



Full Course for Portfolio

윤동진



CONTENTS

About Me

WeightFormer

Wind Power Forecasting

Enhanced Index Tracking

Court Decision Prediction

CONTENTS

About Me



About Me



열정 있는 동료와 함께 **성장**하는 개발자입니다

- **Name** 윤동진(Dongjin Yoon)
- **Email** djyoon0223@gmail.com
- **Blog** <https://alchemine.github.io>
- **GitHub** <https://github.com/alchemine>
- **Docker Hub** <https://hub.docker.com/search?q=alchemine>
- **Skills**
 - Python (TensorFlow, PyTorch, scikit-learn, PyCaret, spaCy, seaborn, Dask, Numba, CUDA), R
 - Git, Docker, Poetry, SQL Server, MySQL, Sphinx, Spotfire
- **Interests**
 - Statistics, Data visualization, Feature engineering, Refactoring, Parallel computing
- **Work Experiences**
 - (주큐햇지 (2020/04 ~ 2022/02)
 - 머신러닝 투자 알고리즘 개발, Multi-node, multi-GPU distributed/parallel computing
 - (주)디셈버앤팍퍼니자산운용 (2023/02 ~ 2023/06)
 - AI 투자로직 연구 및 금융데이터 빌더 운영
- **Education**
 - 인하대학교 컴퓨터공학과 학사 졸업 (2019/02)
 - 인하대학교 전기컴퓨터공학과 석사 졸업 (2022/02)
윤동진, 이주홍, 최범기, 송재원, “부분복제 지수 상향 추종을 위한 진화 알고리즘 기반 3단계 포트폴리오 선택 양상법 학습”, 스마트미디어저널, 제10권, 제3호, 39-47쪽, 2021년 9월

About Me

1. Experiences

1) 금융 투자 알고리즘 개발

- Enhanced Index Tracking: 유전 알고리즘 기반 지수 상향 추종 알고리즘
 - Project Managing
 - Scheme Definition
 - EDA / Algorithm Modeling
 - Multi-node, multi-gpu distributed parallel computing
 - WeightFormer: BERT 기반 포트폴리오 예측 알고리즘
 - EDA / Algorithm Modeling
- Sequence Diagram(PlantUML), WBS
SQL Server, MySQL
Python(Dask, Numba, CUDA, scikit-learn)
Dask, Numba, CUDA
Python(PyTorch, Transformers, BERT)

2) 경진대회 참가

- Wind Power Forecasting: 풍력 발전량 예측 알고리즘
 - EDA / Algorithm Modeling
 - Court Decision Prediction: 법원 판결 예측 알고리즘
 - EDA / Algorithm Modeling
- Python(PyTorch, Transformers, GRU)
Python(PyTorch, spaCy, Transformers, T5, vicuna, LoRA)

3) 개발환경 구축 및 배포

- base: Prepared CUDA based Docker Image for Machine Learning Project
 - Dockerhub
 - analysis-tools: PyPI Analysis tools package for Machine Learning Project
 - PyPI
- <https://hub.docker.com/repository/docker/alchemine/base>
<https://pypi.org/project/analysis-tools>

—○—

CONTENTS

WeightFormer

WeightFormer

1. 프로젝트 소개

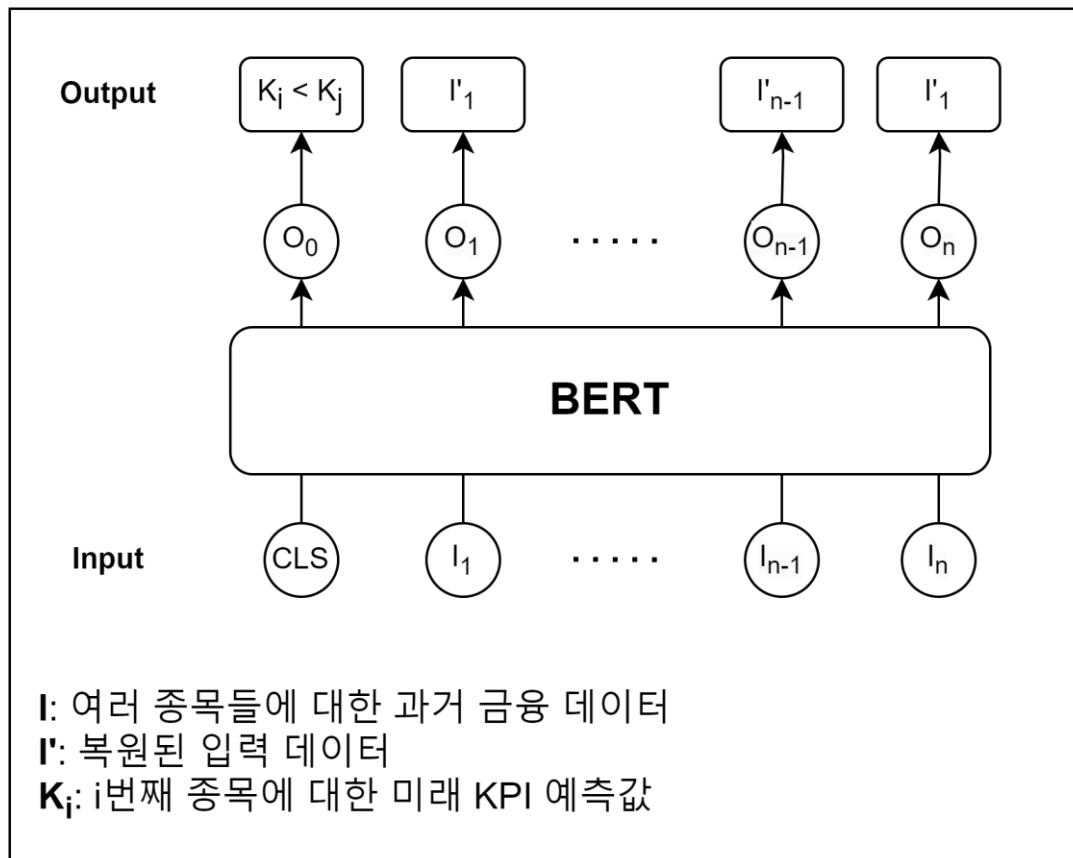
NLP에서 좋은 성과를 보여준 Transformer(BERT)의 아이디어를 비시계열 금융 데이터에 적용한 파일럿 프로젝트

2. 상세 설명

- 과거 비시계열 금융 데이터를 입력으로 하여, 미래의 최적 투자비율(포트폴리오)을 예측
- BERT에서 Language Modeling을 위해 사용한 두 가지 pretraining task—NSP(Next Sentence Prediction), MLM(Masked Language Modeling)—의 의미를 비시계열 금융 데이터에 적용할 수 있을지 확인하는 것이 목적
- 최종결과를 얻기 전에 프로젝트가 중단되어, 모델 구현과 학습 과정에 집중하여 설명

2. 상세 설명(continued)

1) 문제 정의



1. 입력 데이터

여러 종목들에 대한 과거 금융 데이터 (multi factor)
(sequence: 종목)

2. NSP \rightarrow r2s (rank2stock)

Sequence 중 2개의 종목(i, j)에 대한 미래 KPI 예측값을 비교

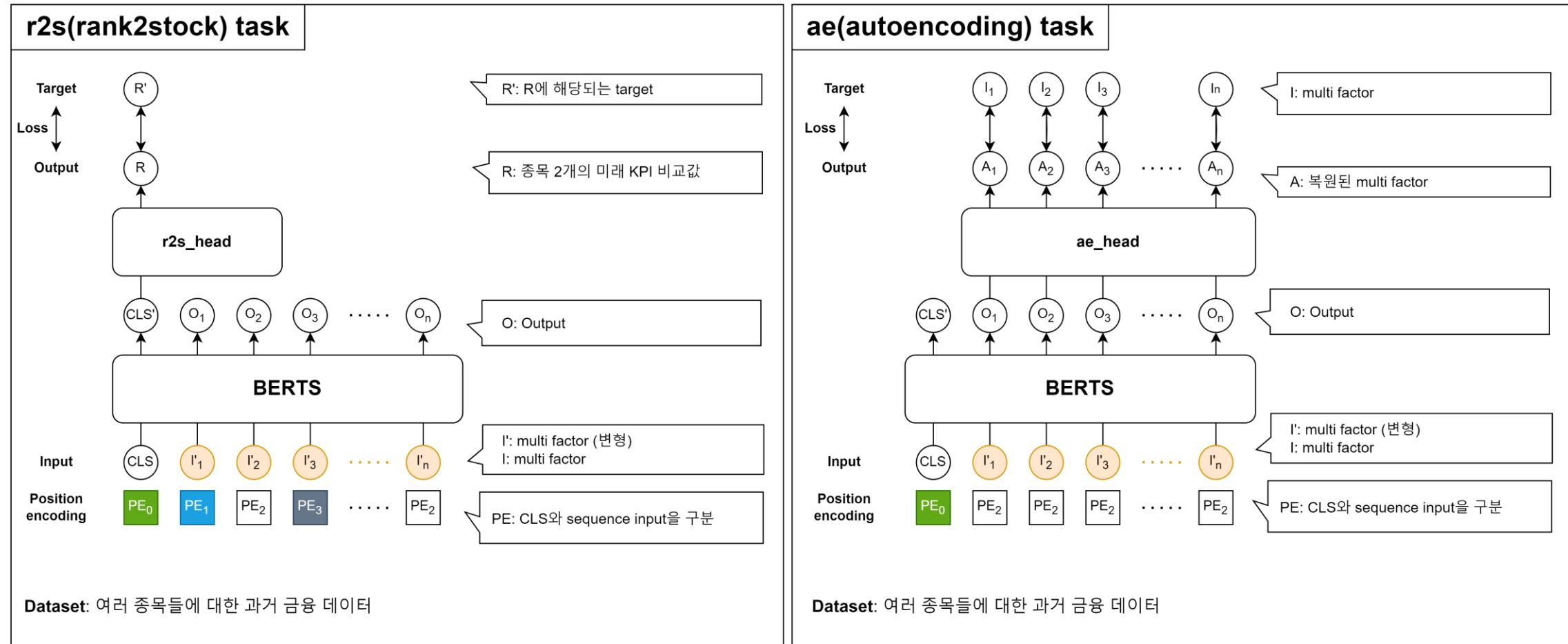
3. MLM \rightarrow ae (autoencoding)

Masking된 입력값들에 대한 출력값으로부터 입력값을 복원

WeightFormer

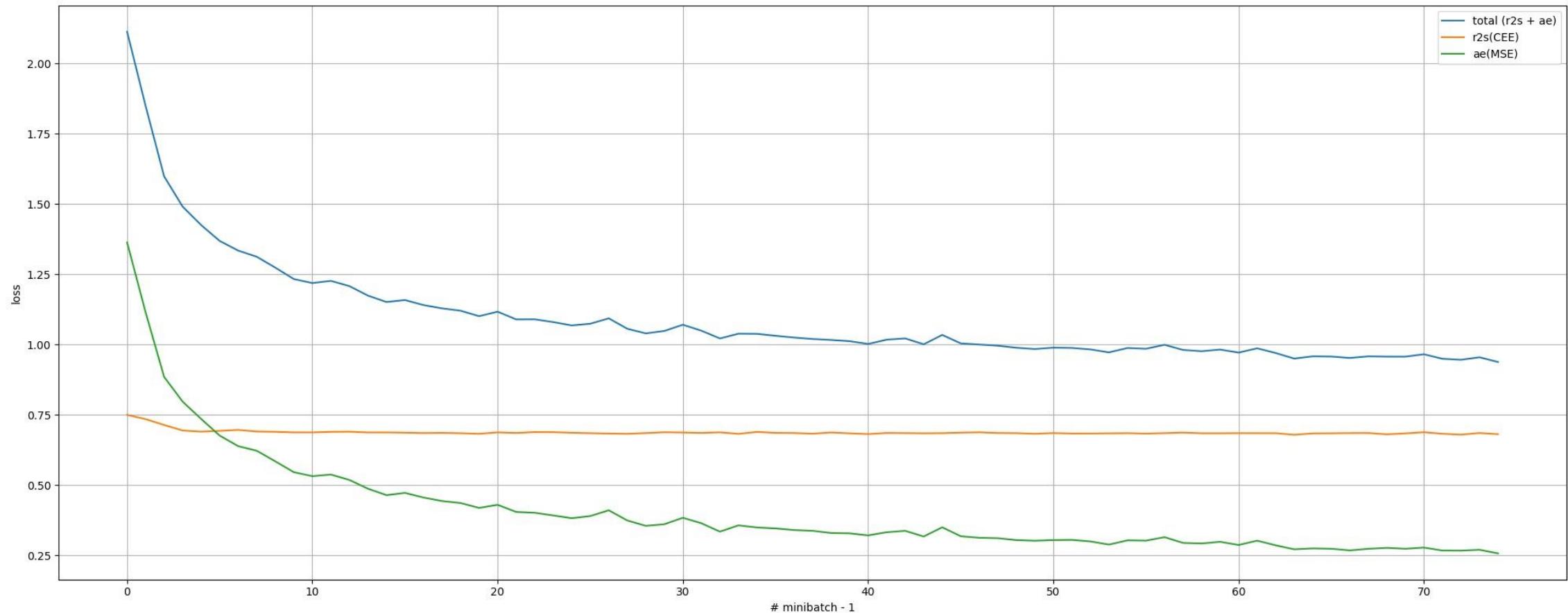
2. 상세 설명(continued)

2) 모델 구현 (base)



2. 상세 설명(continued)

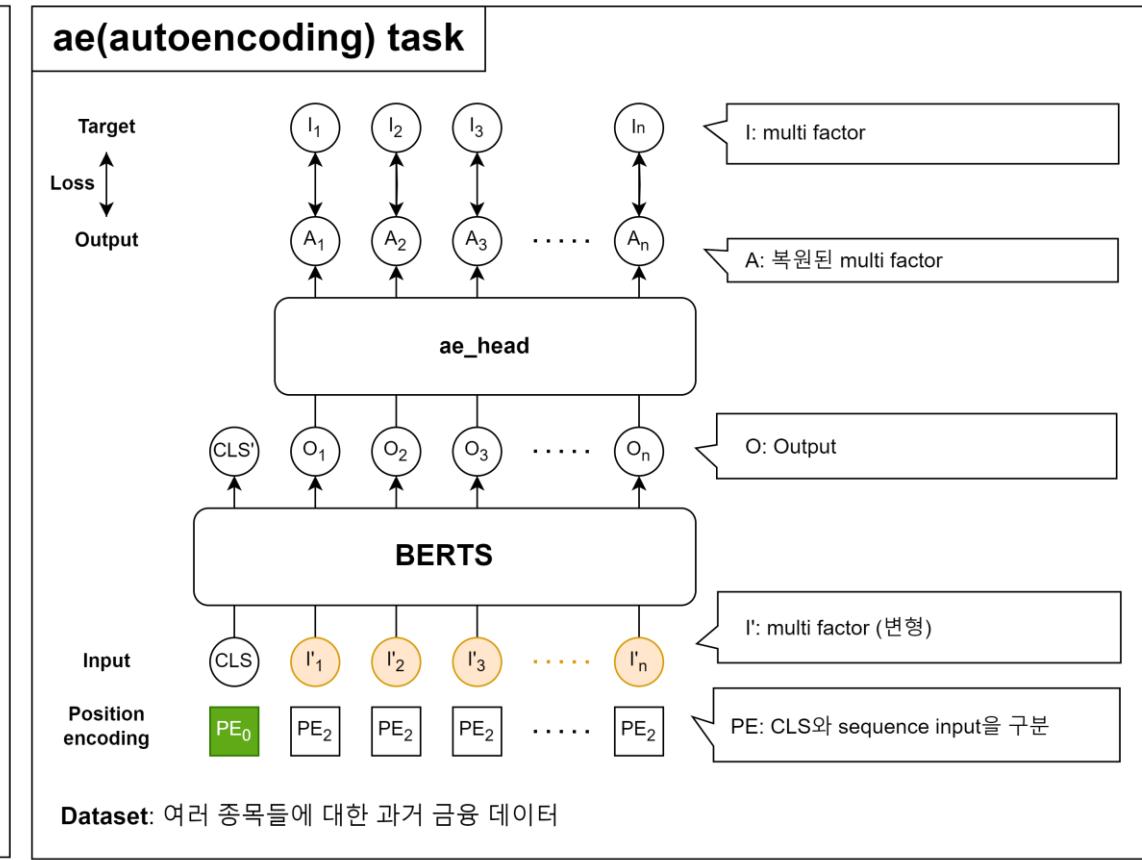
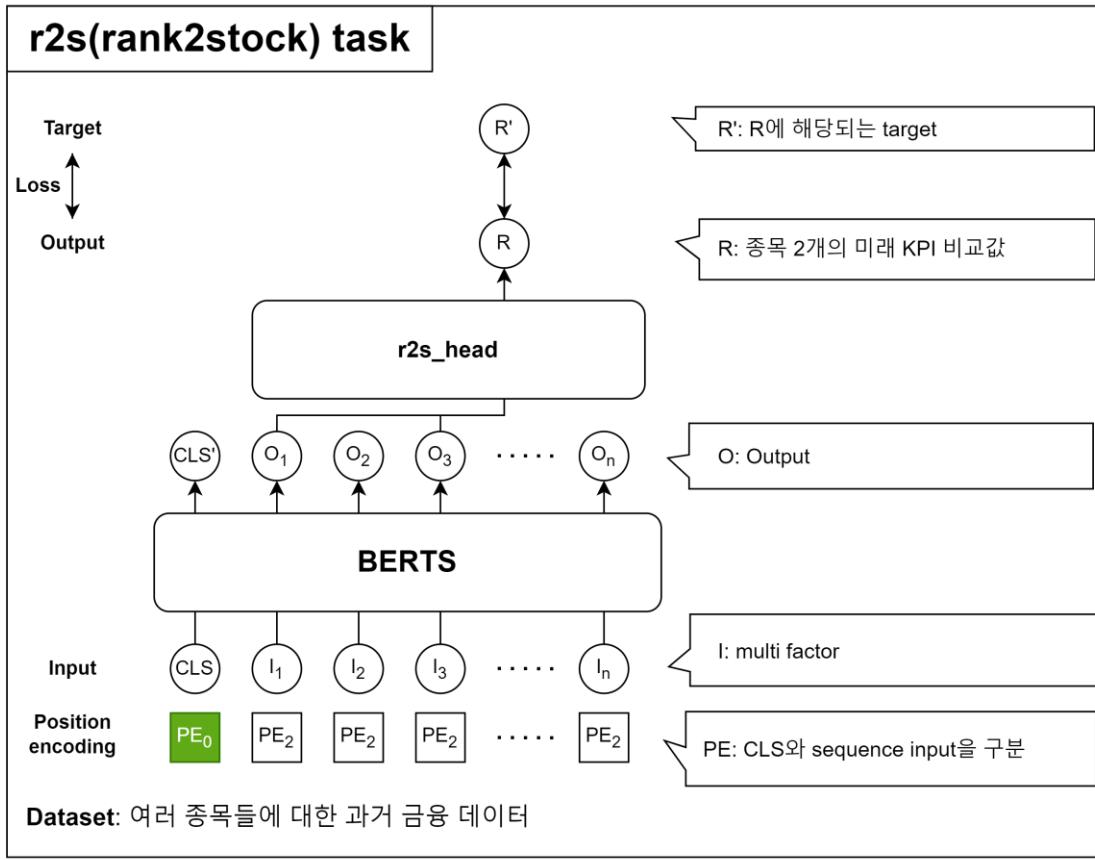
2) 모델 구현 (base)



WeightFormer

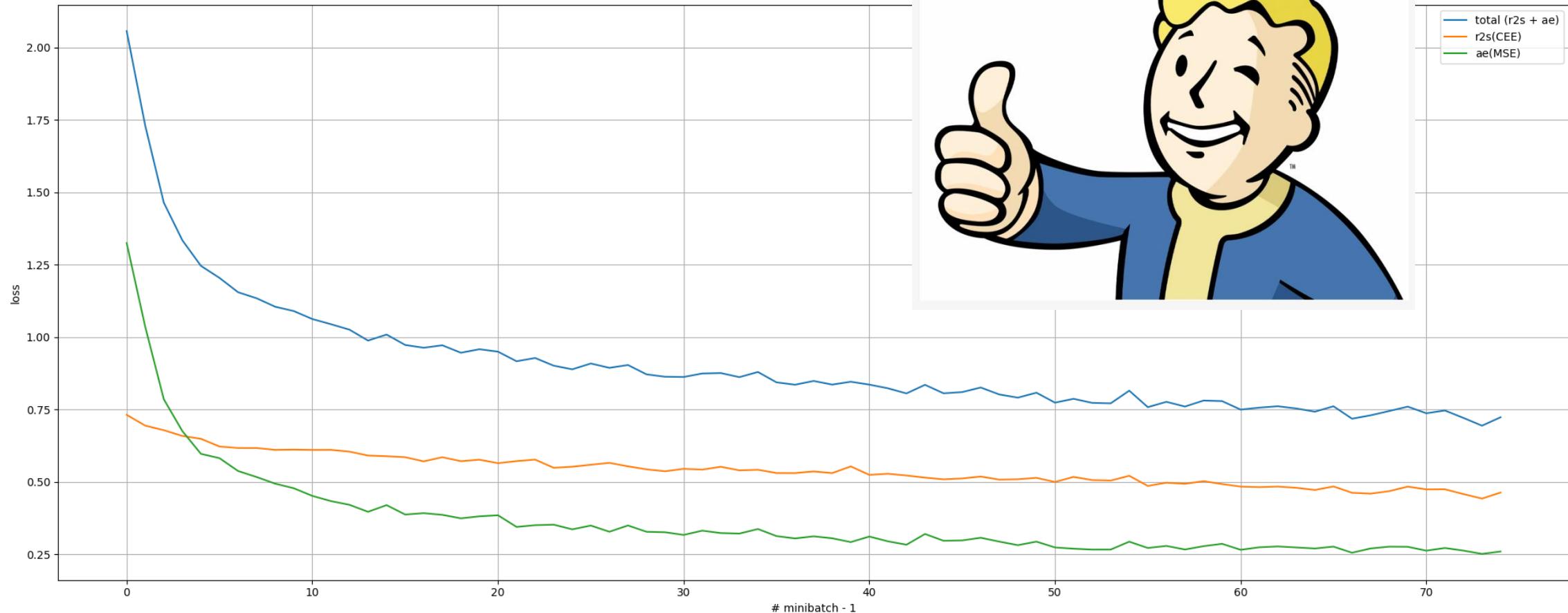
2. 상세 설명(continued)

3) 모델 구현 (modified)



2. 상세 설명(continued)

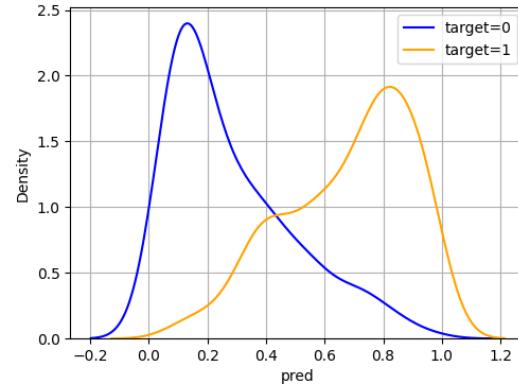
3) 모델 구현 (modified)



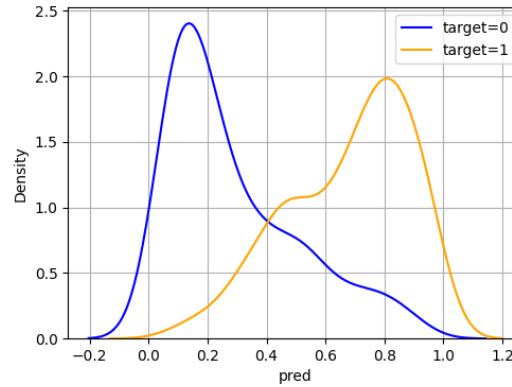
2. 상세 설명(continued)

3) 모델 구현 (modified)

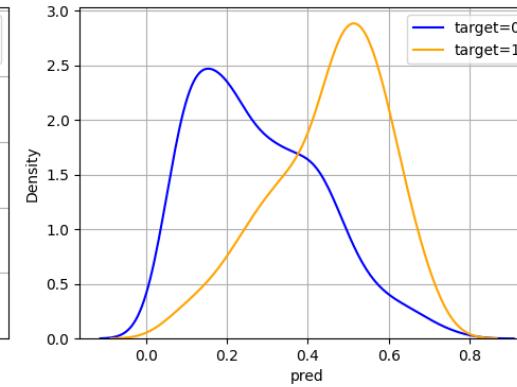
누적수익률



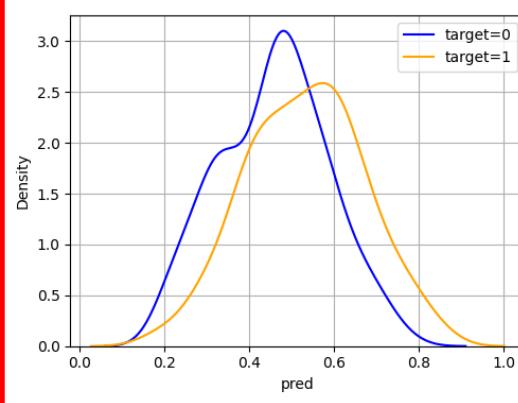
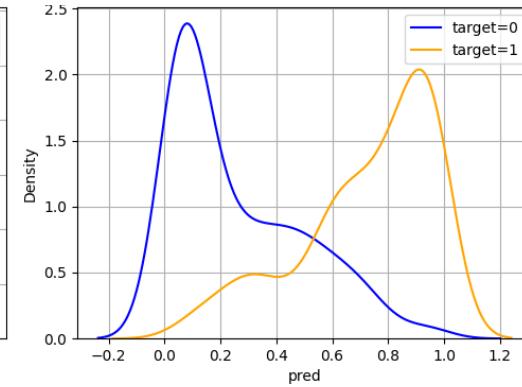
Sharpe Ratio



연평균수익률

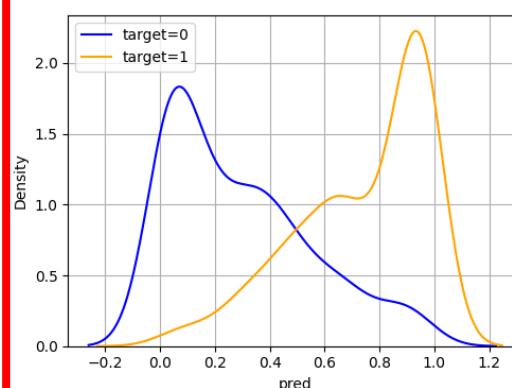


변동성

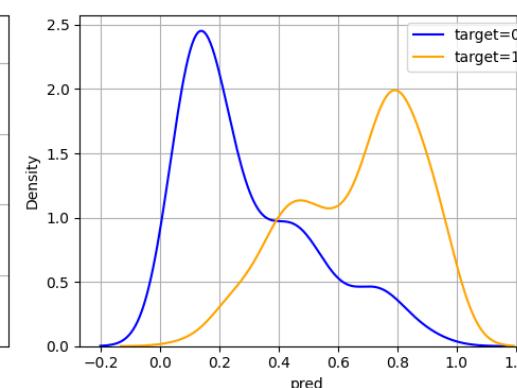


MDD

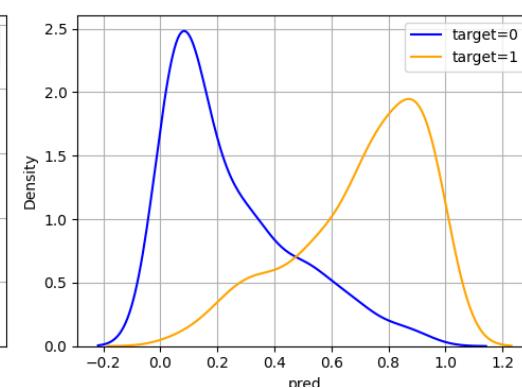
평균 기반이 아닌 불연속적인 특성으로 인해 학습이 어려움



VaR



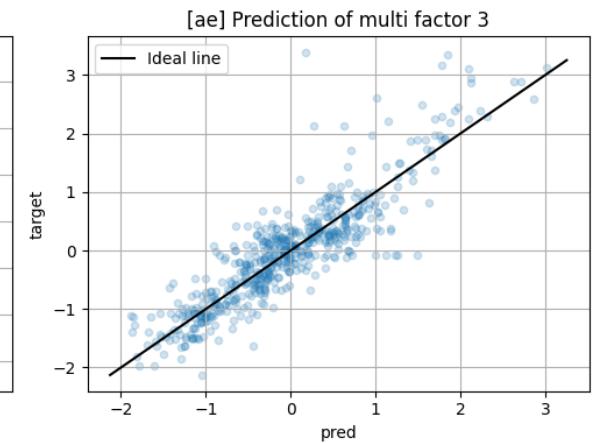
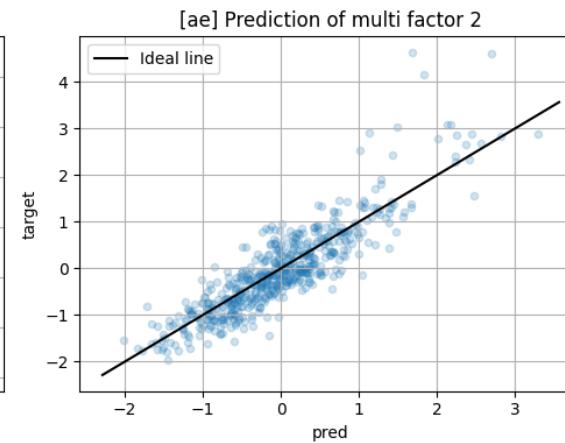
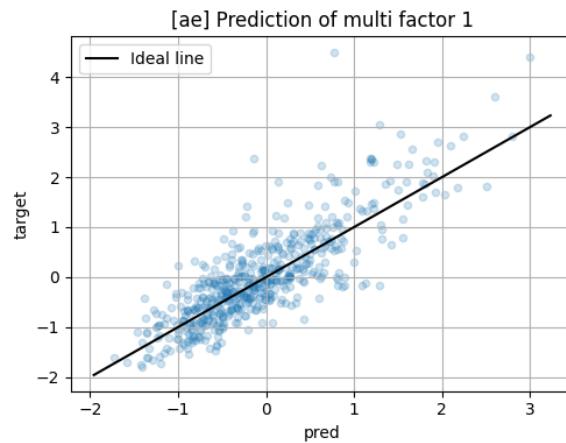
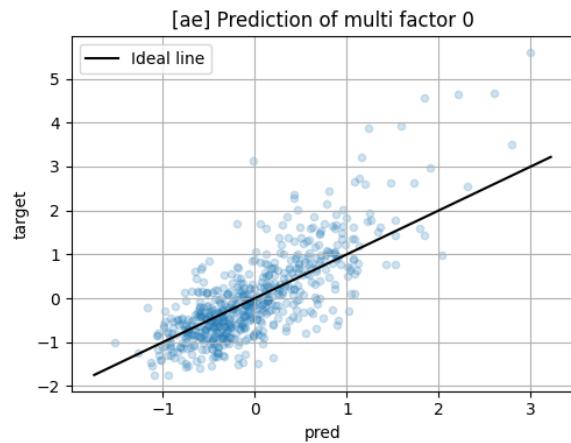
Information Ratio



추적 오차

2. 상세 설명(continued)

3) 모델 구현 (modified)



3. 결론

- Language modeling에서 word(token) 간의 관계를 모델링하기 위해 사용된 기법들을 금융 데이터에서 종목 간의 관계를 모델링하기 위해 변경/적용해본 결과, 학습이 가능한 것을 확인할 수 있었음
- 그러나, 학습 데이터에 대한 결과만으로는 실질적으로 유의미한 결과인지 검증하기 어려움
- 시장 데이터 등을 학습한 모델들과 fusion한 후, 최종적으로 포트폴리오를 출력하는 decoder 등에 대한 구상/구현 작업을 통해 실질적으로 본 프로젝트의 유의미성을 검증할 수 있을 것으로 기대

CONTENTS

Wind Power Forecasting



Wind Power Forecasting

1. 프로젝트 소개

풍력 발전 터빈의 위치, 온도와 풍속 등의 시계열 데이터를 학습하여 미래의 유효전력을 예측하는 프로젝트

- GitHub <https://github.com/alchemine/spatio-temporal>
- Dacon <https://dacon.io/competitions/official/235926/overview/description>

2. 상세 설명

- 시계열의 특성을 유지하기 위하여 데이터의 연속성을 보장하고 많은 결측치와 에러값(전체 데이터의 23%)을 처리하는 것이 핵심
- 데이터를 집계된(aggregated) 수치로만 바라보는 것이 아니라 ‘왜 이런 값이 나왔을까?’ 데이터의 근본에 대하여 깊게 고찰하는 시간을 가질 수 있었고, 데이터를 가공하기 위해 다양한 기법들을 시도하여 많은 것들을 배우고 성장할 수 있었던 계기가 되었음
- 데이터 분석을 통해 인사이트를 얻고 적용하는 방식에 대하여 설명

1) 문제 정의

사용되는 feature는 다음과 같습니다.

TurbID	- Wind turbine ID, 발전기 ID
Day	- Day of the record, 날짜
Tmstamp	- Created time of the record, 시간 (10분 단위)
Wspd	- The wind speed recorded by the anemometer, 풍속(m/s)
Wdir	- wind direction, 터빈이 바라보는 각도와 실제 바람 방향 각도 차이(°)
Etmp	- Temperature of the surrounding environment, 외부 온도(°C)
Itmp	- Temperature inside the turbine nacelle, 터빈 내부 온도(°C)
Ndir	- Nacelle direction, i.e., the yaw angle of the nacelle, 터빈이 바라보는 방향 각도(°)
Pab	- Pitch angle of blade, 터빈 당 3개의 날이 있으며 각각의 각도가 다름(°)
Prtv	- Reactive power, 무효전력 : 에너지원을 필요로 하지 않는 전력(kW)
Patv	- Active power(target variable), 유효전력 : 실제로 터빈을 돌리는 일을 하는 전력(kW)

주어진 문제는 총 134개의 터빈(TurbID)에 대하여 미래 2일 후까지의 유효전력(Patv)를 예측하는 것입니다.

데이터는 10분 단위로 sampling 되어 134x288개의 유효전력에 대한 MSE와 MAE의 평균을 통해 평가됩니다.

- 학습 데이터 Day: 1 ~ 243일
- 테스트 데이터 Day: 244, 245일

1) 문제 정의 (continued)

다음의 4가지 조건에 해당하는 경우(abnormal), 해당 데이터에 대한 예측값은 평가 지표에 합산되지 않습니다.

조건 1. Wspd > 2.5 이고 Patv <= 0

바람이 부는데 발전량이 없는 경우

조건 2. Pab1 > 89 혹은 Pab2 > 89 혹은 Pab3 > 89

바람과 날개의 각도 차이가 커 바람의 영향을 받지 못하는 경우

조건 3. Wdir < -180 혹은 Wdir > 180 혹은 Ndir < -720 혹은 Ndir > 720

허용된 기기의 정상 범위를 넘어가는 경우

조건 4. Patv가 null

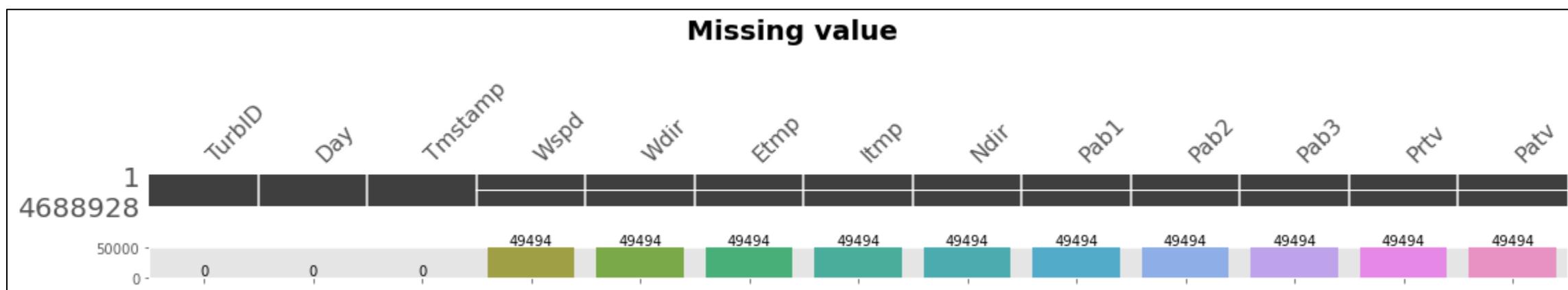
Target이 존재하지 않는 경우

Wind Power Forecasting

2) Explanatory Data Analysis (Training data)

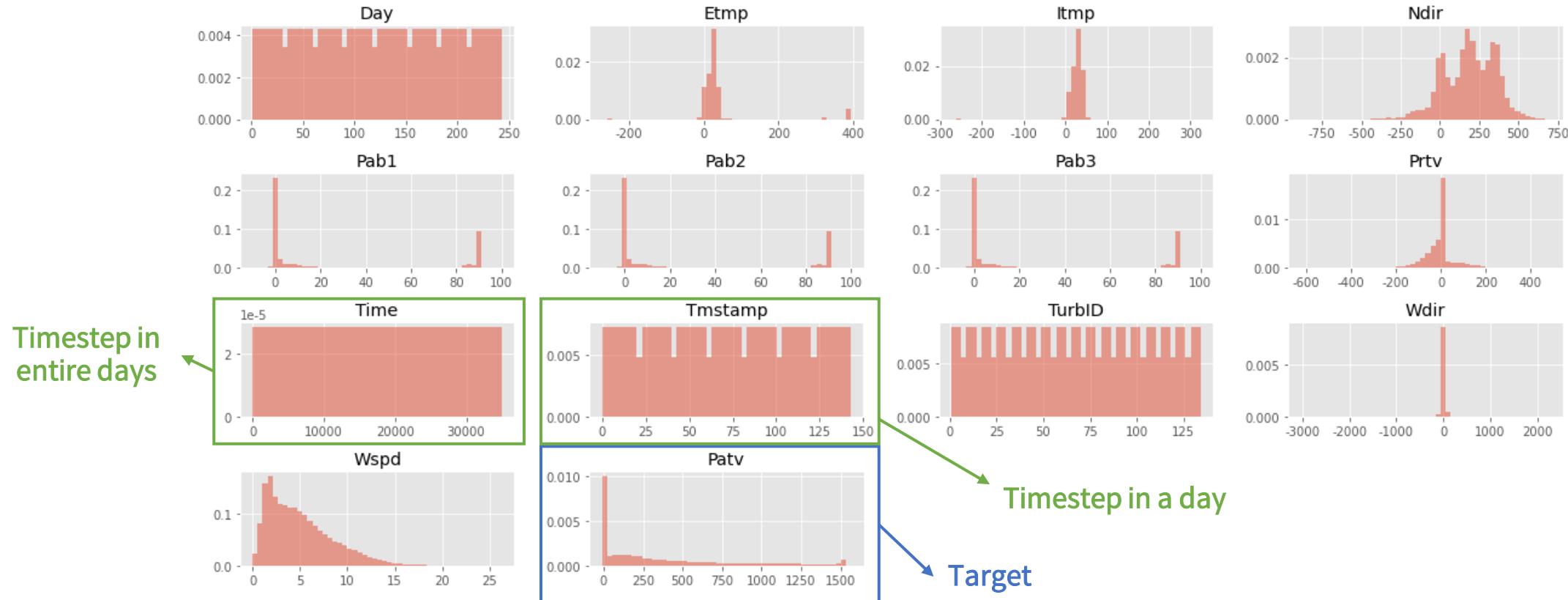
	TurbID	Day	Tmstamp	Wspd	Wdir	Etmp	Itmp	Ndir	Pab1	Pab2	Pab3	Prtv	Patv
0	1	1	00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	1	00:10	6.17	-3.99	30.73	41.80	25.92	1.00	1.00	1.00	-0.25	494.66
2	1	1	00:20	6.27	-2.18	30.60	41.63	20.91	1.00	1.00	1.00	-0.24	509.76
3	1	1	00:30	6.42	-0.73	30.52	41.52	20.91	1.00	1.00	1.00	-0.26	542.53
4	1	1	00:40	6.25	0.89	30.49	41.38	20.91	1.00	1.00	1.00	-0.23	509.36
...
4727227	134	243	23:10	10.98	-1.96	-5.11	-0.67	345.57	8.82	8.82	8.82	136.49	1152.60
4727228	134	243	23:20	11.82	-3.18	-5.46	-0.54	345.57	13.87	13.87	13.87	84.43	681.65
4727229	134	243	23:30	11.91	-1.42	-5.21	-0.42	345.57	10.69	10.69	10.69	145.72	1118.35
4727230	134	243	23:40	11.86	-0.95	-5.40	-0.38	345.57	13.94	13.94	13.94	89.56	683.49
4727231	134	243	23:50	11.72	0.04	-5.23	-0.37	345.57	10.90	10.90	10.90	120.18	1026.93

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4688928 entries, 0 to 4727231
Data columns (total 13 columns):
 #   Column      Dtype  
 --- 
 0   TurbID     int64  
 1   Day         int64  
 2   Tmstamp    object  
 3   Wspd        float64 
 4   Wdir        float64 
 5   Etmp        float64 
 6   Itmp        float64 
 7   Ndir        float64 
 8   Pab1       float64 
 9   Pab2       float64 
 10  Pab3       float64 
 11  Prtv        float64 
 12  Patv        float64 
dtypes: float64(10), int64(2), object(1)
memory usage: 500.8+ MB
```



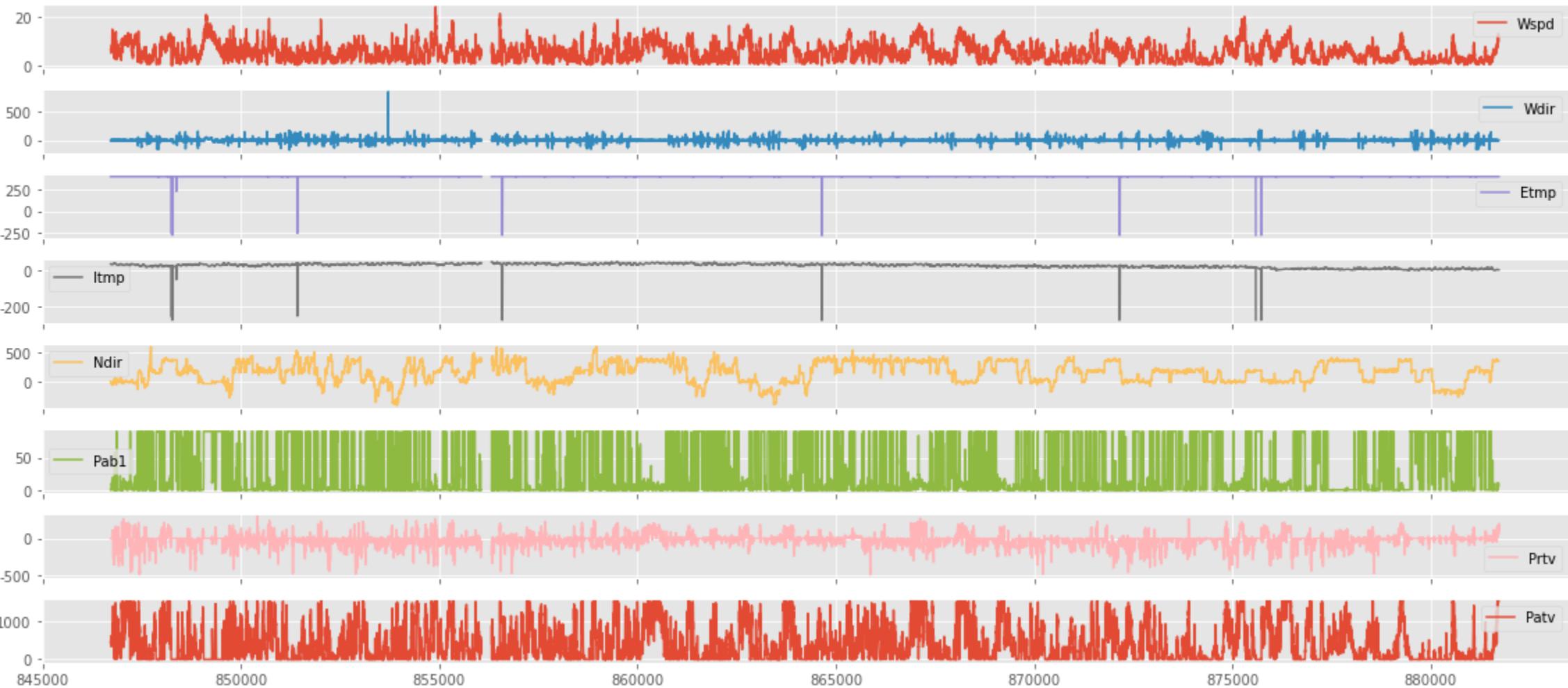
Wind Power Forecasting

Features



Wind Power Forecasting

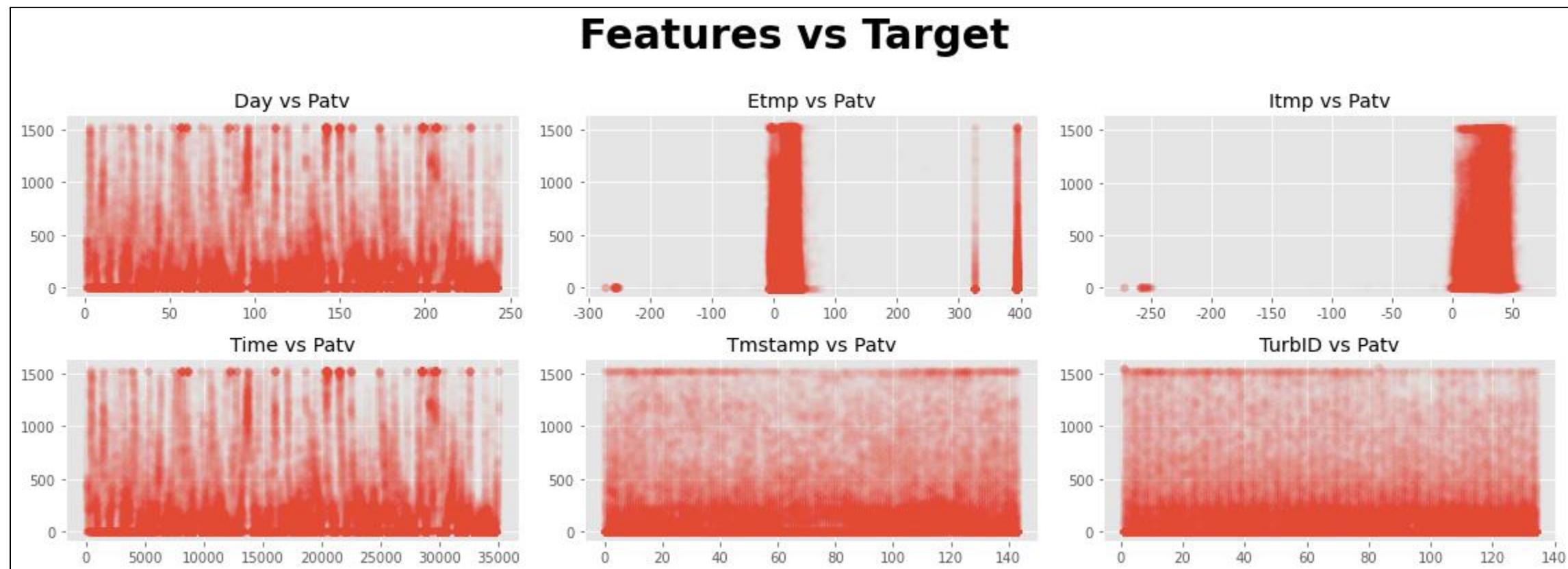
Features of TurbID=25



Wind Power Forecasting

① Day, Ettmp, Itmp, Time, Tmstamp, TurbID

- 뚜렷한 관계가 보이지 않음
- Ettmp, Itmp: 이상치 처리가 필요



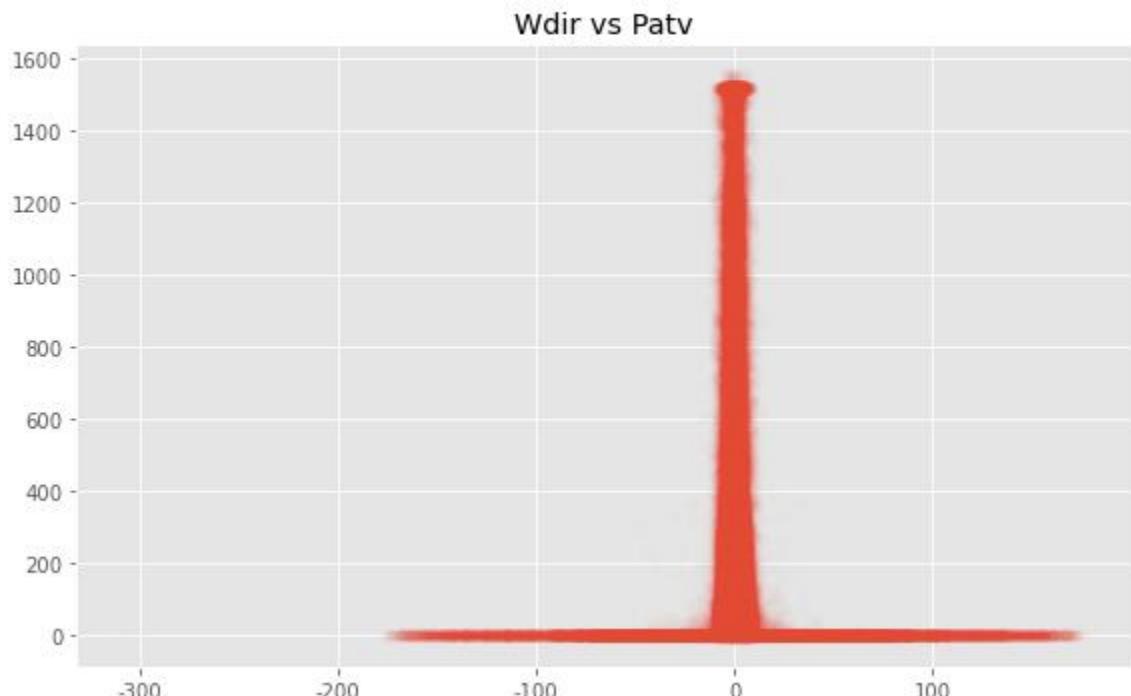
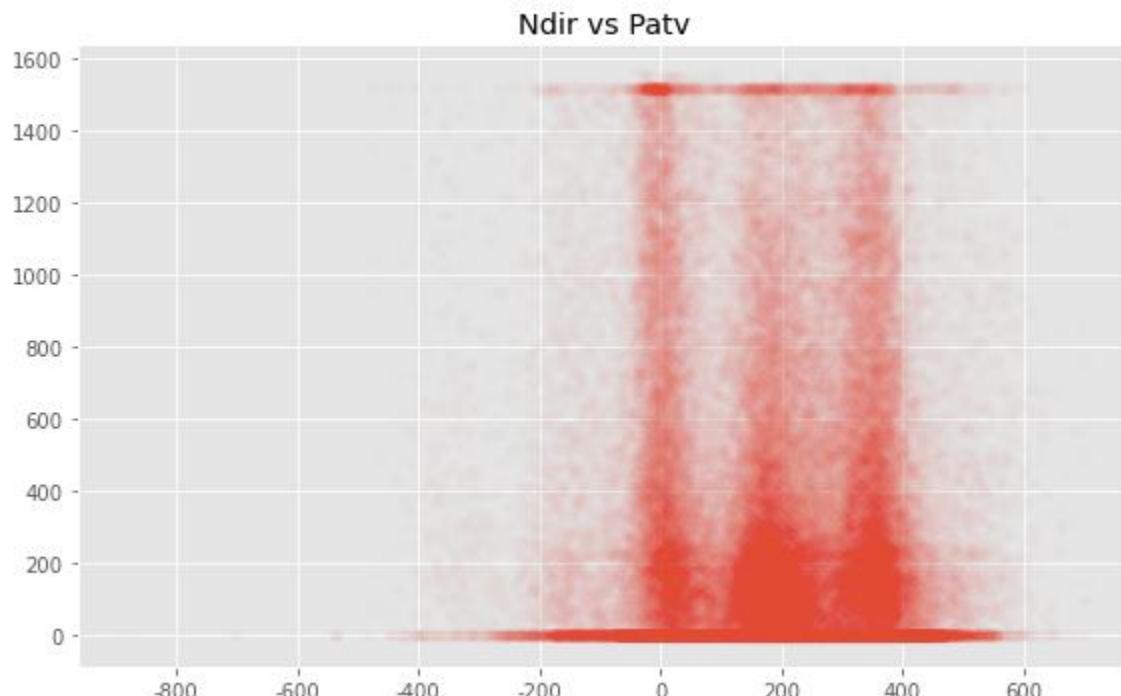
② Ndir

- 0도, 180도, 360도에서 기둥이 보임 (180도의 배수인 것처럼 보이나, -180도는 기둥이 없기 때문에 음수에 대한 구분이 필요)

③ Wdir

- 절댓값이 10도 이상이면, Patv \approx 0

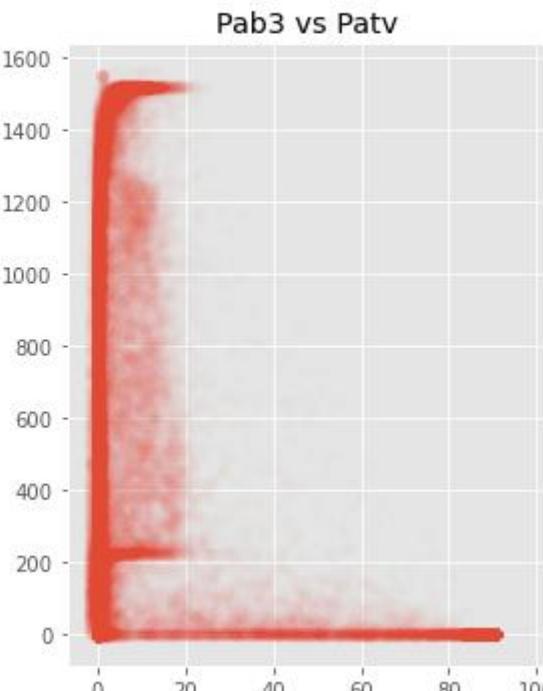
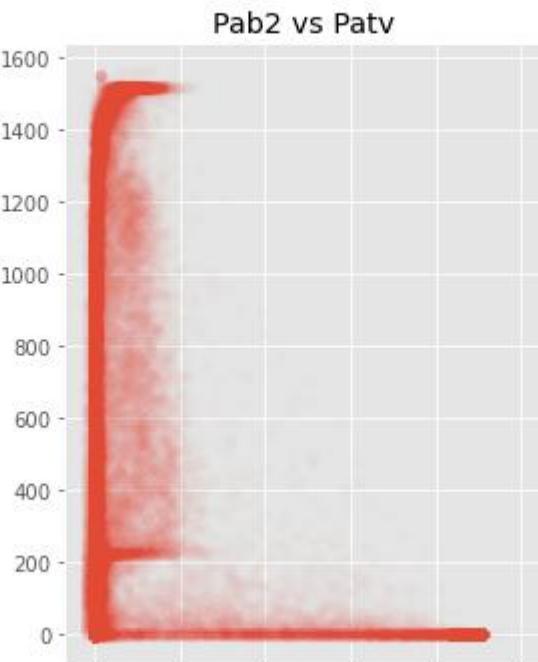
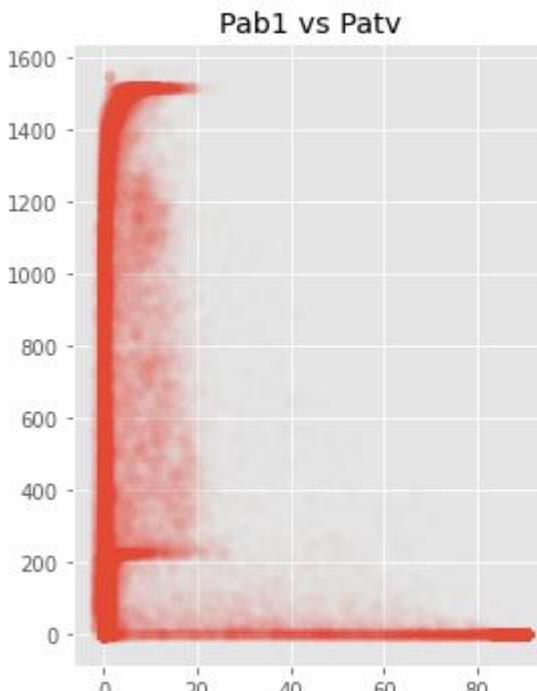
Features vs Target



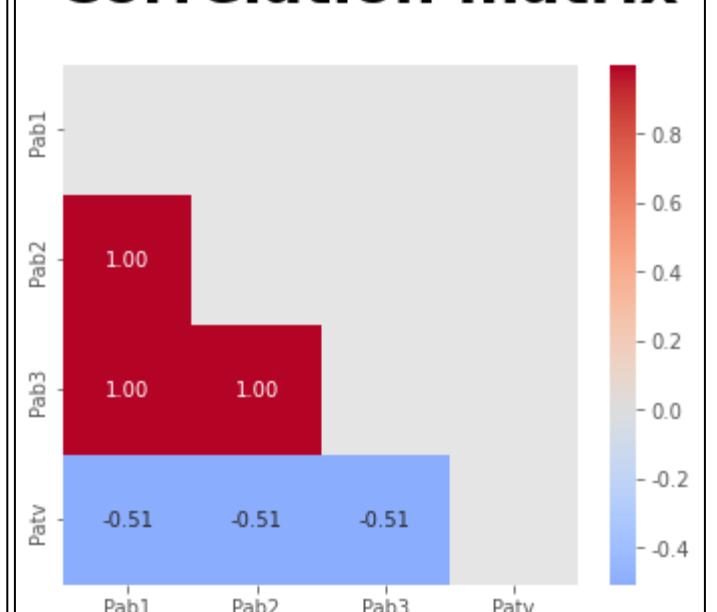
④ Pab1, Pab2, Pab3

- 날개 3개의 각도로 서로 간의 correlation이 1이기 때문에 평균값 Pab를 사용
- 0.03도, 20도를 기준으로 층이 보임

Features vs Target



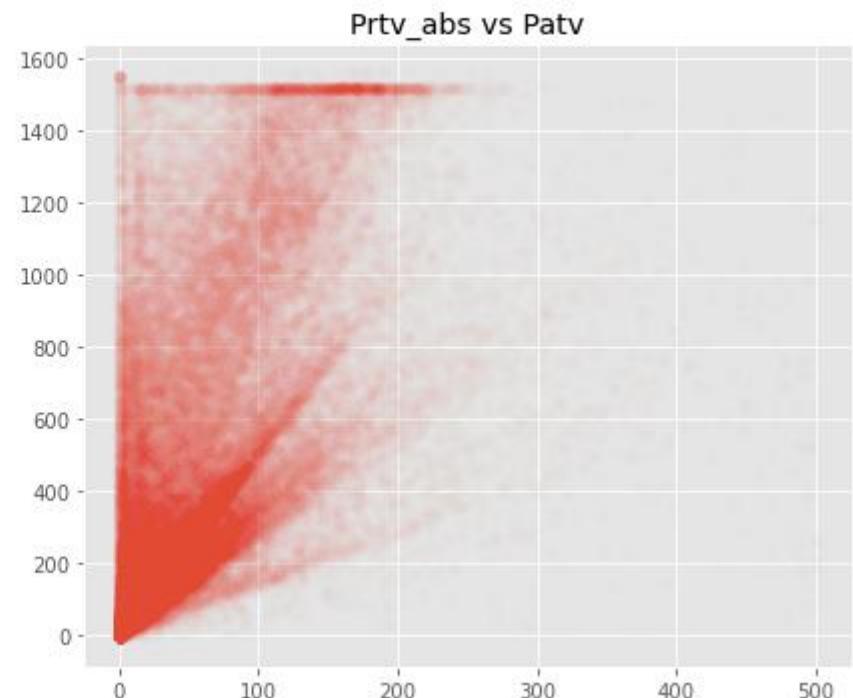
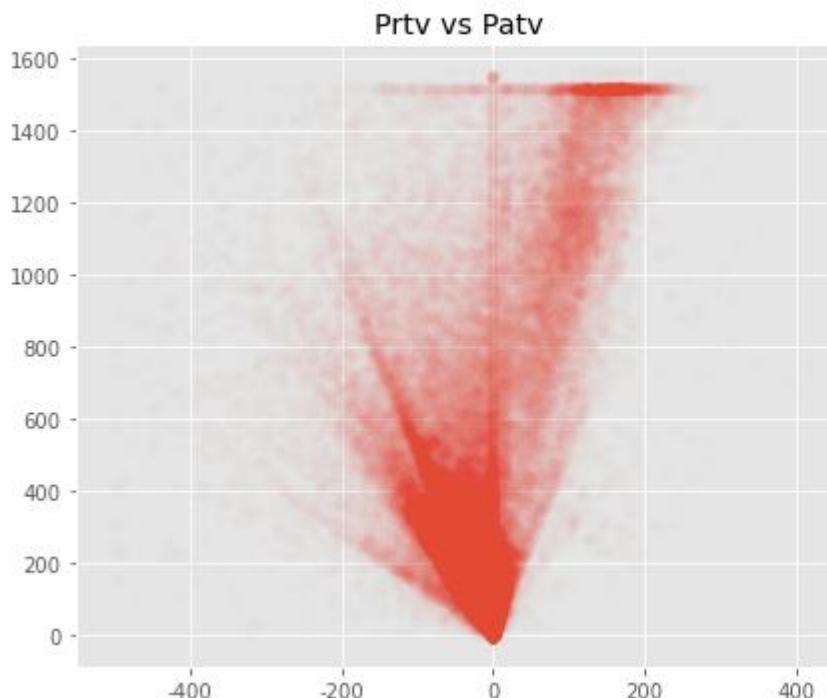
Correlation matrix



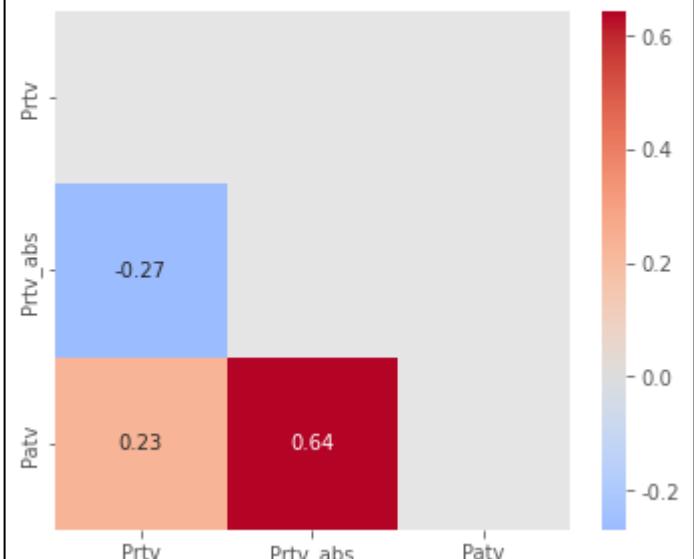
⑤ Prtv

- 음수인 경우와 양수인 경우에 Patv와 다른 관계를 가지는 것 같음
- 절대값이 Patv와 강한 양의 상관관계를 가짐(0.64)

Features vs Target

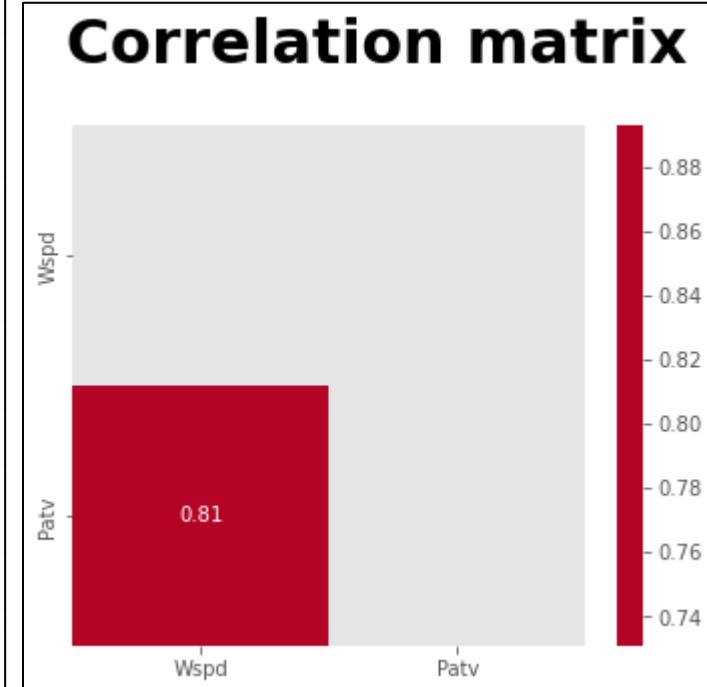
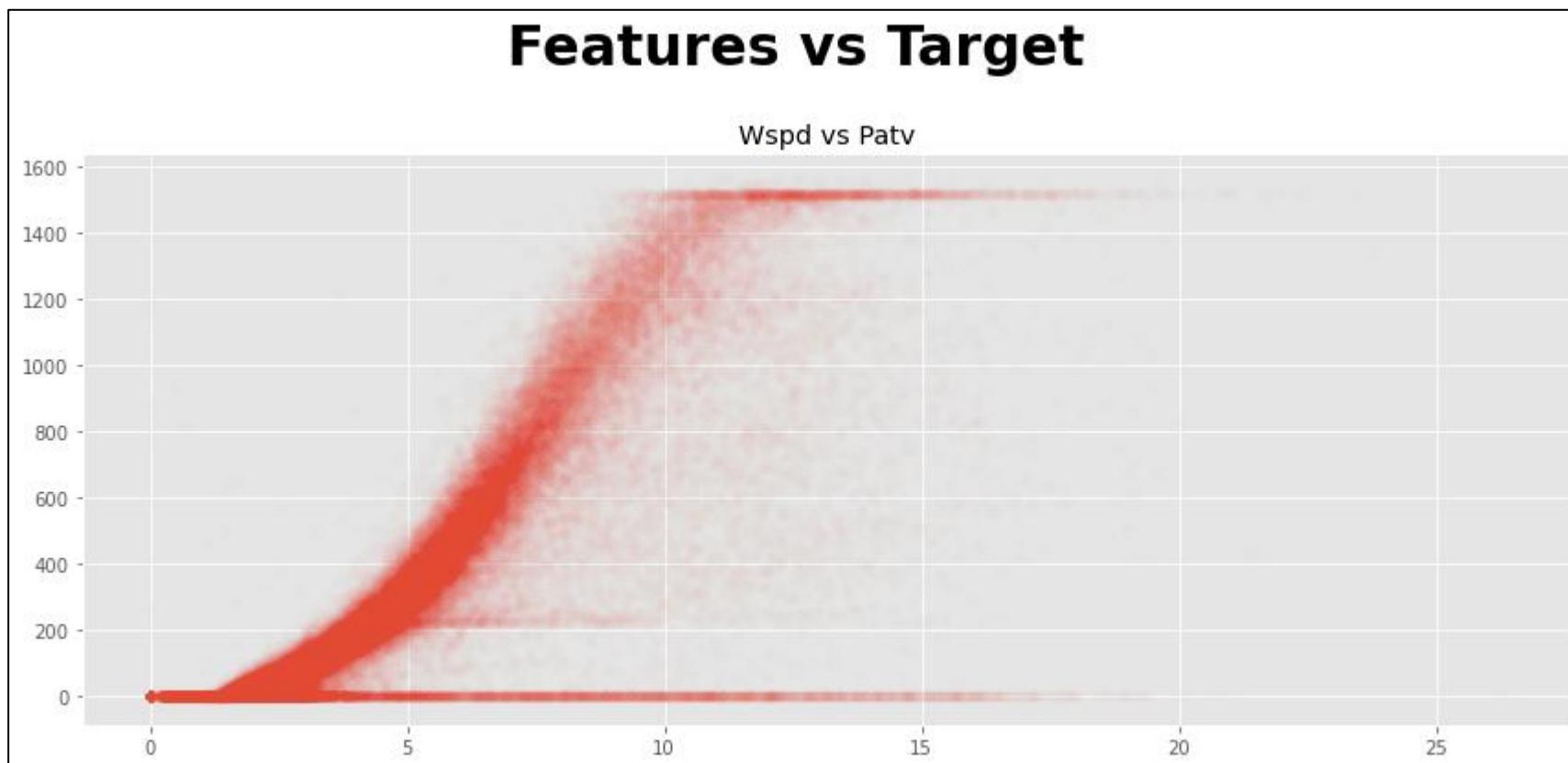


Correlation matrix



⑥ Wspd

- Patv와 강한 양의 상관관계를 가짐(0.81)
- 1m/s 이하이면, Patv = 0
- 1m/s 이상일 때, 다른 요인으로 인해 Patv = 0 인 데이터를 구분할 수 있다면 Wspd의 correlation을 더 높일 수 있음



3) Preprocessing

① Mark abnormal Patv

조건 1. Wspd > 2.5 이고 Patv <= 0

조건 2. Pab1 > 89 혹은 Pab2 > 89 혹은 Pab3 > 89

조건 3. Wdir < -180 혹은 Wdir > 180 혹은 Ndir < -720 혹은 Ndir > 720

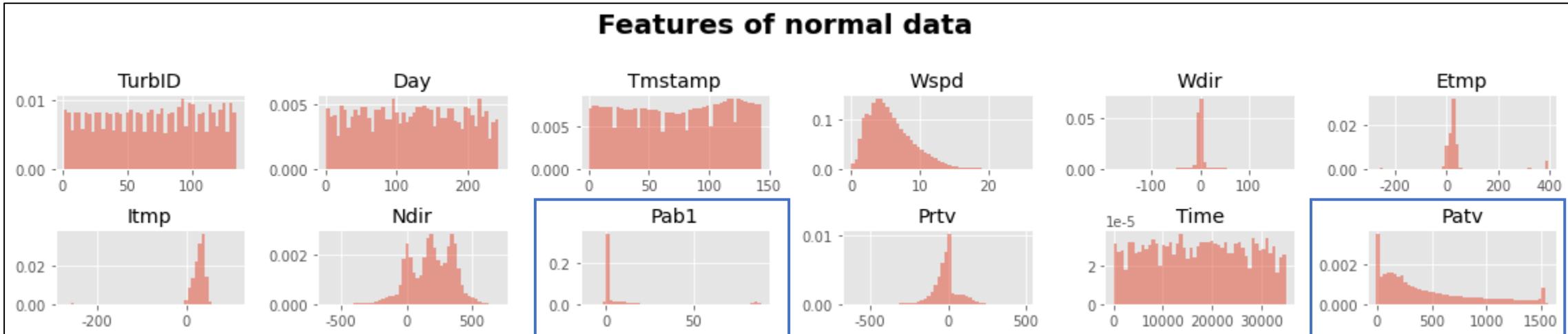
조건 4. Patv가 null

하나라도 해당된다면 Abnormal = 1 o.w. 0

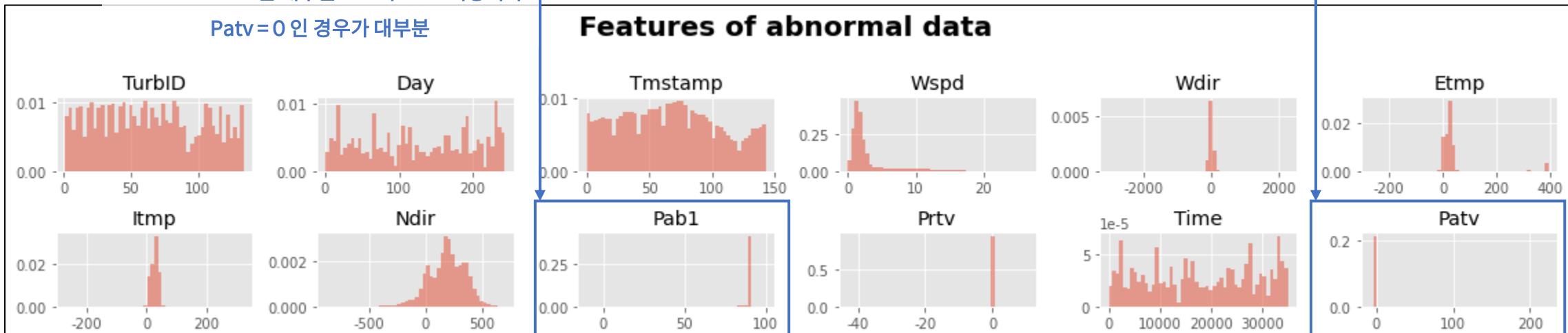
	TurbID	Day	Tmstamp	Wspd	Wdir	Etmp	Itmp	Ndir	Pab1	Pab2	Pab3	Prtv	Time	Patv	Abnormal
0	1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	NaN	1
1	1	1	1	6.17	-3.99	30.73	41.80	25.92	1.00	1.00	1.00	-0.25	2	494.66	0
2	1	1	2	6.27	-2.18	30.60	41.63	20.91	1.00	1.00	1.00	-0.24	3	509.76	0
3	1	1	3	6.42	-0.73	30.52	41.52	20.91	1.00	1.00	1.00	-0.26	4	542.53	0
4	1	1	4	6.25	0.89	30.49	41.38	20.91	1.00	1.00	1.00	-0.23	5	509.36	0
...
4688923	134	243	139	10.98	-1.96	-5.11	-0.67	345.57	8.82	8.82	8.82	136.49	34988	1152.60	0
4688924	134	243	140	11.82	-3.18	-5.46	-0.54	345.57	13.87	13.87	13.87	84.43	34989	681.65	0
4688925	134	243	141	11.91	-1.42	-5.21	-0.42	345.57	10.69	10.69	10.69	145.72	34990	1118.35	0
4688926	134	243	142	11.86	-0.95	-5.40	-0.38	345.57	13.94	13.94	13.94	89.56	34991	683.49	0
4688927	134	243	143	11.72	0.04	-5.23	-0.37	345.57	10.90	10.90	10.90	120.18	34992	1026.93	0

Wind Power Forecasting

① Mark abnormal Patv (continued)



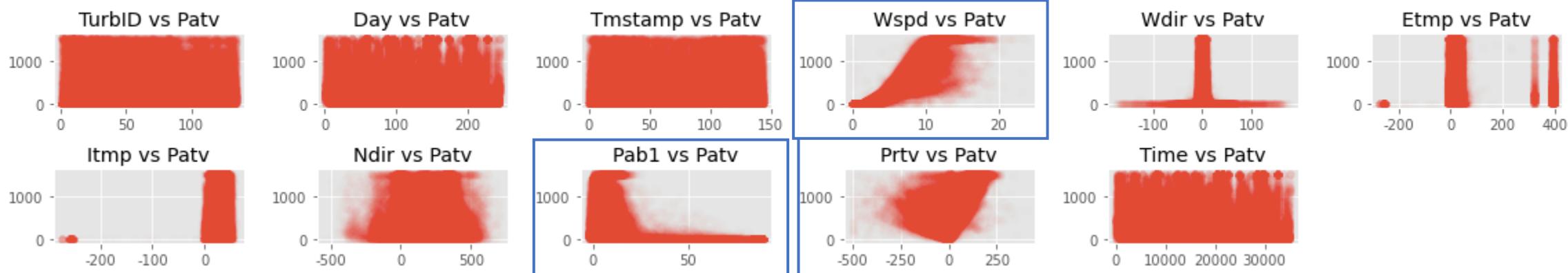
Abnormal data는 대부분 Pab가 20도 이상이라



Wind Power Forecasting

① Mark abnormal Patv (continued)

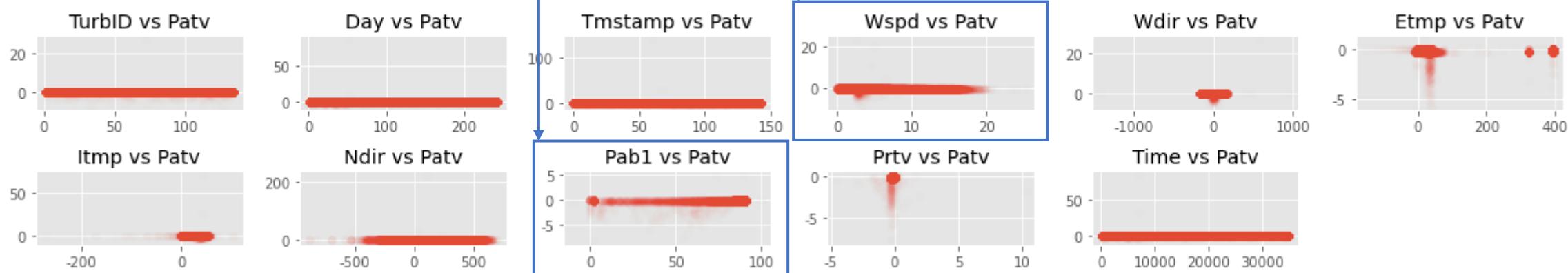
Features vs Target of normal data



Pab로 인한 Patv=0 분리 후,

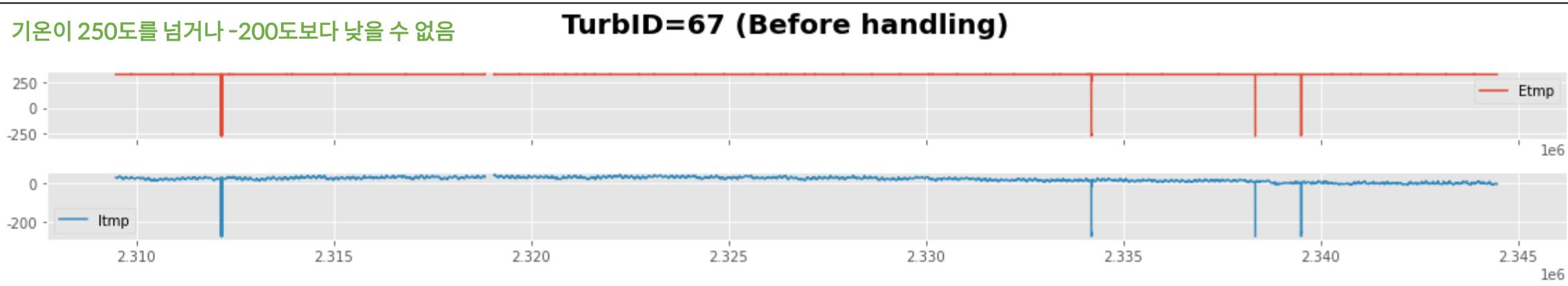
corr(Wspd, Patv) 상승: $0.81 \rightarrow 0.88$

Features vs Target of abnormal data

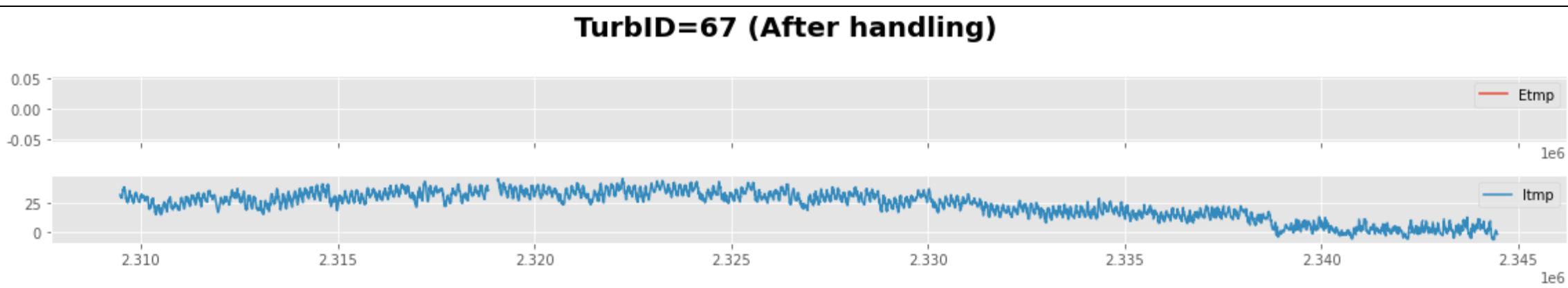


② Outlier handling

1. 다음과 같이 연속적으로 outlier가 이어지는 경우는 직접 null값으로 처리



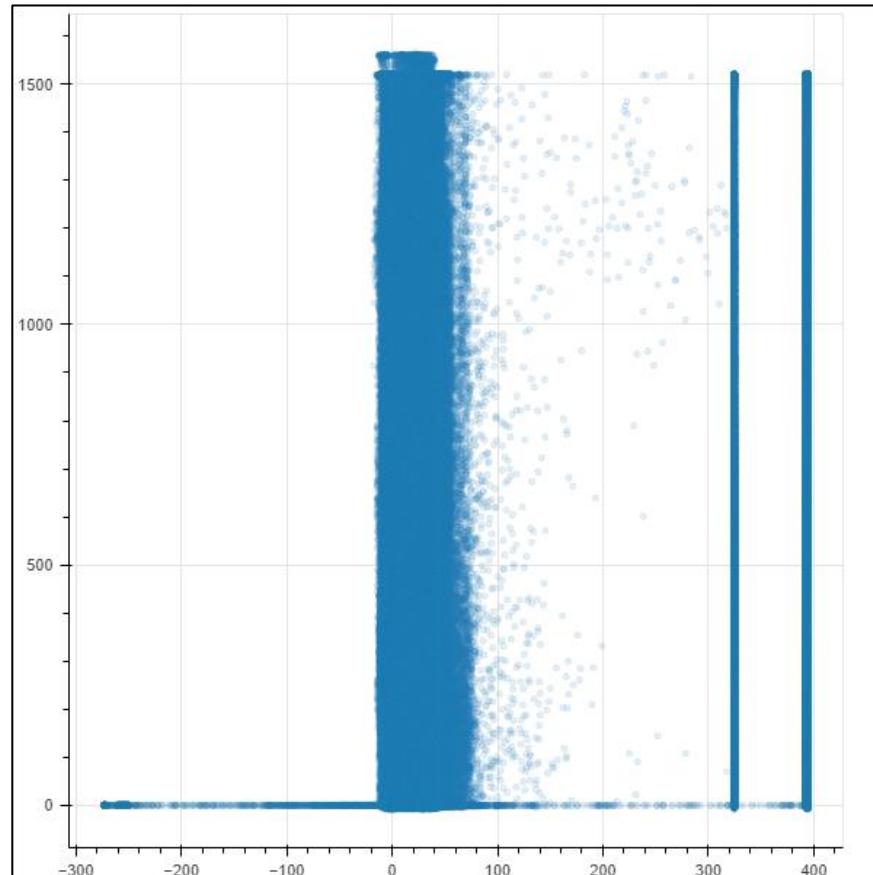
TurbID=67 (After handling)



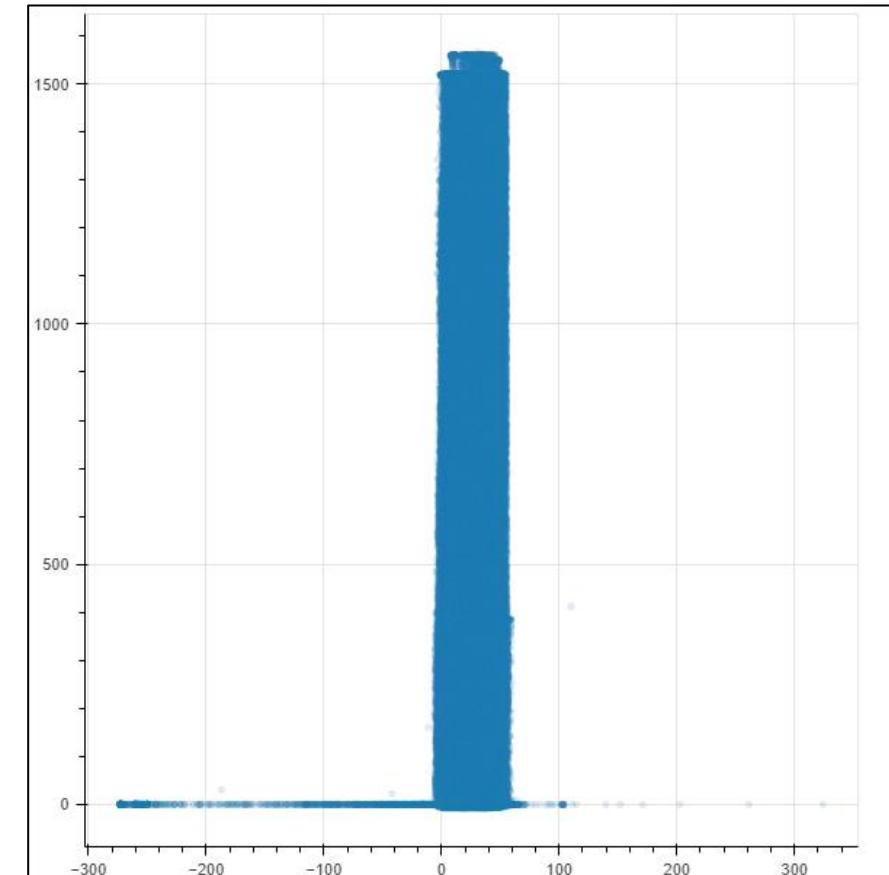
② Outlier handling

2. Ndir, Wdir, Pab: 주어진 정상 범위(not abnormal)로 clipping

Etmp: -20도~80도 안에 들도록 clipping

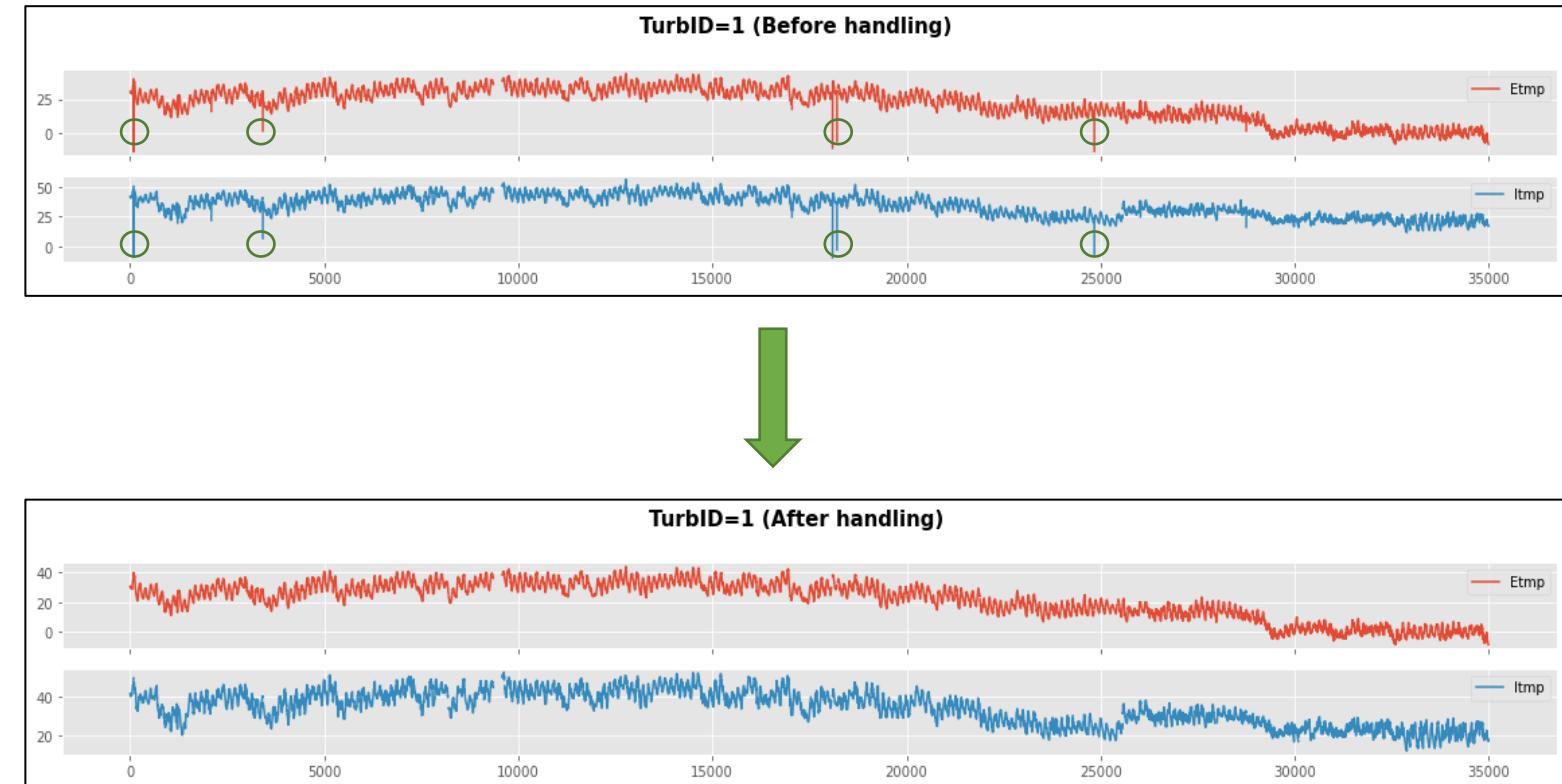
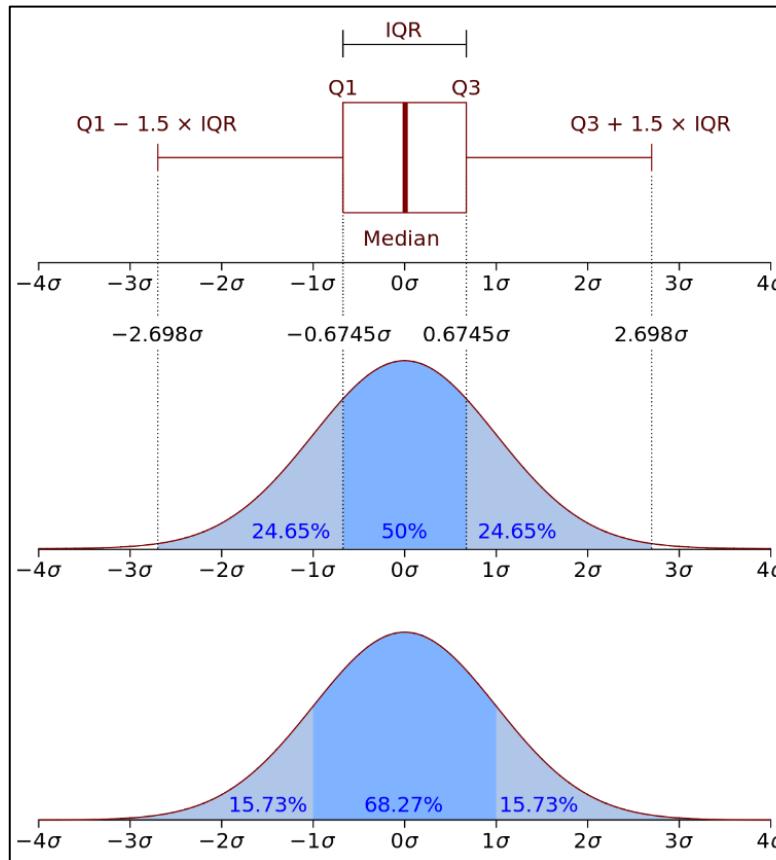


Itmp: -10도~65도 안에 들도록 clipping



② Outlier handling

3. 각 sample에 대하여 해당값 혹은 1차 차분값이 주변 2일 동안의 값들에 대하여 이상치라고 판단되는 경우 null로 채움
(Etmp와 Itmp는 종모양의 분포를 가지고 있기 때문에 $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$ 이외의 값을 이상치로 설정하는 것이 적절)



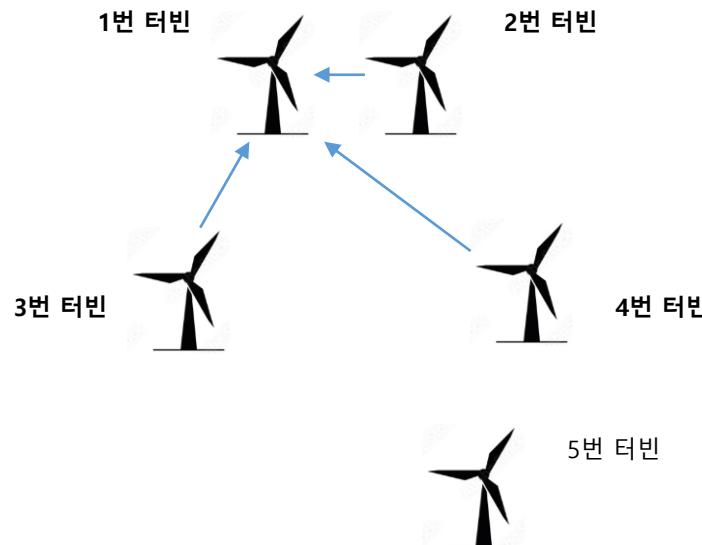
③ Imputing

1. Greedy imputing

이상치를 처리하고 난 후, 터빈의 데이터가 상당수 소실되는 경우 다음 알고리즘을 통해 값을 채움

Algorithm Greedy imputing

1. Imputing하고자 하는 터빈(대상 터빈)과 가장 가까운 k개의 터빈을 선택
 2. 선택된 터빈들을 대상 터빈의 값과 비교하여 유사한 순으로 정렬(MAE or MSE)
 3. 대상 터빈의 값이 존재하지 않은 timestep의 값을 선택된 터빈들에서 정렬된 순서대로 찾아 채워넣음

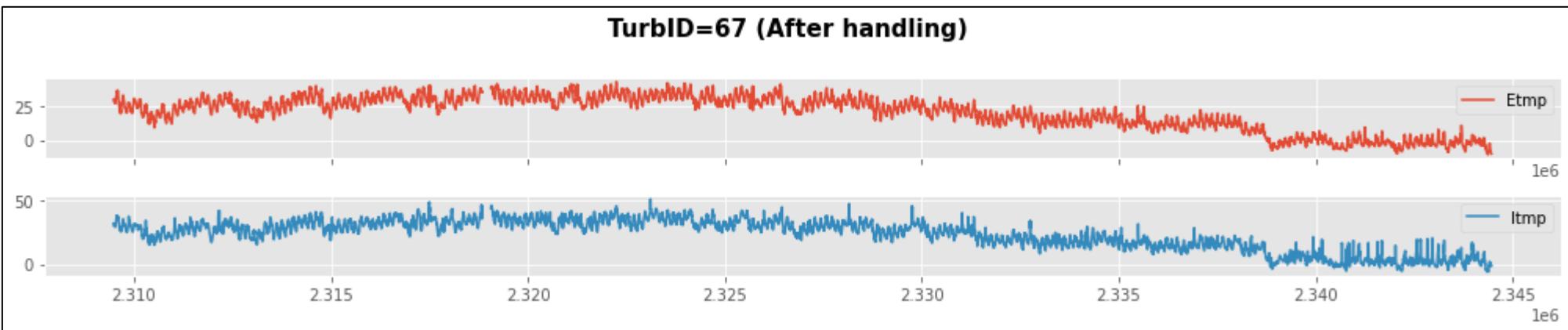
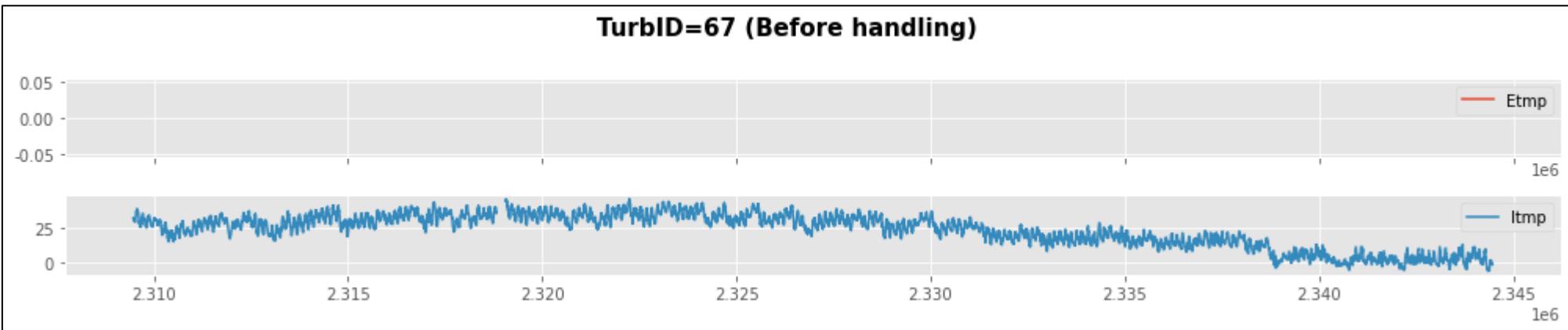


Ex) 1번 터빈에 대한 Greedy imputing 수행과정

터빈 ID	거리	유사도 (MAE)	Time=1	Time=2	Time=3	Time=4	Time=5
1번 (기준)			10	20			
3번	2	0	10			40	
2번	1	1	11	21	31		
4번	3	2		22		44	55
5번	X	3	13	23	34	45	56

③ Imputing

1. Greedy imputing(continued)

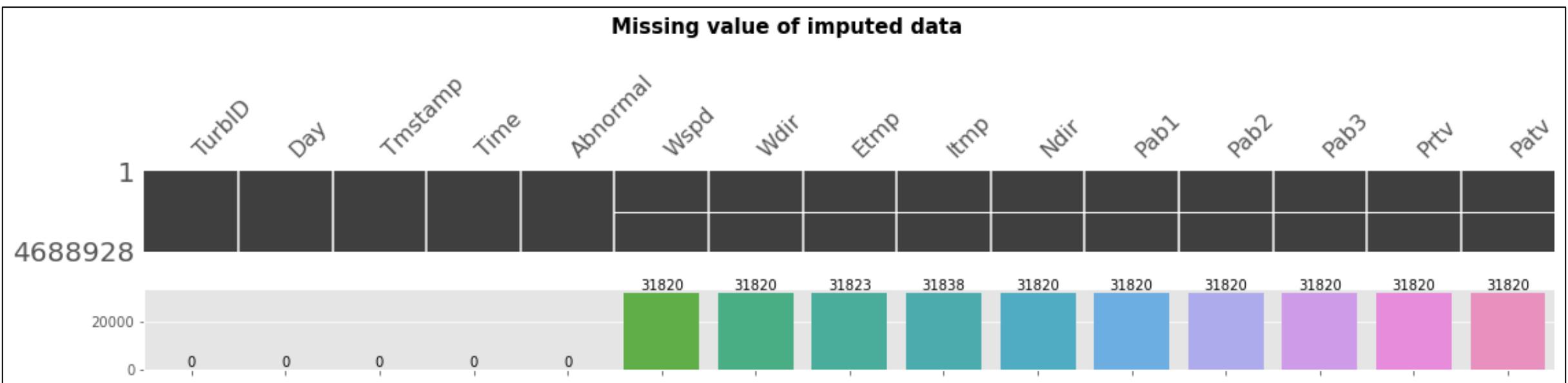


③ Imputing

2. Linear interpolation

각 turbine에 대하여 threshold(e.g. 12시간) 이상 연속적이지 않은 결측치를 linear interpolation으로 채움
(긴 sequence를 얹지로 interpolation하면 시계열 특성이 망가지게 될 염려가 있음)

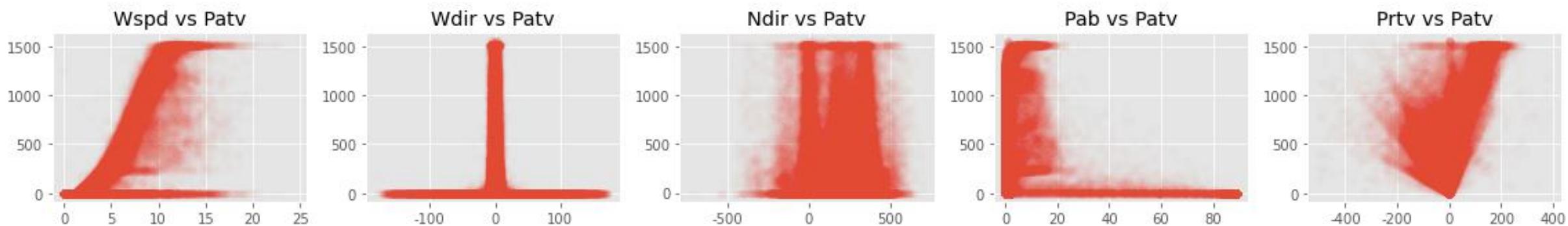
최종 imputing 결과



④ Feature engineering

1. Dummy variables ([insight from EDA](#))

```
data['Wspd_extreme'] = data['Wspd'] < 1  
data['Wdir_extreme'] = data['Wdir'].abs() > 10  
data['Ndir_extreme'] = data['Ndir'] < -90  
data['Pab_extreme1'] = data['Pab'] < 0.03  
data['Pab_extreme2'] = data['Pab'] > 20  
data['Prtv_pos']      = data['Prtv'] > 0
```

Features vs Target

④ Feature engineering

2. Multiplicative variables(interactive effect)

```
data['Wspd_active'] = data['Wspd'] - 1
data['Wdir_active'] = data['Wdir'].abs() - 10
data['Ndir_cos_abs'] = np.abs(np.cos(Ndir_rad)) # 0도, 180도, 360도
data['Prtv_abs']     = data['Prtv'].abs()
```

```
data['Wspd_comb'] = data['Wspd_extreme'] * data['Wspd_active']
data['Wdir_comb'] = data['Wdir_extreme'] * data['Wdir_active']
data['Ndir_comb'] = data['Ndir_extreme'] * data['Ndir_cos_abs']
data['Prtv_comb'] = data['Prtv_pos'] * data['Prtv_abs']
```

④ Feature engineering

3. Feature extraction(domain knowledge)

```
ALPHA          = 40
data['Pab']     = (data['Pab1'] + data['Pab2'] + data['Pab3'])/3
Pab_rad        = np.radians(data['Pab']+ALPHA)
data['TSR']      = 1 / np.tan(Pab_rad)
data['RPM']       = data['Wspd_active'] * data['TSR']
data['Wspd_cube'] = data['Wspd_active']**3
data['Patv_pos']   = np.maximum(data['Patv'], min_val)
data['Patan_abs'] = np.arctan(data['Prtv_abs'] / data['Patv_pos'])

Wdir_rad        = np.radians(data['Wdir'])
data['Wdir_cos'] = np.cos(Wdir_rad)
data['Wdir_sin'] = np.sin(Wdir_rad)
data['Wspd_cos'] = data['Wspd_active'] * np.cos(Wdir_rad)
data['Wspd_sin'] = data['Wspd_active'] * np.sin(Wdir_rad)
data['Ndir_sin_abs'] = np.abs(np.sin(Ndir_rad))
```

④ Feature engineering

4. Time encoding

```
DAY          = 6*24  # 10minute * 6 * 24hour
Time_in_day    = data['Time'] * (2*np.pi) / DAY
data['Day_cos'] = np.cos(Time_in_day)
data['Day_sin'] = np.sin(Time_in_day)

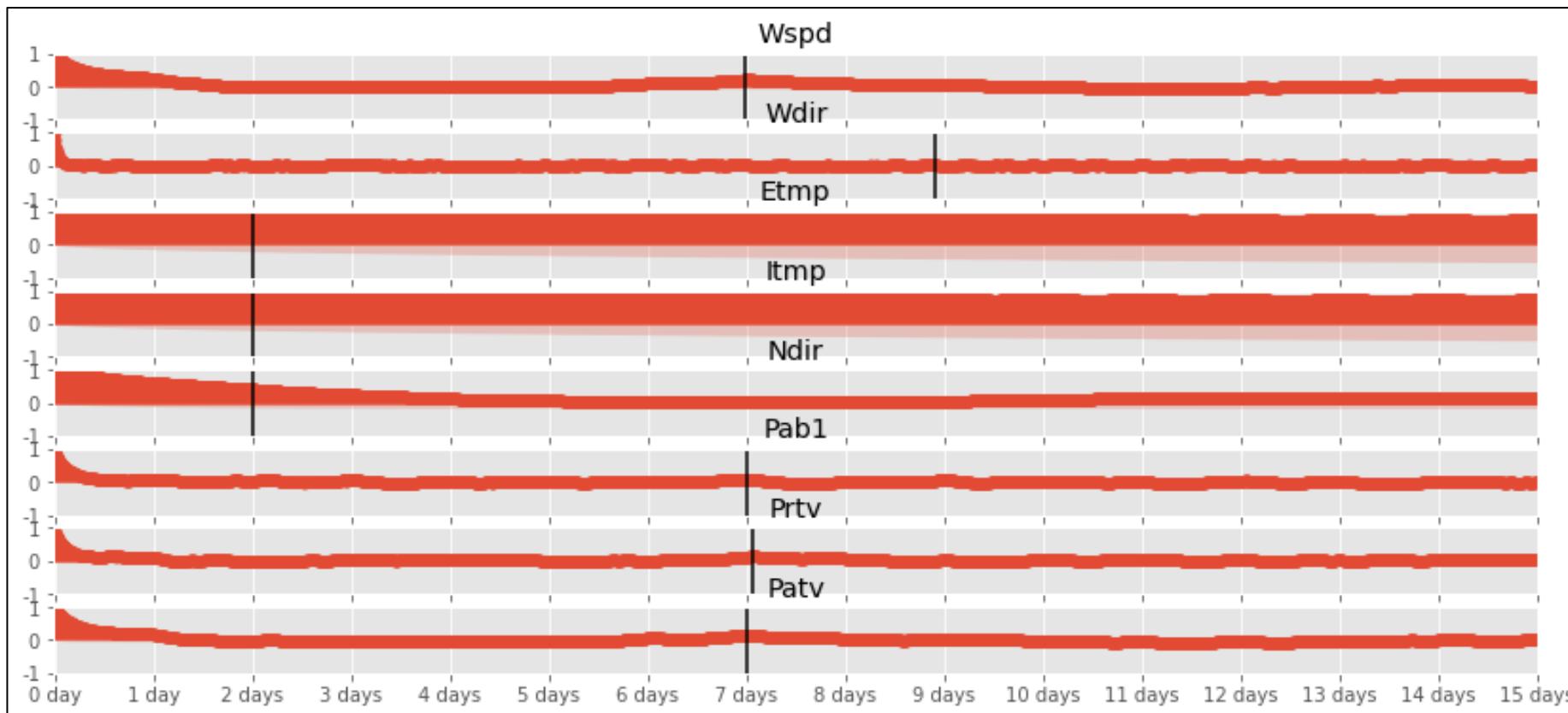
YEAR         = 365*DAY
Time_in_year   = data['Time'] * (2*np.pi) / YEAR
data['Year_cos'] = np.cos(Time_in_year)
data['Year_sin'] = np.sin(Time_in_year)
```

④ Feature engineering

5. Lagged variables

Autocorrelation function을 그려보면 lag=7일(144×7)에서 비교적 상관관계가 높다는 것을 확인할 수 있습니다.

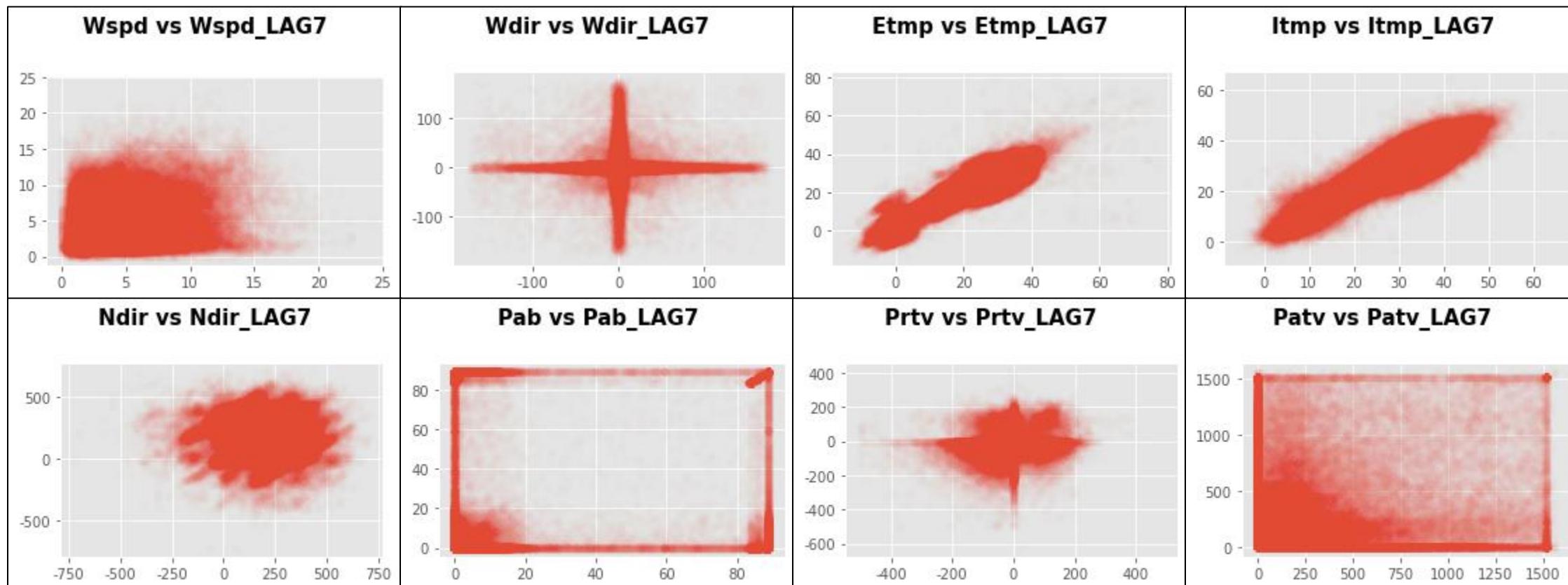
따라서, input에 output의 7일 전 데이터가 포함되도록 lagged feature를 추가하였습니다.



④ Feature engineering

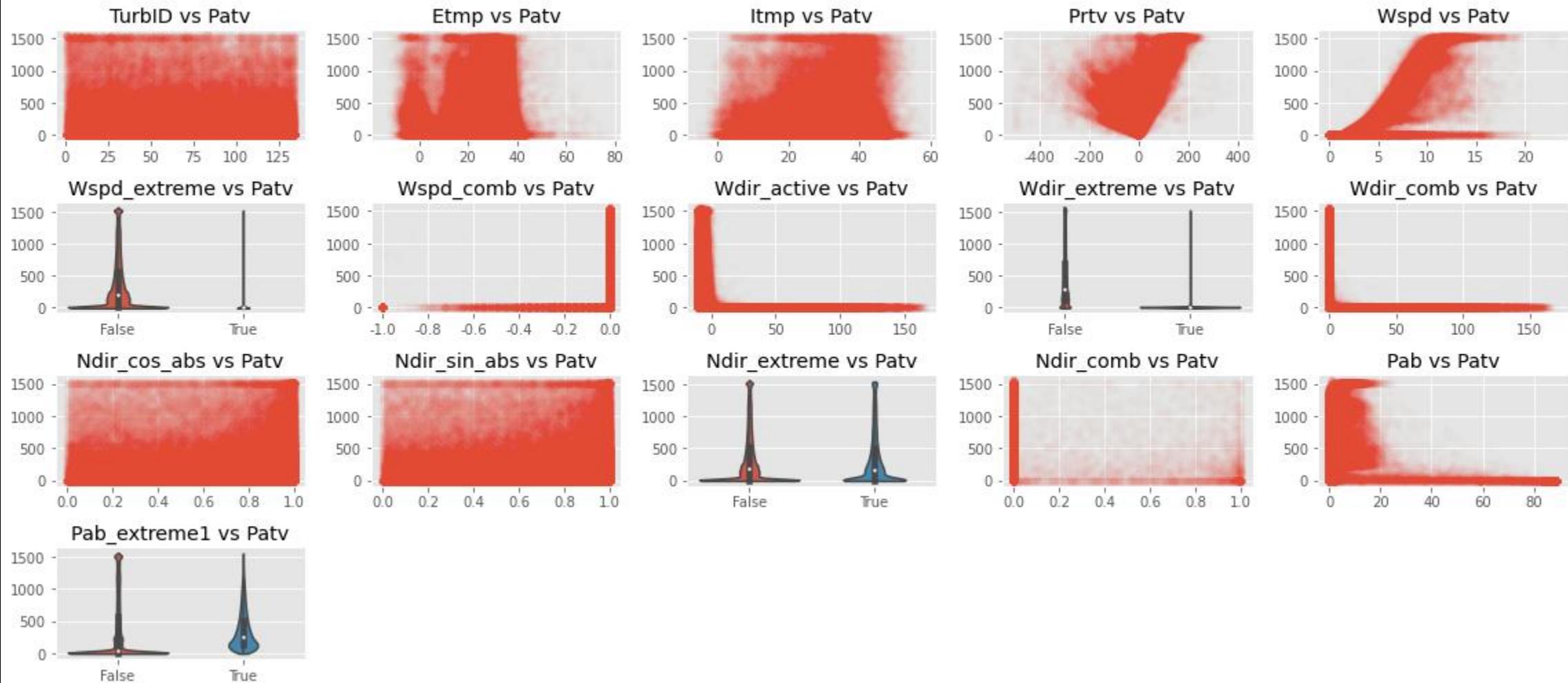
5. Lagged variables (continued)

7일 lagging한 결과



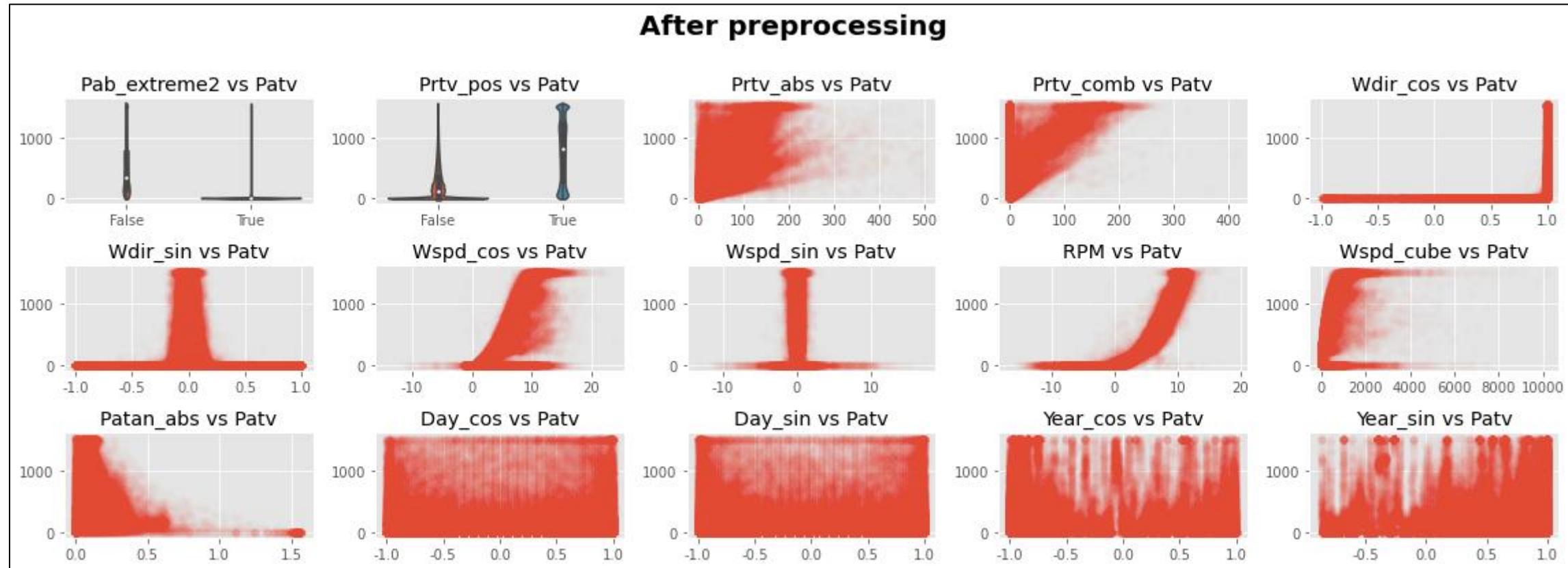
최종 preprocessing 결과

After preprocessing



Wind Power Forecasting

최종 preprocessing 결과



4) Training

① Data split

Training set : Day $\in \{1\text{일}, \dots, 217\text{일}\}$

Validation set : Day $\in \{218\text{일}, \dots, 241\text{일}\}$

Test set : Day $\in \{242\text{일}, 243\text{일}\}$

Input sequence의 길이: 288

Null값은 포함하는 샘플은 제외하여 sliding window로 생성

Outputs : [Wspd, Patv]

Loss : MSE

* Shape of dataset

- Training input : (13266, 288, 63)
- Training output : (13266, 288, 2)
- Validation input : (1474, 288, 63)
- Validation output: (1474, 288, 2)
- Test input : (134, 288, 63)
- Test output : (134, 288, 2)

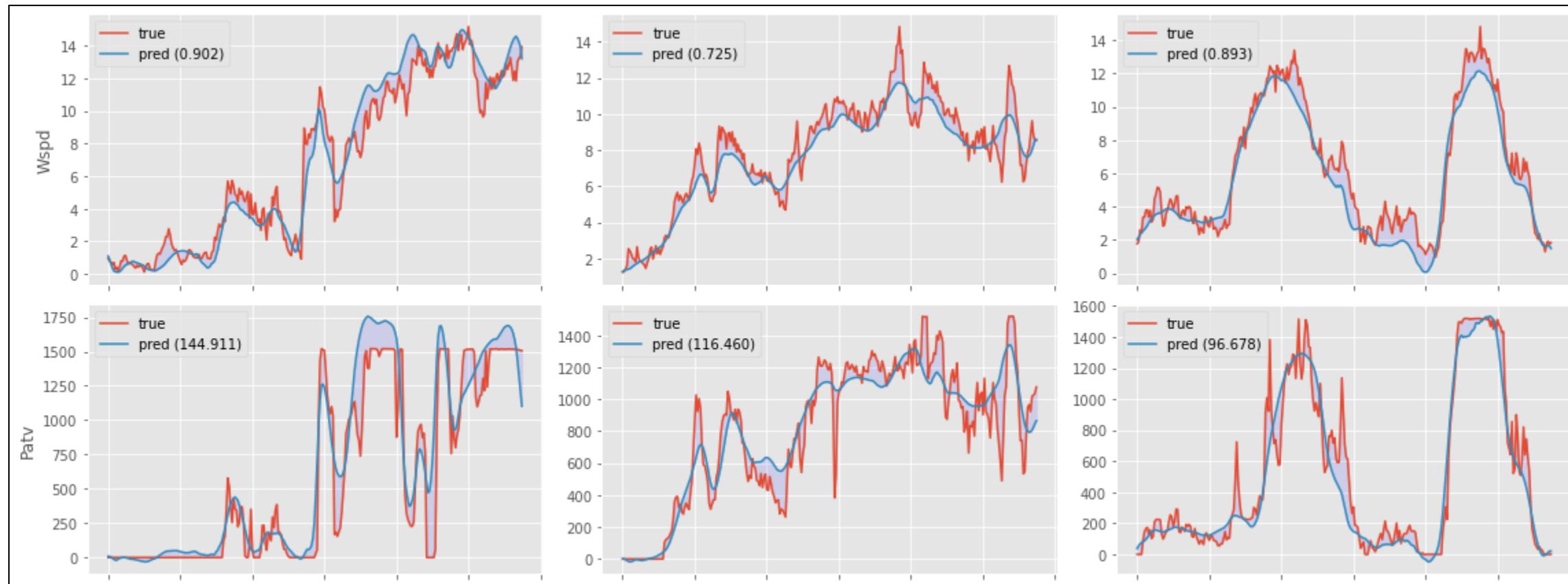
② Model

GRU (# units=32, 1 hidden layer)

5) Result

① Training set

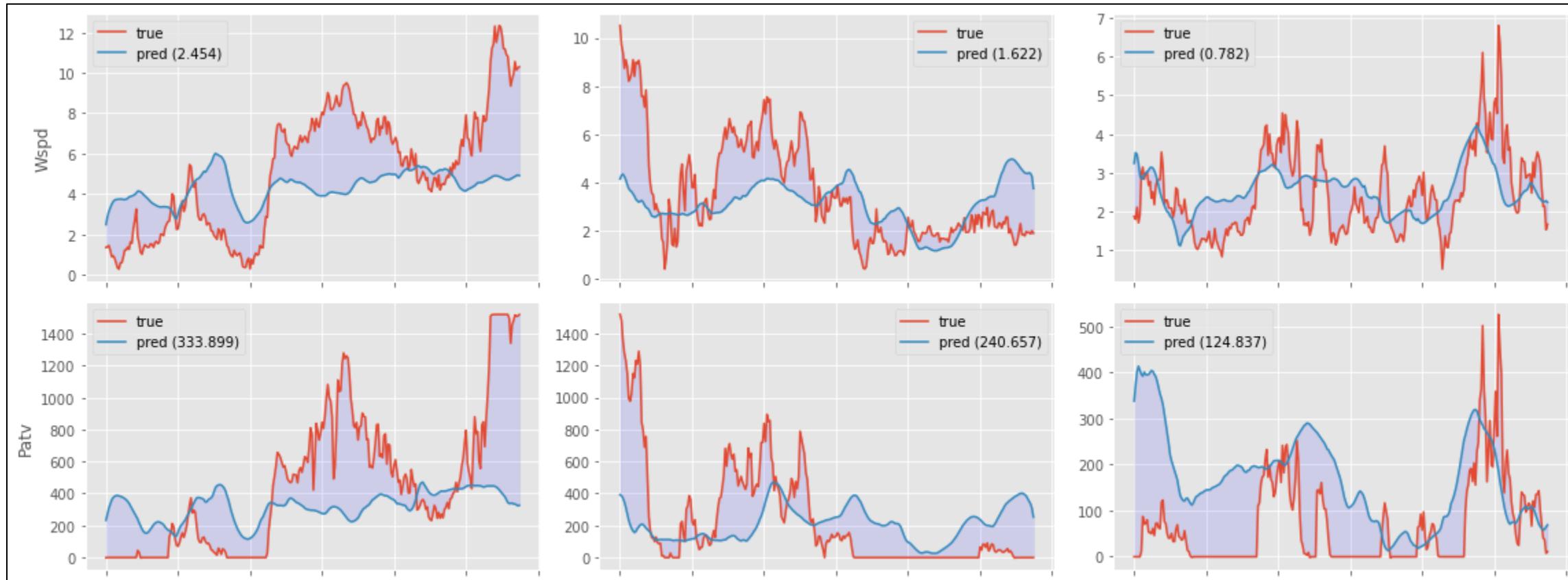
평가점수 = 129.80



5) Result

② Validation set

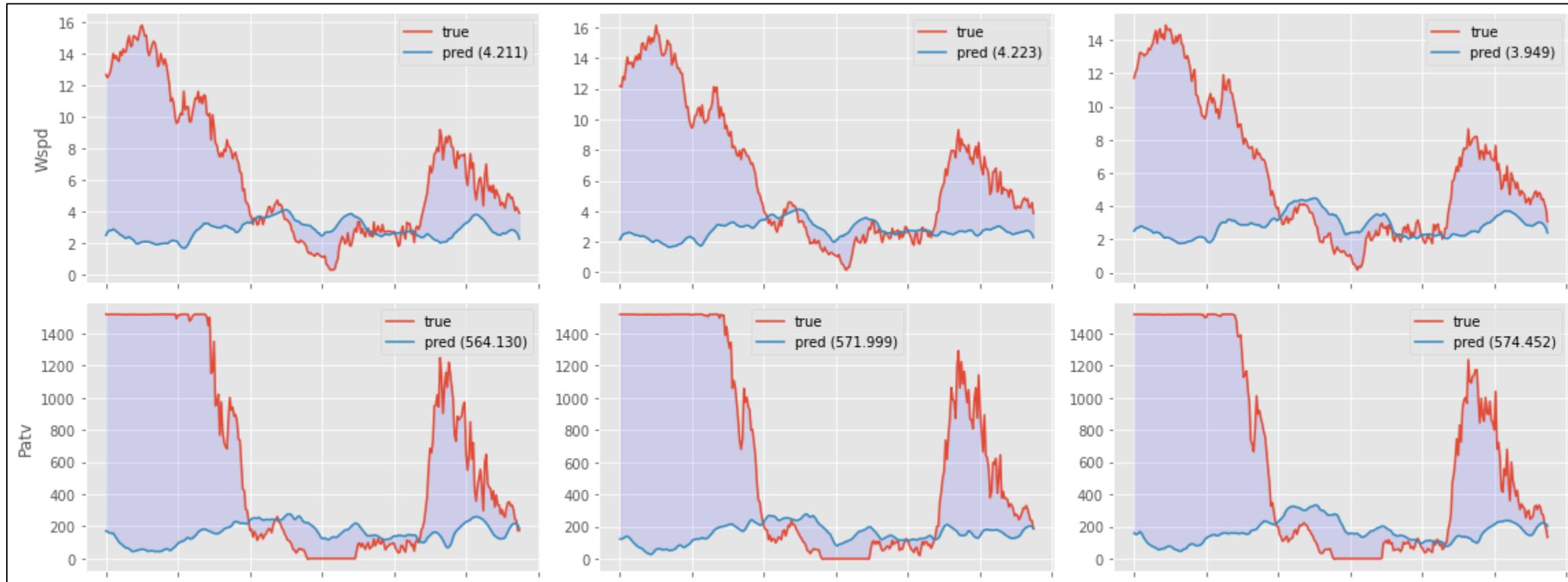
평가점수 = 250.07



5) Result

③ Test set

평가점수 = 614.25



3. 결론

- 학습 데이터에 대해서는 꽤 좋은 결과를 보여주었으나, 학습 데이터와 먼 데이터일수록 좋은 결과를 얻지 못한 것으로 추정
- 모델의 복잡도를 충분히 줄인 경우에도 overfitting이 발생한 것으로 보아,
 - 1) 너무 많은 feature들을 사용하여 데이터를 구성
→ 더욱 유의미한 feature들을 선택
 - 2) Training set의 데이터와 validation set의 분포, test set의 분포의 차이가 심함
→ 학습 구간을 validation set 근처로 축소시키고 validation set 구간의 길이를 줄이는 것도 고려해볼만함
 - 3) 모델이 데이터의 주요한 패턴을 제대로 학습하지 못함
→ Autoformer 등 개선된 시계열 예측 모델을 사용
 - 4) Long-term dependency 문제를 완화
→ Sequence length를 줄여 모델이 효과적으로 데이터를 학습할 수 있도록 함

이러한 문제들을 해결한다면 개선된 결과를 얻을 수 있을 것으로 생각됨

CONTENTS

Enhanced Index Tracking

Enhanced Index Tracking

1. 프로젝트 소개

벤치마크 지수를 구성하는 여러 종목들 중 지수 이상의 수익률을 내는 포트폴리오(종목과 편입비율)을 선택하는 프로젝트

본 프로젝트에 대한 논문은 스마트미디어저널(KCI)에 등재되었으며 석사 학위 논문으로 인정받았습니다.

자세한 내용은 동봉된 발표자료와 논문을 통해 확인하실 수 있습니다.

- GitHub <https://github.com/alchemine/enhanced-index-tracking>
- Paper https://kism.or.kr/file/memoir/10_3_4.pdf

2. 상세 설명

- 비정상적인(nonstationary) 금융 데이터를 사용하는 경우, 데이터의 변동성과 수익성의 trade-off를 제어하고 모델의 일반화 성능을 높이는 것이 핵심 이를 해결하기 위해 3단계 포트폴리오 선택 알고리즘과 앙상블 학습 알고리즘을 제안하여 2016년 ~ 2020년 약 5년간 S&P500 지수에 적용한 결과, 평균적으로 18.9% 높은 샤프지수를 가진 포트폴리오를 선택할 수 있었음

2. 상세 설명(continued)

1) 문제 정의

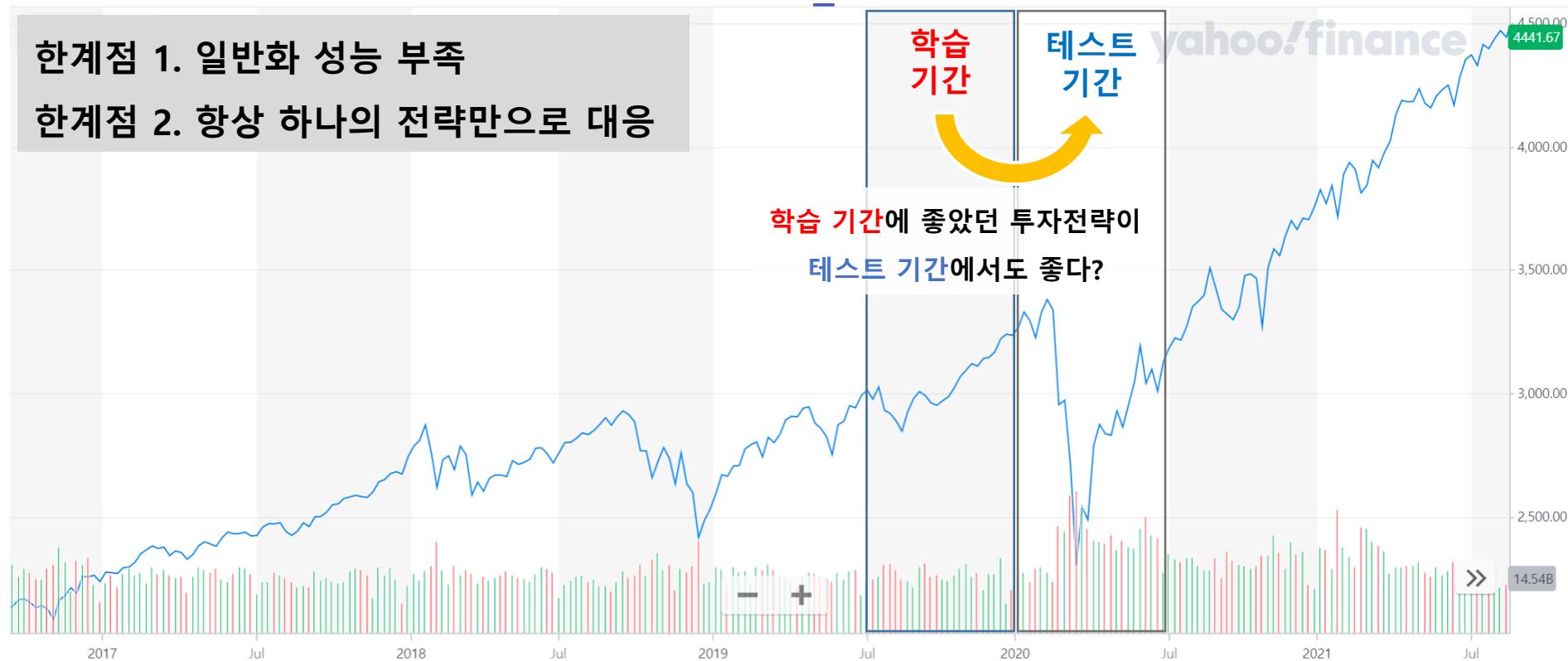
학습 기간의 종가(close) 데이터를 기반으로 학습하여 테스트 기간 동안 벤치마크 지수보다 (안정적으로) 높은 수익률을 얻고자 함



2. 상세 설명(continued)

2) 기존 연구의 한계점

- ① 테스트 기간에서 모델의 성능이 지속되기 어려움 (일반화 성능 부족)
- ② 항상 동일한 목적함수를 사용하여 문제를 해결하려고 함 (한 가지 전략만 고수)



2. 상세 설명(continued)

3) 기존 연구의 한계점에 대한 제안 방법

- ① 테스트 기간에서 모델의 성능이 지속되기 어려움 (일반화 성능 부족)

→ **3단계 포트폴리오 선택 알고리즘**: 목적함수와 별개의 지표를 regularization penalty로 고려하여 포트폴리오를 선택



- **포트폴리오 선택 방법**

Step 1 몬테카를로-유전 알고리즘을 이용한 목적함수(e.g. downside risk) 최적화

Step 2 차분 진화(differential evolution)를 이용한 목적함수 최적화

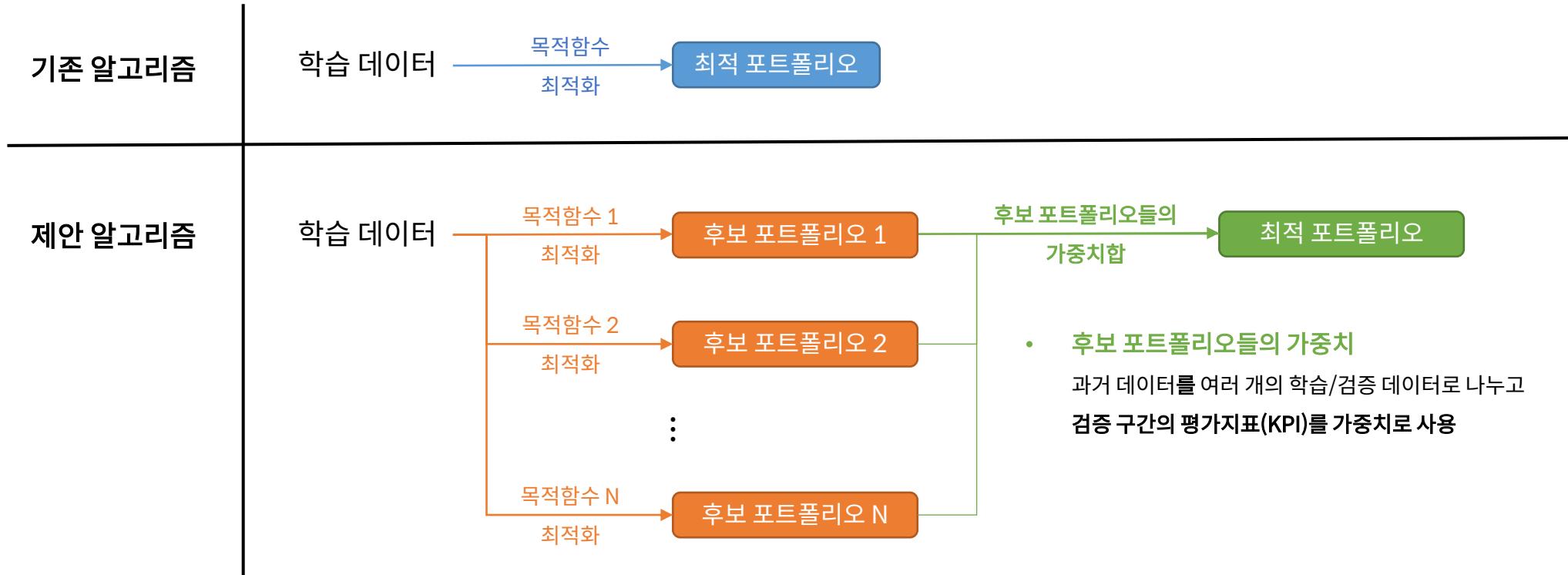
Step 3 후보 포트폴리오들 중 최적의 regularization penalty(e.g. 시가총액, 투자분산도, 종목 간 correlation, portfolio beta)를 가진 포트폴리오를 선택

2. 상세 설명(continued)

3) 기존 연구의 한계점에 대한 제안 방법

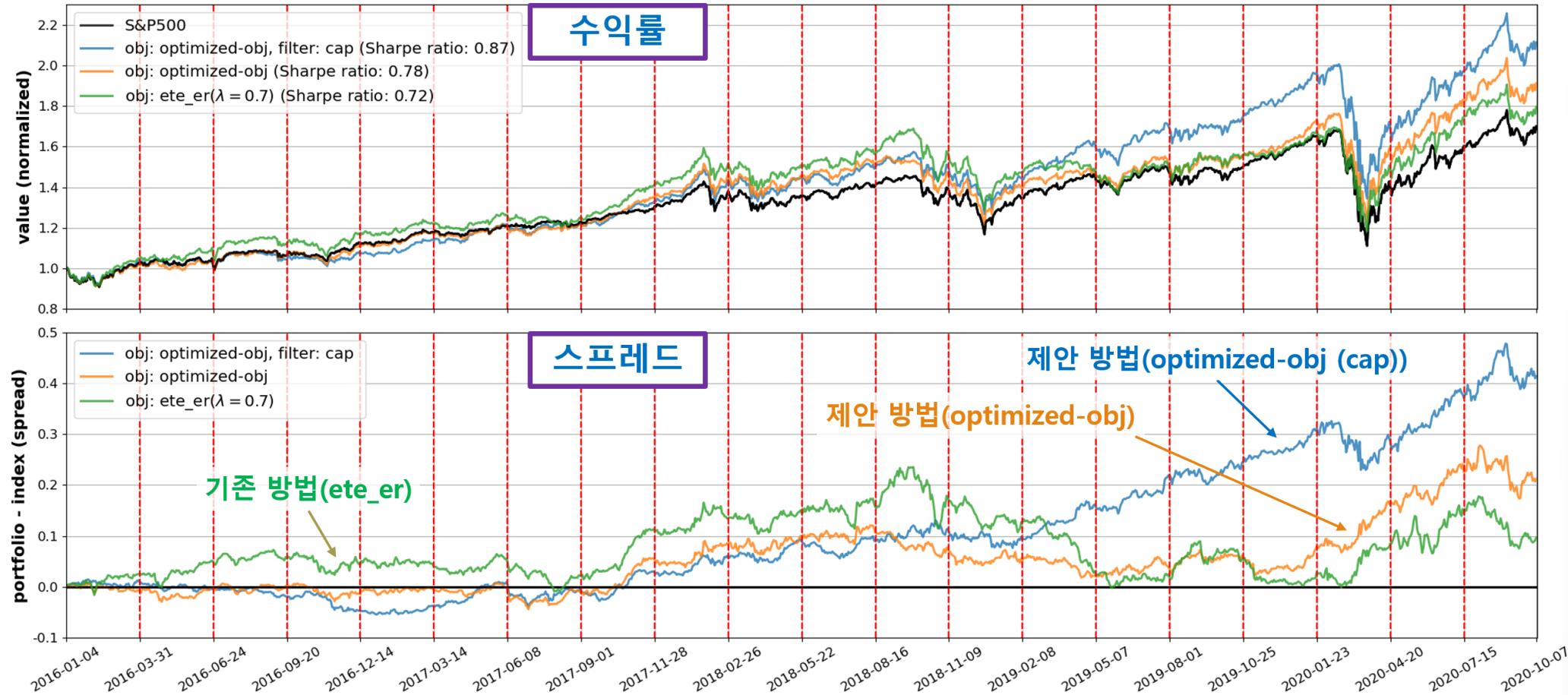
- ② 항상 동일한 목적함수를 사용하여 문제를 해결하려고 함 (한 가지 전략만 고수)

→ **양상을 학습 알고리즘**: 투자 시점마다 목적함수로 선택된 포트폴리오들의 양상을 적응적으로(adaptively) 선택



2. 상세 설명(continued)

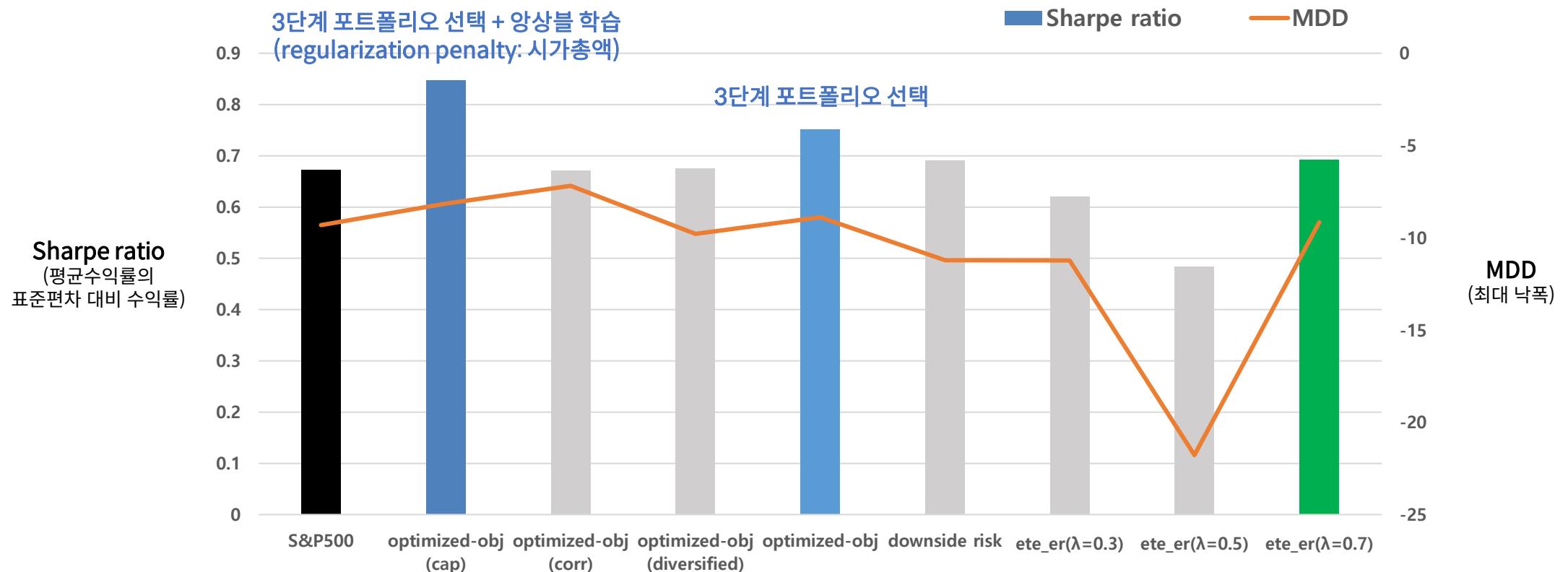
4) 실험 결과



2. 상세 설명(continued)

4) 실험 결과

S&P500 종가(2016/1/4 ~ 2020/10/7)에 대하여 3단계 포트폴리오 선택 알고리즘과 양상블 학습 알고리즘을 적용한 결과, 평균적으로 지수 대비 18.9% 더 높은 샤프지수(Sharpe ratio)를 얻을 수 있었습니다.



3. 결론

- 비정상적인(nonstationary) 금융 데이터의 변동성을 다루기 위해 새로운 regularization technique를 추가하여 2016년 ~ 2020년 약 5년간 S&P500 지수에 적용한 결과, 평균적으로 지수대비 18.9% 높은 샤프지수를 가진 포트폴리오를 선택할 수 있음을
- 진화 알고리즘은 과거 패턴을 학습하는 데에는 좋은 성능을 보여주지만, 미래의 수익률 혹은 변동성을 모델링(forecasting)하는 데에는 적합하지 않다는 한계점이 존재
- NN의 uncertainty를 활용하여 변동성을 측정하고, 직접적으로 KPI를 최대화시키는 Safety Reinforcement Learning 기법을 사용한다면 더욱 활용성이 높아질 것으로 기대

Court Decision Prediction

1. 프로젝트 소개

법률 사건에 대한 판결을 예측하는 프로젝트

- GitHub <https://github.com/alchemine/court-decision-prediction>
- Dacon <https://dacon.io/competitions/official/236112>

2. 상세 설명

- 미국 대법원 판결에 대한 원고와 피고, 판결의 근거가 되는 사실관계로부터 판결 결과(원고의 승소 여부)를 예측하는 binary classification
- 각 class의 개수가 약 1:2 정도로 **imbalance**가 존재하여 accuracy 뿐만 아니라 **precision, recall**을 제대로 관리할 수 있는 능력이 필요
- 총 3가지 방법론을 사용
 - Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측
 - 원고와 피고, 사실관계를 전처리 및 TF-IDF embedding 하여 판결 결과를 예측
 - 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

2. 상세 설명 (continued)

1) Data Overview

	ID	first_party	second_party	Features	Target
0	TRAIN_0000	Phil A. St. Amant	Herman A. Thompson	On June 27, 1962, Phil St. Amant, a candidate for public office, made a television speech in Baton Rouge, Louisiana. During this speech, St. Amant accused his political opponent of being a Communist and of being involved in criminal activities with the head of the local Teamsters Union. Finally, St. Amant implicated Herman Thompson, an East Baton Rouge deputy sheriff, in a scheme to move money between the Teamsters Union and St. Amant's political opponent. ¶Thompson successfully sued St. Amant for defamation. Louisiana's First Circuit Court of Appeals reversed, holding that Thompson did not show St. Amant acted with "malice." Thompson then appealed to the Supreme Court of Louisiana. That court held that, although public figures forfeit some of their First Amendment protection from defamation, St. Amant accused Thompson of a crime with utter disregard of whether the remarks were true. Finally, that court held that the First Amendment protects uninhibited, robust debate, rathe...	1

2. 상세 설명 (continued)

2) Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측

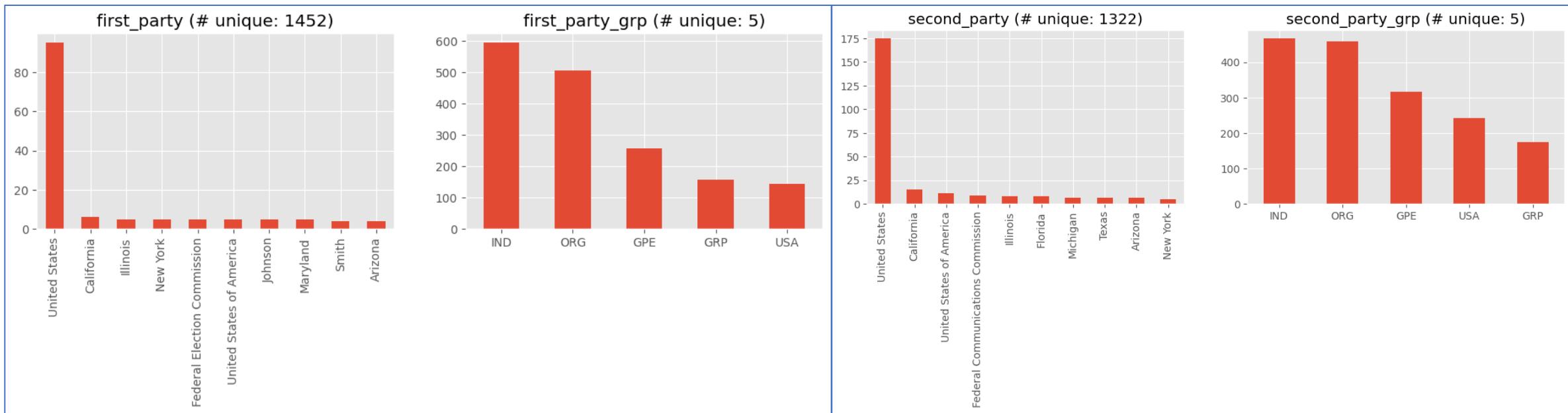
- ① Grouping: first_party, second_party → first_party_grp, second_party_grp



2. 상세 설명 (continued)

2) Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측 (continued)

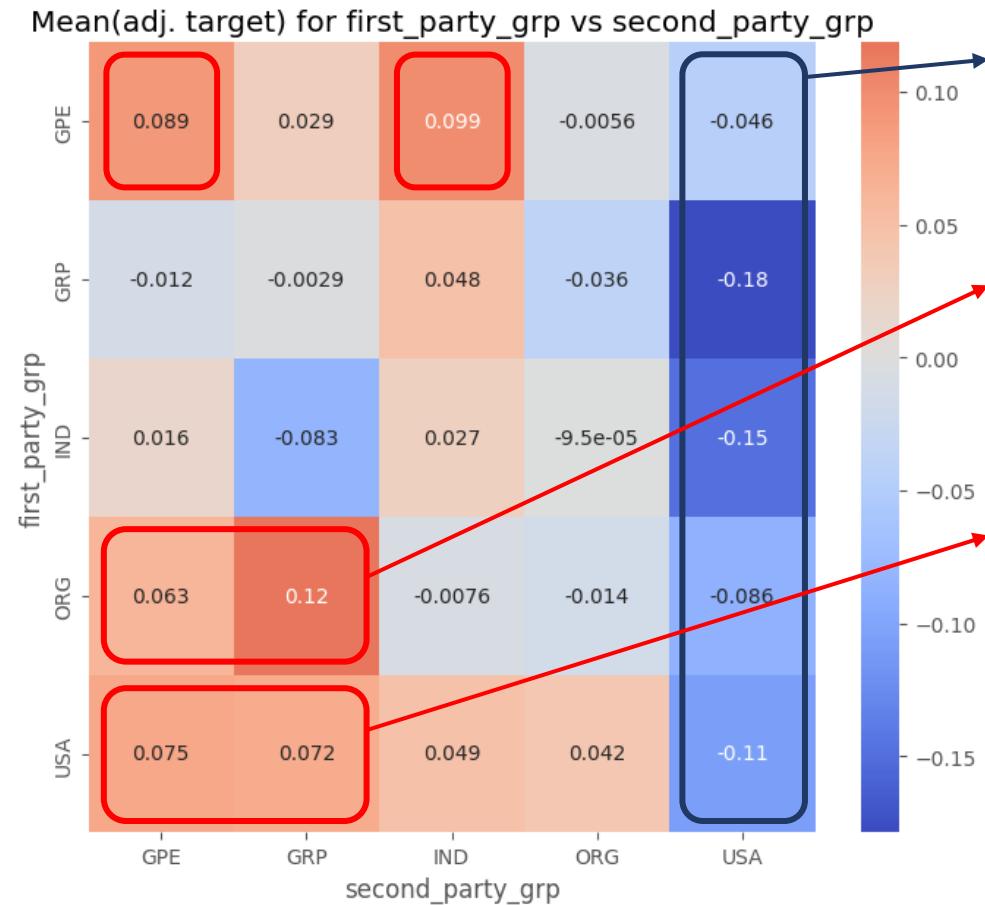
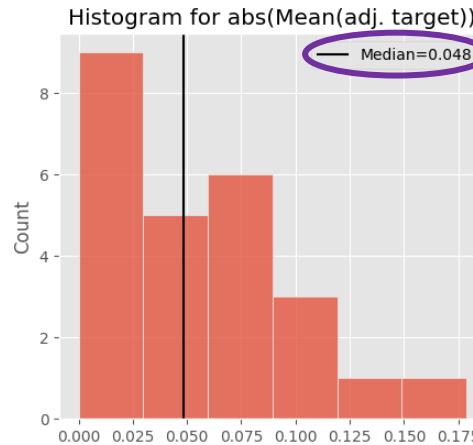
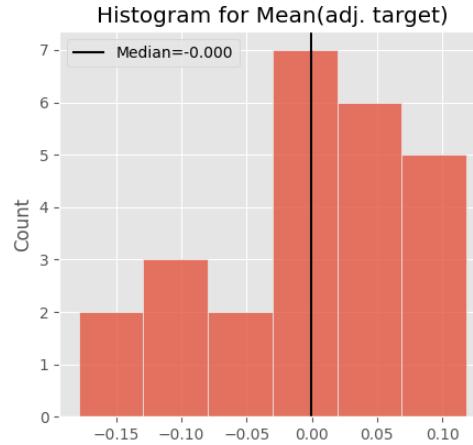
① Grouping: first_party, second_party → first_party_grp, second_party_grp (continued)



2. 상세 설명 (continued)

2) Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측 (continued)

① Grouping: first_party, second_party → first_party_grp, second_party_grp (continued)



피고(second_party)가 연방정부(USA)인 경우,

- 평균적으로 피고가 승소할 가능성이 높음
- 개인집단, 개인(GRP, IND)이 상대인 경우, 특히 높음

원고(first_party)가 기관(ORG)인 경우,

- 지방정부(GPE), 개인집단(GRP)이 상대인 경우, 원고가 승소할 가능성이 높음

원고(first_party)가 연방정부(USA)인 경우,

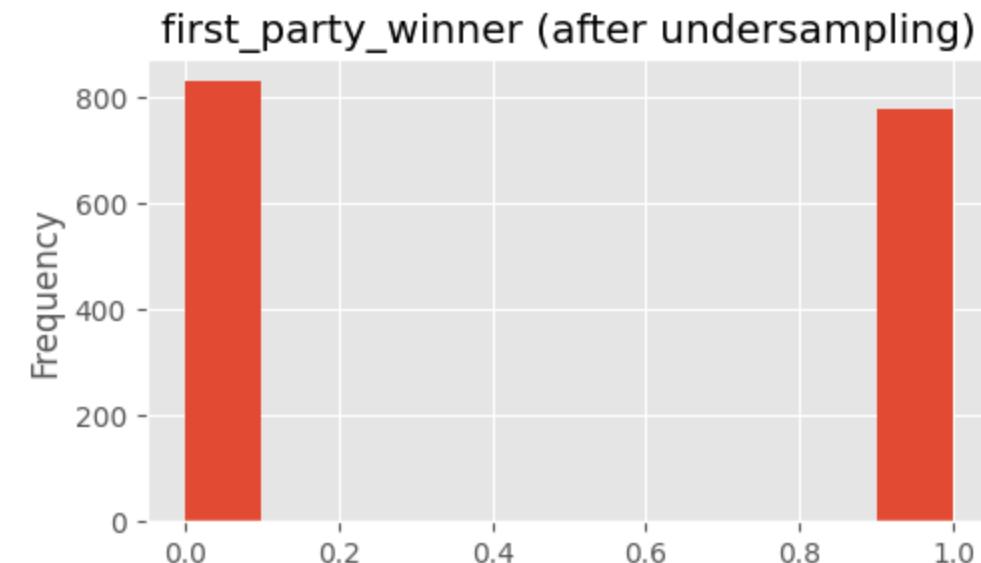
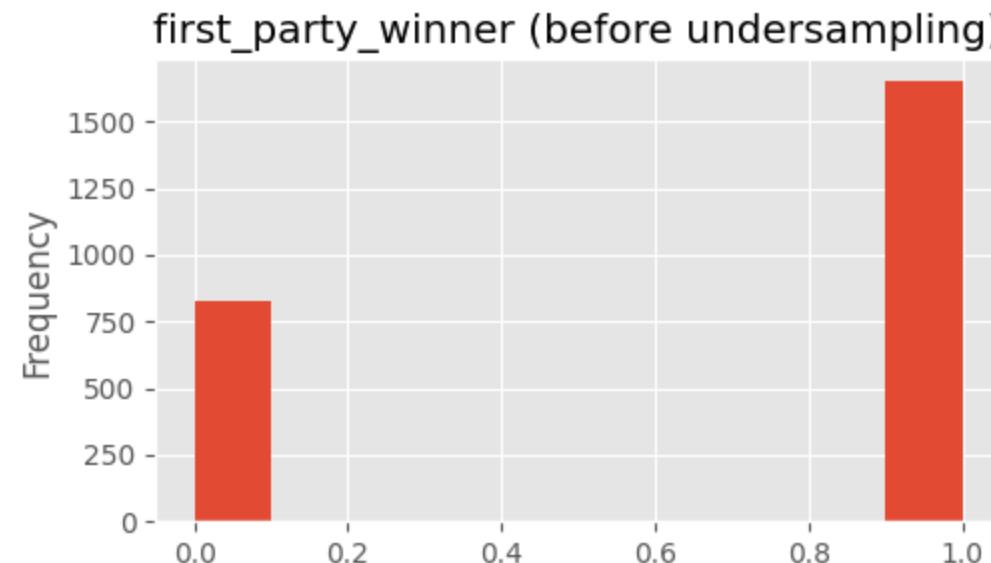
- 지방정부(GPE), 개인집단(GRP)이 상대인 경우, 원고가 승소할 가능성이 높음

2. 상세 설명 (continued)

2) Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측 (continued)

② Undersampling: Neighbourhood Cleaning Rule

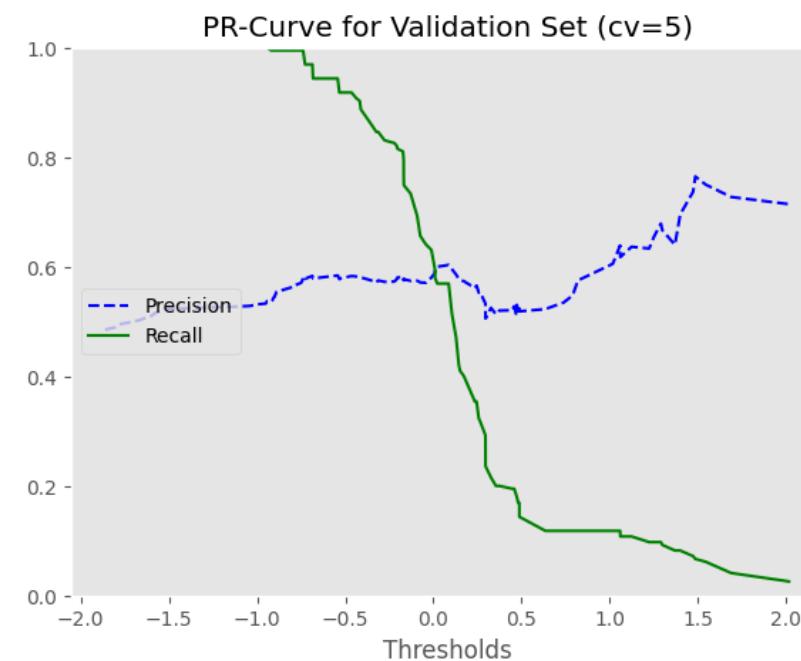
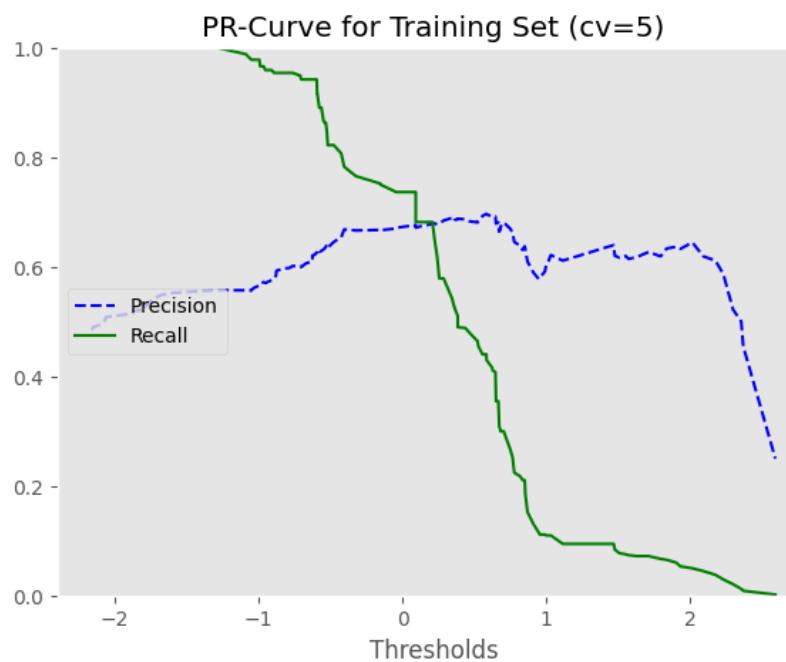
kNN-neighbourhood 기반 노이즈 샘플들을 제거



2. 상세 설명 (continued)

2) Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측 (continued)

③ PCA + LogisticRegression 학습 결과



Validation score

	precision	recall	f1-score	support
0	0.64	0.55	0.59	207
1	0.59	0.68	0.63	195
accuracy			0.61	402
macro avg	0.62	0.61	0.61	402
weighted avg	0.62	0.61	0.61	402

2. 상세 설명 (continued)

3) 원고와 피고, 사실관계를 전처리 및 embedding 하여 판결 결과를 예측

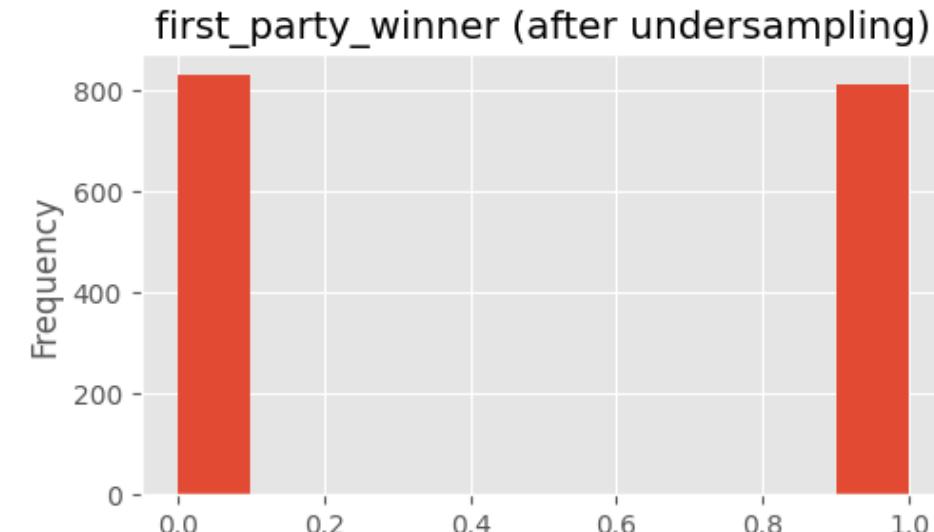
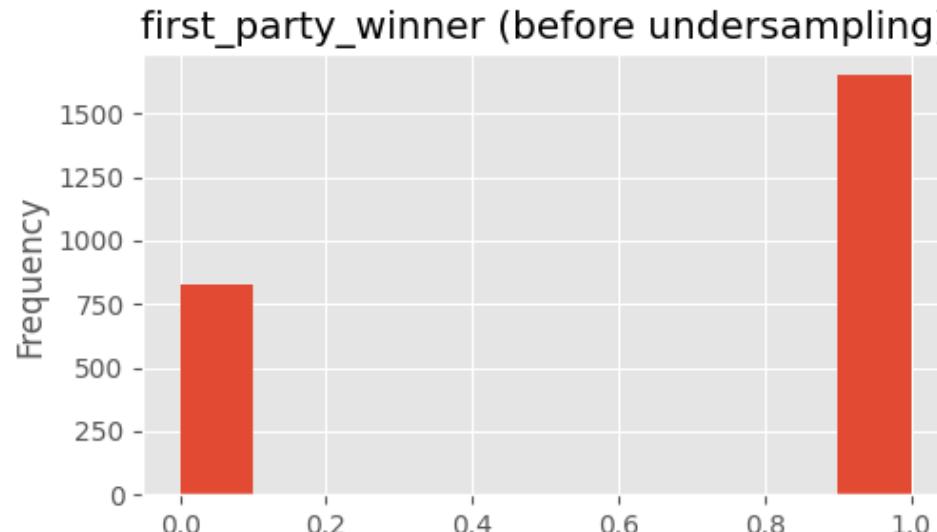
① Preprocessing Text

- Cleaning: Corpus에서 한 글자 단어, 공백, 특수문자 등 노이즈 데이터 제거
- Tokenization: nltk.tokenize.TreebankWordTokenizer 사용
- Stopword: nltk.corpus.stopwords 사용
- Lemmatization: nltk.stem.WordNetLemmatizer 사용
- Vectorization: CountVectorizer(원고/피고), TfidfVectorizer(사실관계)
- Concatenation

2. 상세 설명 (continued)

3) 원고와 피고, 사실관계를 전처리 및 embedding 하여 판결 결과를 예측

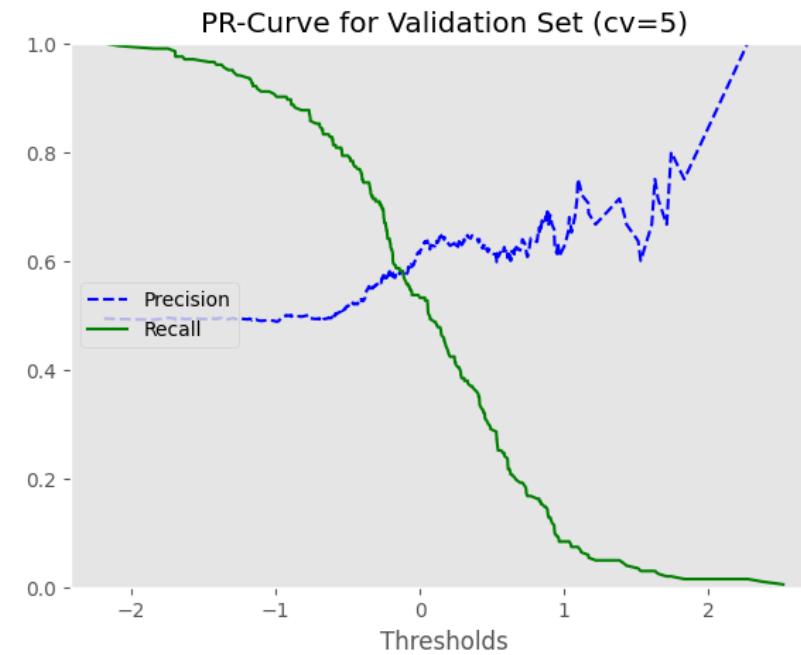
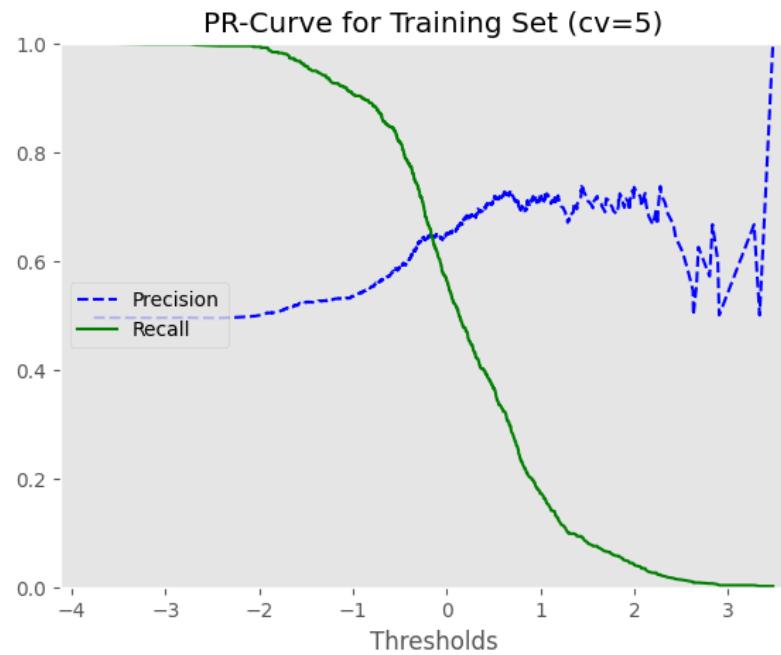
② Undersampling: Neighbourhood Cleaning Rule



2. 상세 설명 (continued)

3) 원고와 피고, 사실관계를 전처리 및 embedding 하여 판결 결과를 예측

③ PCA + LogisticRegression 학습 결과



Validation score

	precision	recall	f1-score	support
0	0.67	0.72	0.69	208
1	0.68	0.63	0.66	203
accuracy			0.67	411
macro avg	0.67	0.67	0.67	411
weighted avg	0.67	0.67	0.67	411

2. 상세 설명 (continued)

4) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

① Prompt 구성

USER:

- first_party: **Phil A. St. Amant (IND)**
- second_party: **Herman A. Thompson (IND)**
- facts:

Baton Rouge, Louisiana. During this speech, St. Amant accused his political opponent of being a Communist and of being involved in criminal activities with the head of the local Teamsters Union. ... Finally, that court held that the First Amendment protects uninhibited, robust debate, rather than an open season to shoot down the good name of anyone who happens to be a public servant.

- Question: Do the first_party win the case? Answer with only 1 or 0. DO NOT ANSWER ANY OTHER WORDS.

ASSISTANT:

- Answer:

Token의 개수를 일정하게 하기 위해, facts에서 다음 조건에 맞는 문장들만을 선택하고 최대 1000자로 한정

1. 피고/원고가 포함된 문장
2. 연결어(finally, however, hence, therefore 등)로 시작하는 문장
3. 핵심단어(court, judge, adjudicate, reverse 등)가 포함된 문장 (lemma 비교)

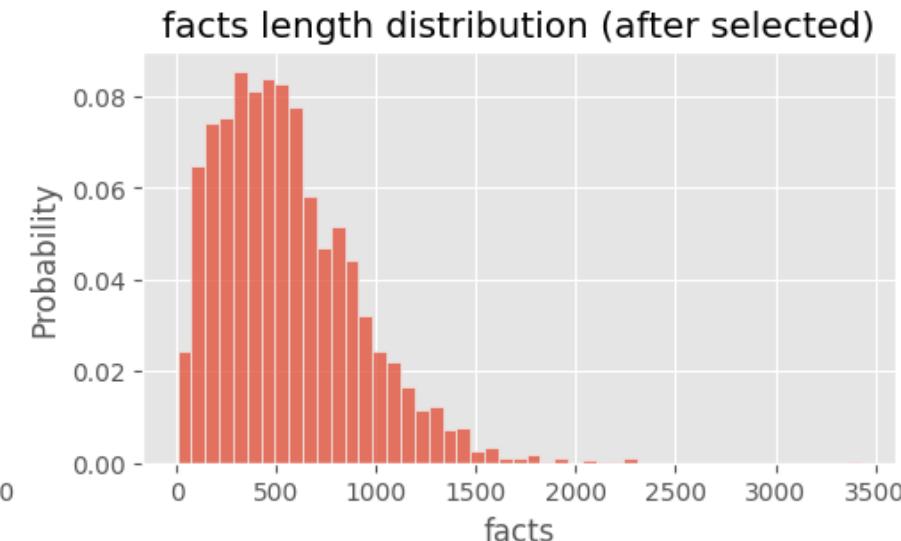
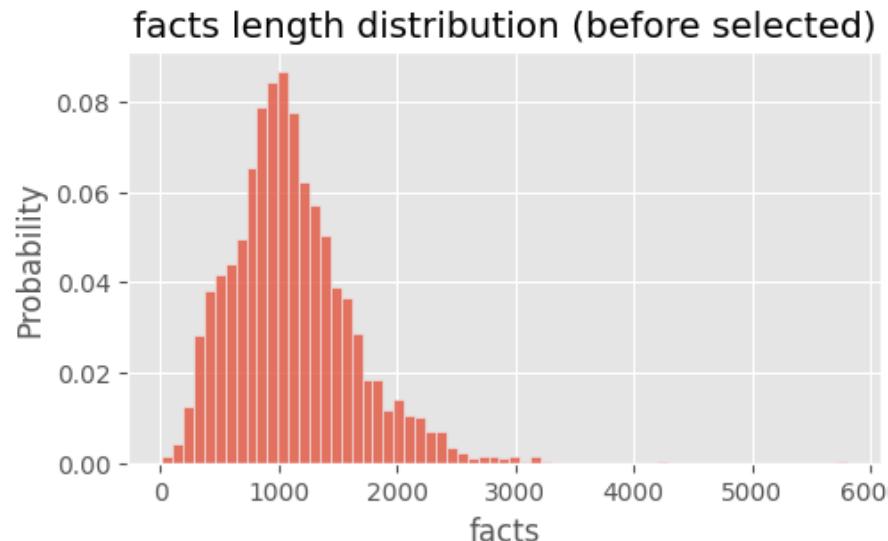
2. 상세 설명 (continued)

- 4) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

① Prompt 구성 (continued)

Token의 개수를 일정하게 하기 위해, facts에서 다음 조건에 맞는 문장들만을 선택하고 뒤에서부터 최대 1000자로 한정 (PAD: EOS token)

1. 피고/원고가 포함된 문장
2. 연결어(finally, however, hence, therefore 등)로 시작하는 문장
3. 핵심단어(court, judge, adjudicate, reverse 등)가 포함된 문장 (lemma 비교)



2. 상세 설명 (continued)

4) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

② LLM Modeling

vicuna-13b 모델 사용

Vicuna Model Card

Model Details

Vicuna is a chat assistant trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT.

- Developed by: [LMSYS](#)
- Model type: An auto-regressive language model based on the transformer architecture.
- License: Non-commercial license
- Finetuned from model: [LLAMA](#)

Downloads last month
27,088

Hosted inference API ⓘ
Text Generation
Inference API has been turned off for this model.

Spaces using lmsys/vicuna-13b-v1.3 2

- alexkueck/ChatBotLI2Klein
- alexshengzhili/calahealthgpt

```

LlamaForCausalLM(
    (model): LlamaModel(
        (embed_tokens): Embedding(32000, 5120, padding_idx=0)
        (layers): ModuleList(
            (0-39): 40 x LlamaDecoderLayer(
                (self_attn): LlamaAttention(
                    (q_proj): Linear(in_features=5120, out_features=5120, bias=False)
                    (k_proj): Linear(in_features=5120, out_features=5120, bias=False)
                    (v_proj): Linear(in_features=5120, out_features=5120, bias=False)
                    (o_proj): Linear(in_features=5120, out_features=5120, bias=False)
                    (rotary_emb): LlamaRotaryEmbedding()
                )
                (mlp): LlamaMLP(
                    (gate_proj): Linear(in_features=5120, out_features=13824, bias=False)
                    (down_proj): Linear(in_features=13824, out_features=5120, bias=False)
                    (up_proj): Linear(in_features=5120, out_features=13824, bias=False)
                    (act_fn): SiLUActivation()
                )
                (input_layernorm): LlamaRMSNorm()
                (post_attention_layernorm): LlamaRMSNorm()
            )
        )
        (norm): LlamaRMSNorm()
    )
    (lm_head): Linear(in_features=5120, out_features=2, bias=True)
)

```

Binary classification 설정에 따라 수정

2. 상세 설명 (continued)

4) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

② LLM Modeling (continued)

bfloat16 자료형을 이용하여 모델의 메모리 감소

Floating Point Formats



2. 상세 설명 (continued)

4) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

② LLM Modeling (continued)

LoRA 학습을 통해 효율적으로 새로운 데이터에 대한 fine-tuning 수행

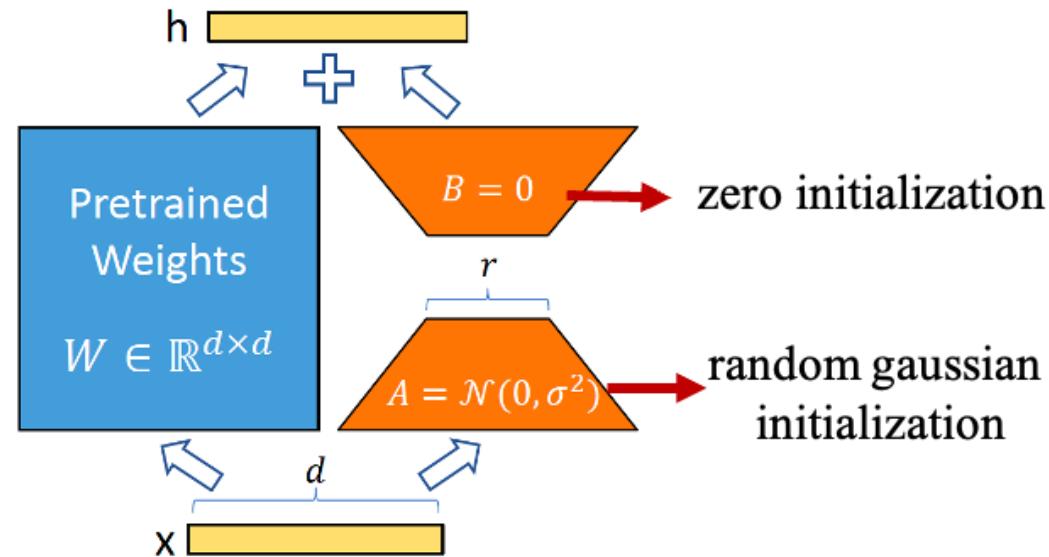


Figure 1: Our reparametrization. We only train A and B .

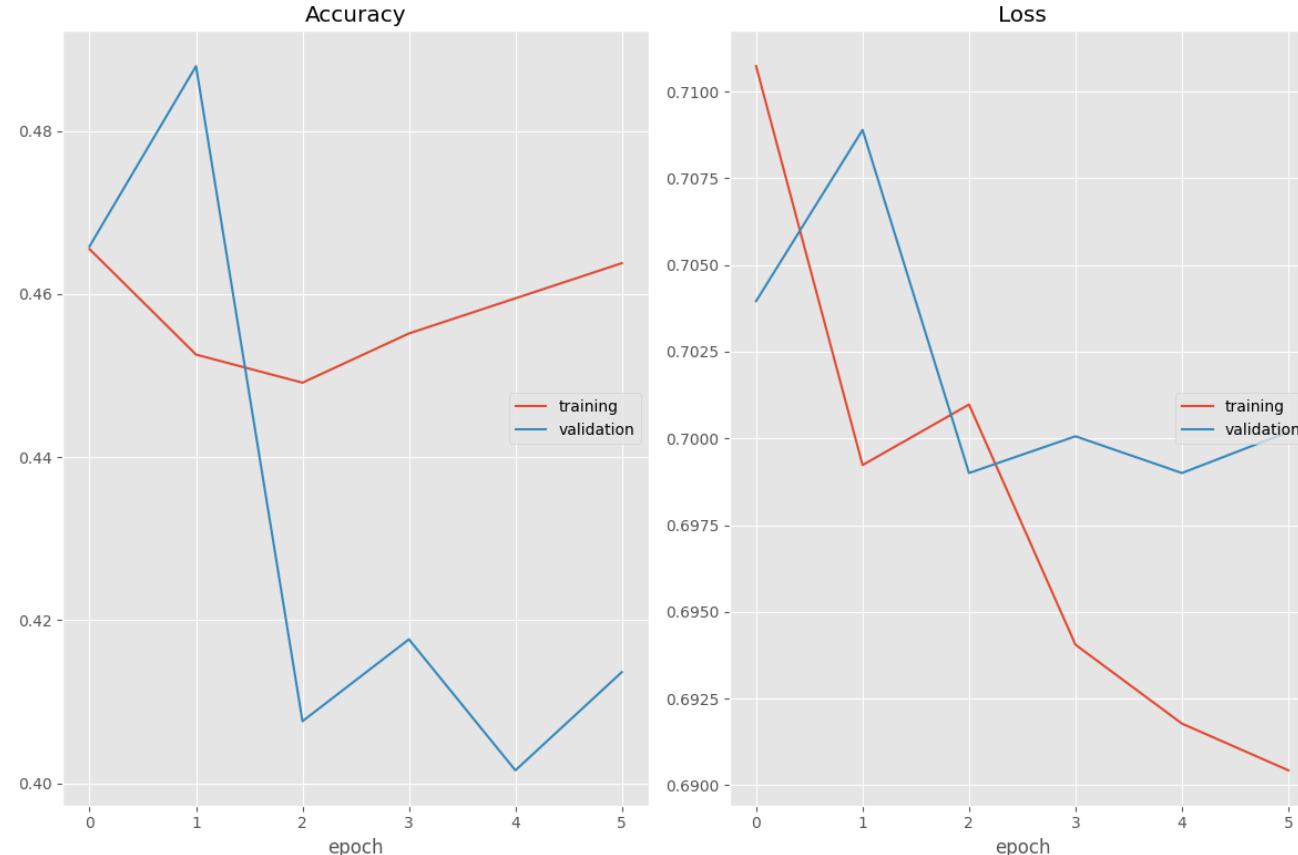
$$h = W_0 x + \Delta W x = W_0 x + B A x$$

2. 상세 설명 (continued)

4) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측

③ 학습 결과

Training loss는 떨어지는 모습을 보여주었으나, validation loss는 제대로 학습이 되지 않았음



3. 결론

- 총 3가지 방법론을 사용하여 판결 결과를 예측하고자 하였음
 - 1) Grouping한 원고와 피고를 입력으로 하여 판결 결과를 예측
 - 2) 원고와 피고, 사실관계를 전처리 및 TF-IDF embedding 하여 판결 결과를 예측
 - 3) 사실관계를 포함한 prompt를 입력으로 하는 LLM을 이용하여 판결 결과를 예측
- 평균적으로 2번 방법론이 1번 방법론에 비하여 더 높은 precision, recall을 보여주었을 뿐만 아니라, 변동이 덜한 그래프의 모습을 보여주어 학습의 안정성을 유추해볼 수 있음
- 한편, LLM을 사용한 3번 방법론은 GPU 자원이 충분치 못한 상황에서 이루어져 추가적인 학습과 hyperparameter tuning이 필요하다는 한계가 있지만, training loss와 달리 적절히 감소하지 않는 validation loss, 오버피팅이 발생한 것을 확인할 수 있었음
- 상대적으로 복잡도가 낮은 LLM을 사용한 경우도 학습의 형태가 비슷한 것으로 보아, 데이터의 노이즈가 많아 오버피팅이 발생한 것으로 추정됨
- 1번 방법론과 2번 방법론, 더욱 정제된 데이터셋을 사용한 3번 방법론은 각각 독립적인 모델들로, 양상을 학습 시, 높은 수준의 성능 향상을 기대해볼 수 있을 것으로 예상됨

THANK YOU

감사합니다