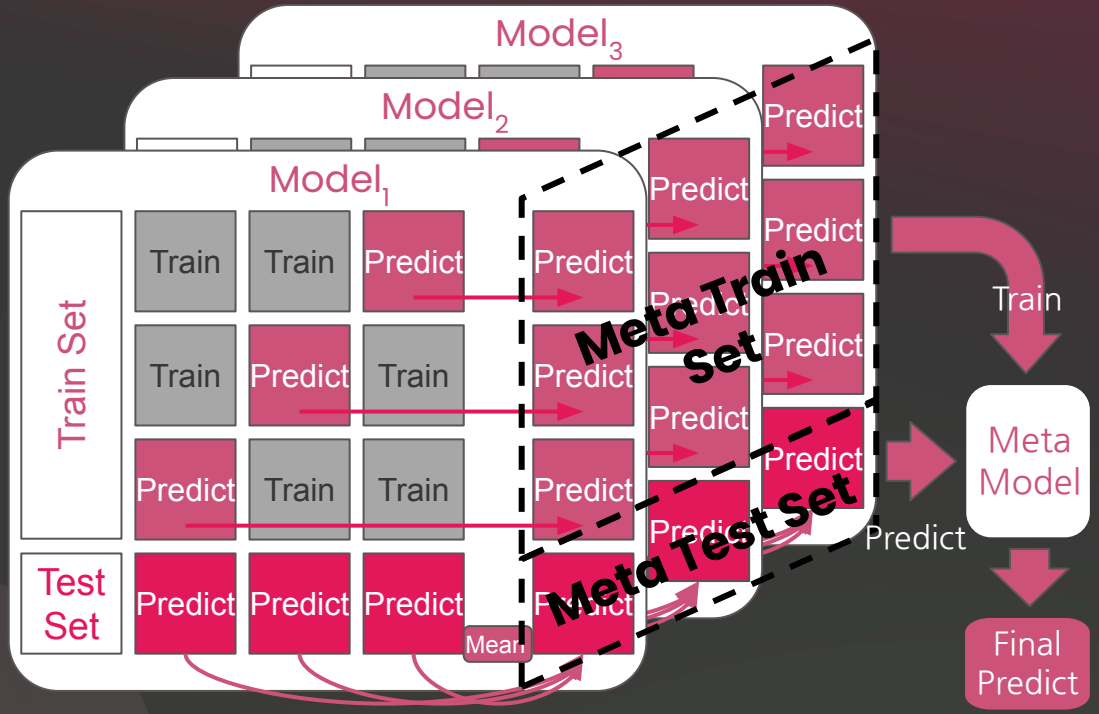
⇩

메타 데이터 세트로 메타 모델의 최종 학습을 수행

⇩

테스트 데이터를 기반으로 최종 예측을 수행

K-fold CV 스태킹 앙상블



Train set을 K개의 fold로 나눔

⇩

1개 fold를 검증용 데이터 폴드로 사용 나머지 폴드 이용 개별 모델 학습

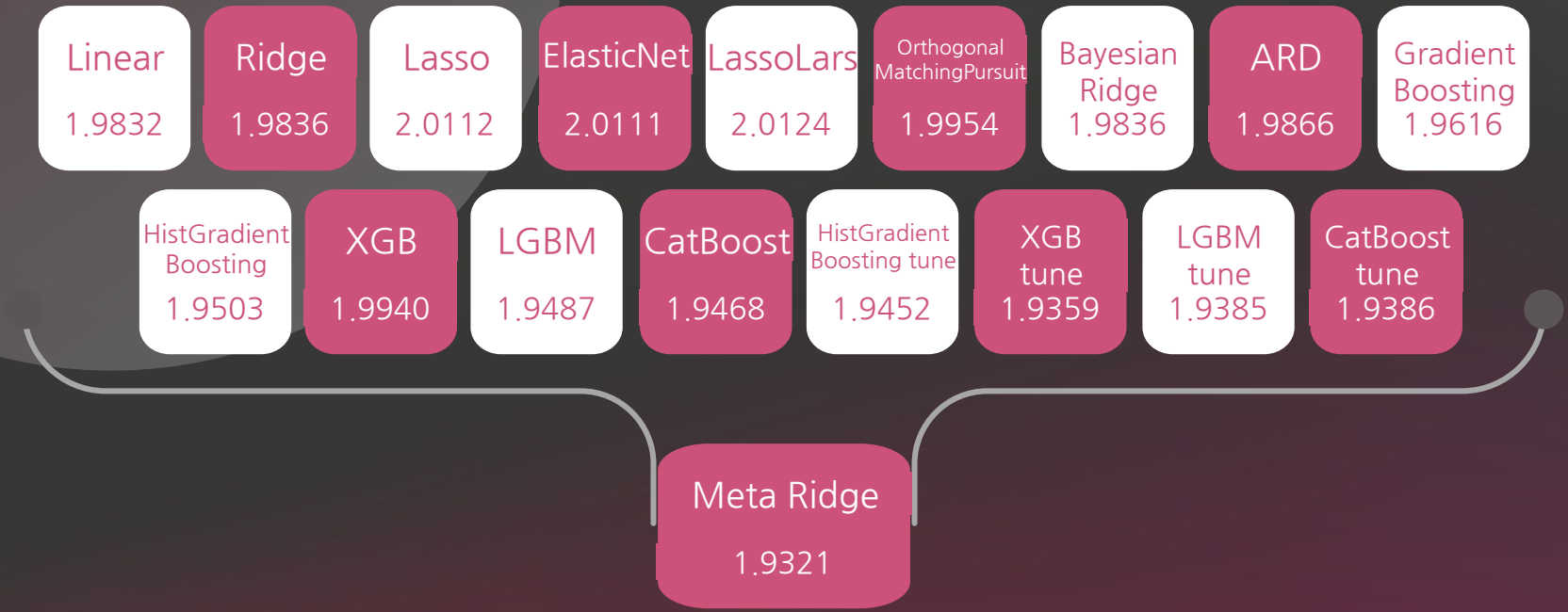
⇩

위 로직 K번 반복 및 Meta Data Set에 결과 저장

⇩

위에서 생성된 Meta Data Set을 메타 모델에 학습 및 예측 수행

스태킹 앙상블 CV Score



속도 향상 전략

CV(K-Fold) 기반 스태킹 앙상블

기반 모델 생성시 Train set을 K번 학습 후 메타 모델 학습
예측시 학습된 모델로 K번 예측 후 평균값으로 메타 모델 예측

K개의 모델이 생성되어 훈련된 모델의 용량과
예측 소요시간이 K배로 증가함

모델 경량화 전략

기반 모델 생성시 Train set을 K번 학습 후 메타 모델 학습
Train set 전체를 학습한 기반 모델 하나를 저장하여 예측시
1개의 예측값으로 메타 모델 예측

메타 모델 학습시 오버피팅 방지와 동시에
용량과 예측 소요시간을 1/K로 줄일 수 있음

04

적용 가능성

소요 시간 & 자원 소모

System info

Ubuntu

18.04.6 LTS

Python

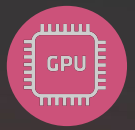
3.7.13

RAM

24GB

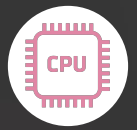
Model Size

718MB



GPU

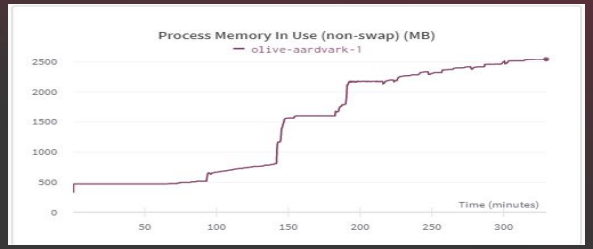
Tesla T4

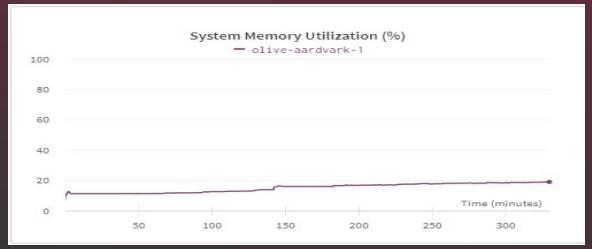


CPU

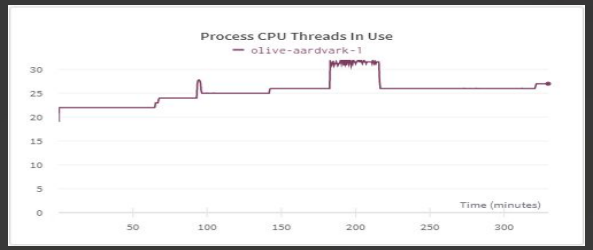
2CPU

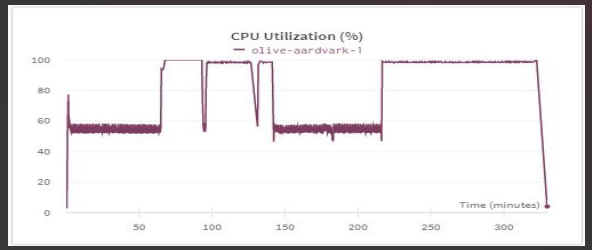
자원 소모량 (모델 훈련 시)



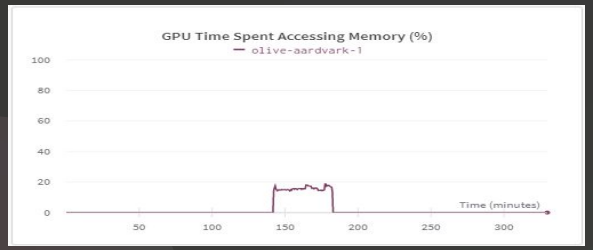


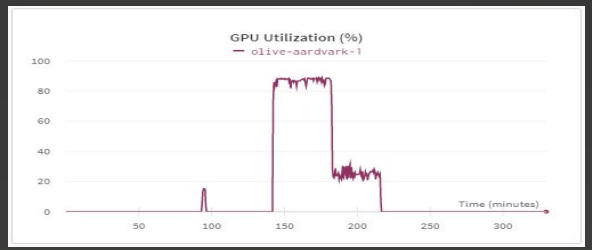
총 소요 시간 : 5시간 18분 10초
총 메모리 사용량 : 2212 MB
(초기 메모리 334MB)





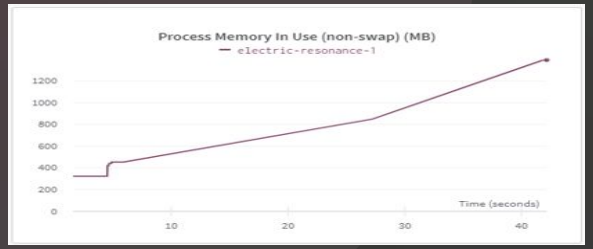
최대 CPU thread 사용량 : 32
CPU Utilization 은 50% ~ 100%
으로 대부분 일정

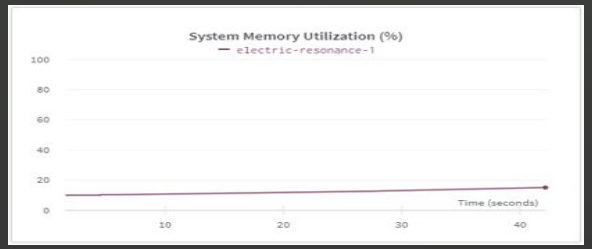




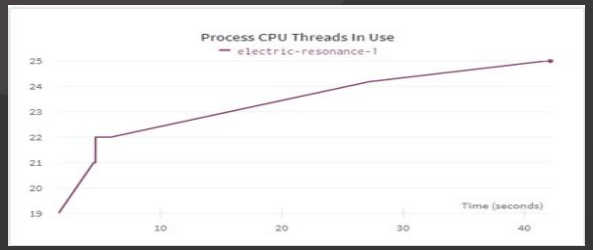
XGB, LGBM의 경우, GPU 최대
사용량 20% 로 40분간 사용

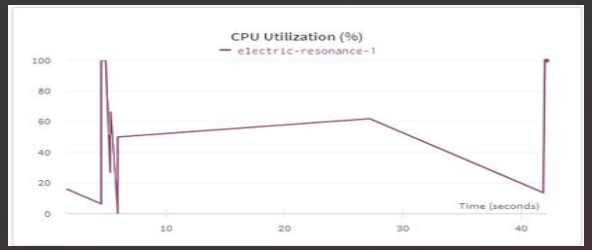
자원 소모량 (훈련된 모델 불러오기 시)



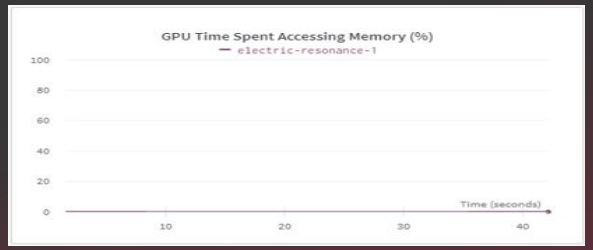


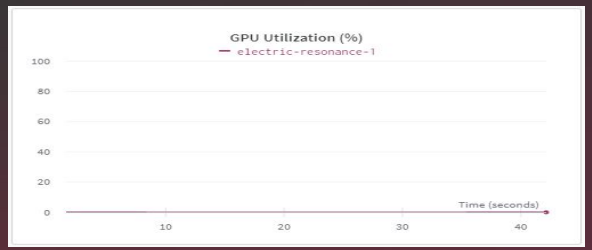
총 소요 시간 : 37초
총 메모리 사용량 : 1071 MB
(초기 메모리 322MB)





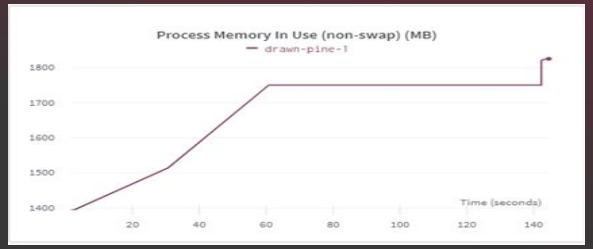
최대 CPU thread 사용량 : 25

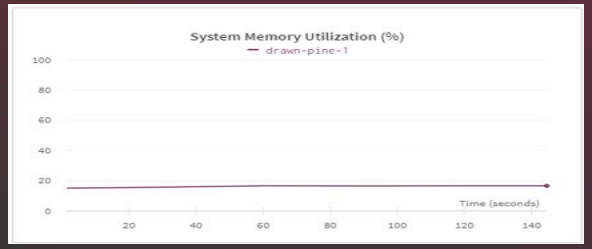




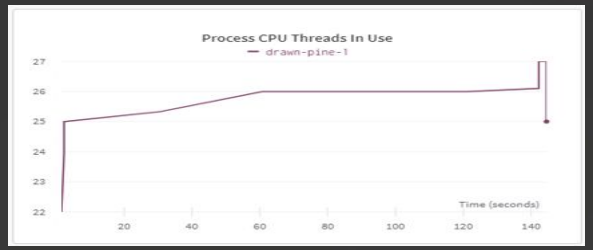
학습이 다 된 모델을 불러오기 때문에 GPU를 사용하지 않음

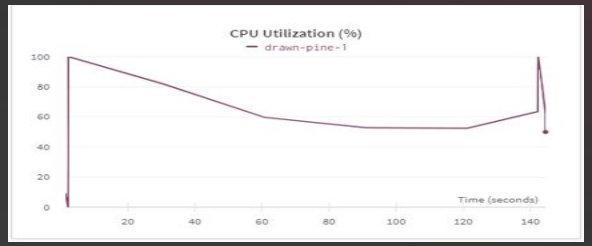
자원 소모량 (예측 실행 시)



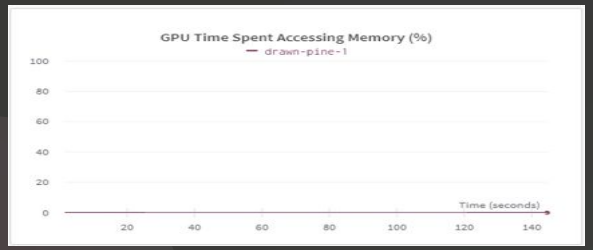


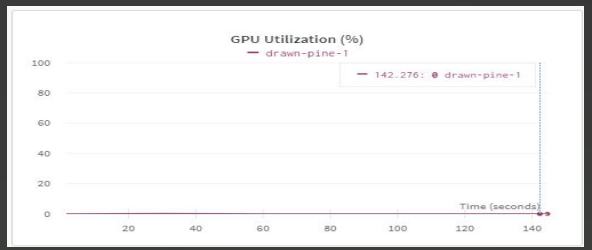
총 소요 시간 : 2분 32초
총 메모리 사용량 : 431 MB
(초기 메모리 1394MB)





최대 CPU thread 사용량 : 27



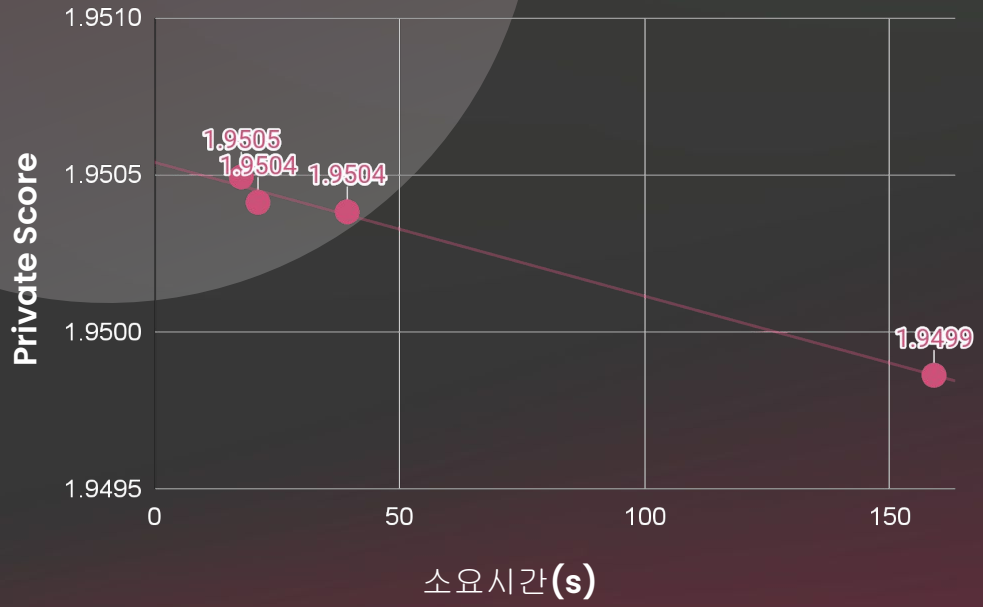


예측에는 GPU가 쓰이지 않음

예측 소요 시간 : 2min 32s

훈련시 소요시간 : 5시간 18분 10초
훈련된 모델 로드시 소요시간 : 37초

속도-정확도 트레이드 오프



Score : 1.9499

17개 모델로 예측 시 평균 2min 39s

Score : 1.9505

14개 모델로 예측 시 평균 17.7s

감사합니다!

김민석 : kmsk96@naver.com
박정기 : jgpark1998@gmail.com
오윤주 : chunkuk1994@gmail.com
장시연 : siyeonjang@gmail.com