

태양광 패널을 사용하는 고객의 에너지 소비량 및 생산량 예측

[최종발표] ESTsoft Wassup 1기 2회차 프로젝트 - 5조



- 팀원 소개
- 주제 및 문제 정의
- 데이터 전처리
- 모델 비교
- 회고 및 앞으로의 방향
- Q&A

목차

팀원 소개



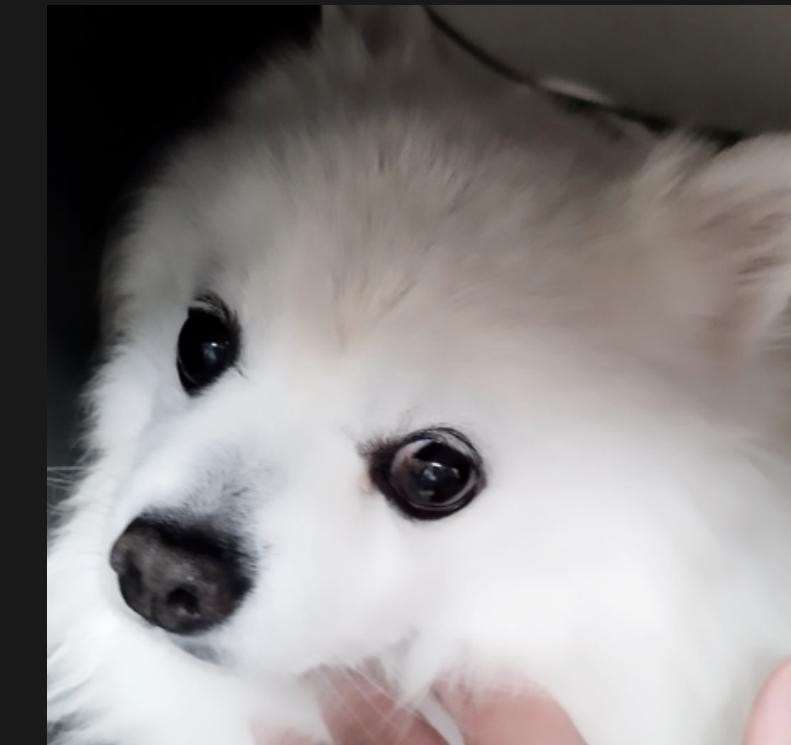
김선들



김종성



안지우

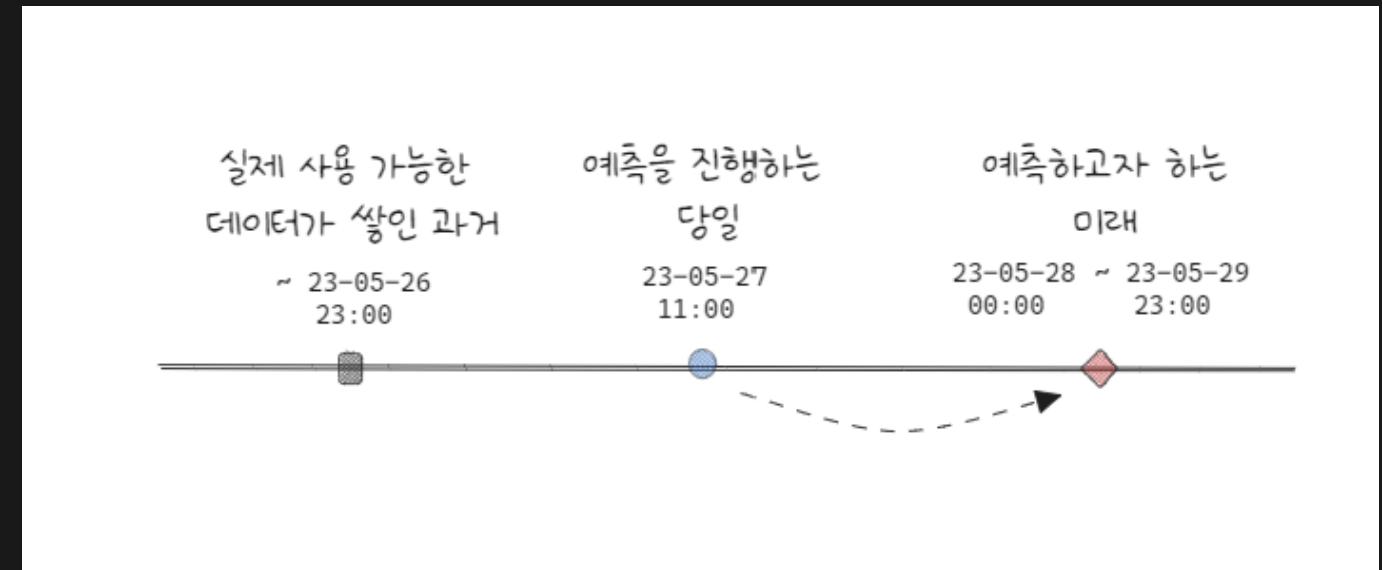


최세준

문제 정의

주제

- 생산량과 소비량의 불균형 해소와 비용 절감을 위해 태양광 패널을 사용하는 고객의 에너지 소비량 및 생산량 예측
 - kaggle competition, by Enefit
- 대회 기준과는 다르게, 하나의 세그먼트(0번)를 기준점으로 잡고 24시간 후와 96시간 후의 소비량 및 생산량 예측으로 문제를 새롭게 정의
- 주최 측에서는 시간대별로 주어진 소비량 및 생산량 데이터 외에, 고객의 지역 정보, 기상 데이터, 전력 가격, 가스 가격 등의 데이터를 추가적으로 제공
 - 주어진 데이터 외에, 에스토니아의 공휴일 데이터를 추가적으로 사용
 - 데이터 병합(merge)은 대회에서 제공한 'data_block_id'를 활용



문제 정의

문제

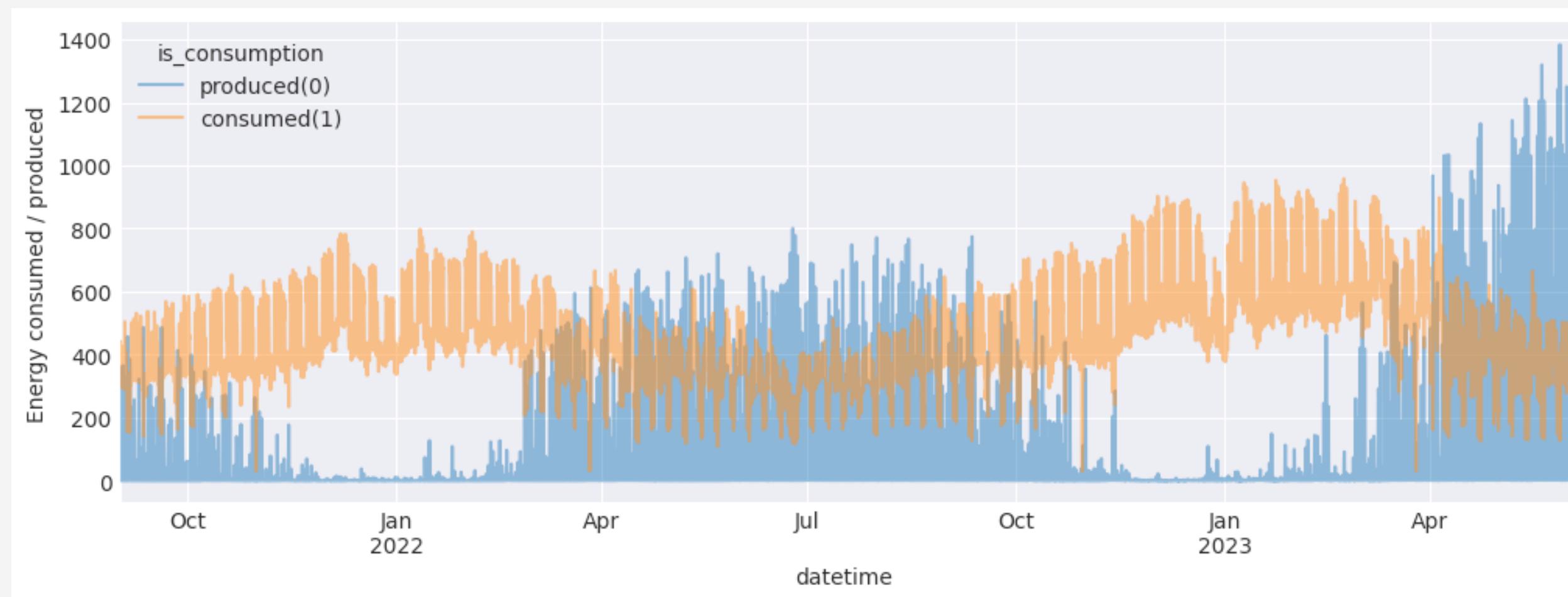
- 목표: 24시간 및 96시간 후의 전력 소비량과 생산량 예측
 - 1) 24시간 후 예측
 - dynamic
 - ARIMA vs. ANN (Pred_size = 1)
 - 통으로(pred_size = 24)
 - ANN vs. PatchTST
 - 2) 96시간 후 예측
 - dynamic
 - ARIMA vs. ANN (Pred_size = 1)
 - 통으로(pred_size = 96)
 - ANN vs. PatchTST

실험 방법

- Consumption/Production 각각 별개의 모델로 예측
- prediction size
 - 1) 24시간 후, 96시간 후를 통으로 예측
 - 2) 1시간 단위로 dynamic하게 예측
- 피쳐
 - 전체 피쳐, 선택 피쳐 (ANN, PatchTST)
 - 타겟 1개 피쳐 (ARIMA, SARIMA)
- 활용 모델: ARIMA, SARIMA, ANN, PatchTST
- 평가 지표: MAE, MAPE, R2 score

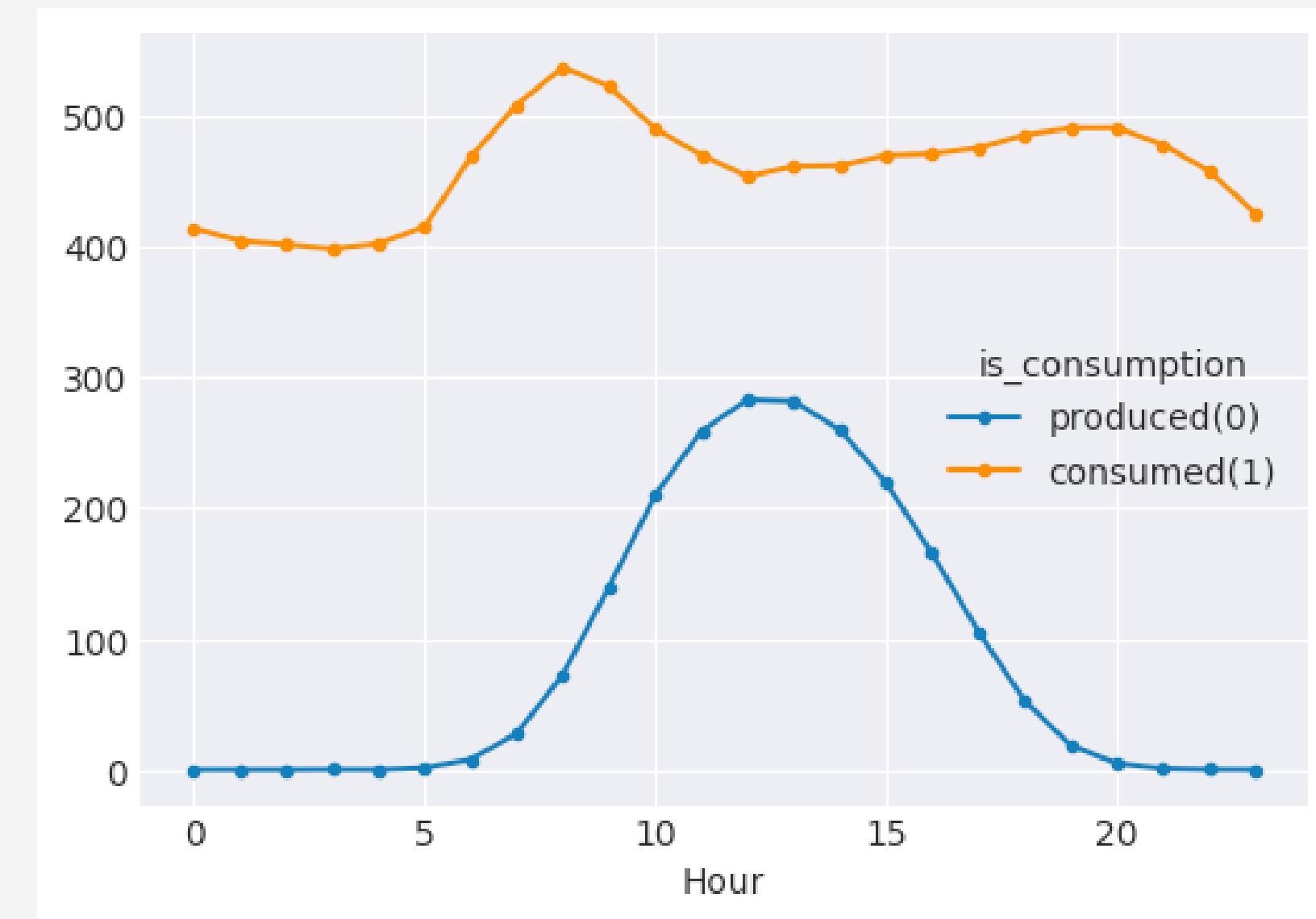
데이터 전처리 - 데이터 살펴보기

| county | is_business | product_type | prediction_unit_id |
|--------|-------------|--------------|--------------------|
| 0 | 0 | 1 | 0 |



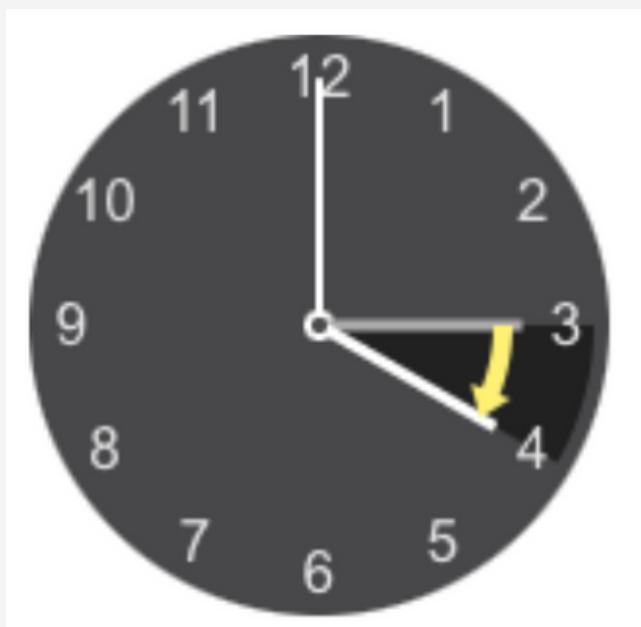
전체 기간에 대한 target plot

데이터 전처리 - 데이터 살펴보기



시간대별 target 평균

데이터 전처리 - 결측값



| | |
|----------------|---------------------|
| target | NaN |
| is_consumption | 1 |
| datetime | 2022-03-27 03:00:00 |



| | |
|----------------|---------------------|
| target | NaN |
| is_consumption | 1 |
| datetime | 2021-10-31 03:00:00 |

| | |
|----------------|---------------------|
| target | NaN |
| is_consumption | 1 |
| datetime | 2023-03-26 03:00:00 |

| | |
|----------------|---------------------|
| target | NaN |
| is_consumption | 1 |
| datetime | 2022-10-30 03:00:00 |

특이점 - 써머타임

데이터 전처리 - 결측값 처리

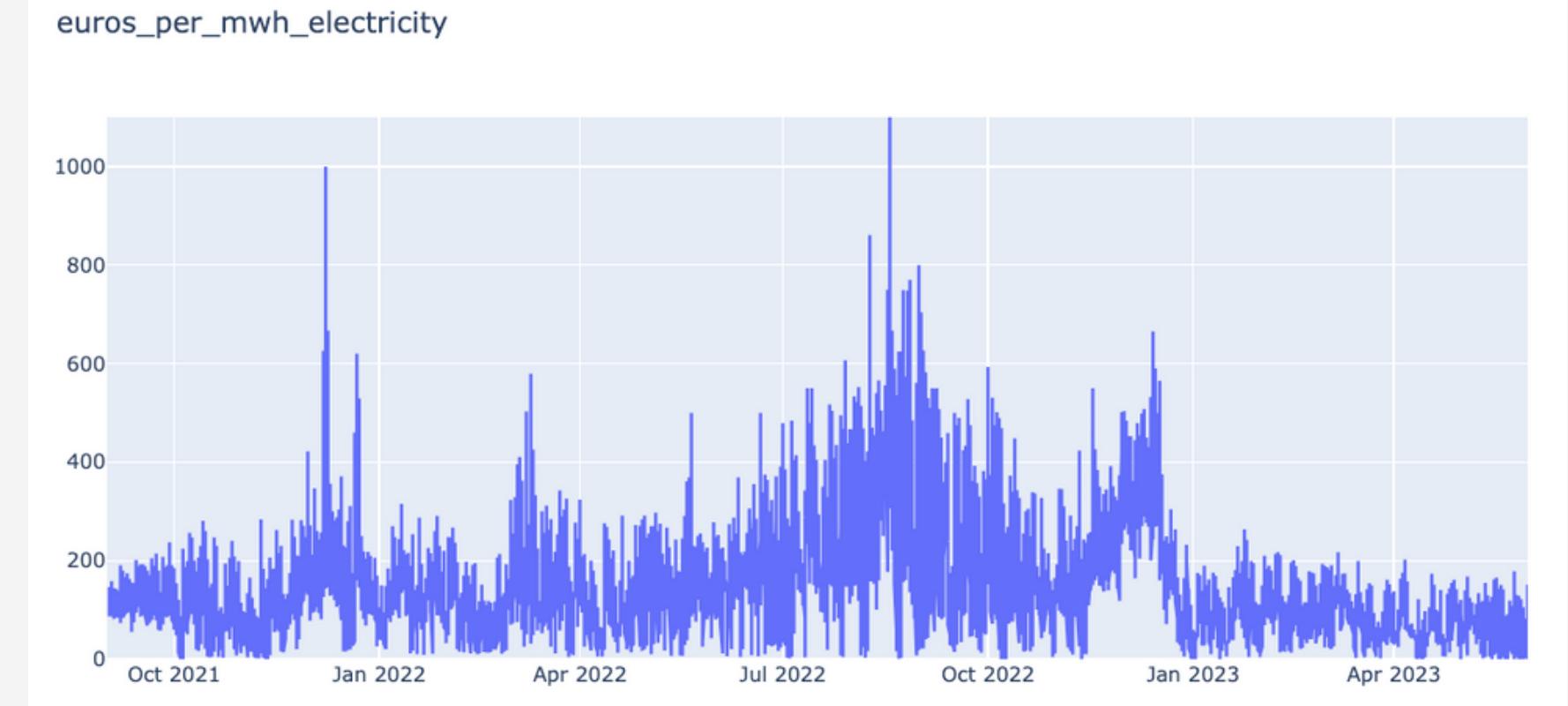
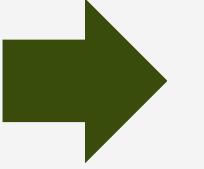
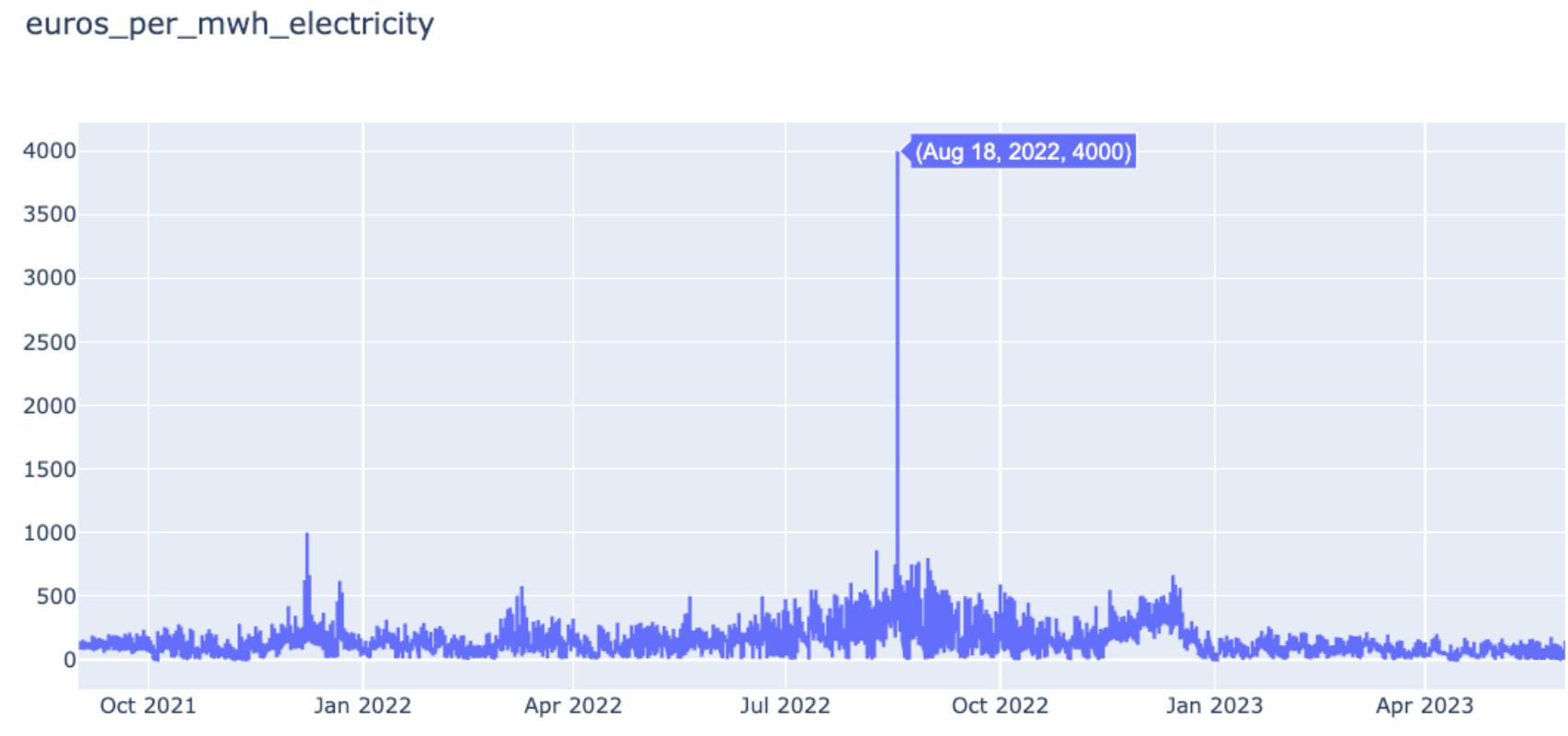
처리방법1 - drop

- 해당 결측치가 있는 행을 모두 제거
- 처음 2일 공통적으로 삭제 -> 30528행
 - 추가데이터 중, 예측 데이터를 포함하는 column에서는 21-09-01 ~ 21-09-02의 데이터가 모두 결측값으로 존재
 - 따라서, 결측값을 무리하게 채우기보다는 데이터 양이 이미 충분하기 때문에 첫 2일의 데이터를 drop하는 것으로 결정

처리방법2 - mean

- 결측값 기준 전일 동시간과, 후일 동시간의 평균으로 대체
 - 써머타임

데이터 전처리 - 이상치



electricity price에서 이상치 의심

데이터 전처리 - 이상치

Record high €4,000 on Wednesday

Electricity prices in Nord Pool's Estonian price area reached record highs on Wednesday, with the average price for August 17 reaching €682.05 per megawatt-hour and the hourly price to peak at €4,000 per megawatt-hour from 5-6 p.m. (6-7 p.m. Estonian time), which is also the power exchange's cap.

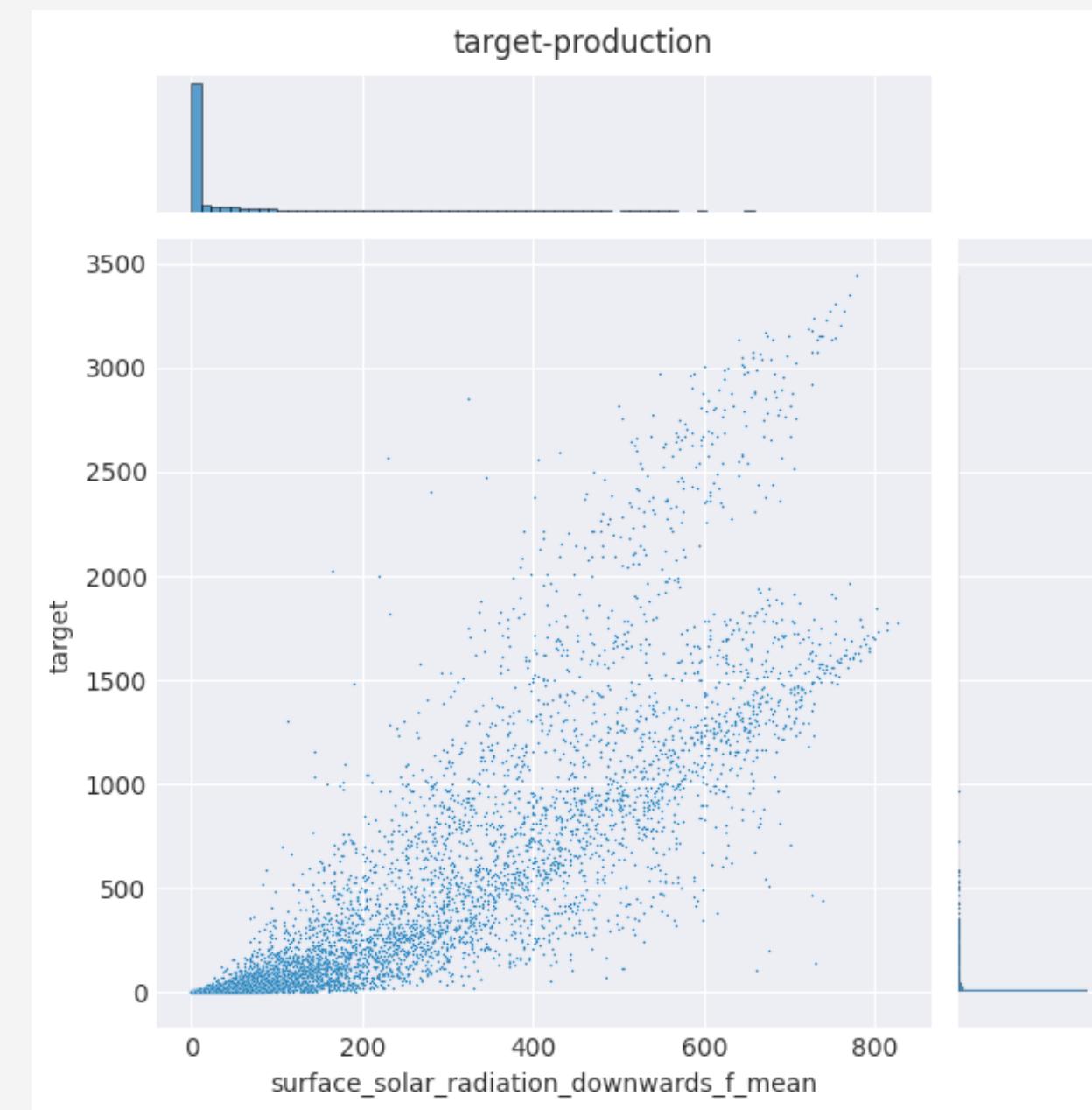
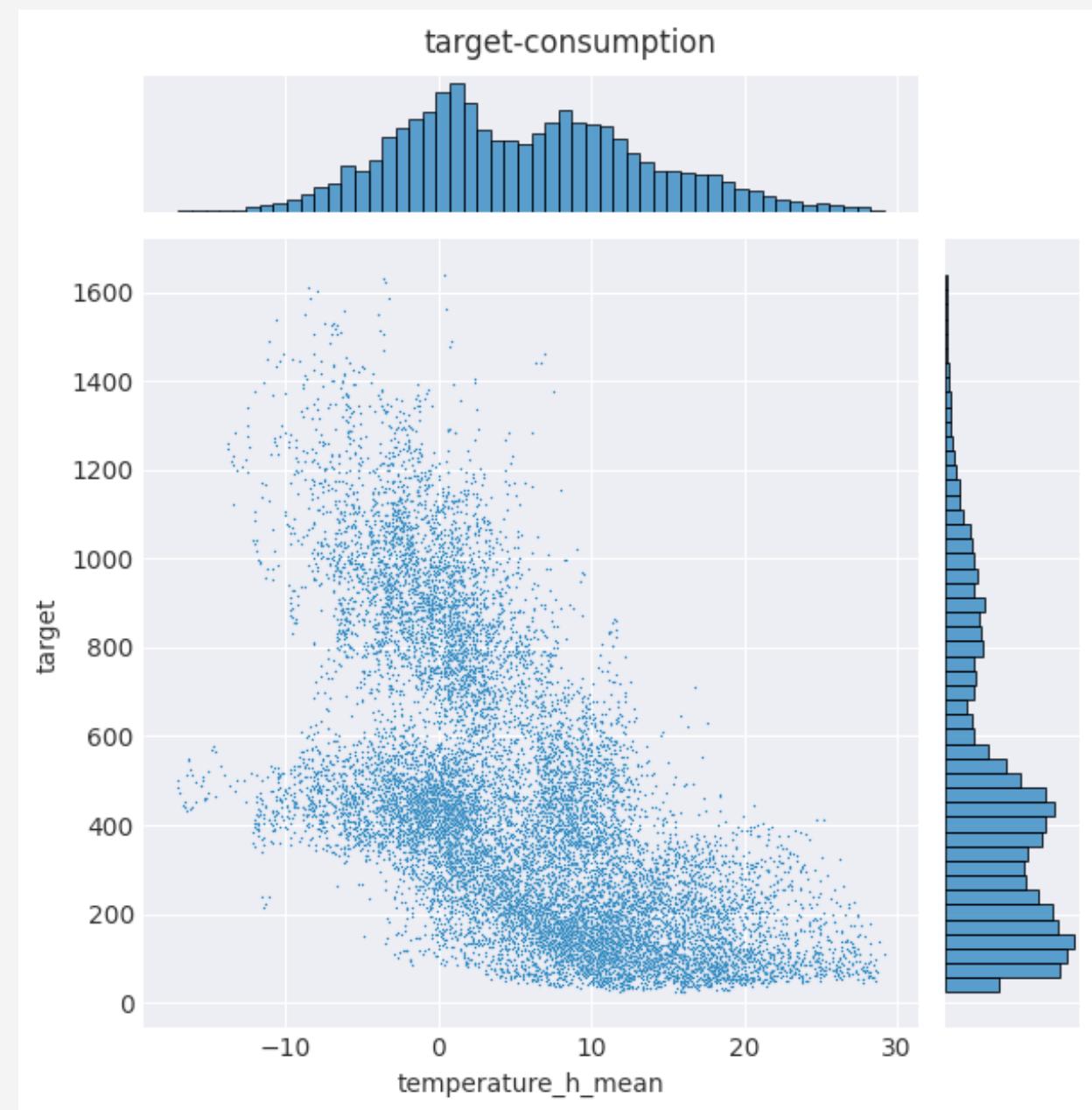
Latvia and Lithuania will both likewise see their electricity prices peak at €4,000 at the same time on Wednesday, however both countries' average daily prices are exceeding Estonia's at €823.98.

Estonia's previous record high average electricity price was set last Monday, when the daily average price reached €501 per megawatt-hour.

The previous hourly electricity price record of €1,000.07 per megawatt-hour was set on December 7, 2021.

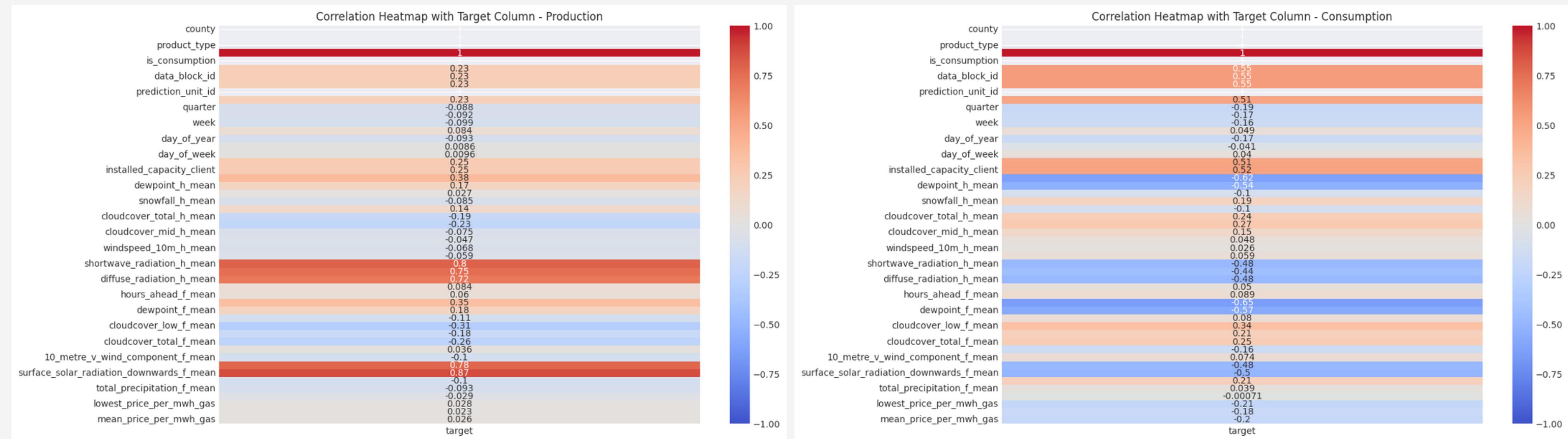
**All times are noted in Central European Summer Time (CEST), one hour behind Estonian time, in accordance with Nord Pool's own data.*

데이터 전처리 - 상관관계

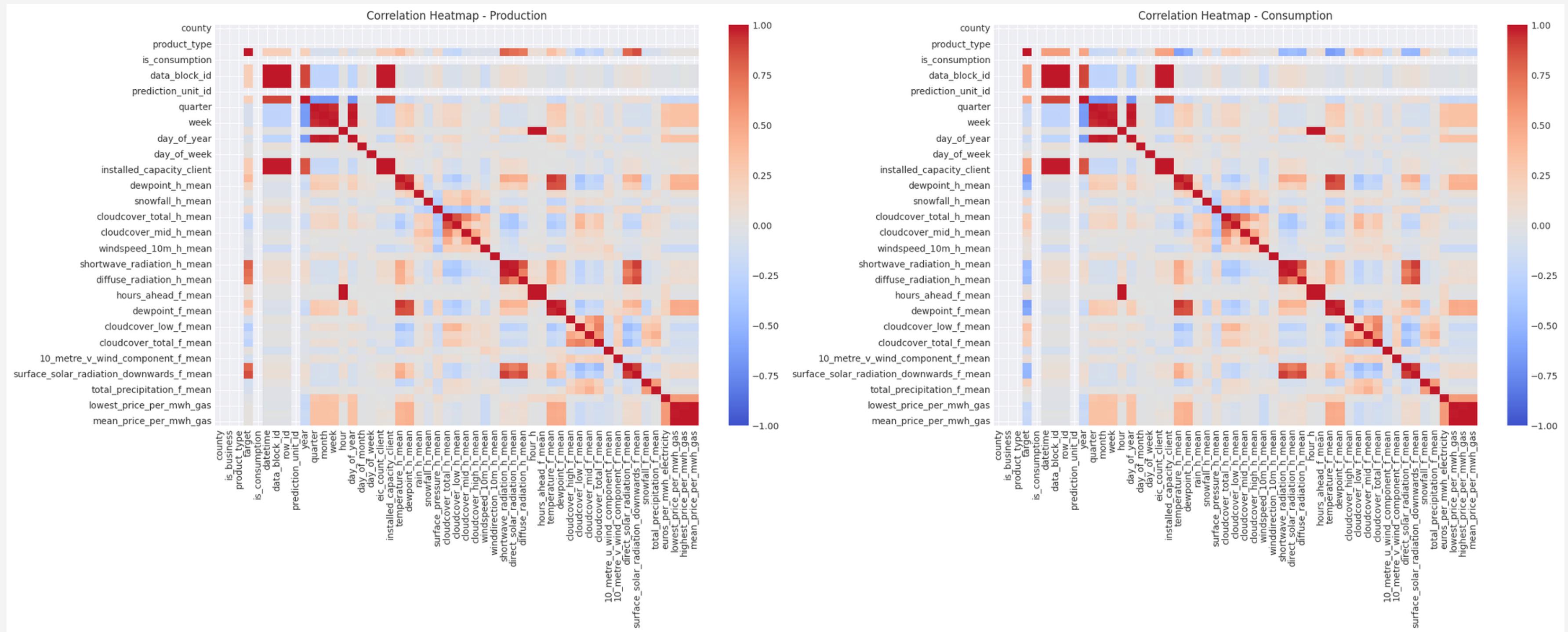


jointplot

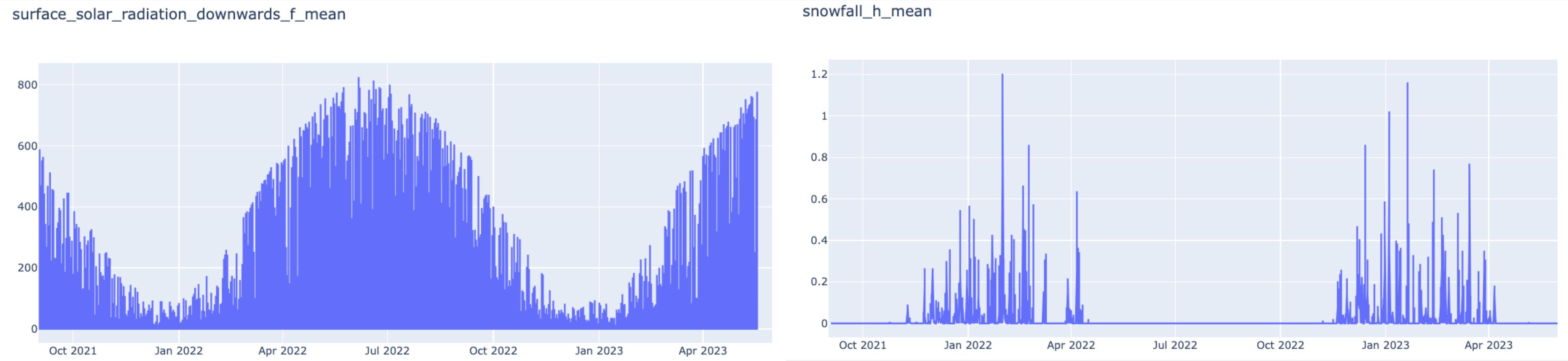
데이터 전처리 - 상관관계



데이터 전처리 - 상관관계

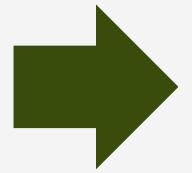
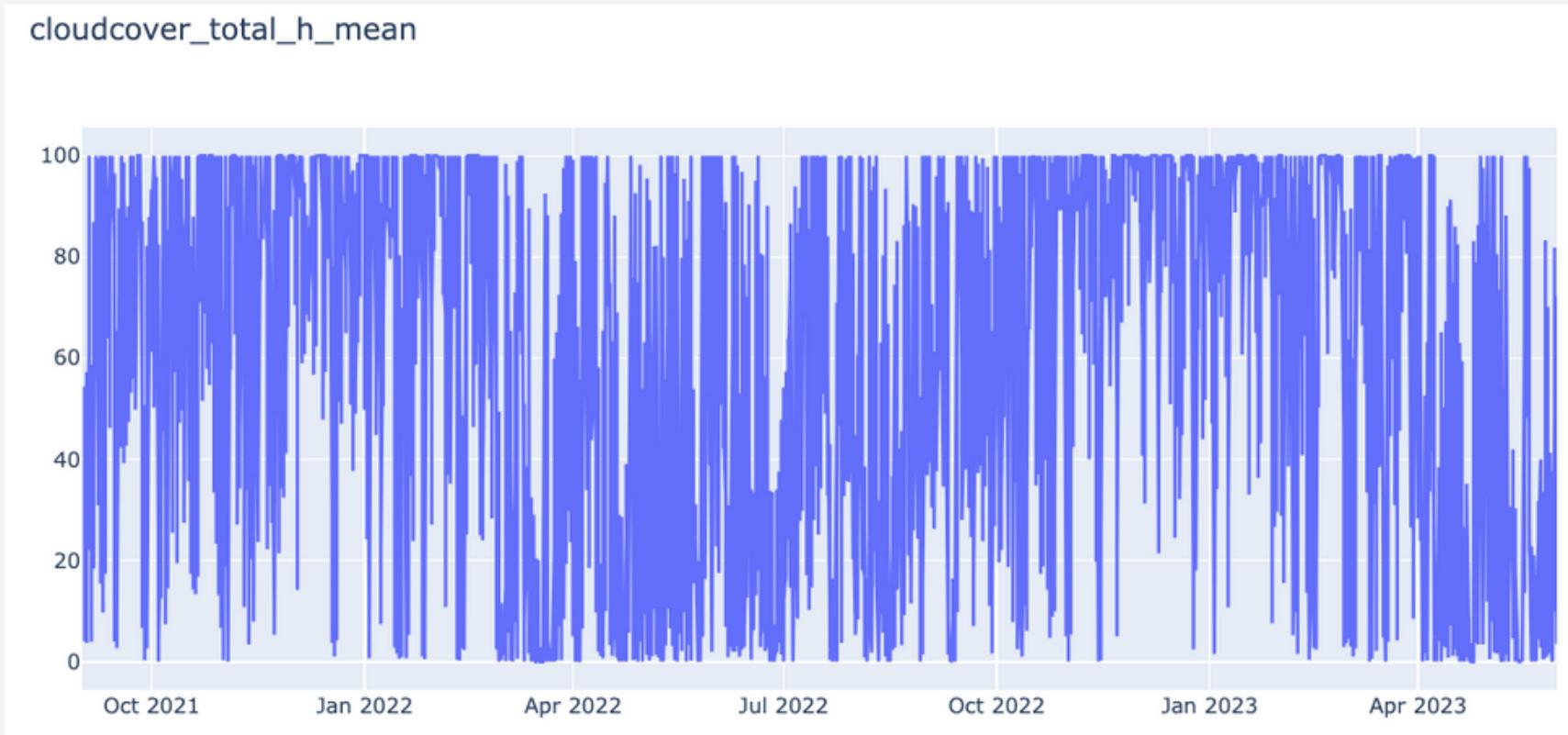


데이터 전처리 - 피쳐 선택



피쳐별 패턴, 특성 확인

데이터 전처리 - 피쳐 선택



일별로 평균을 내서 확인

데이터 전처리 - 피쳐 선택

Full 모델(44개+target: 전체 피쳐)

- "year", "quarter", "month", "week", "day", "hour", "day_of_year", "day_of_month", "day_of_week", "eic_count_client", "installed_capacity_client", "temperature_h_mean", "dewpoint_h_mean", "rain_h_mean", "snowfall_h_mean", "surface_pressure_h_mean", "cloudcover_total_h_mean", "cloudcover_low_h_mean", "cloudcover_mid_h_mean", "cloudcover_high_h_mean", "windspeed_10m_h_mean", "winddirection_10m_h_mean", "shortwave_radiation_h_mean", "direct_solar_radiation_h_mean", "diffuse_radiation_h_mean", "hour_h", "hours_ahead_f_mean", "temperature_f_mean", "dewpoint_f_mean", "cloudcover_high_f_mean", "cloudcover_low_f_mean", "cloudcover_mid_f_mean", "cloudcover_total_f_mean", "10_metre_u_wind_component_f_mean", "10_metre_v_wind_component_f_mean", "direct_solar_radiation_f_mean", "surface_solar_radiation_downwards_f_mean", "snowfall_f_mean", "total_precipitation_f_mean", "euros_per_mwh_electricity", "lowest_price_per_mwh_gas", "highest_price_per_mwh_gas", "mean_price_per_mwh_gas", "country_holiday"

축소 모델(8개+target: 선택 피쳐)

- 총 고객 수: eic_count_client
- 평균 기온: temperature_h_mean
- 평균 이슬점: dewpoint_h_mean
- 평균 적설량: snowfall_h_mean
- 평균 구름량: cloudcover_total_h_mean
- 평균 단파복사량: shortwave_radiation_h_mean
- 평균 직달 일조량: direct_solar_radiation_h_mean
- 평균 태양복사량:
surface_solar_radiation_downwards_f_mean
 - 평균: 기상 관측소가 한 county에 여러 개 존재하는 경우

ARIMA / SARIMA

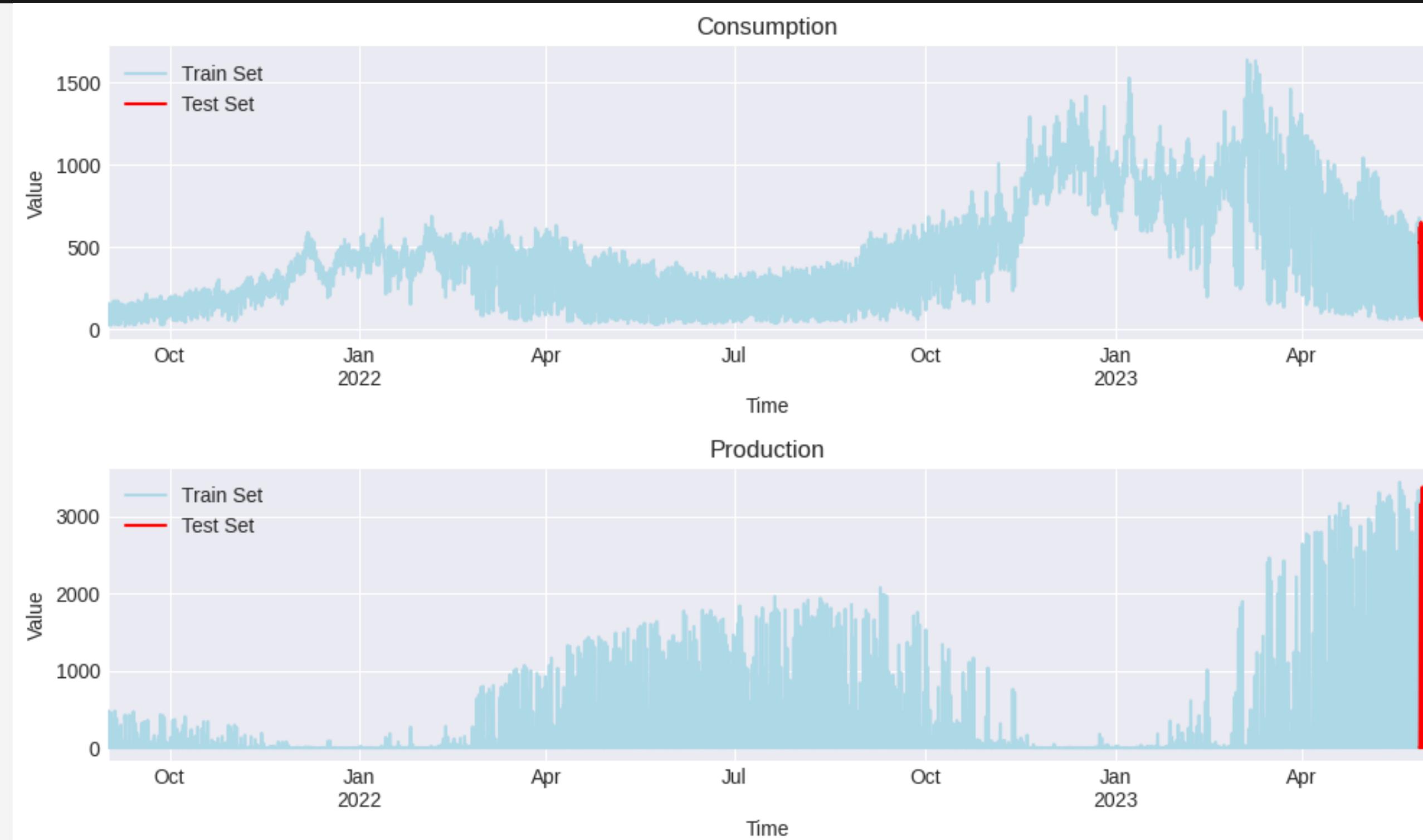
실험 준비

- 데아터 전처리
 - 연속적인 시계열을 갖도록 결측값 처리
- Stationary (정상성) 확인
 - ACF, PACF 그래프 및 Adfuller Test ($p\text{-value} < 0.05$)
 - Consumption -> 1차 차분 후 정상성 확인됨
 - Production -> 정상성 확인됨
- Base parameters 선정
 - ACF, PACF 그래프를 살펴보고 초기 p, q 값을 선정함.
 - auto_arima
 - 데이터가 커서, 커널이 죽거나 신뢰하기 어려운 값을 배출함

실험 과정

- Consumption, Production 분리
- test_size = 24, 96
 - 1 step(1시간) 씩 Dynamic하게 예측해서 24시간 후와 96시간 후를 예측하는 실험 진행
- Scaler: None/minmax
- Parameters 최적화
 - Base parameters를 기준점으로 삼고 조금씩 변경해 서 실험 진행
- 가설
 - 데이터의 추세를 보면, consumption은 이전 날의 값, production은 상대적으로 계절의 영향을 크게 받는 것으로 보임.

ARIMA / SARIMA

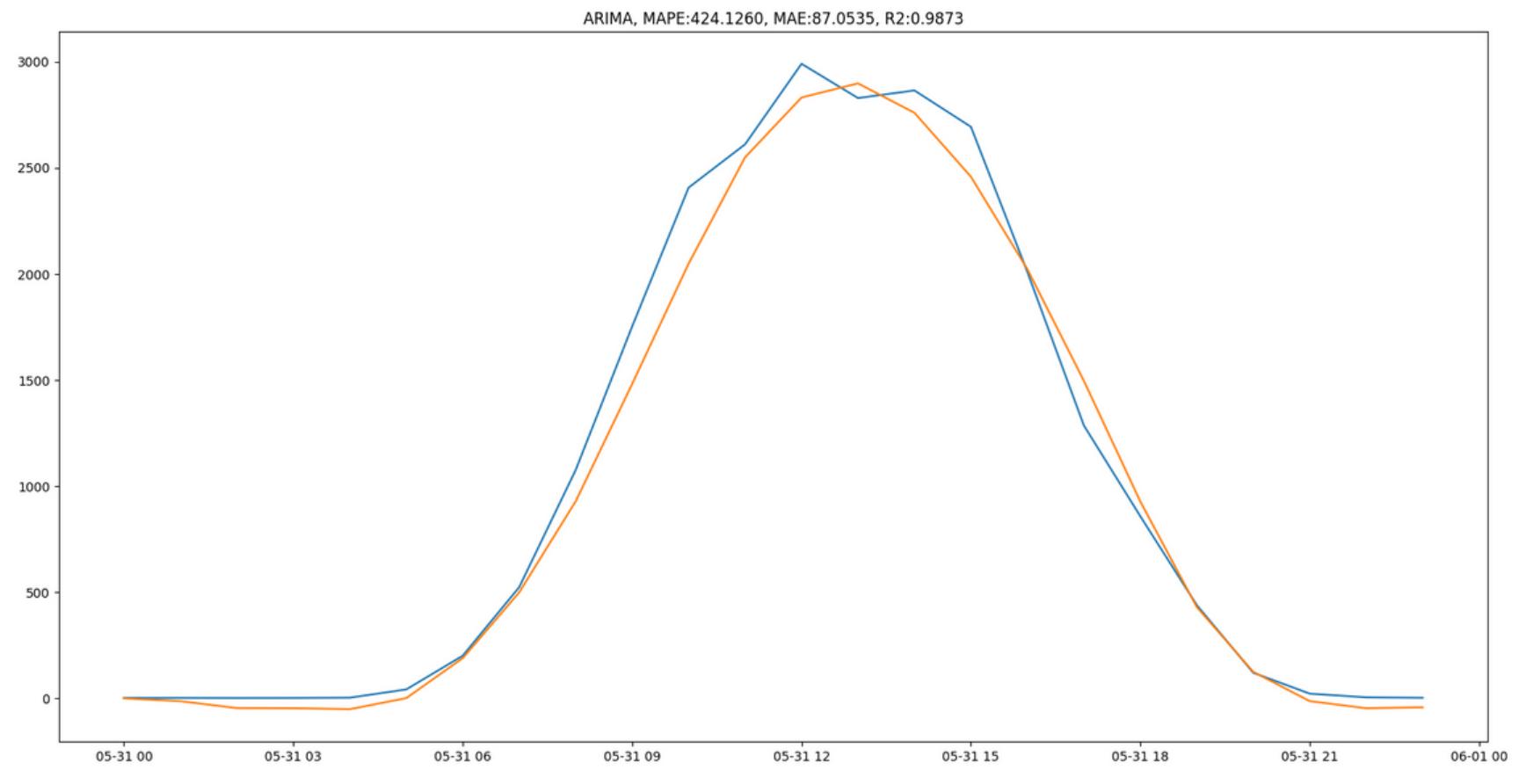
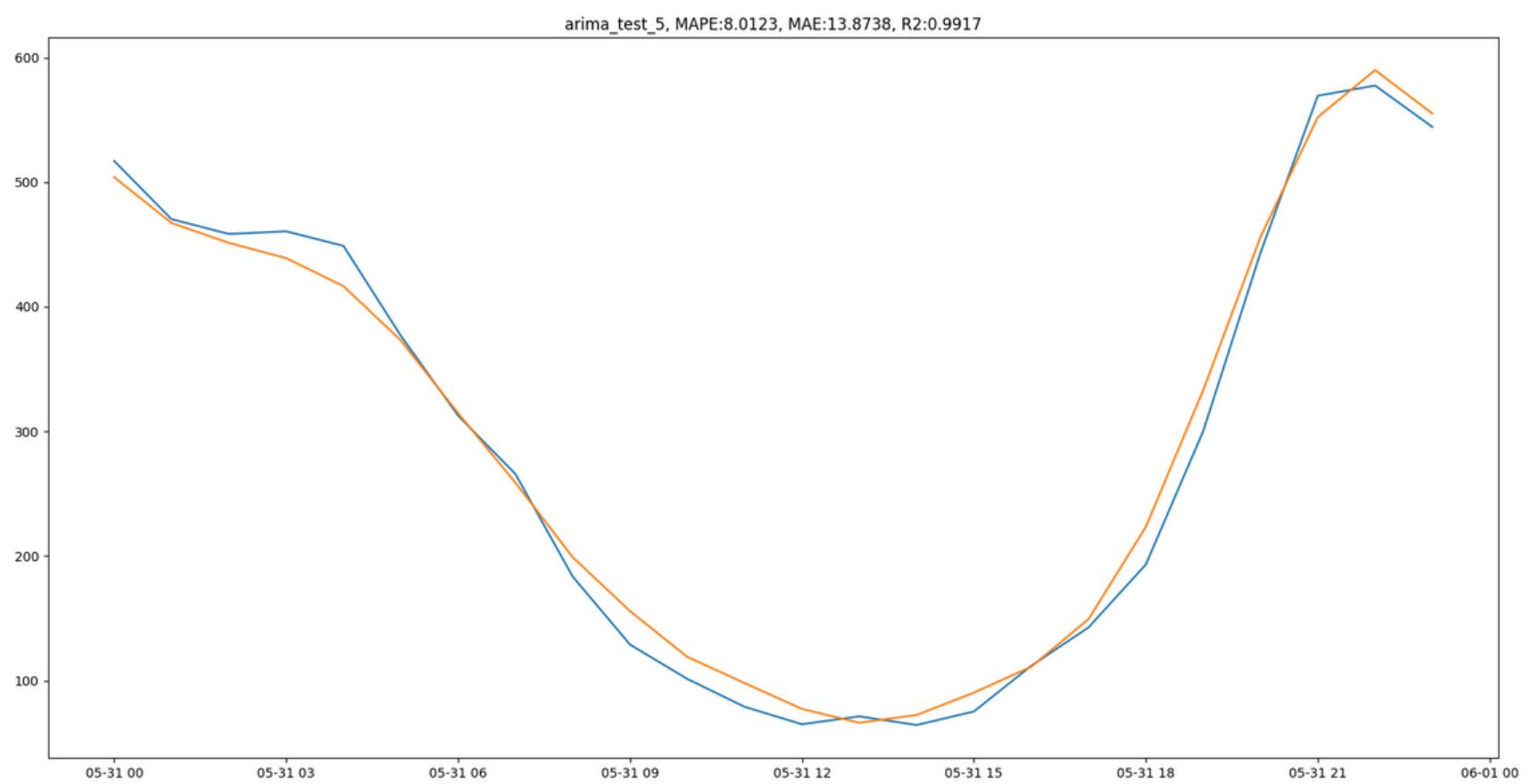


ARIMA / SARIMA

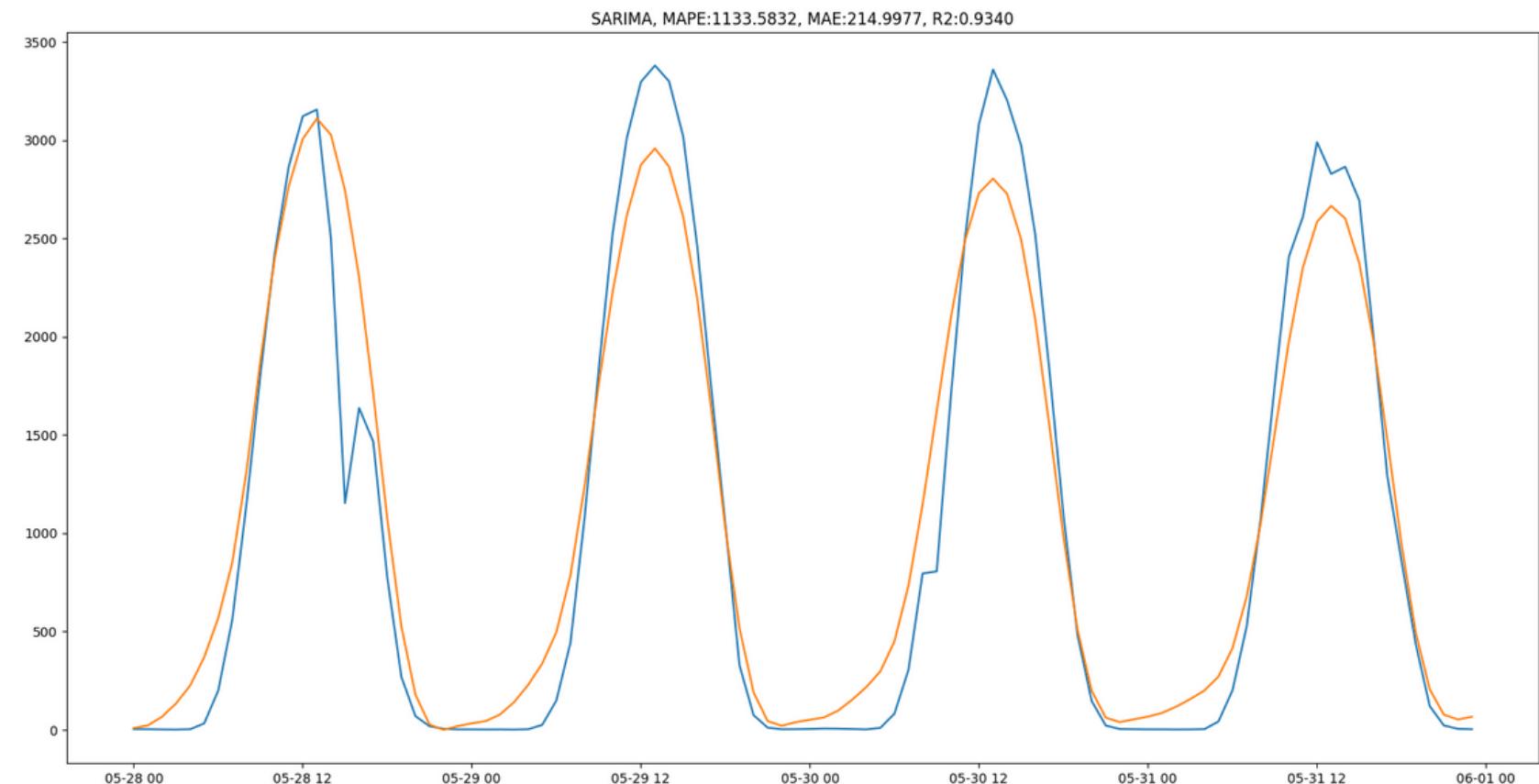
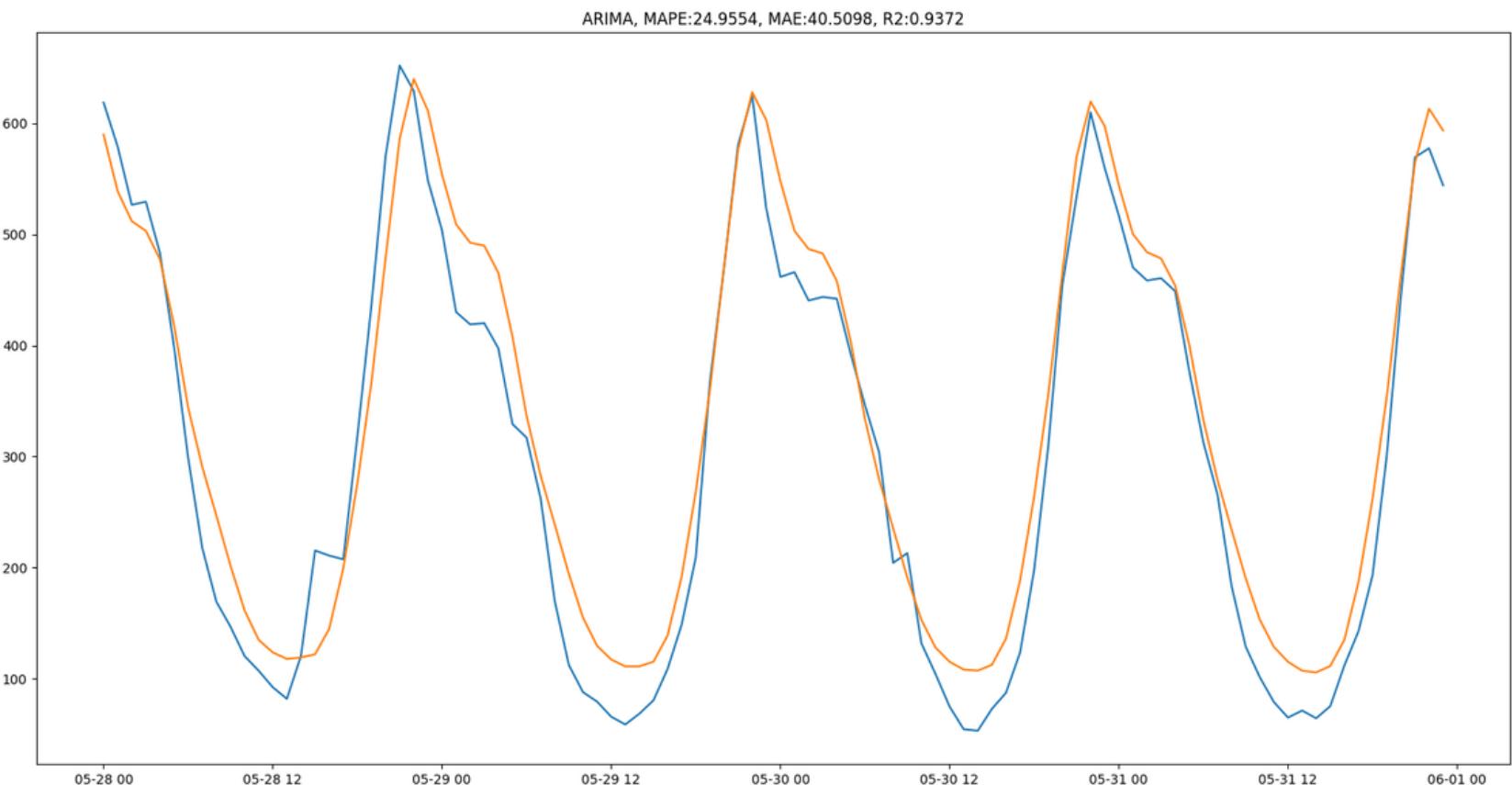
실험 결과

- 24시간 후 예측(test_size = 24)
 - 소비량(consumption)
 - Best: MAE = **13.9550**
 - ARIMA (24,1,24) No scaler
 - Best: MAE = **87.0535**
 - ARIMA (31,0,31) No scaler
- 96시간 후 예측 (test_size = 96)
 - 소비량(consumption)
 - Best: MAE = **40.5098**
 - ARIMA (24,1,30) No scaler
 - 생산량(production)
 - Best: MAE = **214.9977**
 - SARIMA (24,0,24), (0,0,0,24) No scaler

ARIMA / SARIMA - 24시간 Best



ARIMA / SARIMA – 96시간 Best



ANN

실험 준비

- 데이터 전처리
 - 연속적인 시계열을 갖도록 결측값 처리
 - 사용 가능한 데이터 Merge 작업
 - prediction date 의 이틀 전 시간으로 맞춰서 병합
 - Merge 작업 후 결측값 처리
 - 24시간 전과 후의 평균값
- 실험 데이터
 - 전체 피쳐 - 45개 컬럼
 - 데이터 전처리 후 중복되는 날짜 데이터를 제외한 모든 데이터 컬럼
 - 추가 데이터 - 공휴일 피쳐만 포함/제외한 경우
 - 선택 피쳐 - 9개 컬럼
 - 데이터 전처리 후 target 과의 관계를 분석하여 선택된 데이터 컬럼

실험 과정

- Consumption, Production 분리
- test_size = 1, 24, 96
 - 1 step(1시간) 씩 Dynamic하게 예측해서 24시간 후와 96시간 후를 예측하는 실험 진행
 - prediction size를 24, 96으로 통으로 예측하는 실험도 진행
- Scaler: None/minmax
- Parameters 최적화
 - learning rate를 시작으로 5개씩 나눠서 파라미터를 수정
 - 경향성을 예상하여 최적의 파라미터를 유도

ANN

실험 진행 시 Parameters 선정 과정

- Window size = 168
 - window_size를 늘리고 pred_size를 줄일 수록 정확해지는 경향을 확인
 - pred size = 1일때, window size는 168일때 가장 성능이 좋았고 그 이상으로는 다시 낮아졌음
- Learning rate = 0.00001 (e^4)
 - Learning rate 0.00001과 0.00005는 별 차이 없었지만 lr이 더 커질수록 MAE 증가
- Epochs = 500
 - Early stop 기준 이하로 epochs를 낮추는 것보다는, epochs = 500일때 성능이 안정적임.
- 전체 피쳐 vs. 선택 피쳐
 - 신경망 모델의 특성 상, 피쳐가 많을 수록 좋은 성능을 보여줄 것으로 가정했고, 실험 결과에서도 선택 피쳐만 활용하는 것보다는 전체 피쳐를 활용했을 때에 성능이 좋았음.
 - Consumption 과 Production 모두 마찬가지

ANN – Best

Dynamic

소비량(consumption)

- test size = 24, pred size = 1
 - MAE : 12.6166 (전체 피쳐)
- test size = 24, pred size = 1
 - MAE : 15.0453 (선택 피쳐)

생산량(production)

- test size = 24, pred size = 1
 - MAE : 56.1514 (전체 피쳐)
- test size = 24, pred size = 1
 - MAE : 69.1928 (선택 피쳐)

통으로 예측

소비량(consumption)

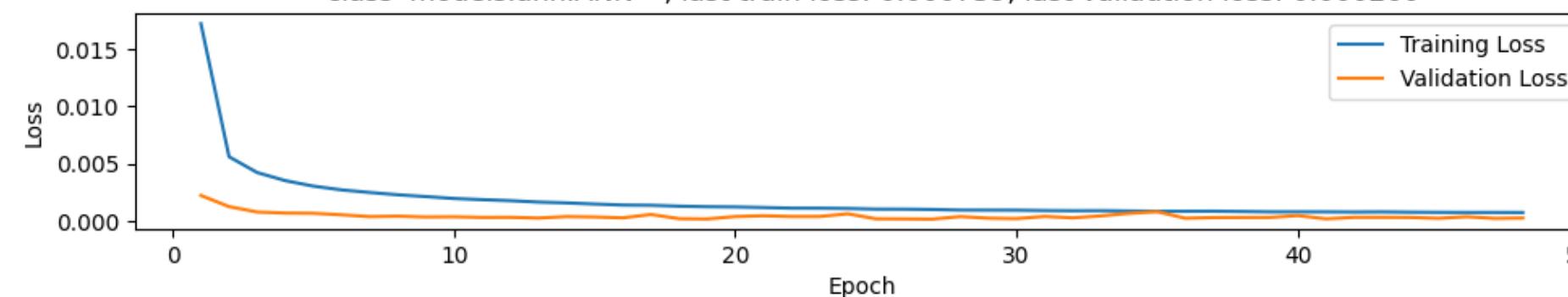
- test size = 96, pred size = 96
 - MAE: 38.7249 (선택 피쳐)
- test size = 24, pred size = 24
 - MAE: 14.4762 (선택 피쳐)

생산량(production)

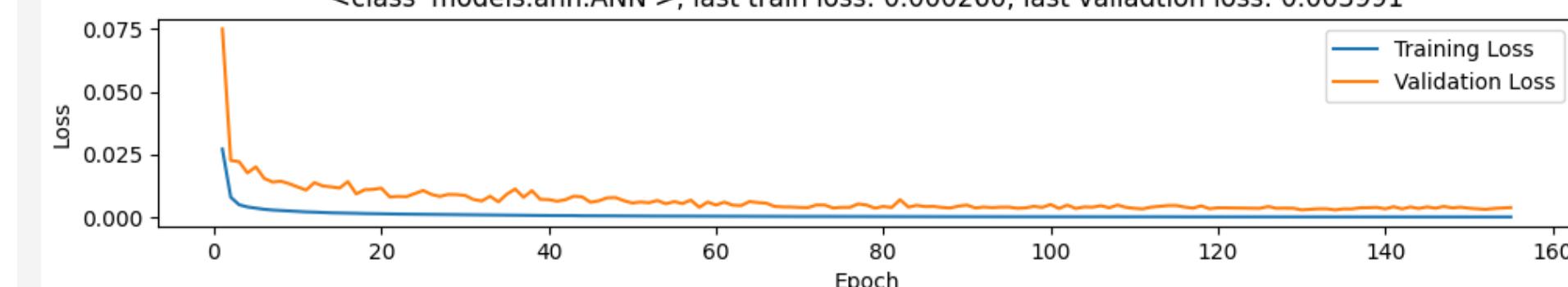
- test size = 96, pred size = 96
 - MAE: 205.1572 (전체 피쳐)
- test size = 24, pred size = 24
 - MAE: 59.2724 (전체 피쳐)

ANN - 24시간 Best

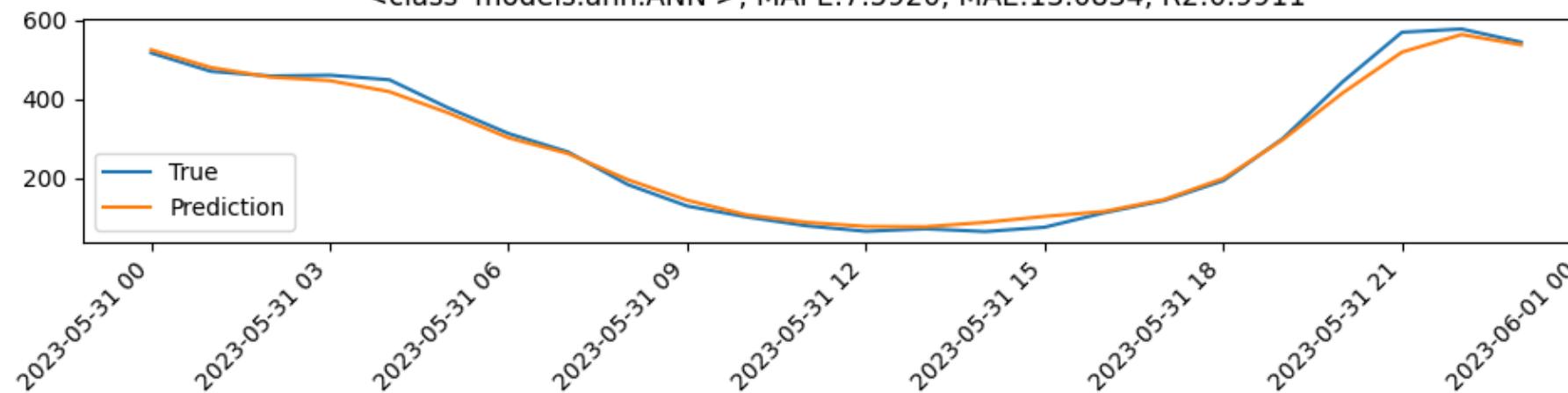
<class 'models.ann.ANN'>, last train loss: 0.000739, last validation loss: 0.000260



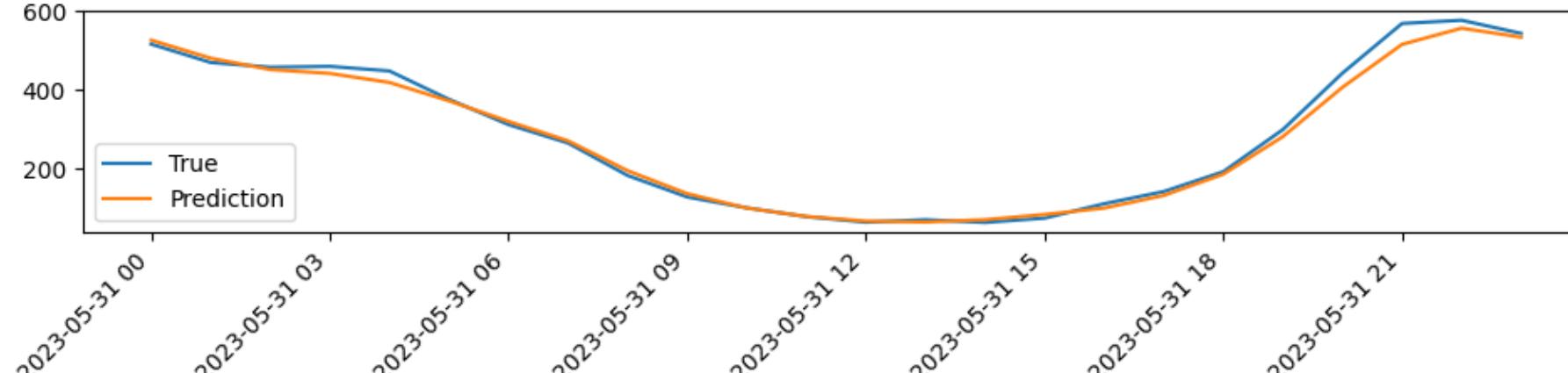
<class 'models.ann.ANN'>, last train loss: 0.000260, last validation loss: 0.003991



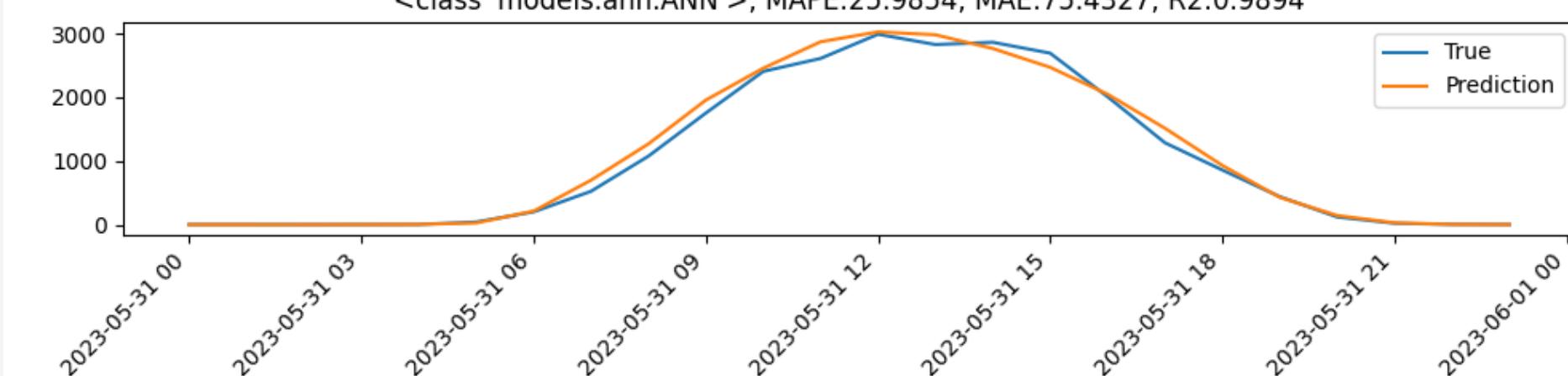
<class 'models.ann.ANN'>, MAPE:7.5920, MAE:13.0834, R2:0.9911



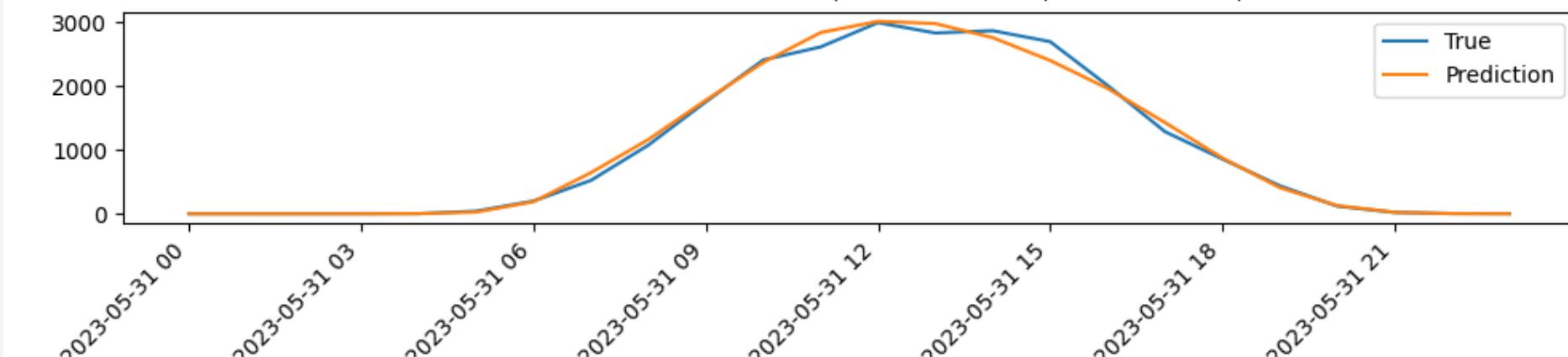
Best model: <class 'models.ann.ANN'>, MAPE:5.0451, MAE:12.6166, R2:0.9911



<class 'models.ann.ANN'>, MAPE:25.9854, MAE:75.4327, R2:0.9894

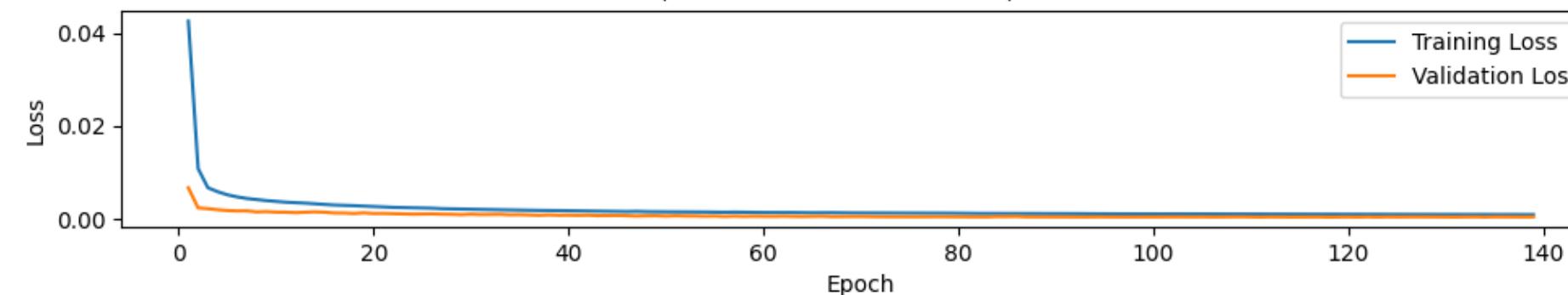


Best model: <class 'models.ann.ANN'>, MAPE:22.4277, MAE:56.1514, R2:0.9929

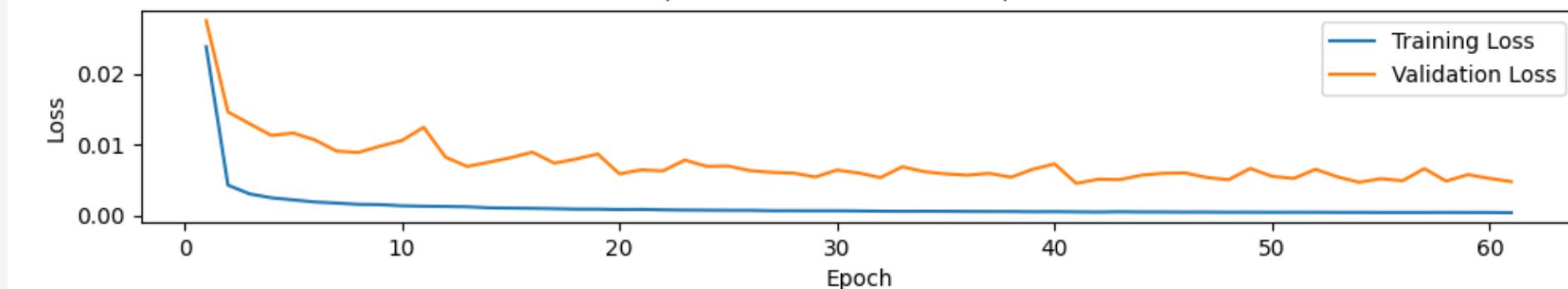


ANN - 96시간 Best

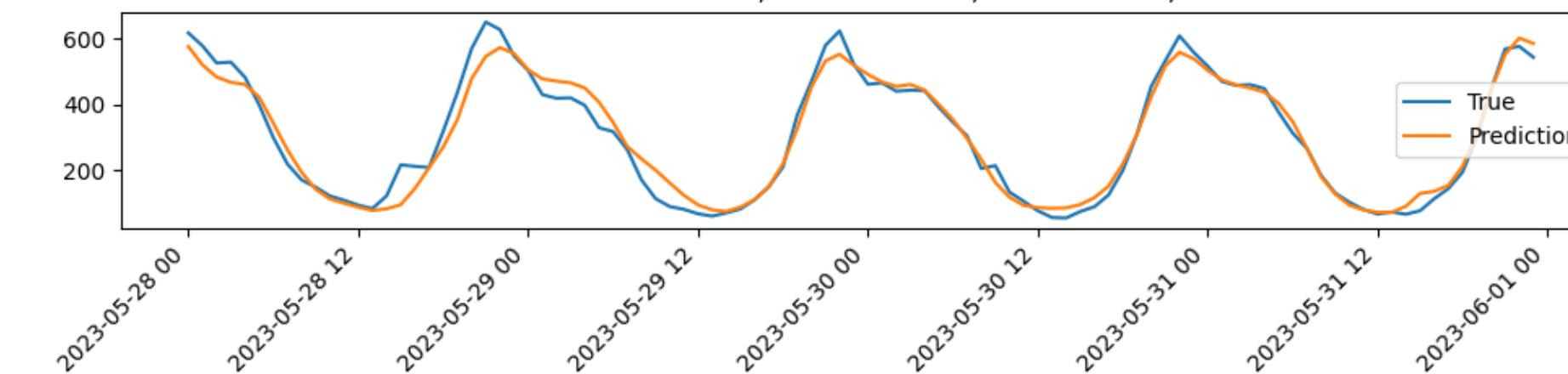
<class 'models.ann.ANN'>, last train loss: 0.000921, last validation loss: 0.000451



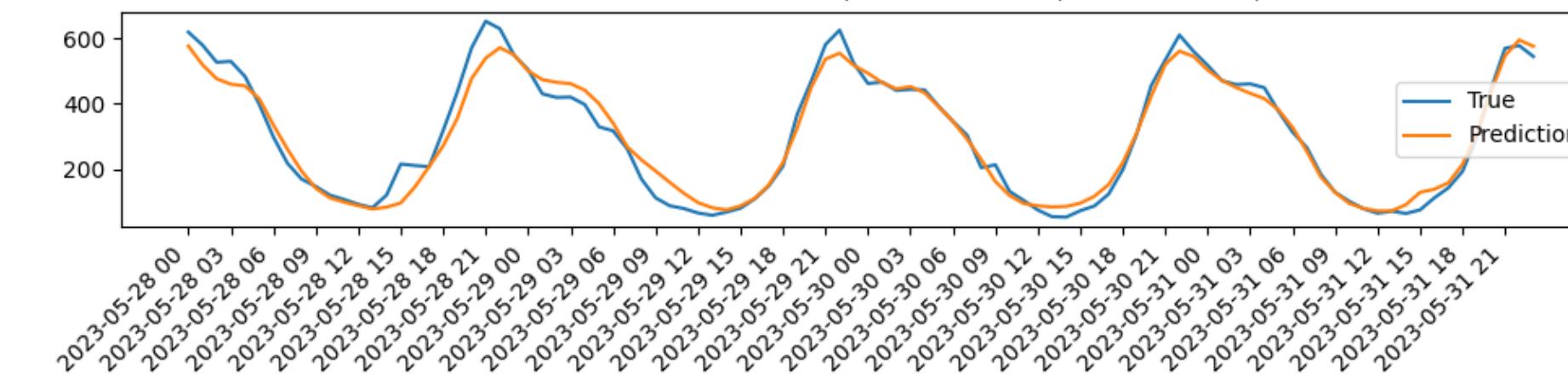
<class 'models.ann.ANN'>, last train loss: 0.000435, last validation loss: 0.004799



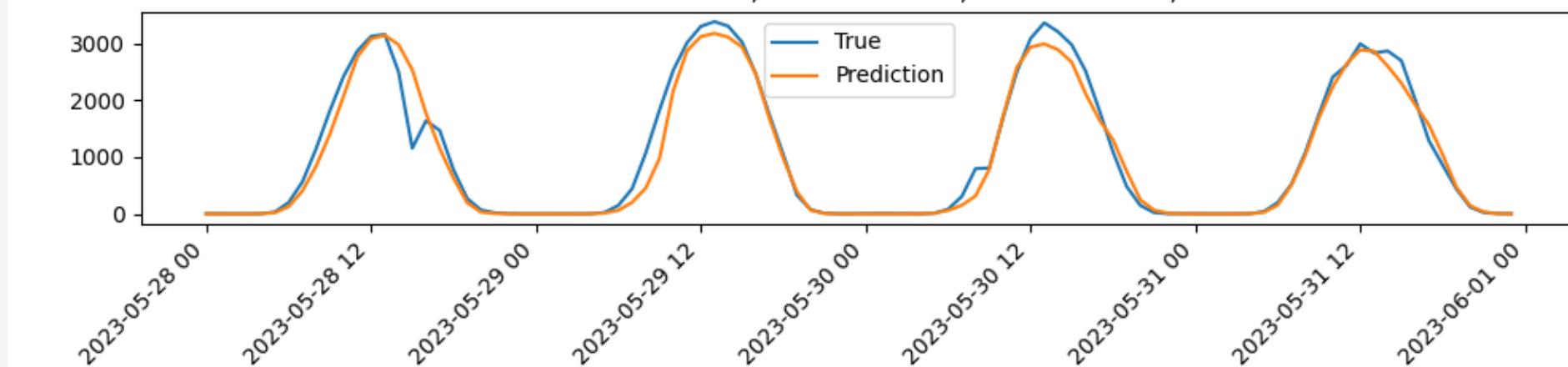
<class 'models.ann.ANN'>, MAPE:13.8413, MAE:27.9459, R2:0.9582



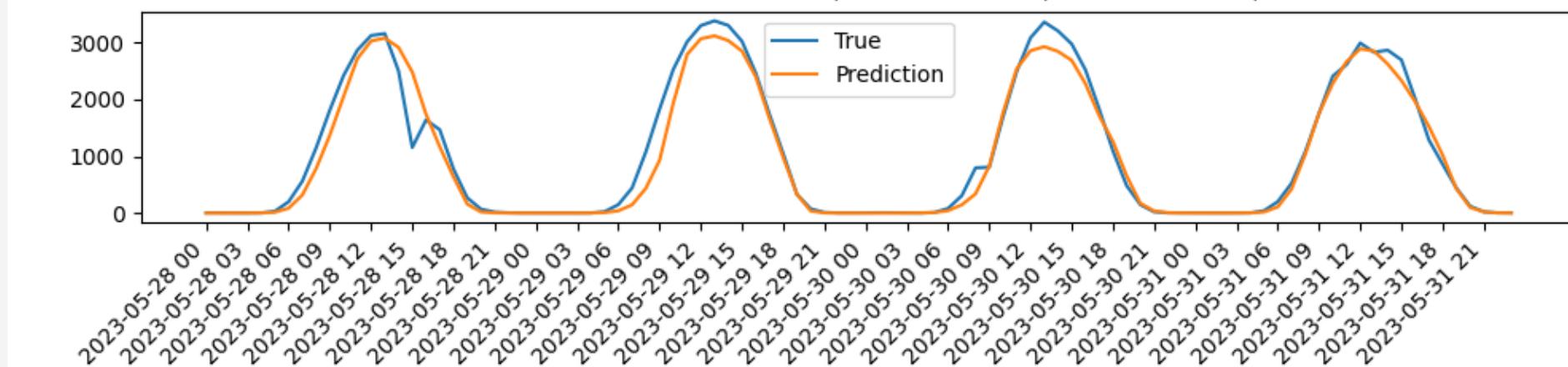
Best model: <class 'models.ann.ANN'>, MAPE:13.9917, MAE:27.5893, R2:0.9591



<class 'models.ann.ANN'>, MAPE:33.7109, MAE:127.1683, R2:0.9598



Best model: <class 'models.ann.ANN'>, MAPE:34.3261, MAE:133.7886, R2:0.9579



PatchTST

실험 준비

- 데이터 전처리
 - 연속적인 시계열을 갖도록 결측값 처리
 - 사용 가능한 데이터 Merge 작업
 - prediction date 의 이틀 전 시간으로 맞춰서 병합
 - Merge 작업 후 결측값 처리
 - 24시간 전과 후의 평균값
- 실험 데이터
 - 전체 피쳐 - 45개 컬럼
 - 데이터 전처리 후 중복되는 날짜 데이터를 제외한 모든 데이터 컬럼 (공휴일 포함)
 - 선택 피쳐 - 9개 컬럼
 - 데이터 전처리 후 target 과의 관계를 분석하여 선택된 데이터 컬럼

실험 과정

- Consumption, Production 분리
- test_size = 24, 96
 - 24시간 후와 96시간 후를 통으로 예측하는 실험 진행
- Scaler: None/minmax
- Parameters 최적화
 - Patch TST 파라미터 실험 및 고정 (논문 참고)
 - Learning rate 를 다양하게 하여 실험 진행

PatchTST

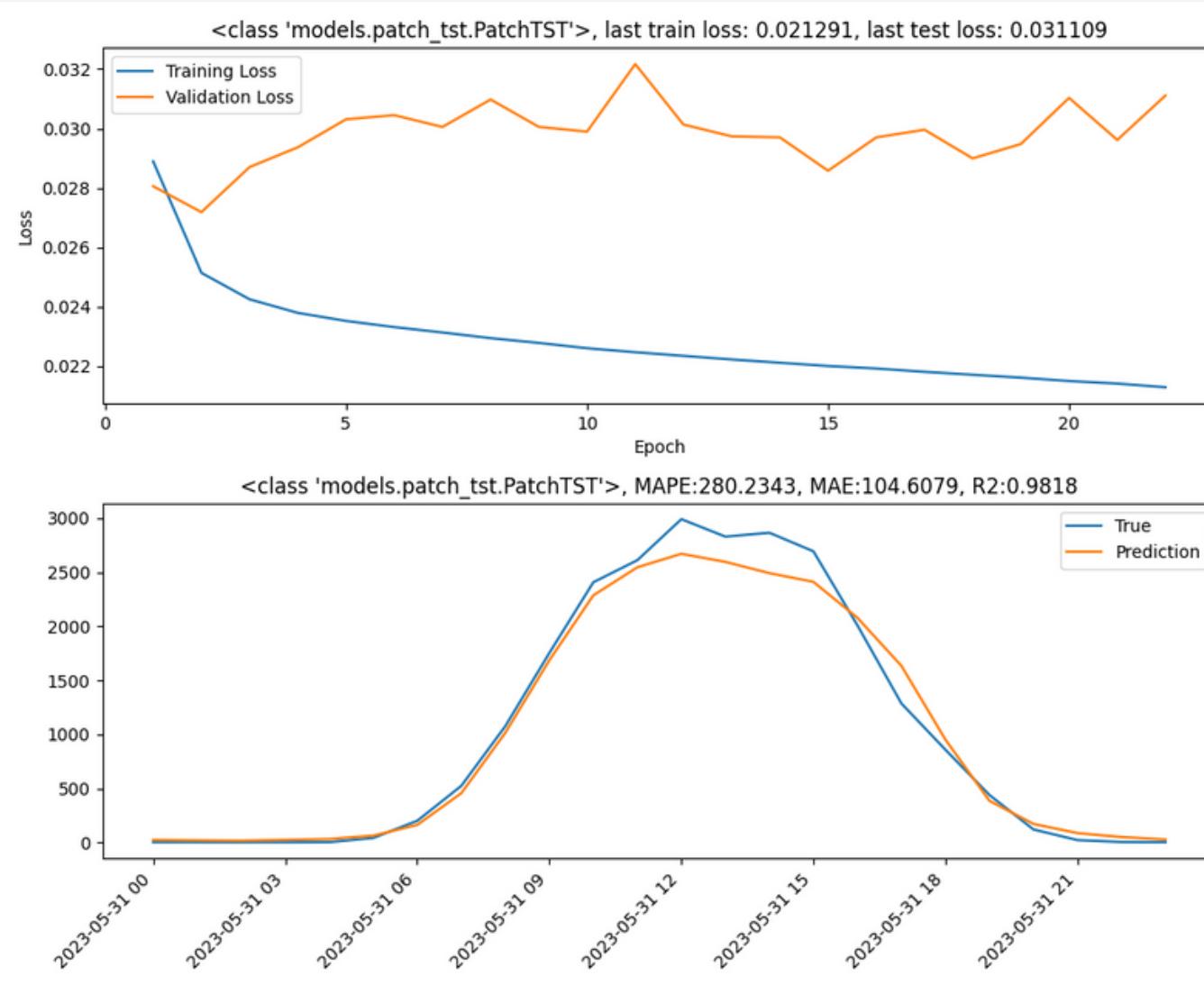
Patch TST Best 파라미터

- Patch length: 16
- Num of patches: 64
- Model dimension: 128
- Num of heads: 8
- Num of layers: 3
- Dim feed forward: 256
- Batch size: 32
- Learning rate:
 - 0.001(24/prod) (96/prod)
 - 0.00001 (24/cons) (96/cons)
- Epochs: 5
- Output layer: F.sigmoid
- LR Scheduler: CosineAnnealingLR
 - eta_min: lr*0.01 (prod), No scheduler (cons)
 - T_max: 500 (prod)

Best 데이터 전처리

- 전체 피쳐 데이터셋
- X scaler: minmax
- Y scaler: minmax

PatchTST - 실험(1)



Learning rate 0.001 ~ 0.00001 을 실험한 결과 (전체 피쳐)

- LR scheduler 는 CosineAnnealing 을 사용, 각 LR*0.1 과 LR*0.01 케이스 실험

Production & 24시간 기준,

Best Score: MAE: 104.6079, R2: 0.9818

- LR Scheduler: None
- Learning rate: 1e-5
- Epochs: 500
- Early-stop-patience: 20
- early-stopped: 21

하지만 Validation loss 를 봤을 때, 이 실험뿐 아니라 전체적으로, Epoch 5 이하에서 최저 Validation loss 가 찍힌것을 확인함.

-> 전체 평균 MAE: 208.5914

-> 추가 실험 진행: 같은 config + Epochs 5

PatchTST - 실험(2)

Epochs = 5 실험

- 전 실험에서 Epoch 5 미만에서 최소 val loss 가 확인되어 진행한 실험

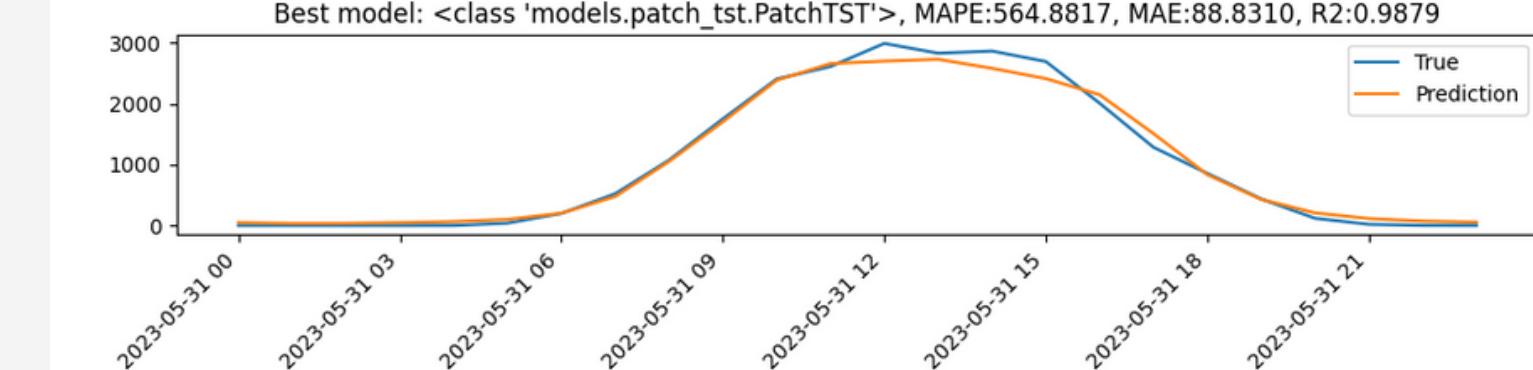
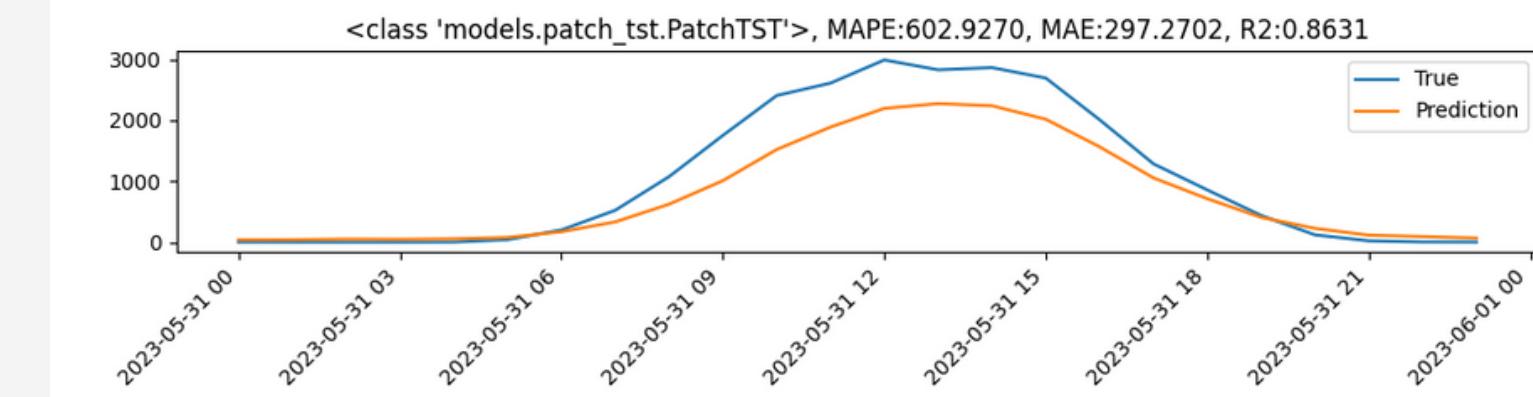
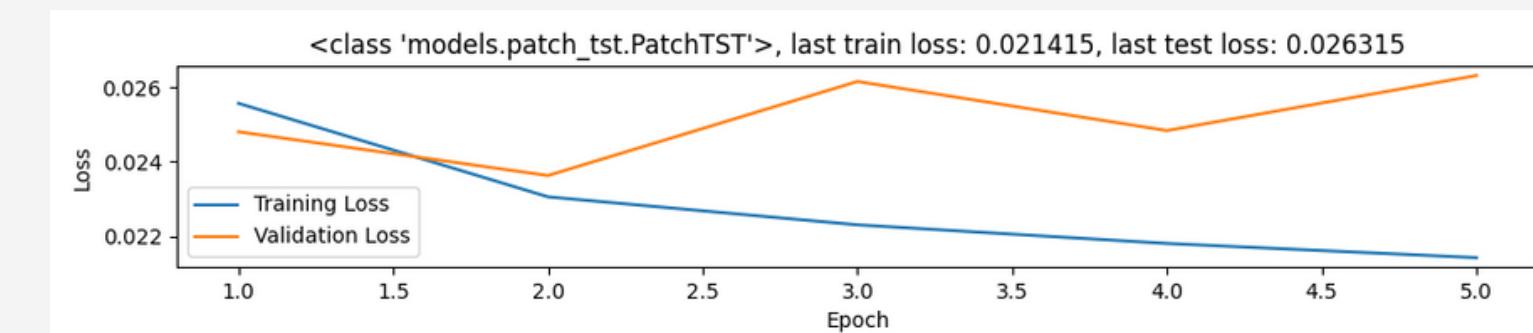
Production data & 24시간 기준,

Best Score: MAE: 88.8310, R2: 0.9879

- LR Scheduler
 - CosineAnnealing
 - eta_min: LR*0.01
- Learning rate: 0.001
- Epochs: 5

평균 MAE 가 (전 실험) 200대에서 170대로 떨어지며 전체적인 성능 향상 단, Validation loss가 여전히 불안정

-> 추가 실험 진행: Learning rate 를 줄여서 실험 재진행



PatchTST - 실험(3-1)

Learning rate 1e-6 ~ 1e-9 실험

- 전 실험에서 val loss 그래프가 불안정하여 진행한 실험

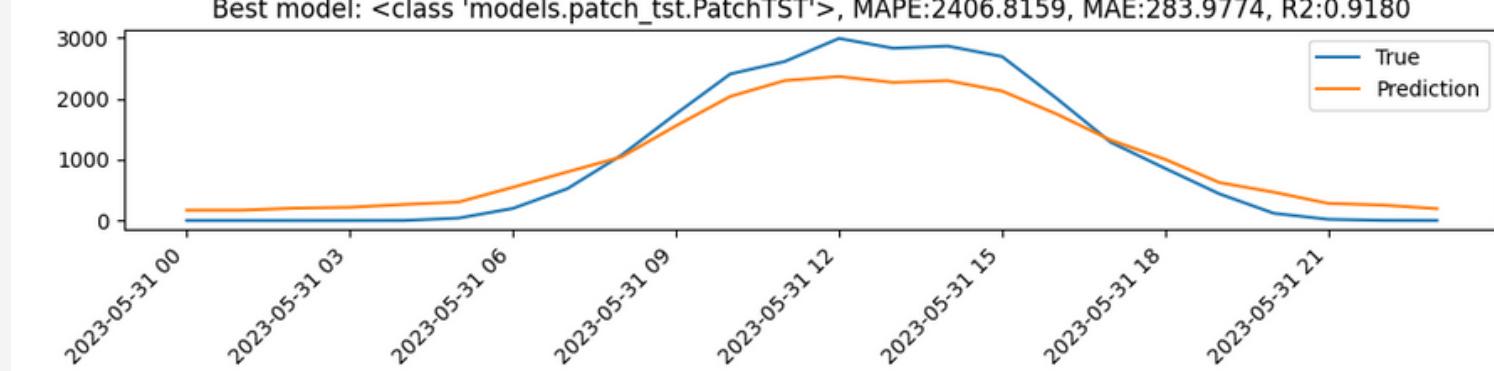
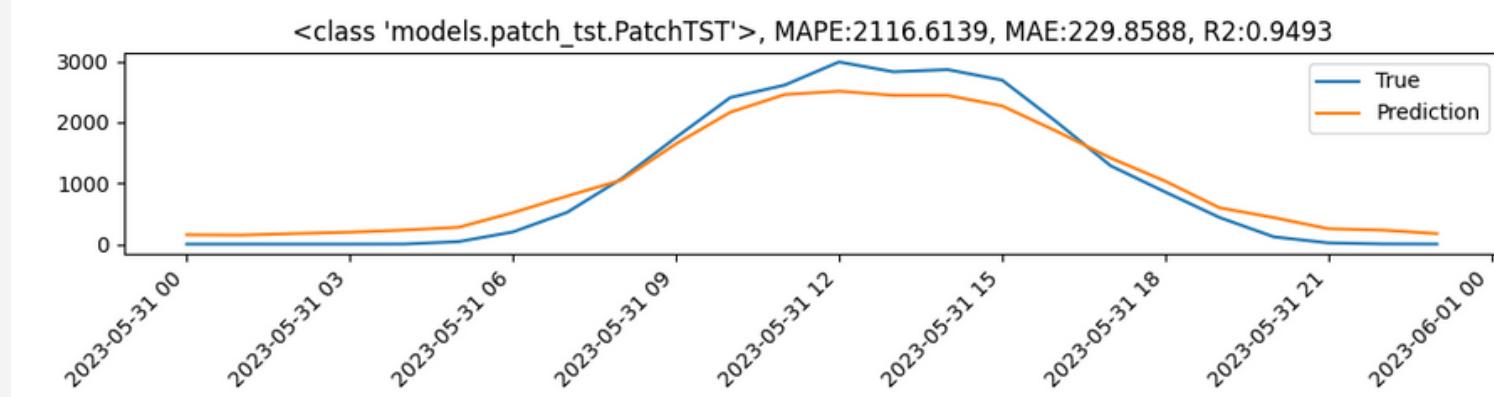
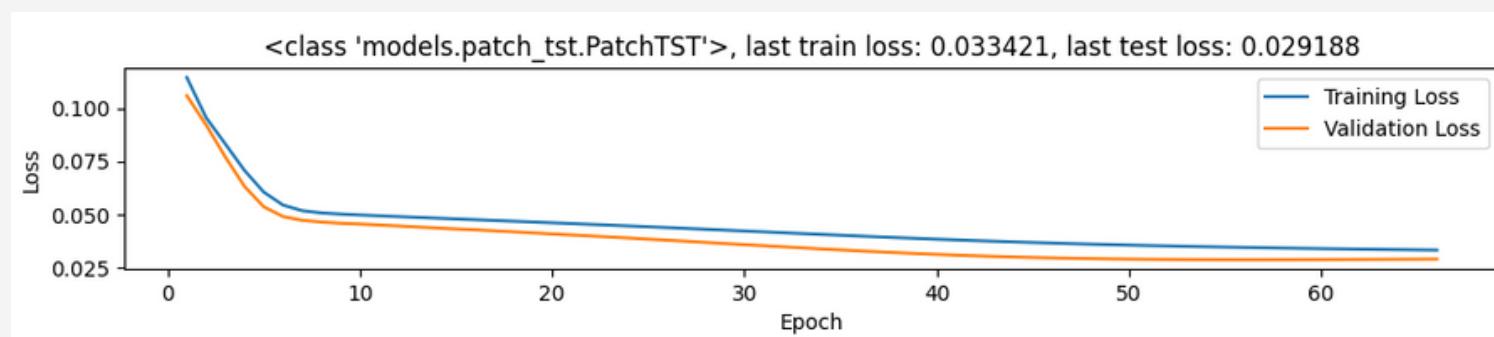
Production data & 24시간 기준,

AVG MAE: 210, Best AVG MAE: 262

- LR Scheduler
 - CosineAnnealing
 - eta_min: LR*0.1 ~ LR*0.01
- Epochs: 500
- Early-stopping-patience: 20

Validation loss는 안정적으로 보이지만 평균 MAE가 210대로, Epoch=5 실험과 비교하여 평균 MAE가 올랐다.
(최소 MAE가 181로 이전 실험들과 비교하여 높은 편)

-> 추가 실험 진행: 선택피쳐로 동일한 실험 진행



PatchTST - 실험(3-2)

Learning rate $1e-6 \sim 1e-9$ 실험

- 선택피쳐 실험 진행

Production data & 24시간 기준,

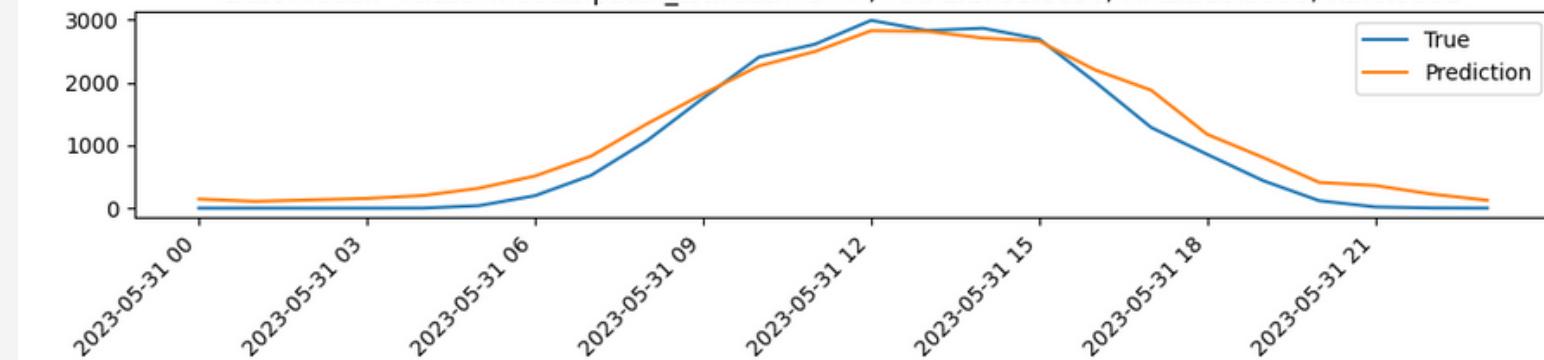
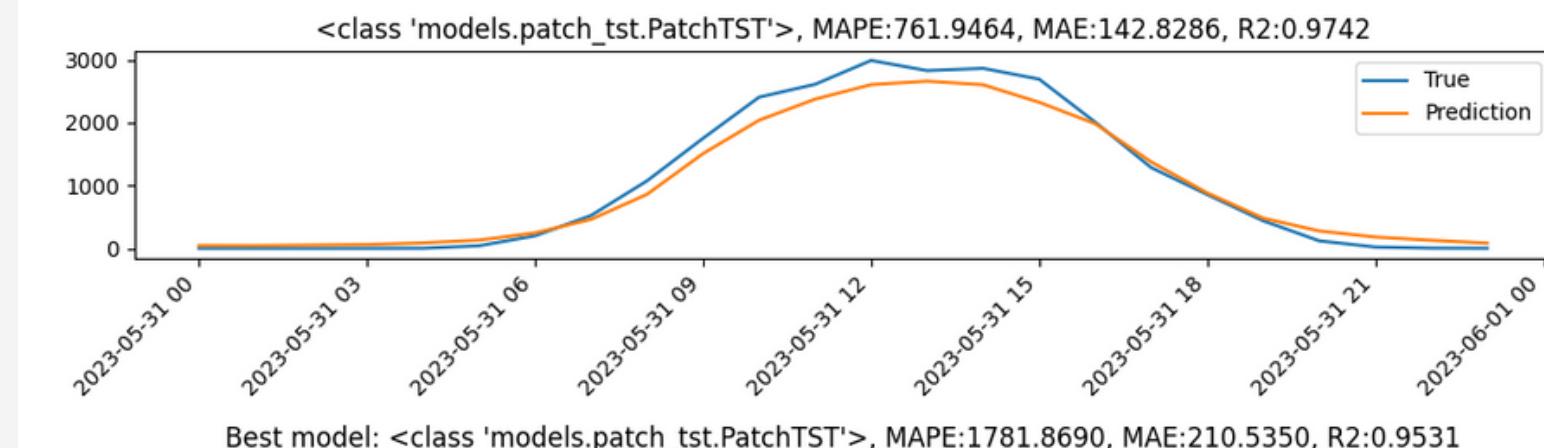
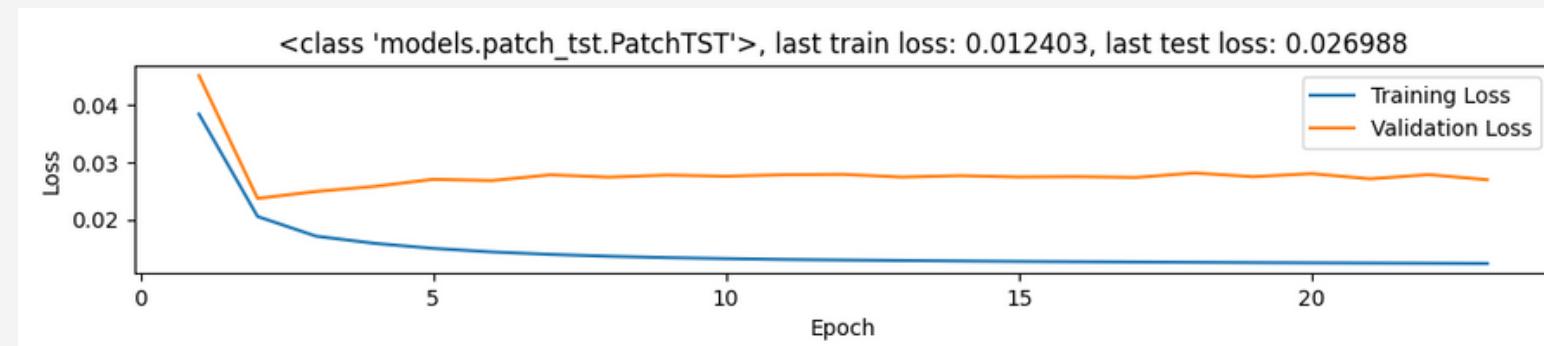
AVG MAE: 155, Best AVG MAE: 187

- LR Scheduler
 - CosineAnnealing
 - eta_min: LR*0.1 ~ LR*0.01
- Epochs: 500
- Early-stopping-patience: 20

평균 MAE 가 155 로 가장 좋은 평균 MAE 를 보여주고 있다.

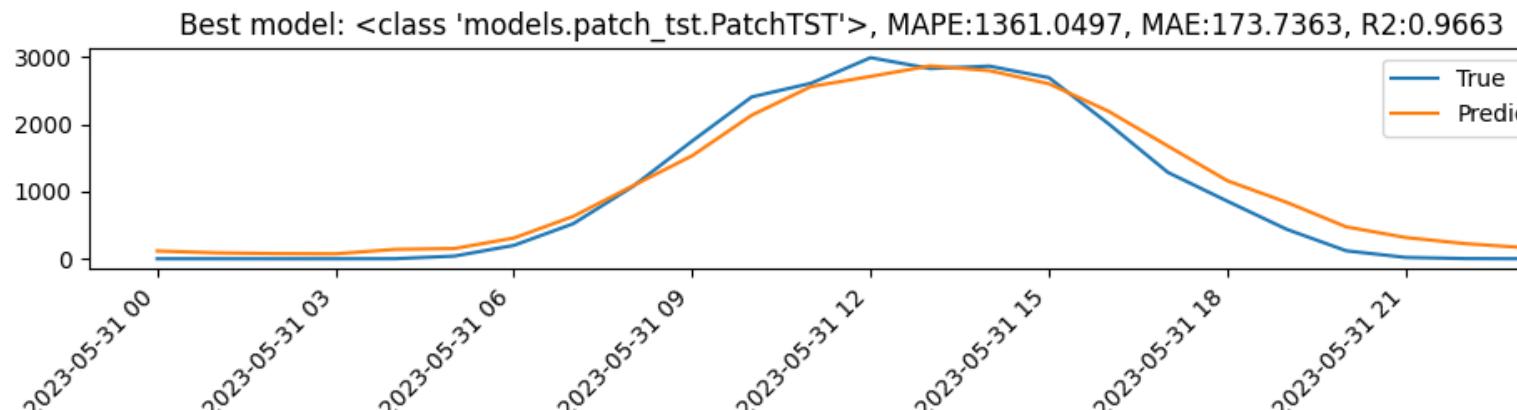
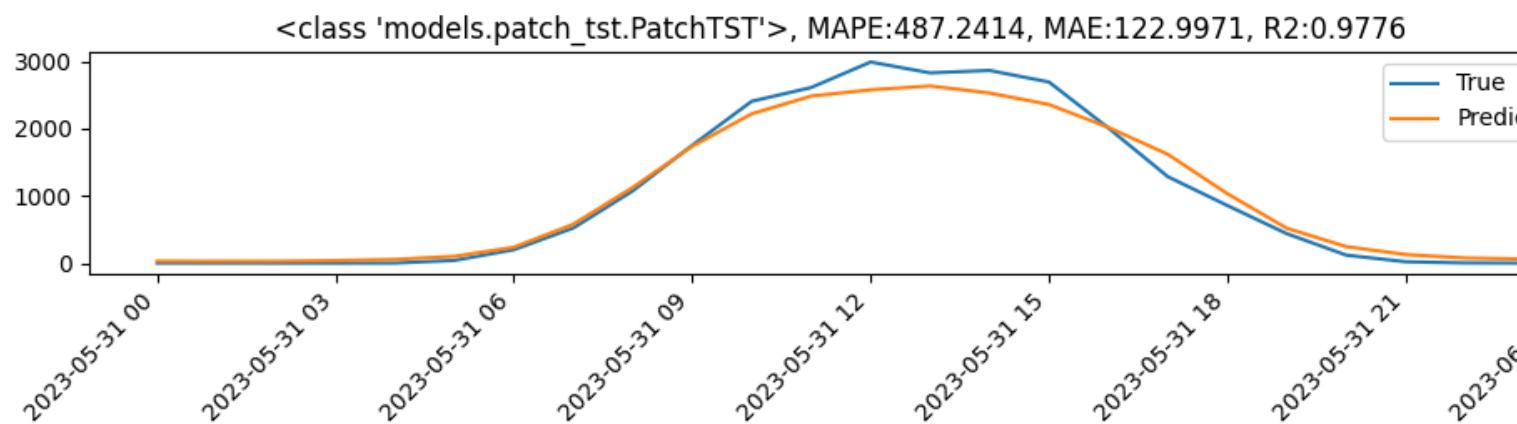
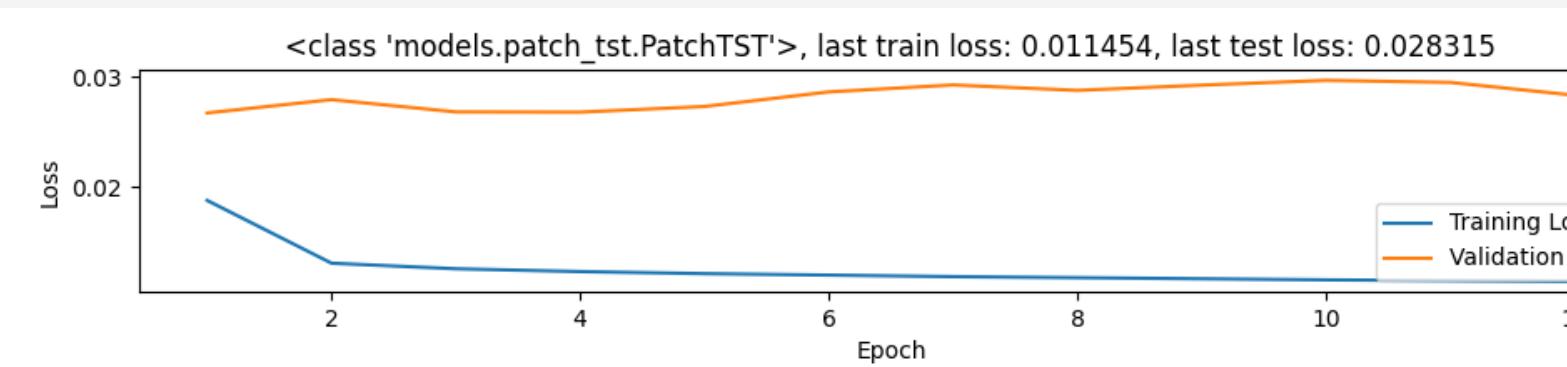
선택피쳐로 추가 실험들을 진행했으나, 최고 성능이 MAE 100 초반으로 전체피쳐 실험 결과보다 못했다. 또한 최적 Learning rate 가 0.00001 로, 전체피쳐 Epoch=5 실험할때의 Learning rate 범위에 포함되었다.

-> 추가 실험 진행: 전체피쳐 기준으로 성능 향상 실험 진행

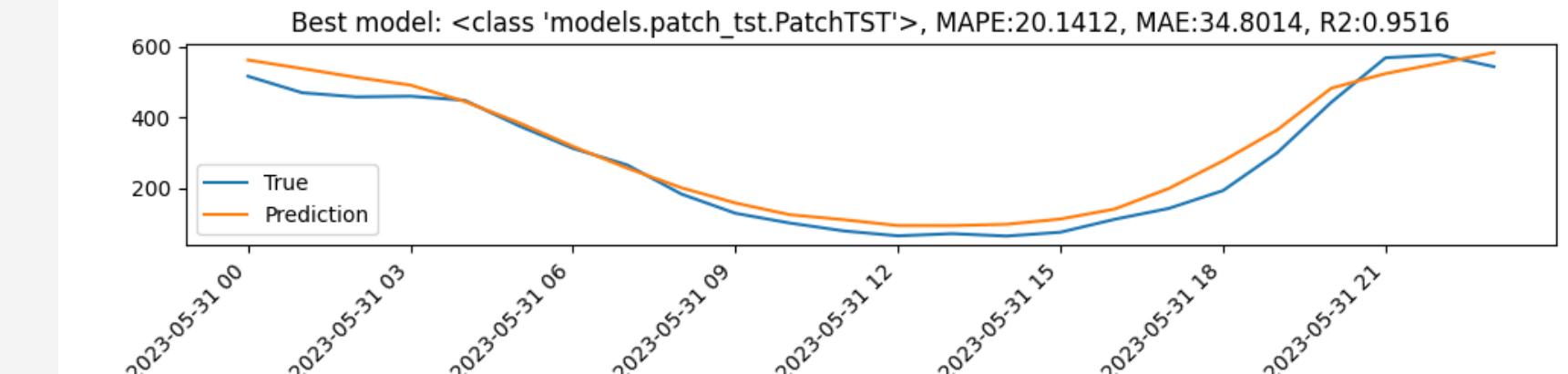
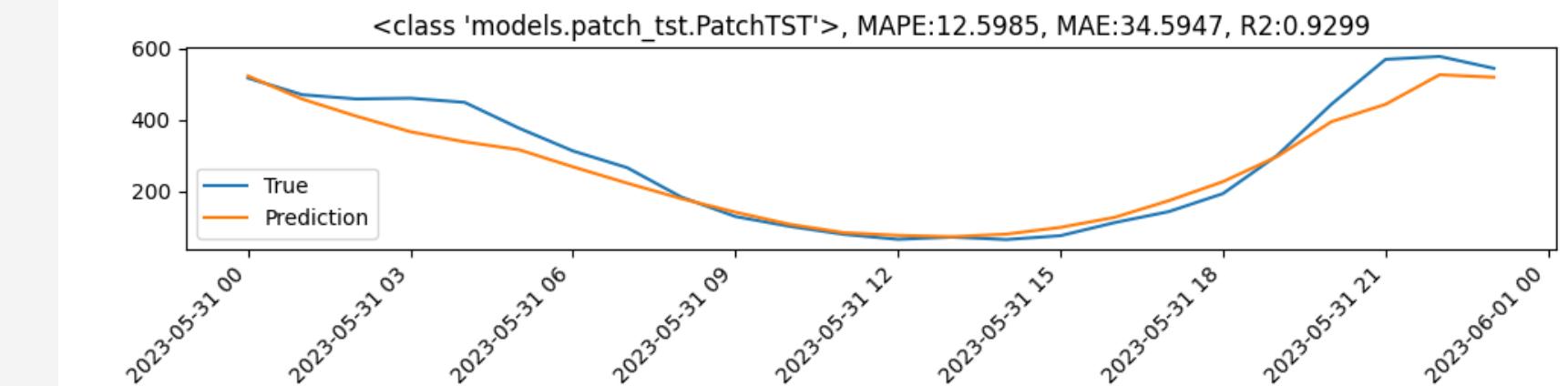
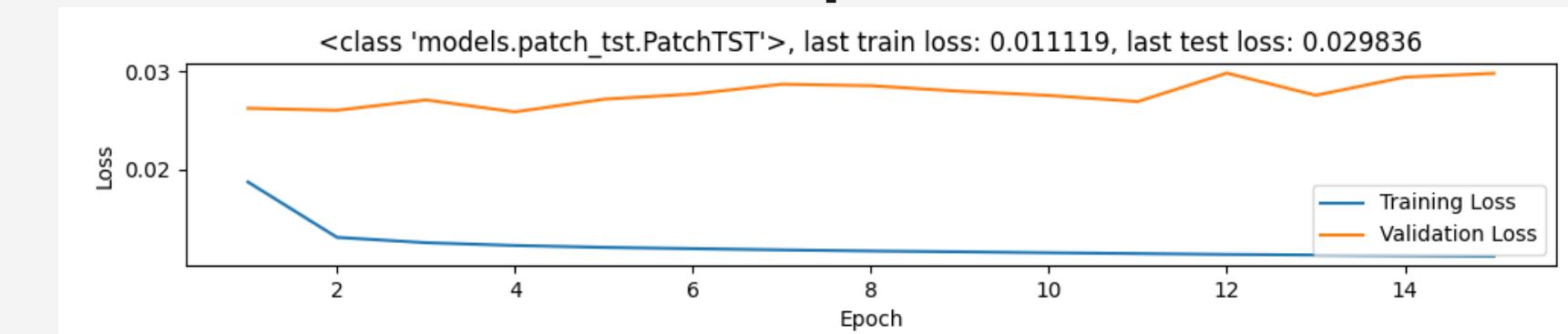


PatchTST - 24시간 선택피처 Best

Production

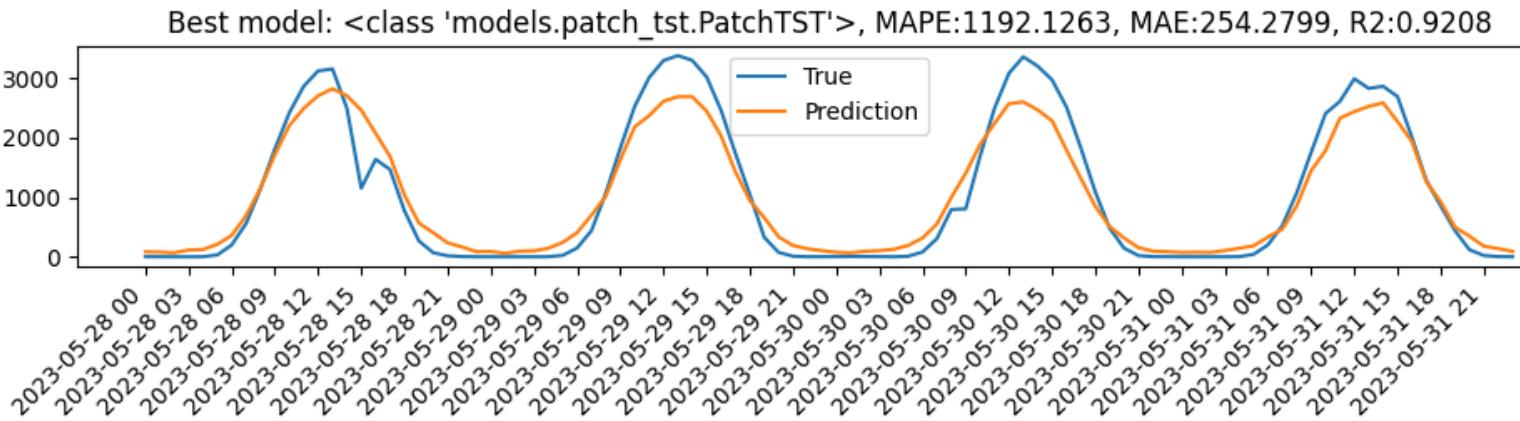
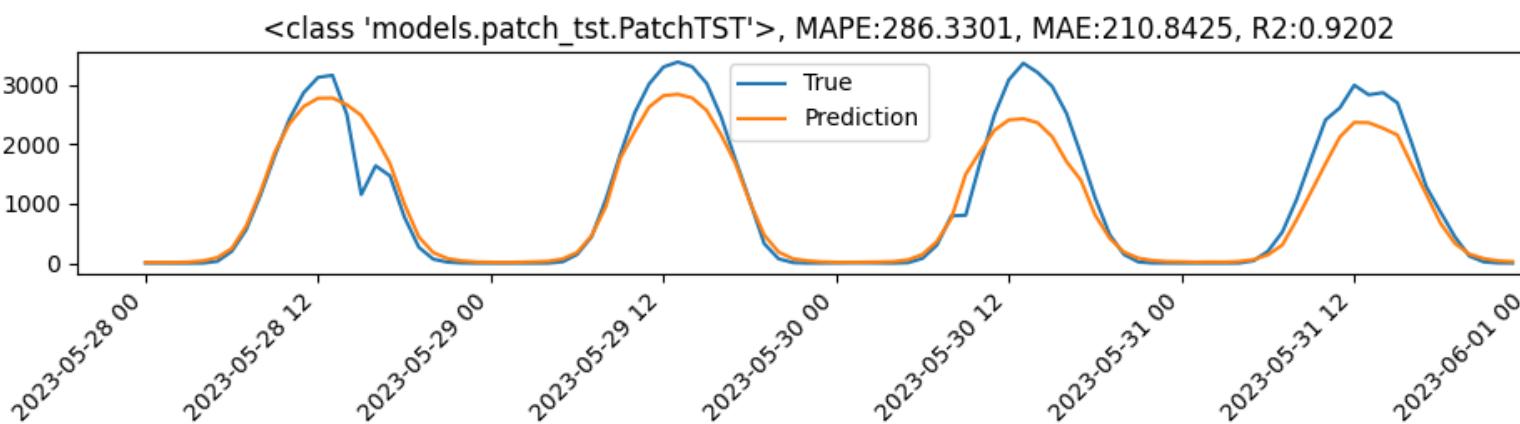
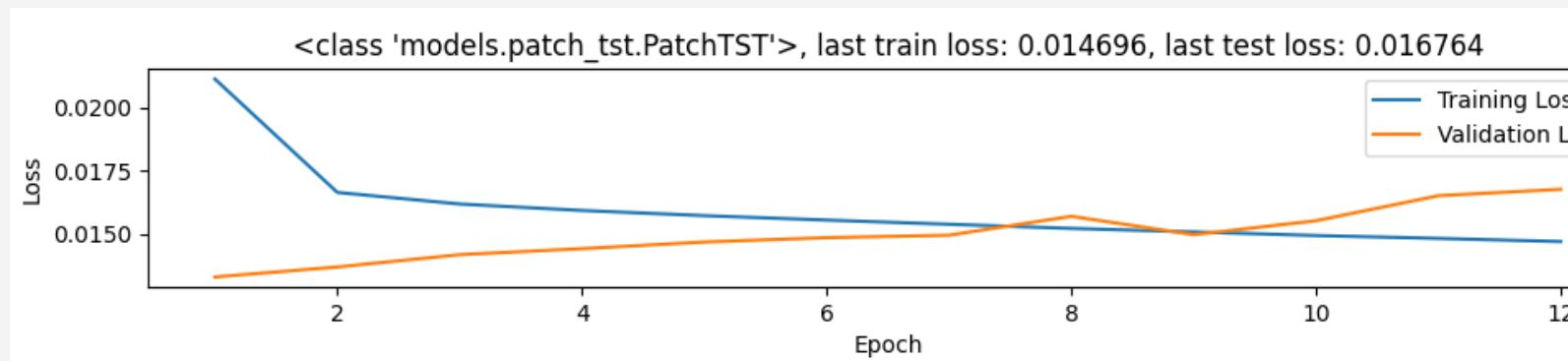


Consumption

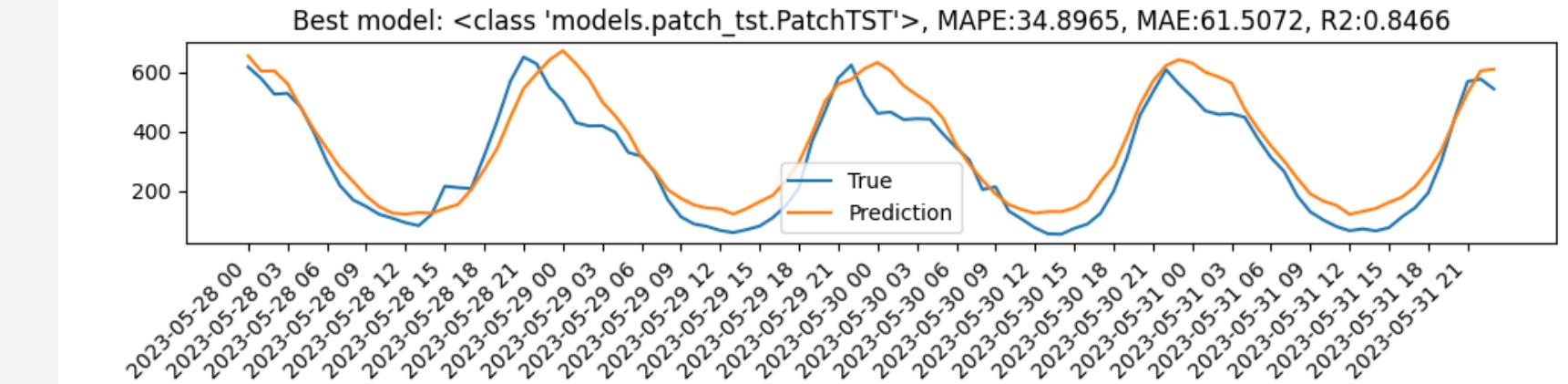
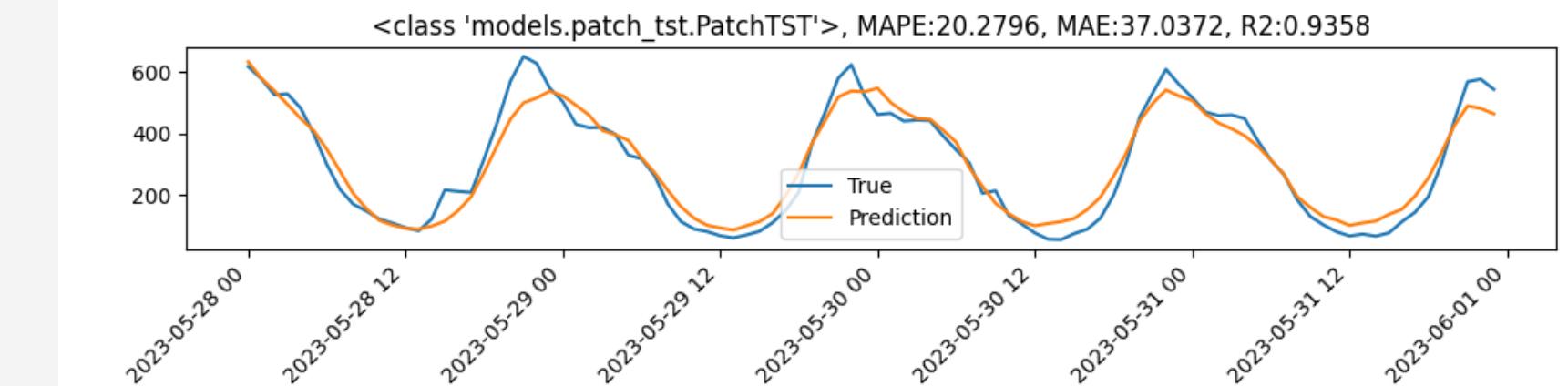
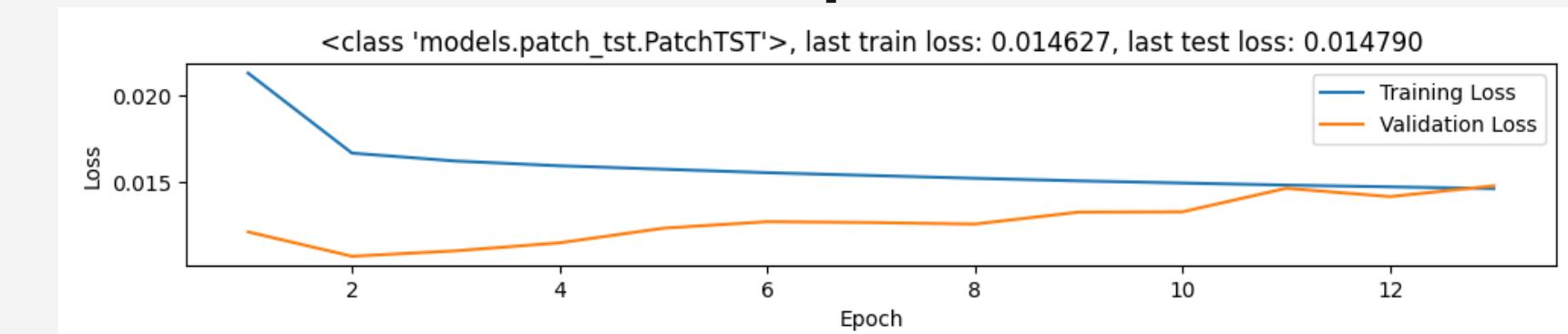


PatchTST - 96시간 선택피쳐 Best

Production

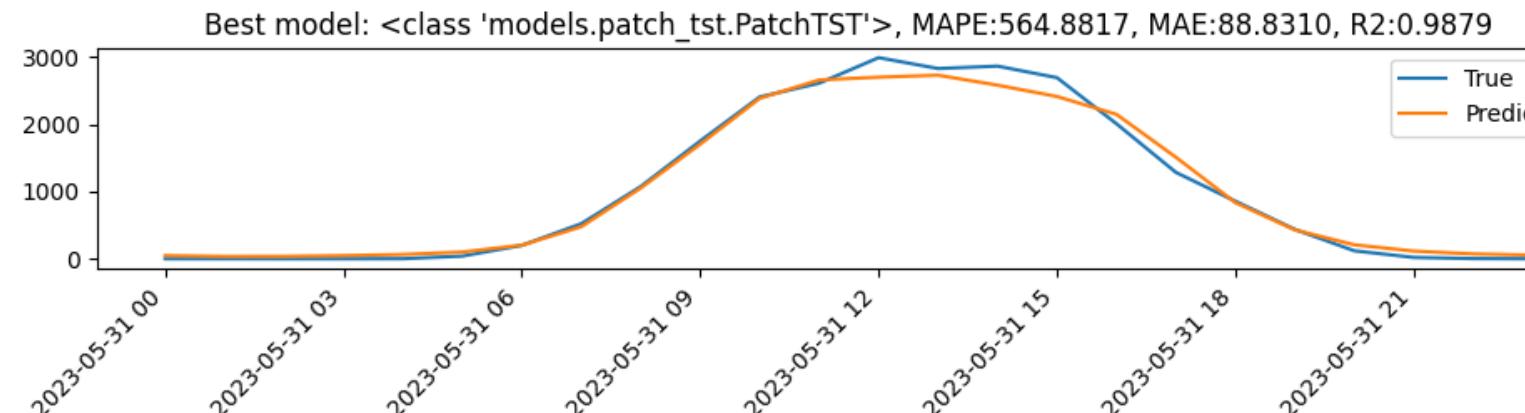
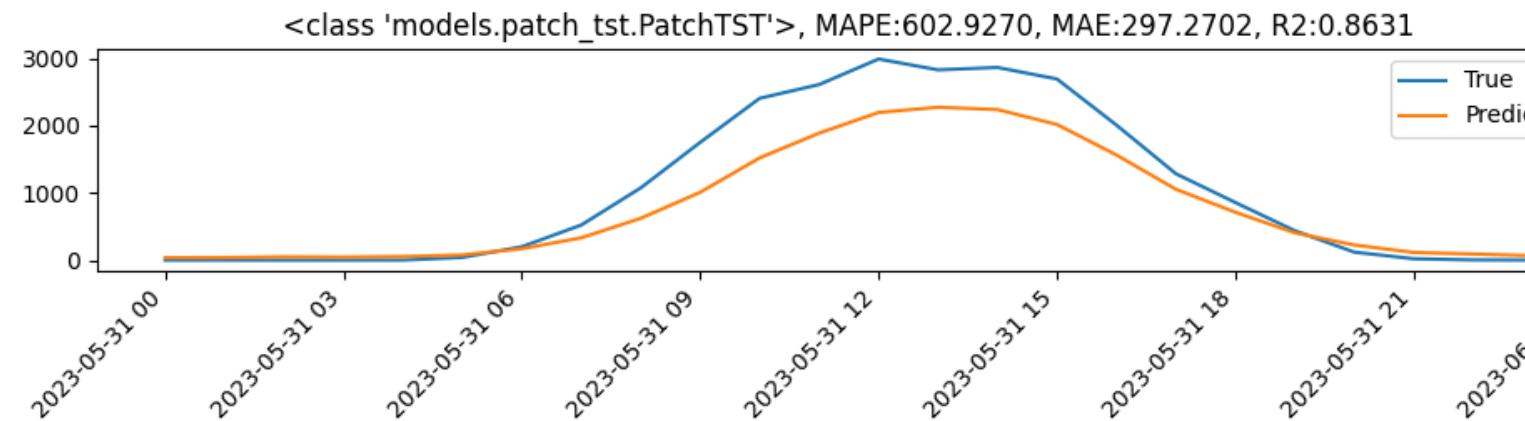
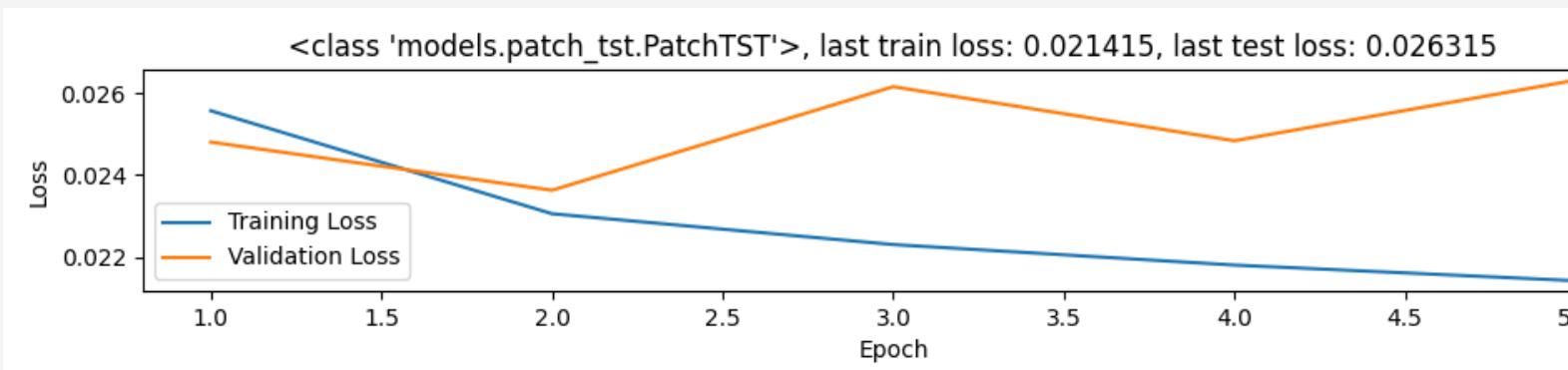


Consumption

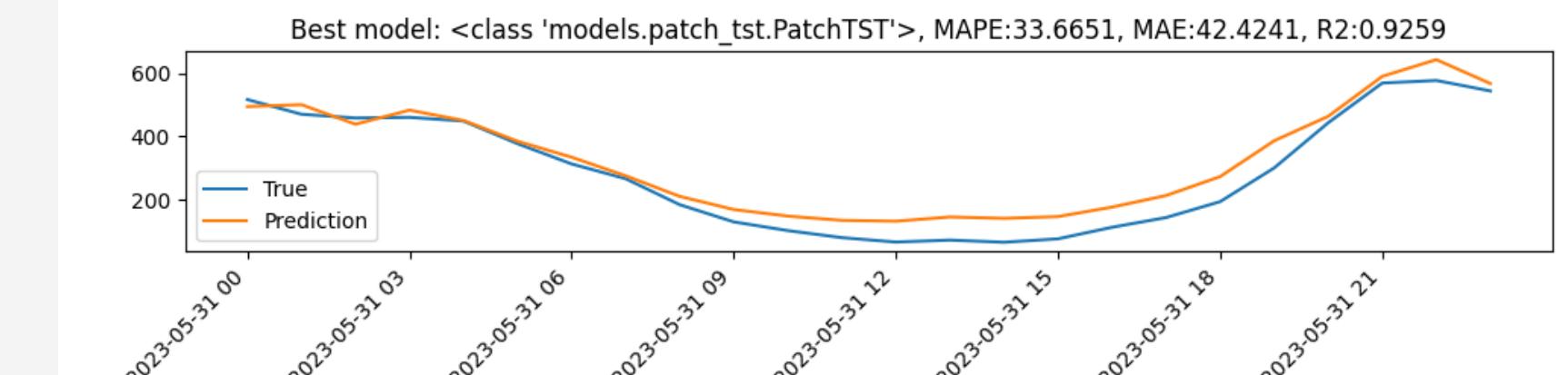
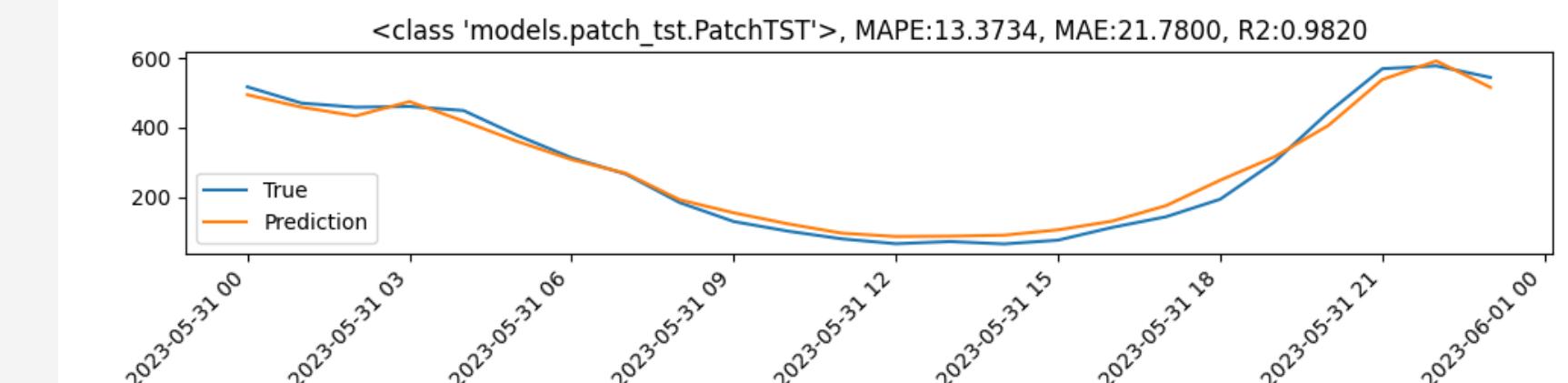
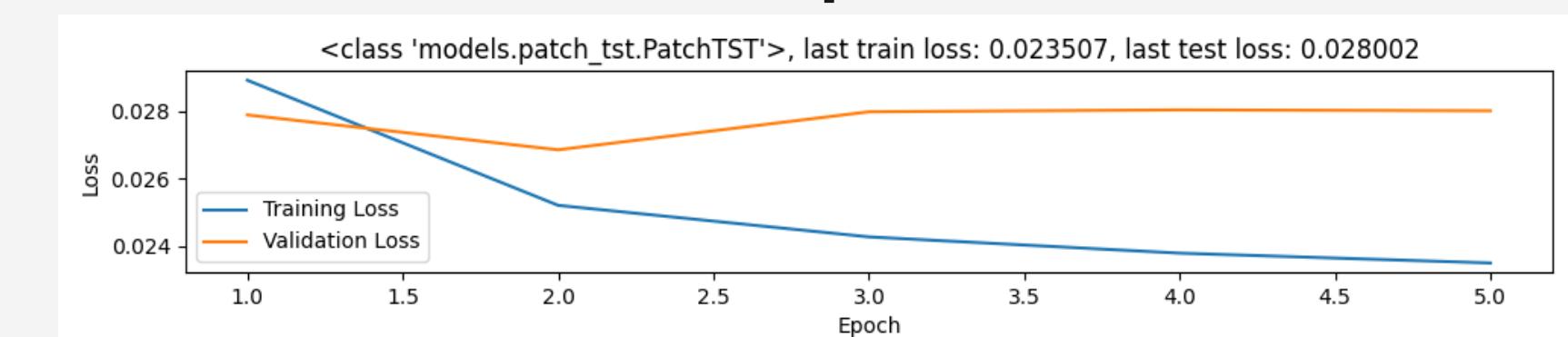


PatchTST - 24시간 전체피쳐 Best

Production

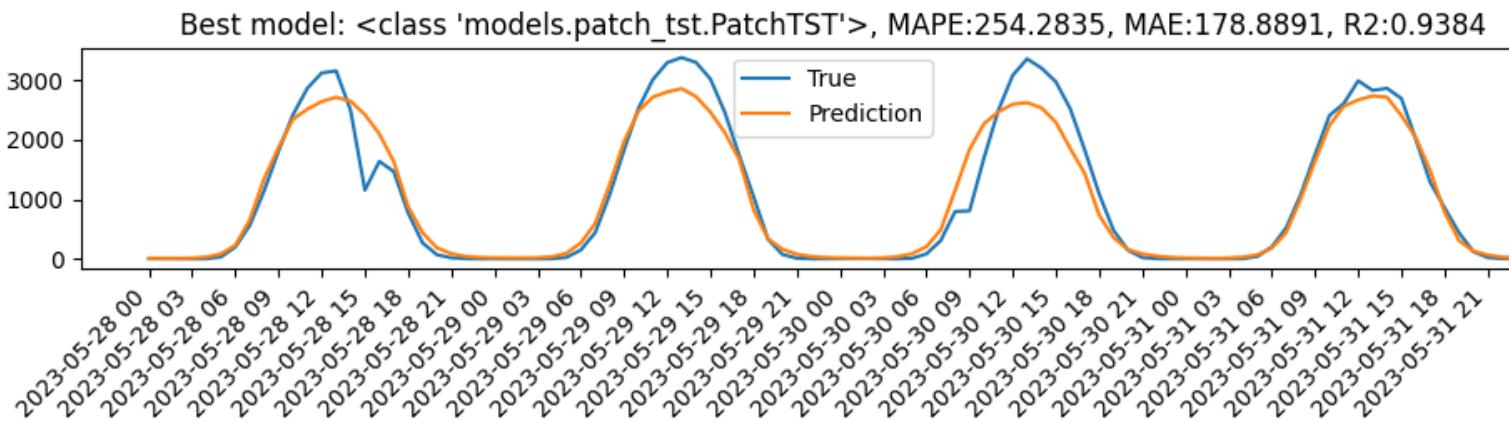
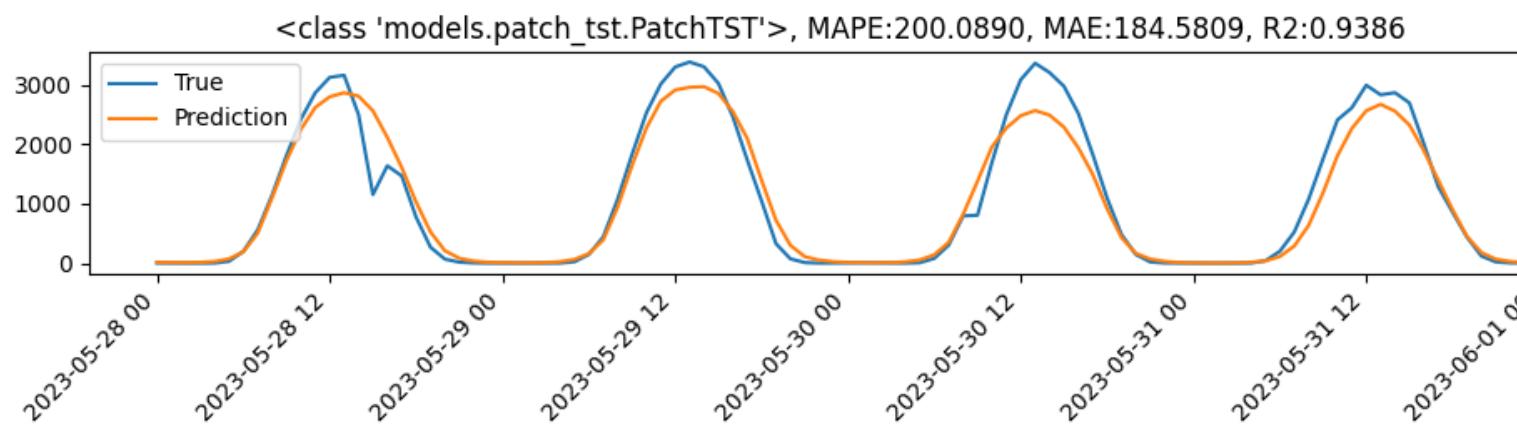
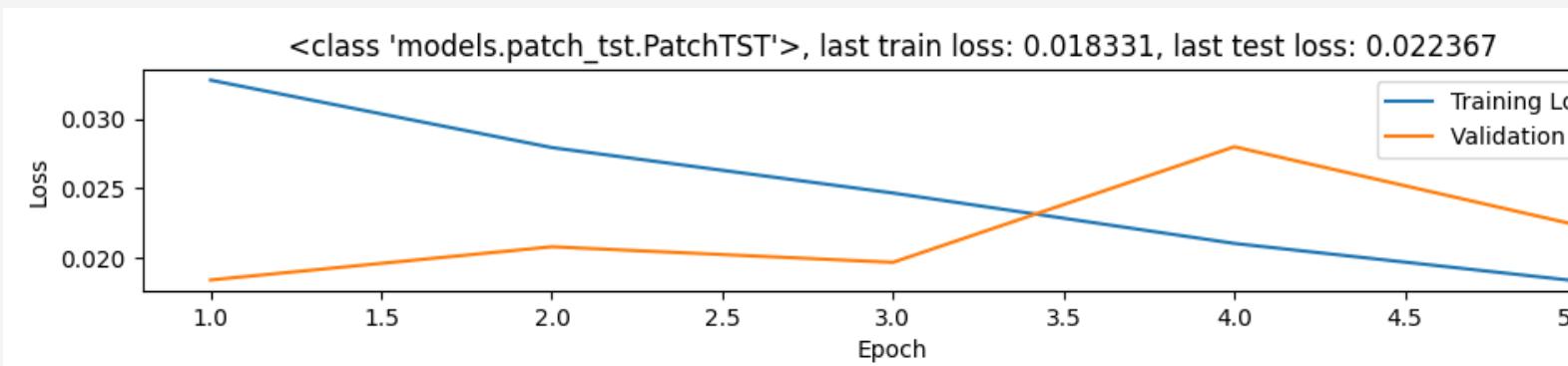


Consumption

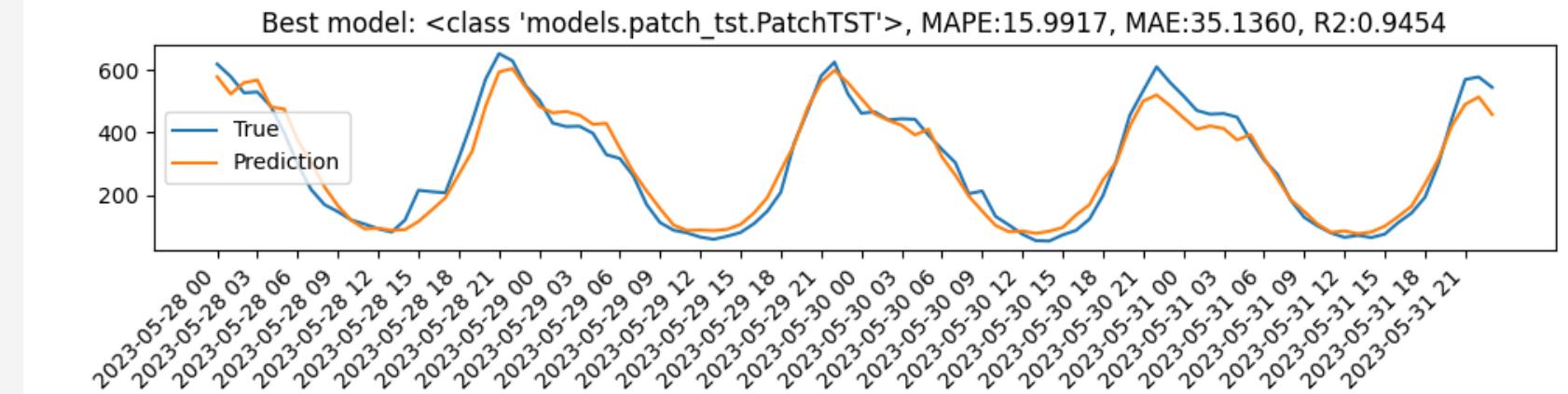
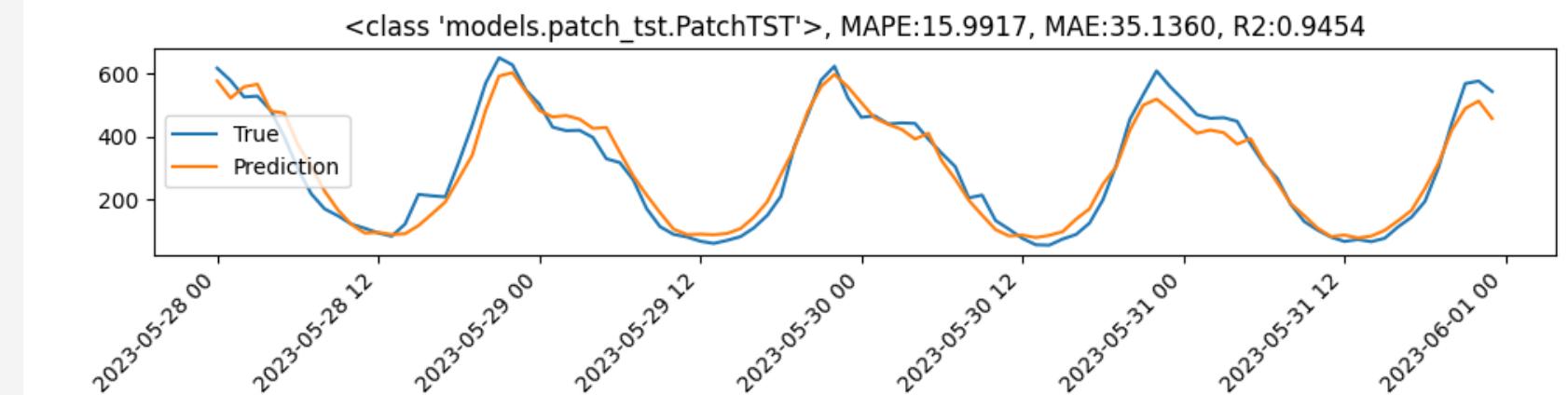
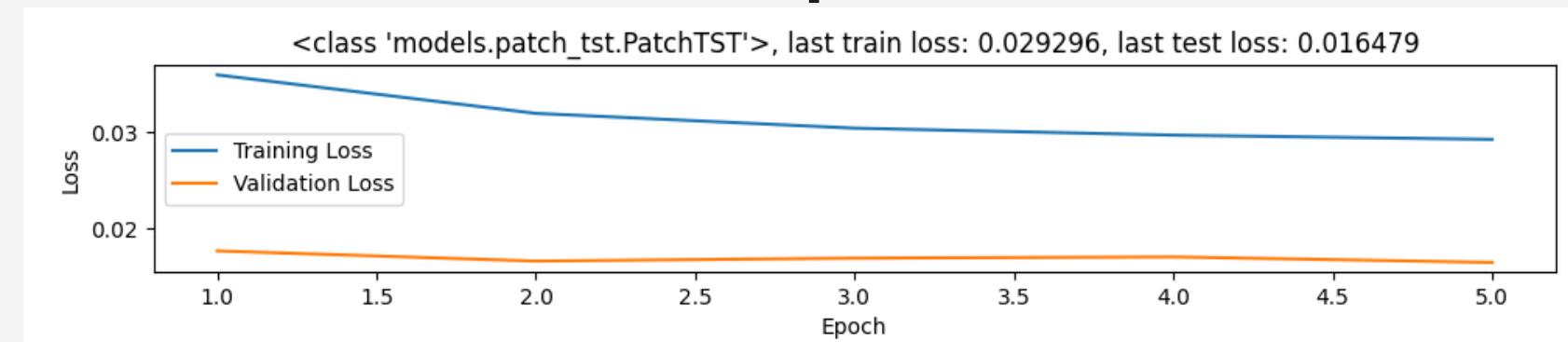


PatchTST - 96시간 전체피쳐 Best

Production



Consumption



PatchTST - 추가 실험했던 내용

속도 절감 및 성능 유지를 위한 실험

- Batch size: 32, 64, 128, 256, 512
 - 속도 절감에 좋았으나 성능은 32 가 가장 좋았다.
- Patch length & N patches
 - 속도 절감에는 거의 영향이 없었다.
 - $(24*14)/2 \Rightarrow (21\text{days} \rightarrow 7\text{days})$ 등 줄여보는 실험들을 진행하였으나, 평균 MAE 는 170 대로 큰 차이 없었다.
 - 21days \rightarrow 28days 또한 성능 차이는 거의 없었다.
- Model dimension: 32, 64, 128, 256
 - 속도 절감에는 거의 영향이 없었다.
 - 성능도 특별히 좋아지지 않았다.
- Dim feed forward: 128, 256, 512, 1024
 - 속도 절감에는 거의 영향이 없었다.
 - 성능도 특별히 좋아지지 않았다.
- N heads & N layers:
 - 속도 절감에는 미미한 영향이 있었다.
 - 성능이 더 좋은 파라미터는 찾지 못했다.

이외의 실험들

- Scaler 실험
 - x scaler=None, y scaler=None
 - AVG MAE=900 대로 나오며 추가 실험 진행 X
 - x scaler=minmax, y scaler=None
 - AVG MAE=300 중반으로 나오며 추가 실험 진행 X
- Epoch 5 미만 실험들
 - 불규칙성이 너무 크고, Epoch 5에서보다 나은 성능의 모델이 없었다.
- Epoch 10 실험
 - 성능은 크게 다르지 않으나 시간이 오래걸렸다.

PatchTST - Best

실험 결과

- 24시간 후 예측(test_size = 24)
 - 소비량(consumption)
 - Best: MAE = 21.7800, R2 = 0.9820
 - 생산량(production)
 - Best: MAE = 88.8310, R2 = 0.9879
- 96시간 후 예측 (test_size = 96)
 - 소비량(consumption)
 - Best: MAE = 35.1360, R2 = 0.9454
 - 생산량(production)
 - Best: MAE = 178.8891, R2 = 0.9384

모델 성능 비교

문제

- 목표: 24시간 및 96시간 후의 전력 소비량과 생산량 예측
 - 1) 24시간 후 예측
 - dynamic
 - ARIMA vs. ANN (Pred_size = 1)
 - 통으로(pred_size = 24)
 - ANN vs. PatchTST
 - 2) 96시간 후 예측
 - dynamic
 - ARIMA vs. ANN (Pred_size = 1)
 - 통으로(pred_size = 96)
 - ANN vs. PatchTST

실험 방법

- prediction size
 - 1) 24시간 후, 96시간 후를 통으로 예측
 - 2) 1시간 단위로 dynamic하게 예측
- 피쳐
 - 전체 피쳐, 선택 피쳐 (ANN, PatchTST)
 - 타겟 1개 피쳐 (ARIMA, SARIMA)
- 활용 모델: ARIMA, SARIMA, ANN, PatchTST
- 평가 지표: MAE, MAPE, R2 score

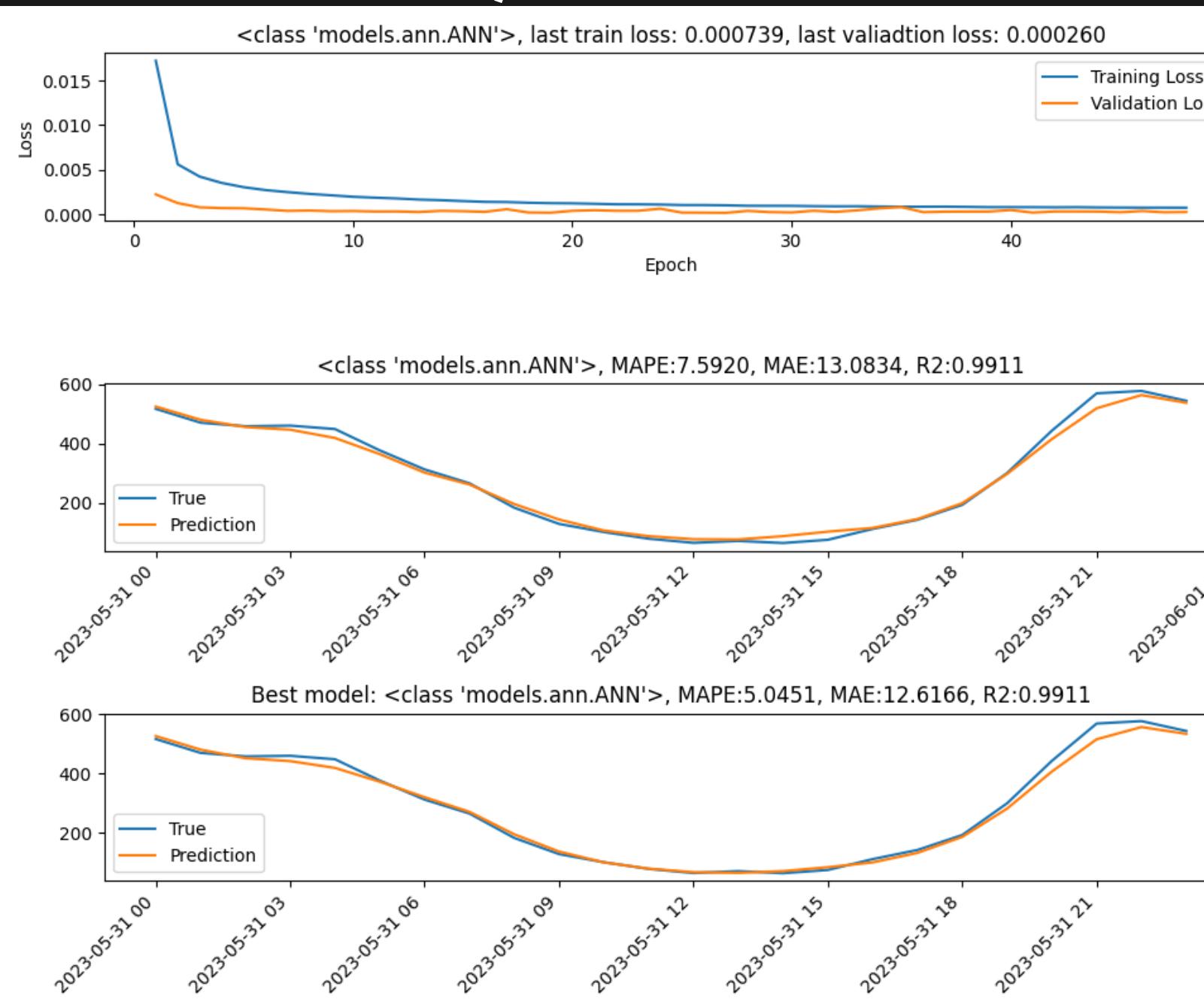
최종 모델 - Consumption

문제

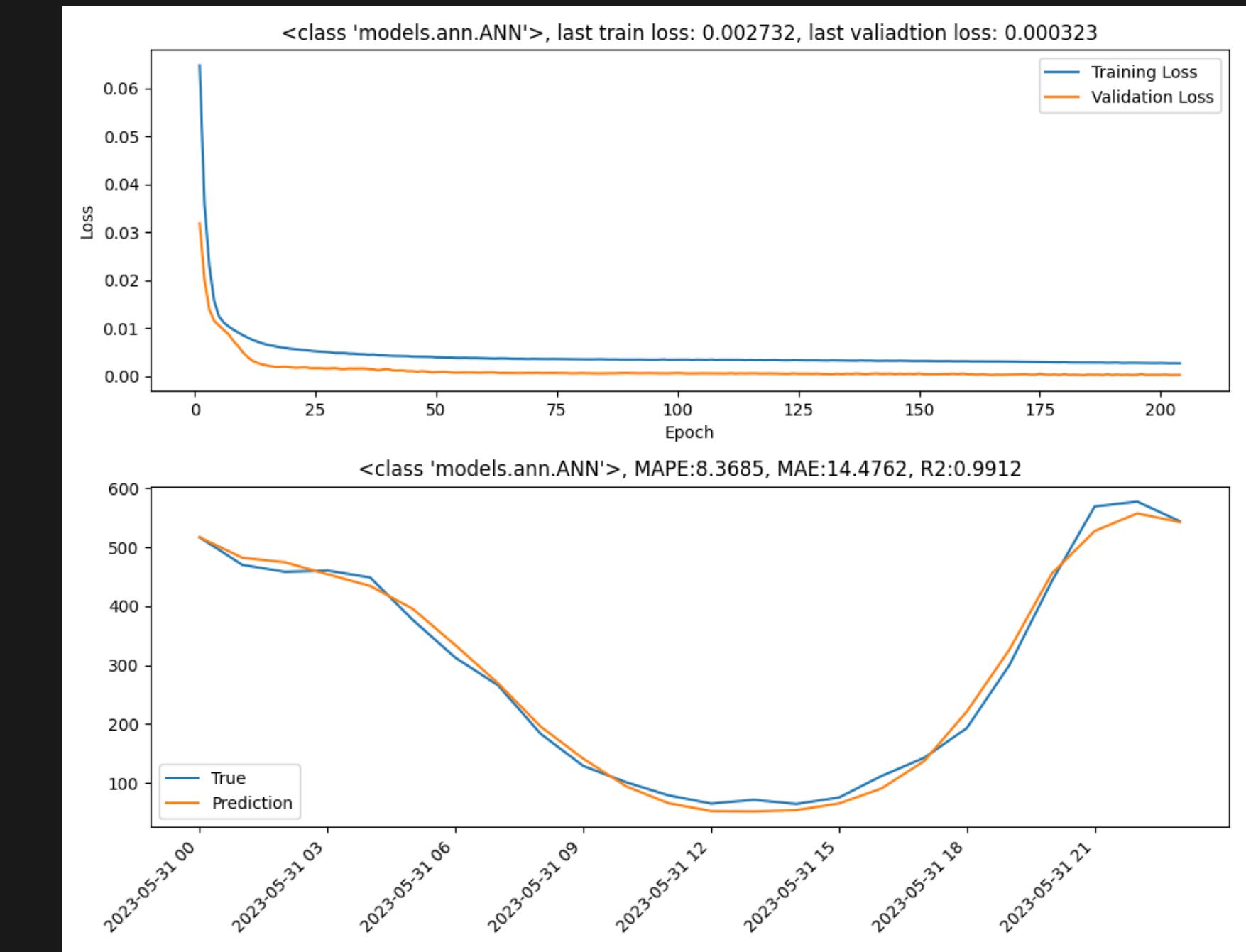
- 목표: 24시간 및 96시간 후의 전력 소비량과 생산량 예측
 - 1) 24시간 후 예측
 - dynamic(pred_size = 1)
 - ARIMA vs. ANN[👑]
 - ARIMA: 13.9550
 - ANN : 12.6166
 - 통으로(pred_size = 24)
 - ANN[👑] vs. PatchTST
 - ANN : 14.4762
 - PatchTST: 21.7800
 - 2) 96시간 후 예측
 - dynamic(pred_size = 1)
 - ARIMA vs. ANN[👑]
 - ARIMA: 40.5098
 - ANN : 27.9459
 - 통으로(pred_size = 96)
 - ANN vs. PatchTST[👑]
 - ANN : 38.7249
 - Patch TST : 35.1360

최종 모델 - Consumption - 24시간

24시간(Dynamic) Best: ANN

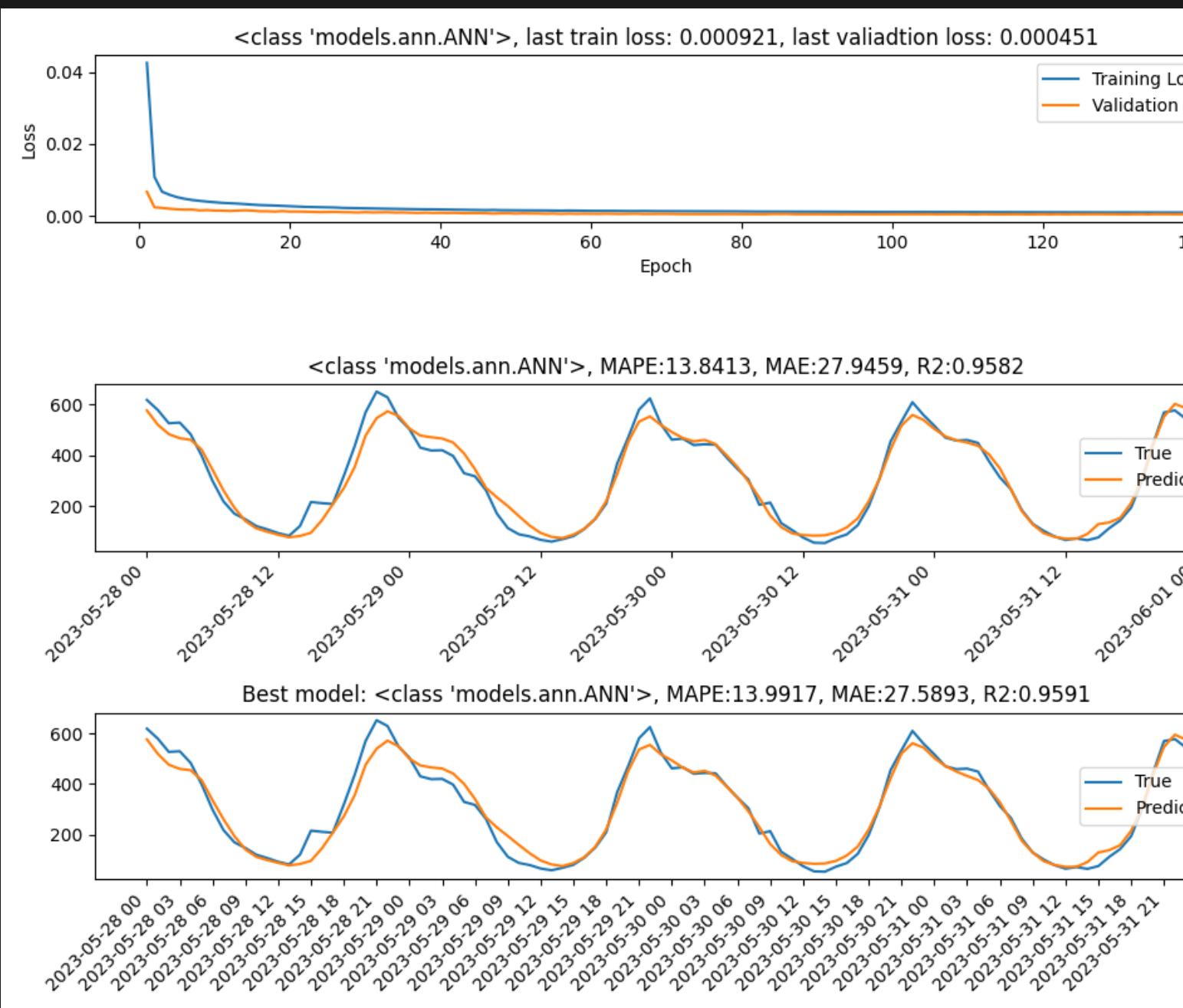


24시간(통으로) Best: ANN

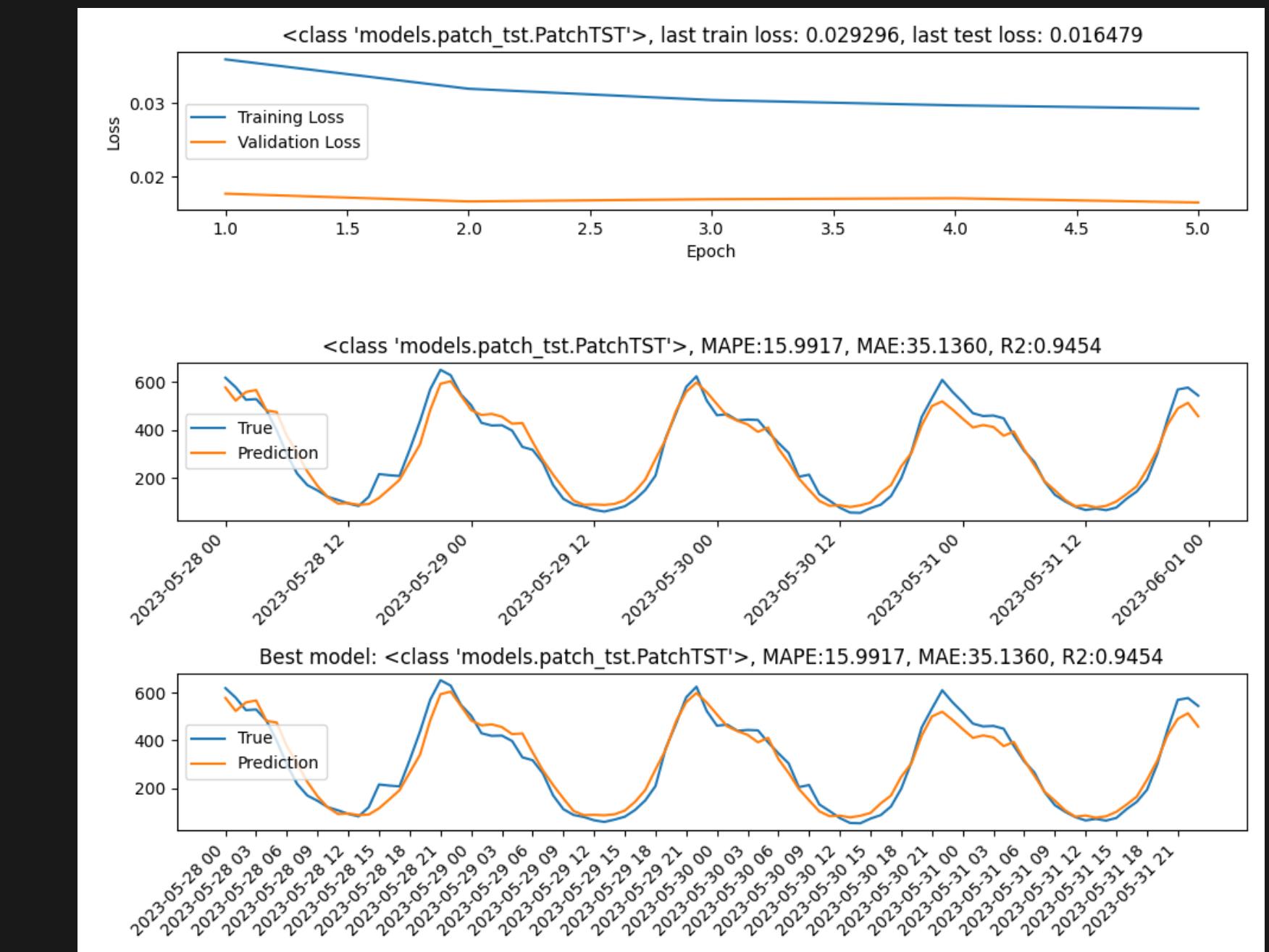


최종 모델 - Consumption - 96시간

96시간(Dynamic) Best: ANN



96시간(통으로) Best: PatchTST



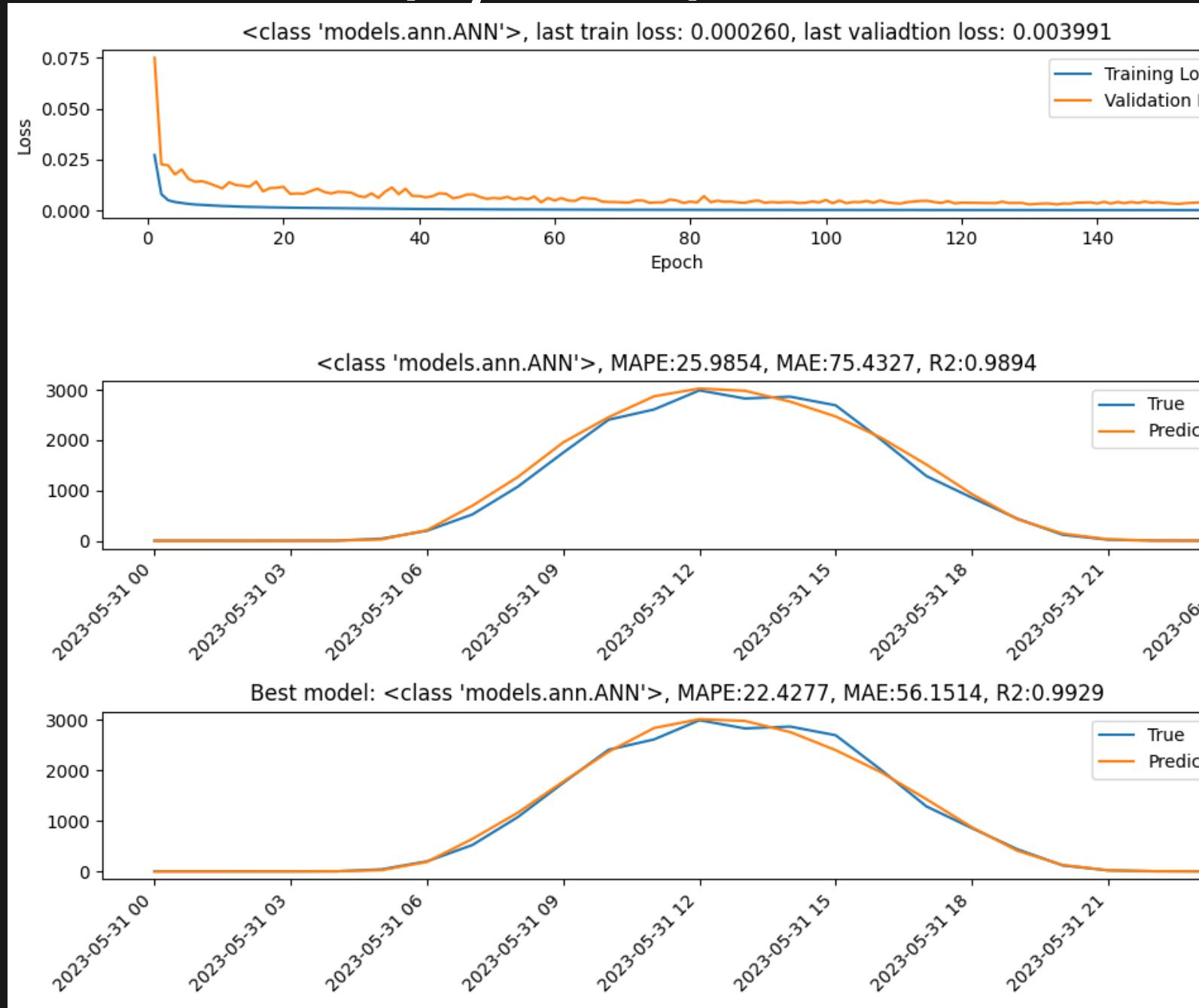
최종 모델 - Production

문제

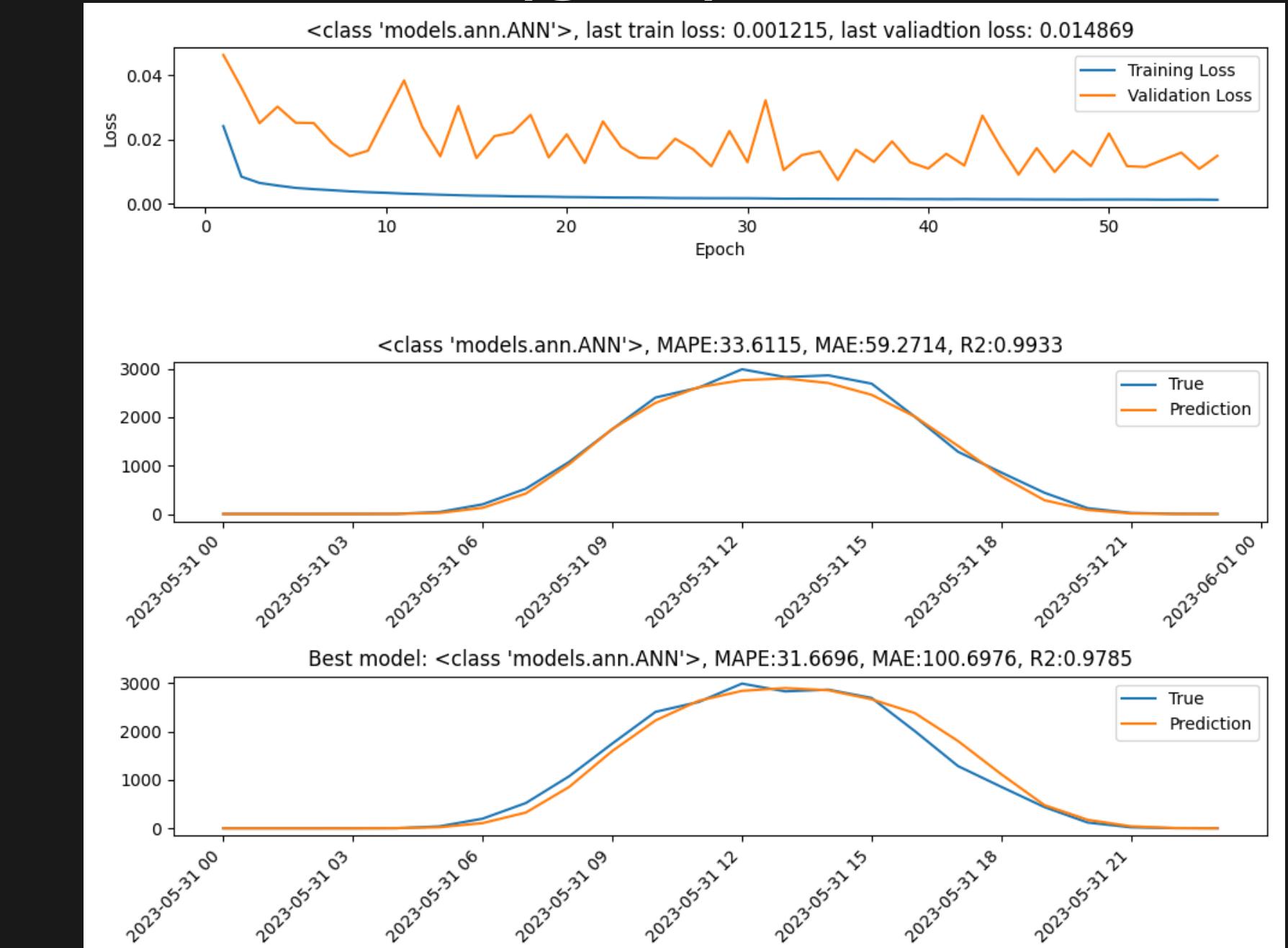
- 목표: 24시간 및 96시간 후의 전력 소비량과 생산량 예측
 - 1) 24시간 후 예측
 - dynamic(pred_size = 1)
 - ARIMA vs. ANN👑
 - ARIMA: 87.0535
 - ANN: **56.1514**
 - 통으로(pred_size = 24)
 - ANN👑 vs. PatchTST
 - ANN: **59.2714**
 - PatchTST: 88.8310
 - 2) 96시간 후 예측
 - dynamic(pred_size = 1)
 - ARIMA vs. ANN👑
 - ARIMA: 214.9977
 - ANN: **127.1683**
 - 통으로(pred_size = 96)
 - ANN vs. PatchTST👑
 - ANN : 205.1572
 - PatchTST: **178.8891**

최종 모델 - Production - 24시간

24시간(Dynamic) Best: ANN

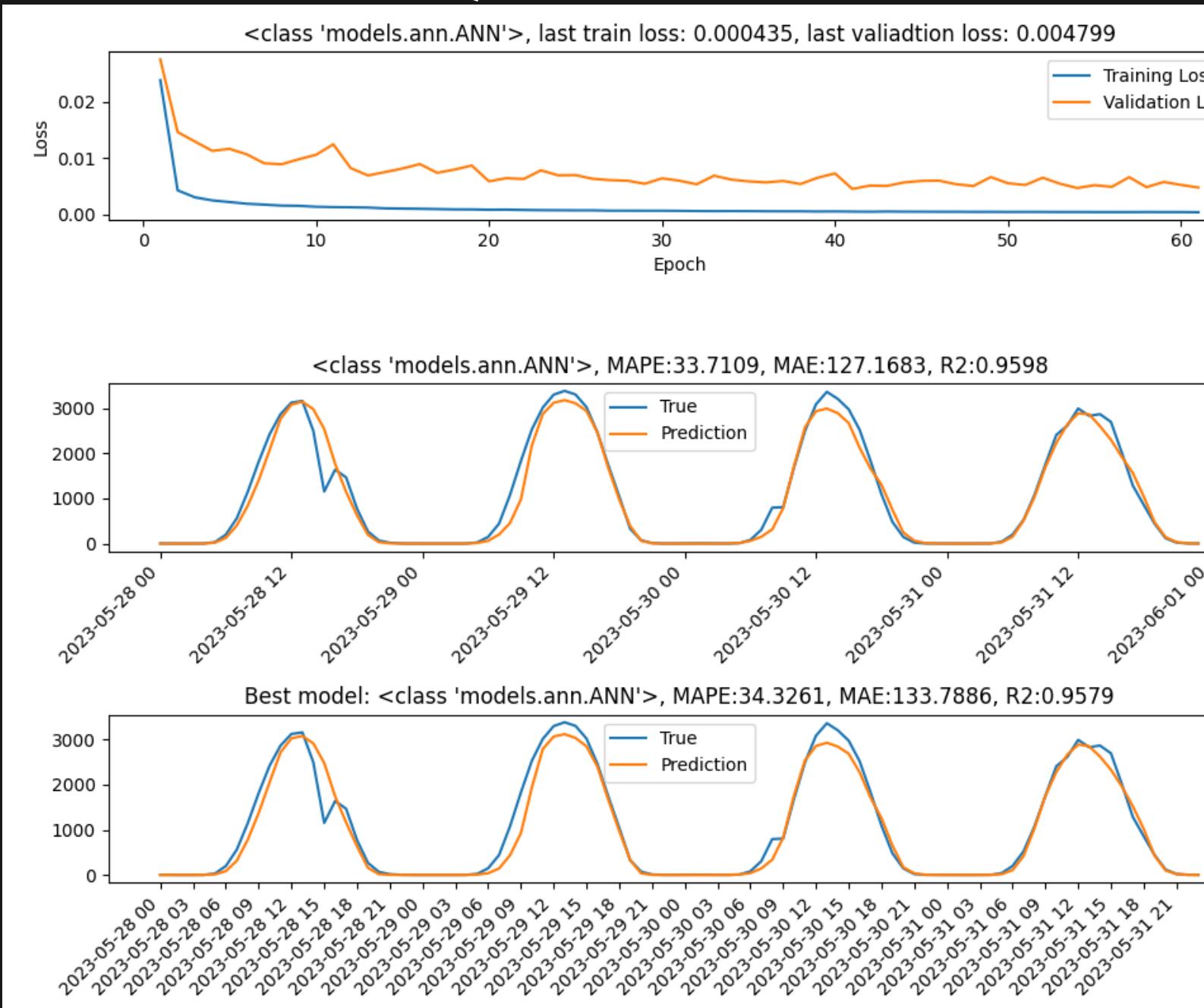


24시간(통으로) Best: ANN

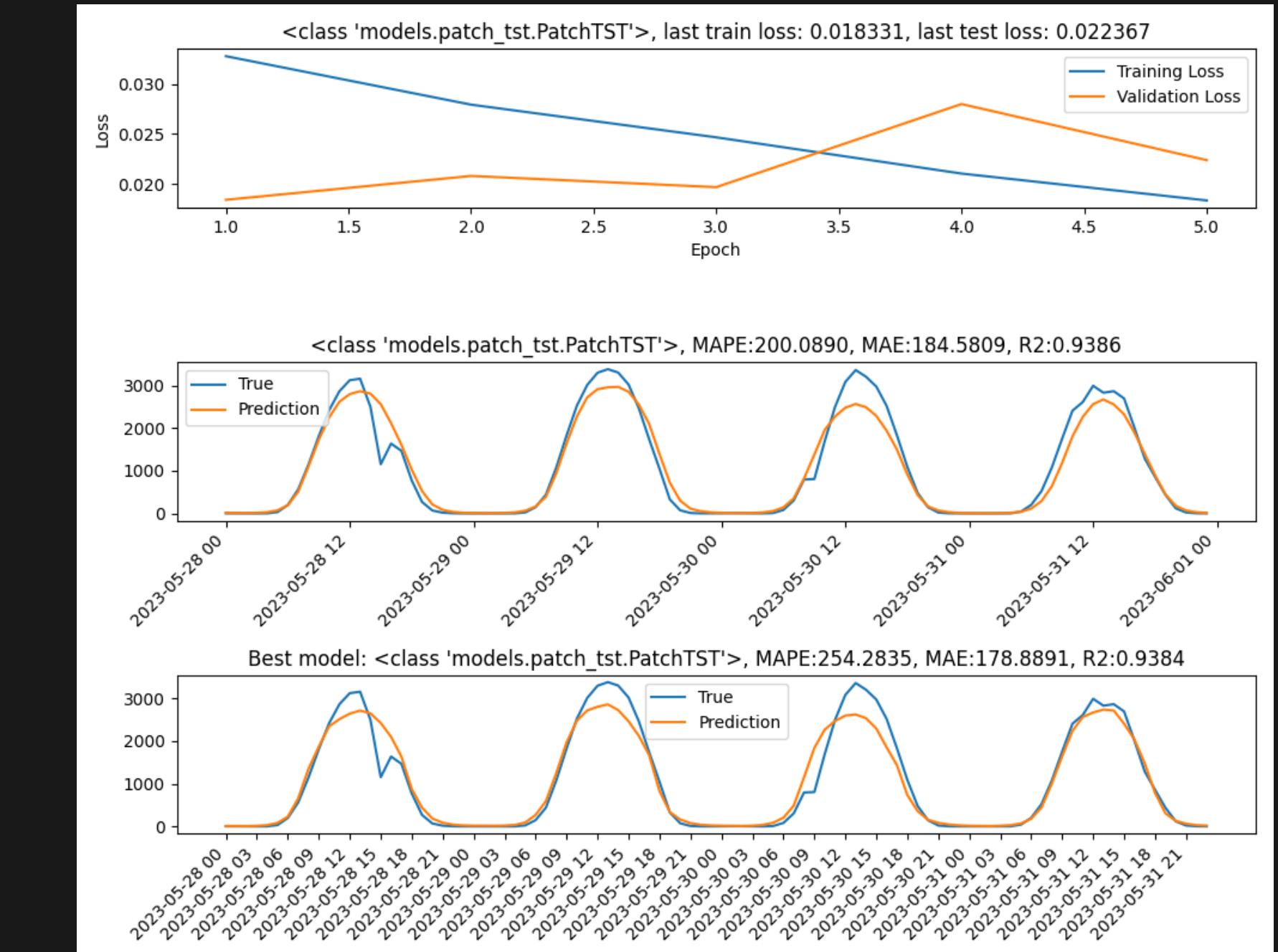


최종 모델 - Production - 96시간

96시간(Dynamic) Best: ANN



96시간(통으로) Best: PatchTST



회고 및 앞으로의 방향

Human Grid Search

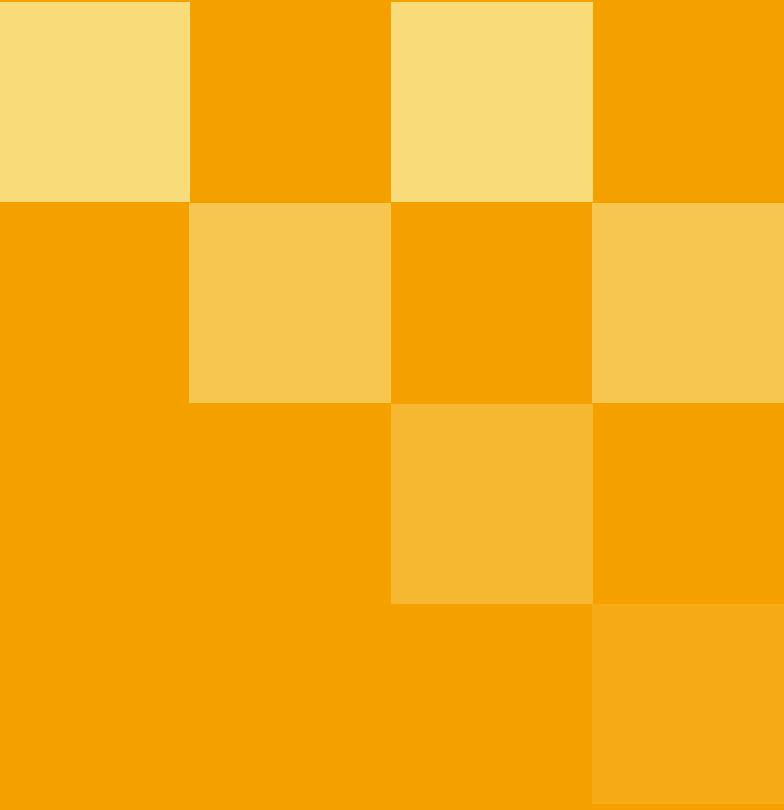
- 한정된 시간 안에서 최대한 많은 실험을 진행
- loss 그래프, prediction 그래프를 눈으로 살펴 보며 파라미터를 조정하는 과정에서 생긴 인사이트

리소스의 한계로 하지 못 한 추가 실험 시도

- window size를 한 달, 분기 단위 등으로 더 크게 늘려보기
- PatchTST 추가 시도
 - reversible instance normalization, strides 조정
 - 다양한 parameter 실험
- LSTM 모델 적용

대회 제출 준비

- 시간 관계 상 생략했던 대회 제출용 모델 준비
- API Submission 준비



Q&A



ESTSOFT