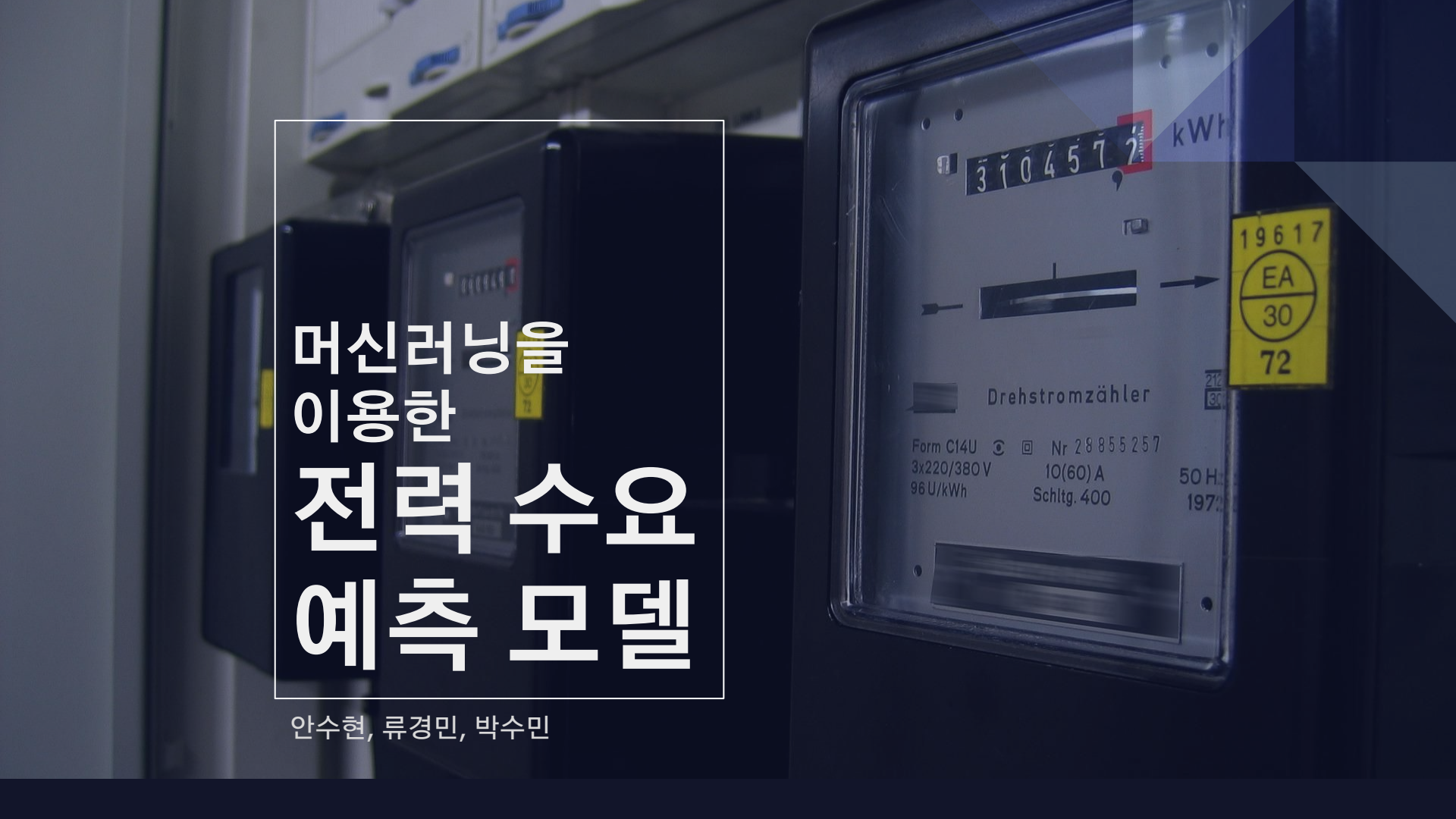


머신러닝을 이용한 전력 수요 예측 모델

안수현, 류경민, 박수민

ORDER

I. DACON COMPETITION

II. EDA

III. METRIC & PRE – PROCESSING

IV. MODELS

V. ACHIEVEMENT & CONCLUSION

DACON COMPETITION

기존 전력 사용 기록과 기상 데이터 등 공공 데이터를 이용하여,
각 가정 및 회사의 시간별, 일별, 월별 전력 사용량을 예측하라.

DACON COMPETITION

A. 예측
- 2018년 7월 1일 00시 ~ 24시: 24시간
- 2018년 7월1일 ~ 7월10일: 10일
- 2018년 7월 ~ 11월: 5개월

➢ 즉 각 세대(또는 상가)의 시간당, 일간, 월간 전력 사용량을 예측(X000의 형태를 가짐)

B. 평가 방법
- 지표(Metric): SMAPE(Symmetric Mean Absolute Percentage Error)
- 임시 랭킹 (Public Score) : 대회 중 Test 데이터의 50%로 채점
- 최종 랭킹 (Private Score) : Public Score에서 사용하지 않은 Test 데이터의 나머지를 합하여 채점(즉, 100%의 데이터 사용)

DATA OVERVIEW

주어진 시간 이후의
시간당, 일별, 월별 전력
사용량을 예측하라

TRAIN		WEATHER
인천 지역의 모 아파트 및 모 상가의 전력 사용량		인천 지역의 기상 예측 데이터
총 1300호의 Meter ID	총 200호의 Meter ID	2016년 7월 20일 00시부터 2018년 7월 1일 23시까지
2016년 7월 26일 11시부터 2018년 6월 30일 24시까지 시간당 전력사용량	2017년 7월 1일 00시부터 2018년 6월 30일 24시까지 시간당 전력사용량	기온, 강수량, 풍속(m/s), 습도(%), 적설(cm), 날씨(구름 많은, 맑음, 비 끝남 등), 전운량(10분위)
Dataset 1	Dataset 2	

WE WANT...

1. **실질적인 데이터 분석** 능력과 **모델 구현** 능력 측정

2. 정확한 전력 **수요 예측**과 그에 따른 **에너지 절약**

EXPLORATORY DATA ANALYSIS

- 결측치 분포도
- 데이터셋 군집화
- 변수 생성
- 상관관계 분석

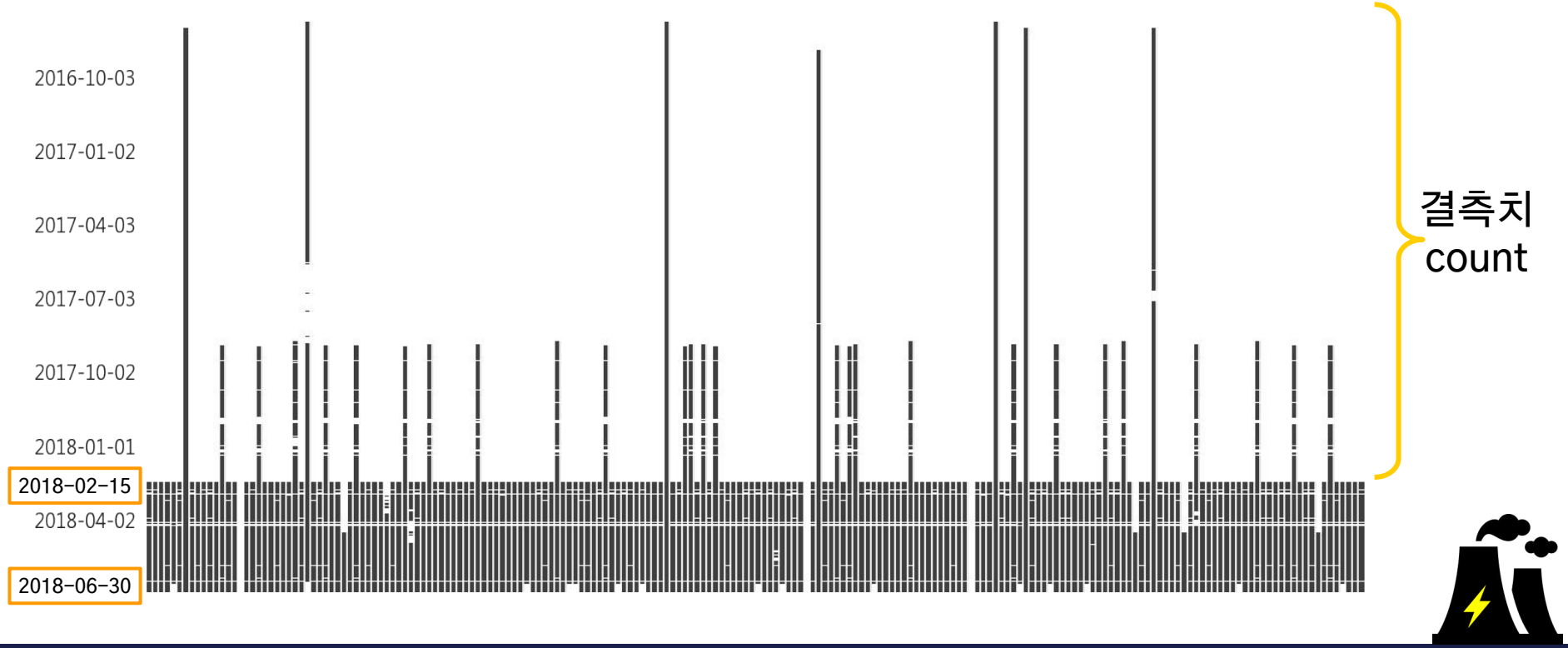
결측치 분포도

Time	X692	X1272	X553	X1299	X4	X598	X1003	X1010	X1216	X1047	X381	X466
2016-07-26 11:00												
2016-07-26 12:00												
2016-07-26 13:00												
2016-07-26 14:00												
2016-07-26 15:00												
2016-07-26 16:00												
2016-07-26 17:00												
2016-07-26 18:00												
2016-07-26 19:00												
2016-07-26 20:00												
2016-07-26 21:00												
2016-07-26 22:00												
2016-07-26 23:00												
2016-07-27 0:00												
2016-07-27 1:00												
2016-07-27 2:00												
2016-07-27 3:00												
2016-07-27 4:00												
2016-07-27 5:00												
2016-07-27 6:00												
2016-07-27 7:00												
2016-07-27 8:00												
2016-07-27 9:00												

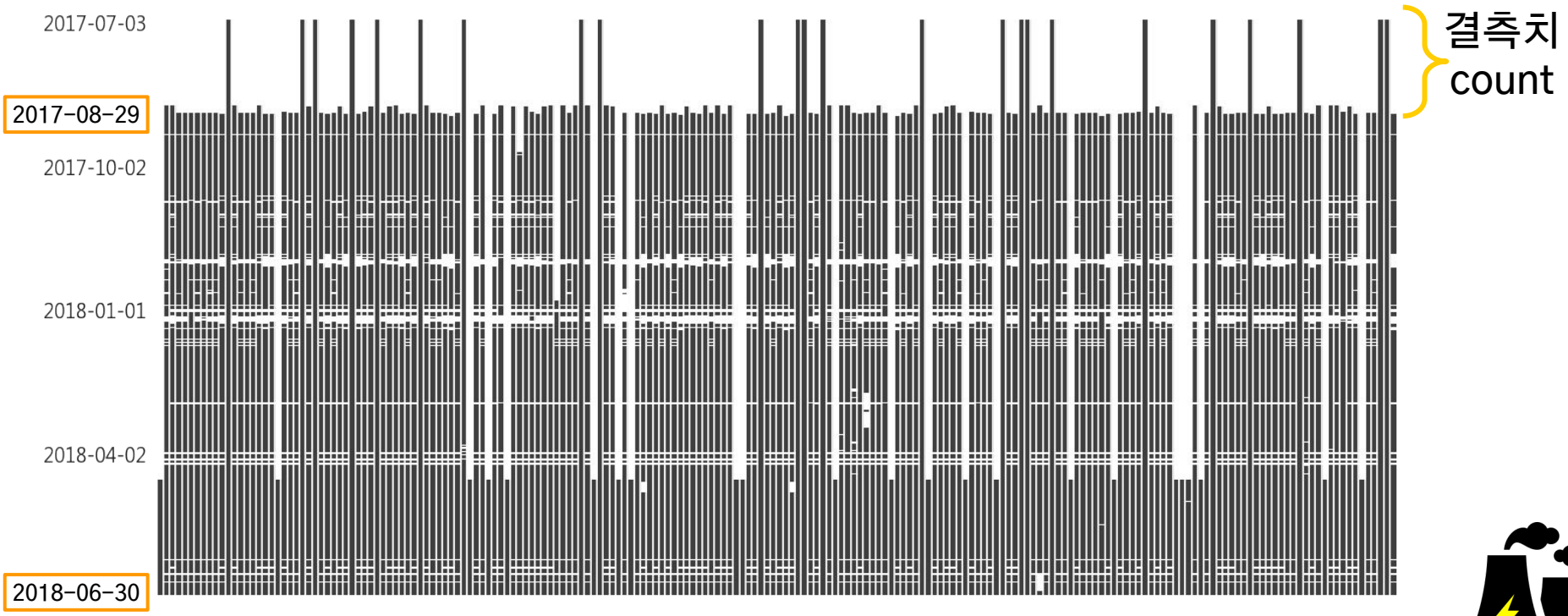
결측치 분포도

	Dataset 1	Dataset 2
총 데이터 수	21,981,700 (16909x1300)	1,752,000 (8760x200)
결측치 데이터 수	16,971,471	471,297
결측치(%)	77.2 %	26.9 %

Dataset 1 결측치 분포도 (200열 무작위 추출)



Dataset 2 결측치 분포도

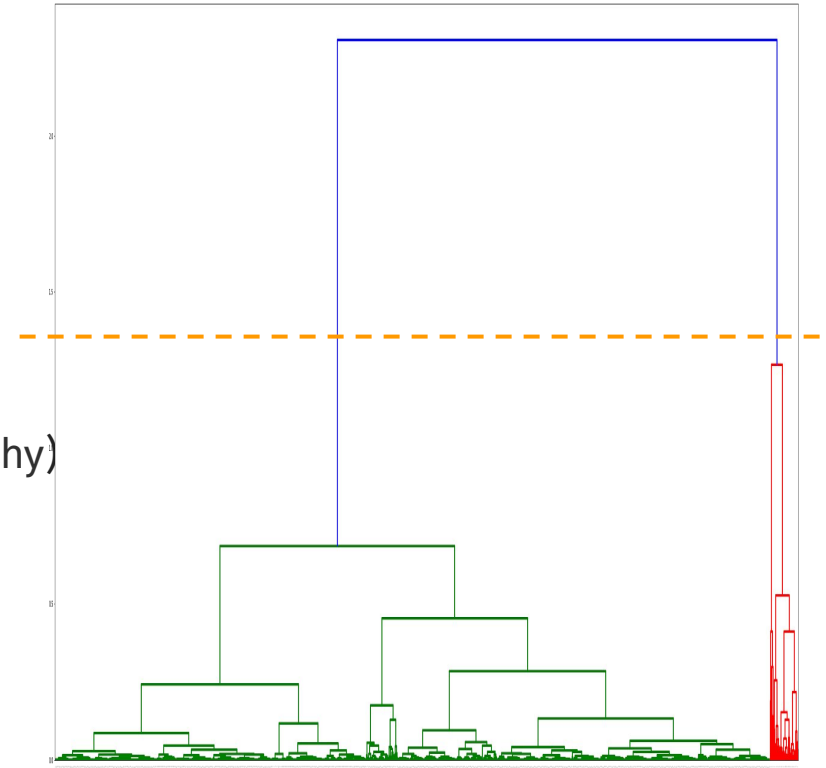


데이터셋 군집화

- Dataset 1, Dataset 2 통합 진행

  * Dataset1 결측치 문제로 2018.02.15에서 분할

- 계층적 군집화
  - Hierarchical Clustering (scipy.cluster.hierarchy)



Step 1. 동일 미터ID내 1일(24시간) 전, 7일(168시간) 전 동일시간 전력수요 비교

Step 2. 각 미터ID 간 1일전 비교값(ratio_1d) & 7일 전 비교값(ratio_7d) 사용하여 Distance 계산

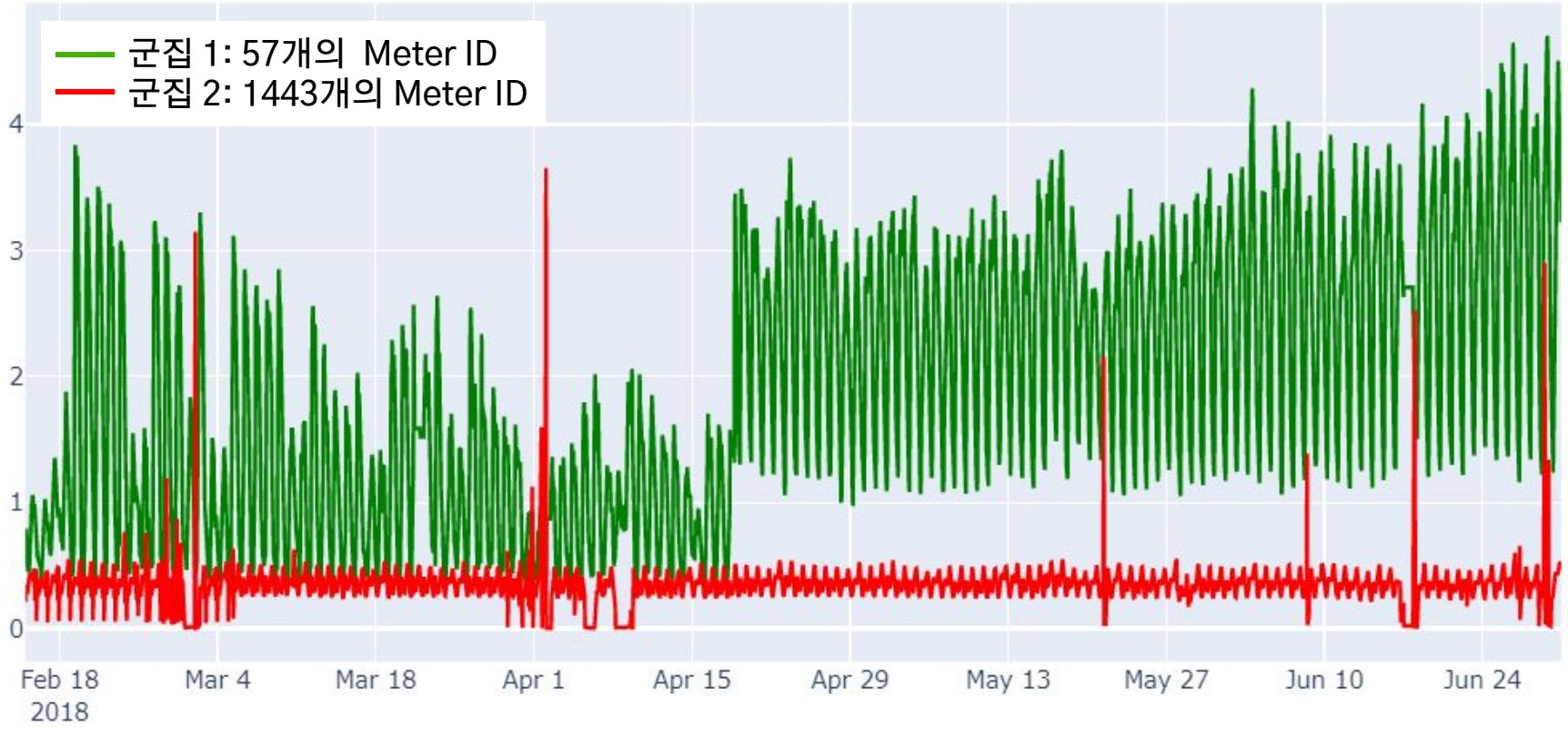
Time	X26
2017.8.10 1:00	0.323
2017.8.10 2:00	0.316
…	…
2017.8.16 1:00	0.407
2017.8.16 2:00	0.421
…	…
2017.8.17 1:00	0.33
2017.8.17 2:00	0.313

lag_24	lag_168
0.407	0.323
0.421	0.316

ratio_1d	ratio_7d
0.945	1.005
0.924	0.998

- ratio_1d = (X26 + 1)/(lag_24 + 1)
- ratio_7d = (X26 + 1)/(lag_168 + 1)

시간별 평균 전력 수요량


군집 1: 57개의 Meter ID
군집 2: 1443개의 Meter ID
4
3
2
1
0
Feb 18
2018
Mar 4
Mar 18
Apr 1
Apr 15
Apr 29
May 13
May 27
Jun 10
Jun 24

Feature Engineering

	Mth	Date	Day	hour	season_1	season_2	season_3	season_4	day_1	day_2	day_3	Temp	Humid	Wind	lag_24	lag_168
2018-02-24 23:00:00	2	24	5	23	0	0	0	1	0	0	1	-0.4	0.0	2.5	1.205	0.348
2018-02-25 00:00:00	2	25	6	0	0	0	0	1	0	0	1	-0.9	0.0	2.6	0.334	0.361
2018-02-25 01:00:00	2	25	6	1	0	0	0	1	0	0	1	-1.1	0.0	2.6	0.338	0.365
2018-02-25 02:00:00	2	25	6	2	0	0	0	1	0	0	1	-1.2	0.0	3.5	0.334	0.334

- 시간 파생 변수
  - 월, 일, 요일, 시간
- 시간 Dummy 변수
  - 계절 (season_1: 봄 ~season 4:겨울) , 요일구분(day_1: 월, day_2: 화~금, day_3: 토, 일)
- Time Lag
  - 24 시간(1일), 168시간(7일)

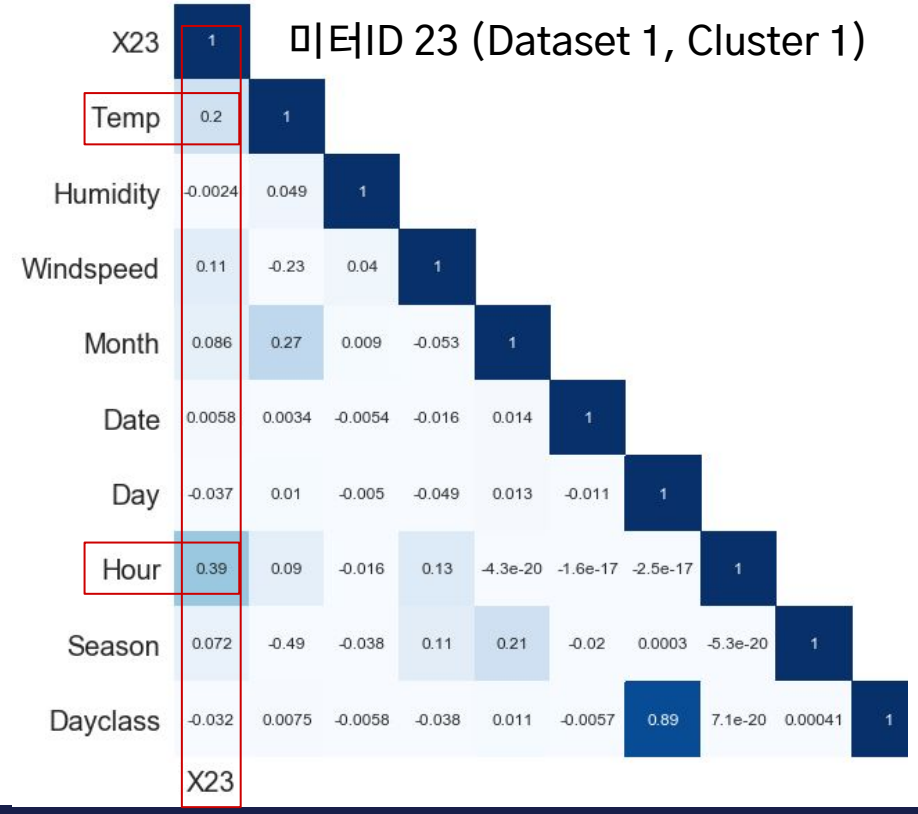
상관관계 분석

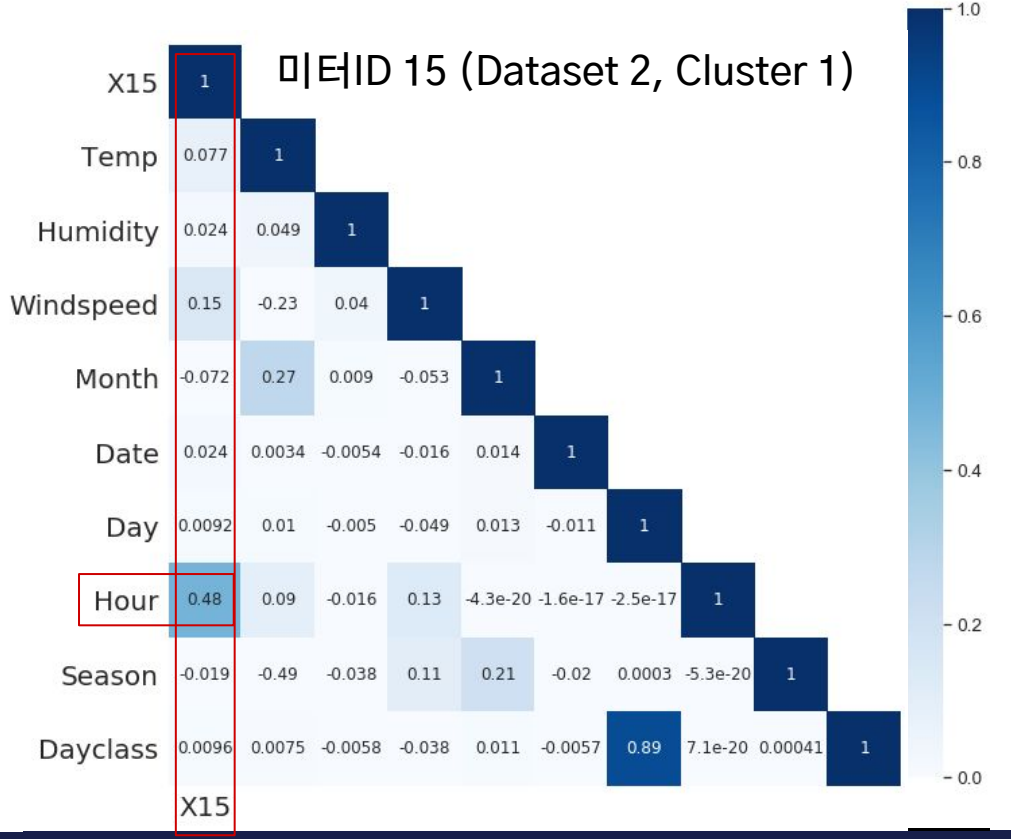
시간 파생변수와 기상 변수를 포함

각 클러스터별, 각 데이터셋에서 1개씩 무작위 추출

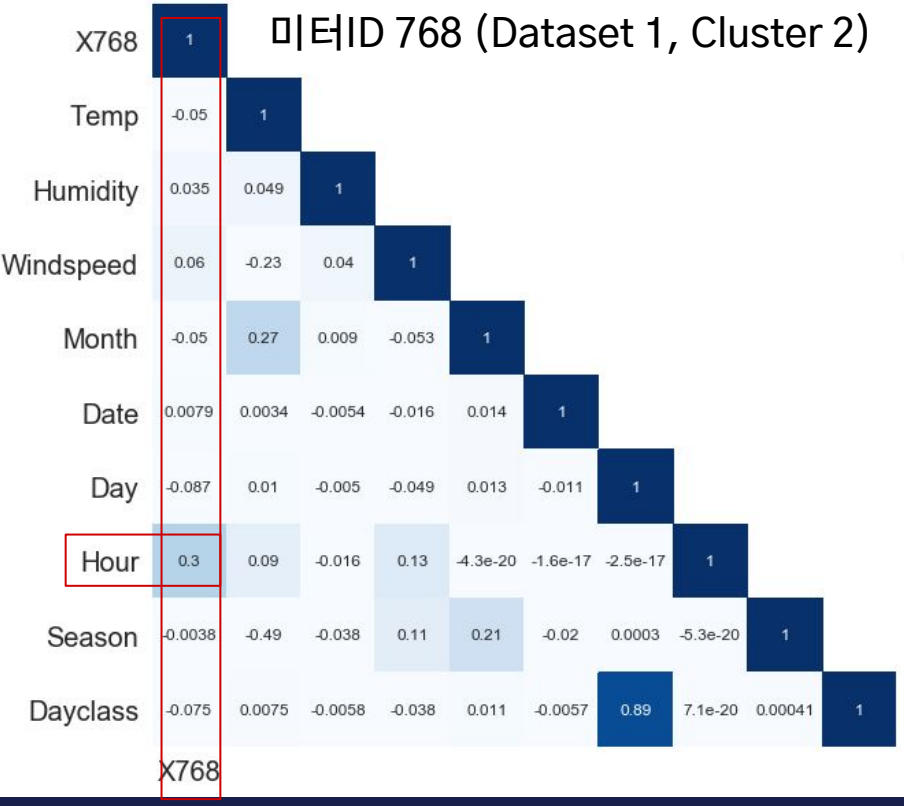
	Dataset 1	Dataset 2
군집 1	X23	X15
군집 2	X768	X231

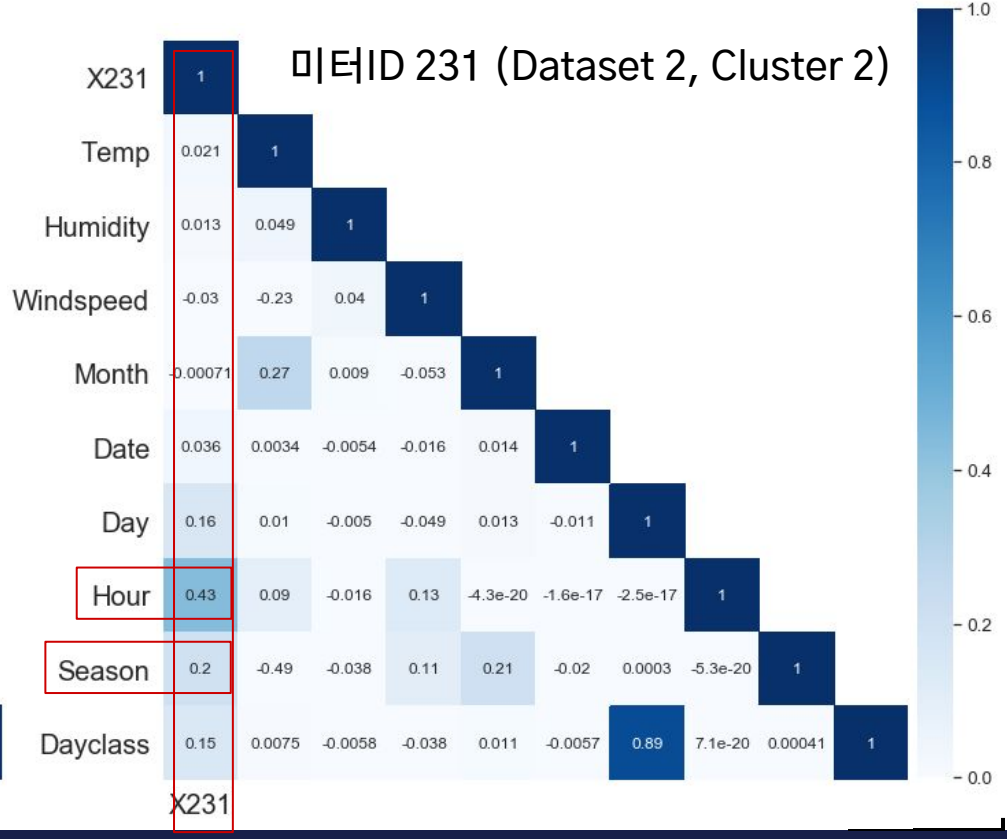
전력수요량과 기타 변수 간 상관관계





전력수요량과 기타 변수 간 상관관계





Small Conclusion

1. **NaN 처리** 방법?
2. **두 개의 군집**
3. 미터기의 **누적 전력량** 해결 방법?
4. ‘**Hour**’ 변수에 중점 맞추기
5. ‘Day’와 ‘Dayclass’의 **다중공선성**이 의심됨

METRIC & PRE-PROCESSING

- SMAPE
- 결측치 처리

METRIC

- SMAPE(Symmetric Mean Absolute Percentage Error) 사용

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

  - n = 모든 예측 개수, A_t = 실제값, F_t = 예측값
  - 백분율 오류를 기반으로 한 정확도 측정 방법
  - 양수 오차보다 음수 오차의 경우, 더 큰 가중치를 준다는 단점이 있음
    - Over-forecasting: A_1 = 100 and F_1 = 110 give SMAPE = 0.0238%
    - Under-forecasting: A_1 = 100 and F_1 = 90 give SMAPE = 0.0263%.

결측치 처리

- 데이터의 대부분이 결측치 (Dataset 1: 77% 이상, Dataset 2: 27%)
- 3가지 가설 제시
  - **최빈값**
  - **이동 평균** (48시간의 중앙값)
  - **요일별로 시간대 평균**
- 30개의 미터ID를 무작위 추출하여, 각각의 가설 적용

가설 평가

Linear Regression(sklearn.linear_model) 사용

2018. 06.30 (24시간) 예측

SMAPE 스코어

	스코어
최빈값	67.987
이동평균	63.696
요일별 시간대 평균	43.708

DACON 추천 전처리 기법

Test Data, 직전 시간의 전력사용량 값 높은 경우 해결 방법

현 대회에 제공이 되는 데이터에는 결측치나 이상치 (NA, 0인 값)이 다수 포함되어 있습니다.

이러한 값들을 처리하는 과정은 이번 대회에서 좋은 예측값을 만들기 위한 핵심적인 사항 중 하나입니다.

현재 예상이 되는 결측치 발생 경우를 살펴보자면

- 이전부터 측정기가 없었던 경우
- 직전 시간대 전력 사용량 값이 높아 이후 값들이 결측치가 되는 문제 (미터링 데이터 수집 시스템의 특징)

등 입니다.

Time	X18
. . .	
2017-08-25 03:00:00	0.28
2017-08-25 04:00:00	2.4
2017-08-25 05:00:00	NaN
2017-08-25 06:00:00	NaN
2017-08-25 07:00:00	NaN
2017-08-25 08:00:00	0.989
. . .	

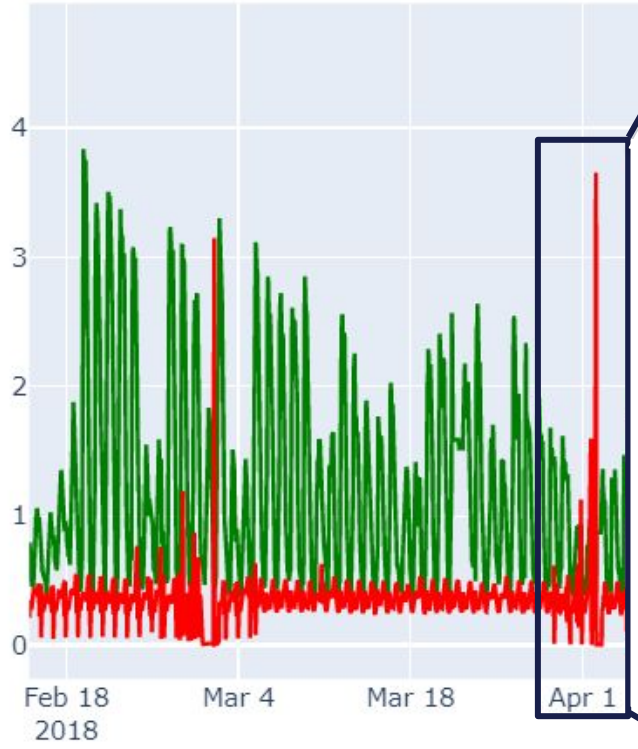
중앙값: 0.7

2.4 > 중앙값

2.4 이후 NaN 수 : 3

2.4 / (3+1) = 0.6

Time	X18
. . .	
2017-08-25 03:00:00	0.28
2017-08-25 04:00:00	0.6
2017-08-25 05:00:00	0.6
2017-08-25 06:00:00	0.6
2017-08-25 07:00:00	0.6
2017-08-25 08:00:00	0.989
. . .	



중앙값	0.434	0.321	0.312	0.271	0.200	0.367	0.311	0.08	0.239

Time	X303	X241	X435	X402	X352	X305	X350	X326	X299
2018.4.1 23:00	0.802	0.354	0.623	0.364	0.082	0.911	0.466	0.03	0.23
2018.4.2 0:00	0.539	0.296	0.427	0.159	0.062	0.638	0.262	0.018	0.138
2018.4.2 1:00	0.676	0.317	0.529	0.239	0.068	0.691	0.234	0.027	0.214
2018.4.2 2:00	4.457	3.987	4.387	2.671	0.901	4.626	3.238	0.518	4.156
2018.4.2 3:00									
2018.4.2 4:00									
2018.4.2 5:00									
2018.4.2 6:00									
2018.4.2 7:00									
2018.4.2 8:00									
2018.4.2 9:00									
2018.4.2 10:00									
2018.4.2 11:00									
2018.4.2 12:00									
2018.4.2 13:00									
2018.4.2 14:00	0.07	0.201	0.204	0.185	0.084	0.203	0.234	0.021	0.346
2018.4.2 15:00	0.344	0.351	0.259	0.226	0.132	0.311	0.331	0.036	0.197

군집 2에 속하는 칼럼의 예시 (데이터셋 2)
2018.04.02 02:00 ~ 12:00
NaN 수: 11

결측치가 지나치게 많을 경우 최종값이 0에 수렴하며 전체적으로 지나치게 일정한 패턴을 보인다.

누적값을 모든 시간에 동일 분할하는 것이 문제가 있다고 판단

Time	X18
. . .	
2017-08-25 03:00:00	0.28
2017-08-25 04:00:00	2.4
2017-08-25 05:00:00	NaN
2017-08-25 06:00:00	NaN
2017-08-25 07:00:00	NaN
2017-08-25 08:00:00	0.989
. . .	

요일별 시간별 평균 구하기

요일 시간 가중치: 결측시간 평균 정규화 (총 합 =1)

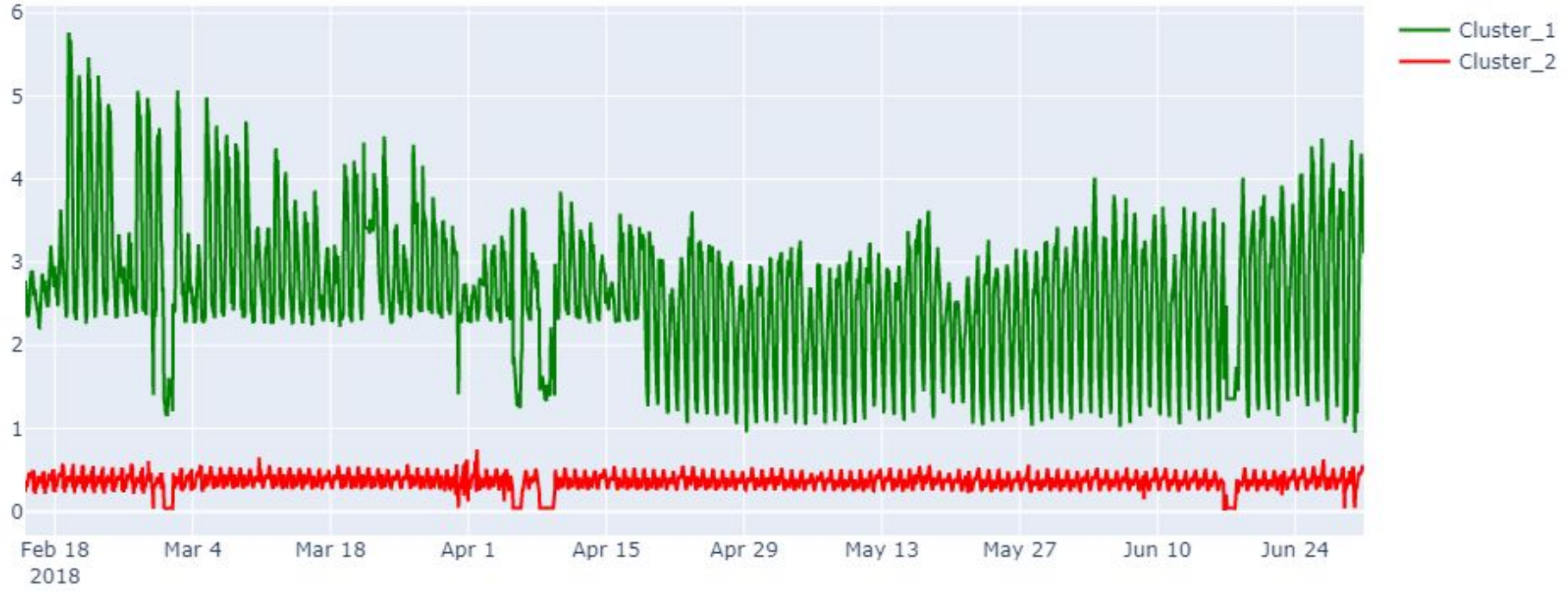
X18_금요일		
시간	평균	가중치
04	0.2	0.13
05	0.25	0.17
06	0.4	0.28
07	0.6	0.41
합계	1.45	1

중앙값: 0.7

2.4 > 중앙값
2.4이후 NaN 수: 3

시간 가중치 x 2.4

Time	X18
. . .	
2017-08-25 03:00:00	0.28
2017-08-25 04:00:00	0.312
2017-08-25 05:00:00	0.408
2017-08-25 06:00:00	0.672
2017-08-25 07:00:00	0.984
2017-08-25 08:00:00	0.989
. . .	


6
5
4
3
2
1
0
Feb 18
2018
Mar 4
Mar 18
Apr 1
Apr 15
Apr 29
May 13
May 27
Jun 10
Jun 24
Cluster_1
Cluster_2

MODELS

- ARIMA
- LSTM
- XGBoost
- LightGBM
- NGBoost

모델 선정 전 고려사항

- 고려사항 :
  - (1) 시계열 데이터에 사용 가능한지. (2) 예측율이 높은지

시계열 데이터 사용 가능	예측율이 높음
1. AR, ARMA, ARIMA 와 같은 머신러닝 모델 2. LSTM 과 같은 딥러닝을 사용한 모델	1. 각종 대회들에서 성과가 좋은 모델들 고려(XGBoost, Lightgbm) 2. 그 외에 새로운 모델은 없는지 탐색

ARIMA

- 시계열을 예측할 때 가장 널리 사용하는 방법
- 과거가 현재에 영향을 미친다는 자기상관

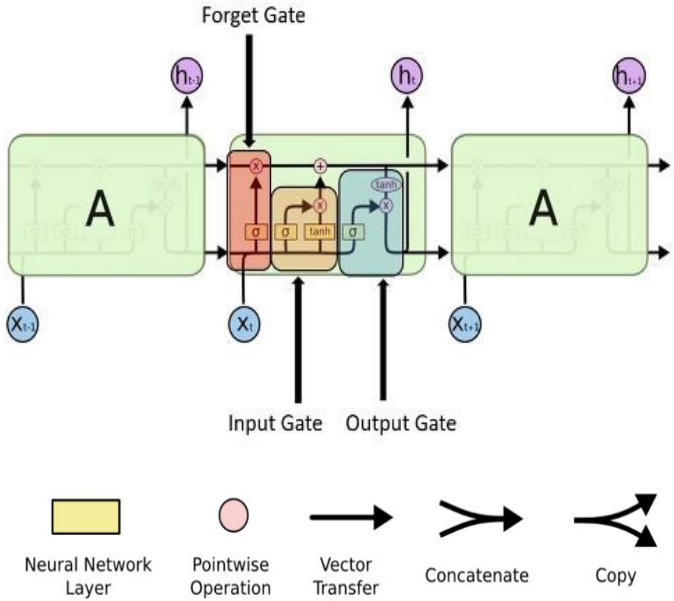
  \+ 시간이 지날수록 나타나는 경향에 대해 파악하는 이동평균

  \+ 추세(모멘텀)

- 증권시장 등 경제분야에서 많이 응용
- 특징
  - 시계열 자료외에 다른 자료가 없더라도 **변동 상태를 확인 가능**
  - 어떤 시계열에도 적용이 가능, 특히 자료의 **변동이 빠를 때 민감하게 반영**
- 단점
  - 활용자의 능력에 따라 **성능 차이 발생**
  - 이상치 발생시 예측 불가능(흐름에 대한 예측이기 때문)

LSTM

- 이전의 정보를 현재의 문제 해결에 활용
- 기간이 긴 데이터를 학습하더라도 과거의 학습내용이 사라지지 않고 예측에 영향을 줌
- 세가지 핵심
  - 무엇을 쓰고 > Input Gate
  - 무엇을 읽고 > Output Gate
  - 무엇을 잊을 것인가 > Forget Gate



XGBoost vs Lightgbm

	XGBoost	LightGBM
장점	- 높은 예측율 - 예측율에 비해 학습속도가 빠름	- XGboost 에 비해 더 빠른 예측 수행 - 더 작은 하드웨어 사용량으로 가능
단점	- 다른 모델에 비해 빠른 편 아님 - 규제에도 불구, 과적합 문제 여전 - 하드웨어 성능 필요	- 데이터가 적을시 과적합 문제 발생

NGBoost

스탠포드 머신러닝 그룹에서 개발한 알고리즘.
확률적 예측을 위해 만들어진 알고리즘.

ACHIEVEMENT & CONCLUSION

- 모델 실행 결과
- 대외 성과
- 결론

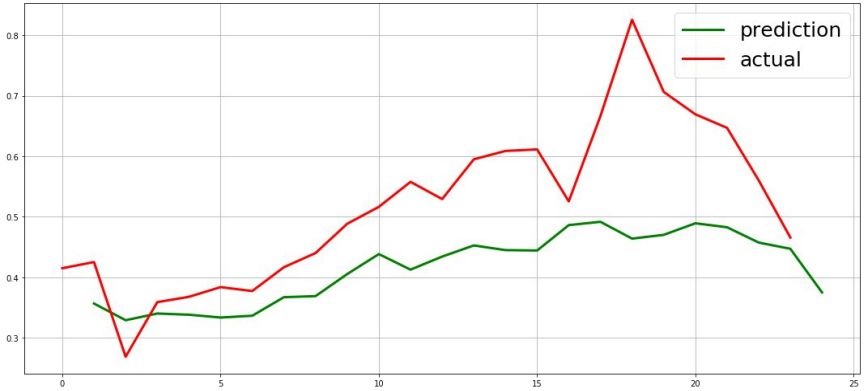
실행 및 결과

- 데이터셋 2에서 30열 무작위 추출
  - 2018.06.30 0시~23시 (24시간) 예측
- 베이스라인
  - Linear Regression (42.919, 00:42)

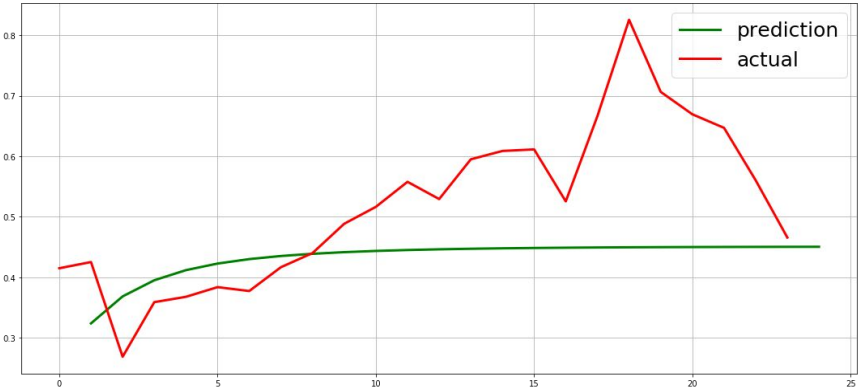
	스코어	시간
ARIMA	44.157	2:22
XGBRegressor	41.976	1:00
LGBMRegressor	34.700	00:25
NGBoost	33.477	11:47

대회 종료 이후 적용

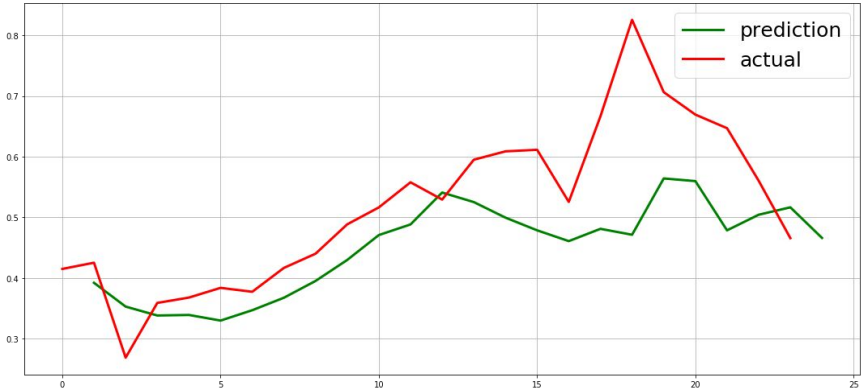
Linear Regression (스코어: 42.919)



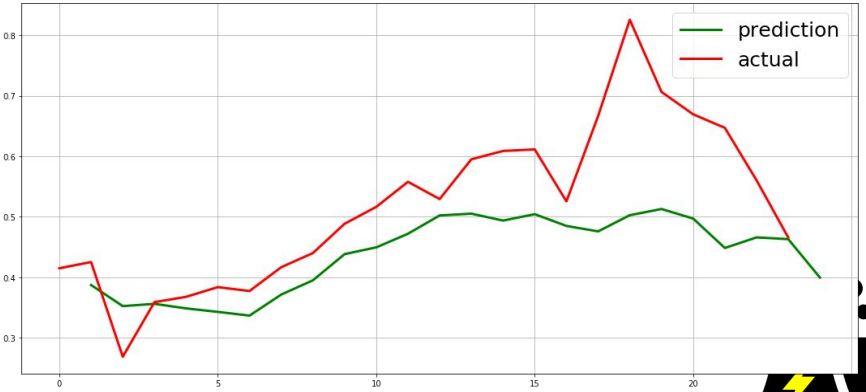
ARIMA (스코어: 44.157)



XGBM Regressor (스코어: 41.976)



Light GBM Regressor (스코어: 34.700)



최종 실행

- 파라미터 최적화 사용
  - GridSearchCV (sklearn.model_selection 패키지)
- 시간별 예측
  - Gradient Boosting 알고리즘 사용 (LGBM / NGBoost)

2018.7.1
00~23시

	Mth	Date	Day	hour	season_2	dayClass_3	season_1	season_3	season_4	dayClass_1	dayClass_2	Temp	Humidity	WindSpeed
2018-07-01 00:00:00	7	1	6	0	1	1	0	0	0	0	0	23.7	0.0	1.7
2018-07-01 01:00:00	7	1	6	1	1	1	0	0	0	0	0	23.7	0.0	2.0
2018-07-01 02:00:00	7	1	6	2	1	1	0	0	0	0	0	23.8	0.0	2.0
...														
2018-07-01 21:00:00	7	1	6	21	1	1	0	0	0	0	0	21.3	4.2	4.5
2018-07-01 22:00:00	7	1	6	22	1	1	0	0	0	0	0	21.3	6.2	4.8
2018-07-01 23:00:00	7	1	6	23	1	1	0	0	0	0	0	21.2	4.6	4.1

- 일별, 월별 예측
  - ARIMA 사용
  - 일별 및 월별 데이터 압축시 Gradient Boosting에서는 overfitting 등의 문제 발생

대외 성과

- 모델 학습 및 최종 제출
  - 데이터셋 2 사용 (최종 예측하고자 하는 200개의 Meter ID)
  - submission.csv

    시간별 : 2018년 7월 1일 00시 ~ 23시
    일별 : 2018년 7월 1일 ~ 7월 10일
    월별 : 2018년 7월 ~ 11월

	대회 종료 당시	이후 업데이트
모델	LGBM Regressor & ARIMA	NGBoost & ARIMA
스코어	33.571	32.430
순위	21 / 124	17 / 127

결론

- ❖ 앞으로의 발전 방향
  - 데이터셋 1의 활용
  - 전이학습
  1. 딥러닝 모델 (상태유지 LSTM 스택)
  2. 각 군집 내 데이터셋 1에서 샘플링한 데이터를 사용하여 군집별 모델 Pre-train
  3. 군집별 모델을 데이터셋 2(최종예측 미터ID)의 각 열에서 Fine-tuning 및 예측 진행

결론

- 클린 데이터의 중요성 (Garbage In Garbage Out)
- 데이터 정제부터 모델 최적화까지 데이터 사이언스 업무 능력 향상
- 효율적인 빅데이터 분석기술을 적용한 전력수요예측 시뮬레이션을 개발하여

  최종 17위, 상위 13% 에 이르는 성과

감사합니다

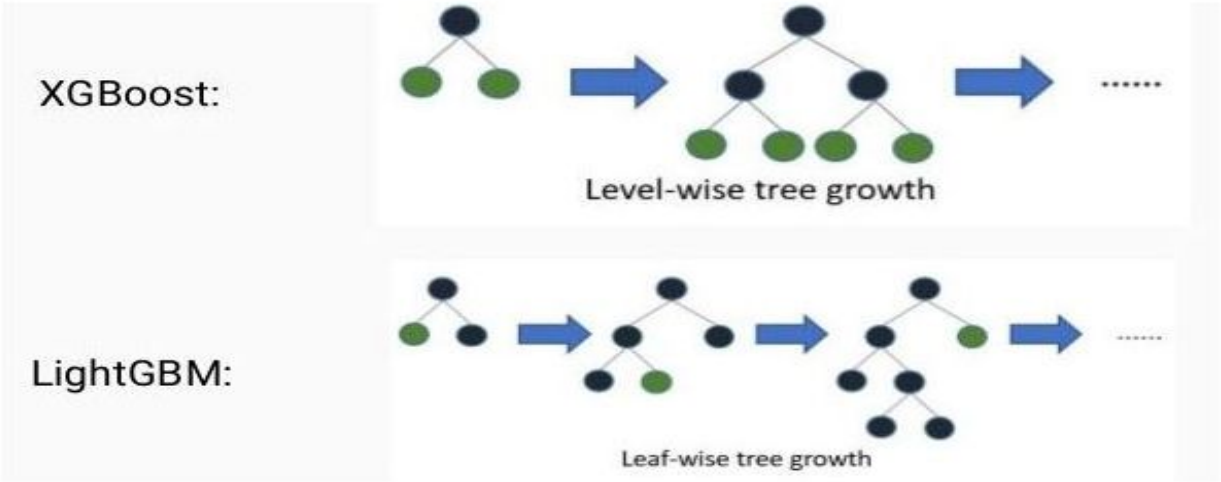
Q & A

부록


탐색적
데이터
분석
STEP 1
데이터
변형
STEP 2
데이터
전처리
STEP 3
모델링
STEP 4
모델
평가
STEP 5

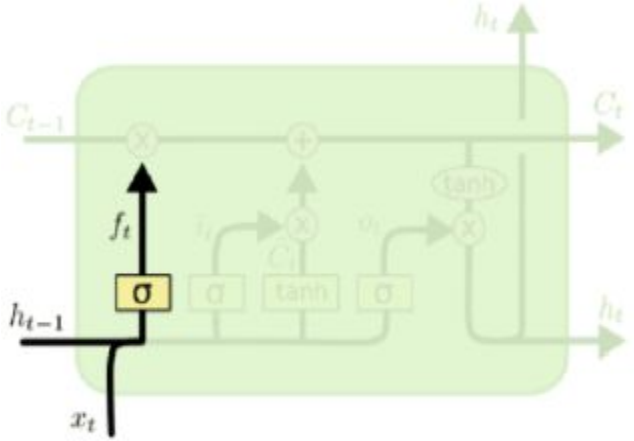
데이터 분석 방법론

XGBoost vs Lightgbm



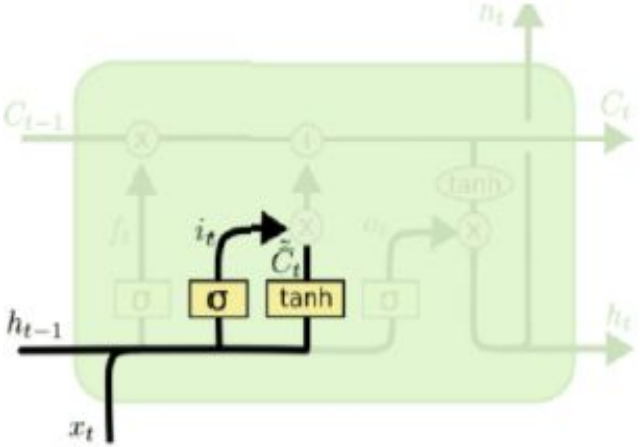
LSTM

- Drop 정보 선택 과정
- Forget Gate Layer
- 시그모이드 레이어로 만들어짐



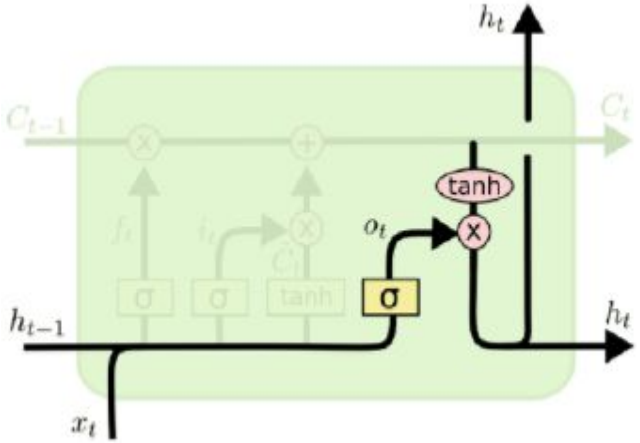
LSTM

- 저장 정보 선택과정
- Input Gate Layer
- 업데이트 Data 결정
- Cell State에 저장



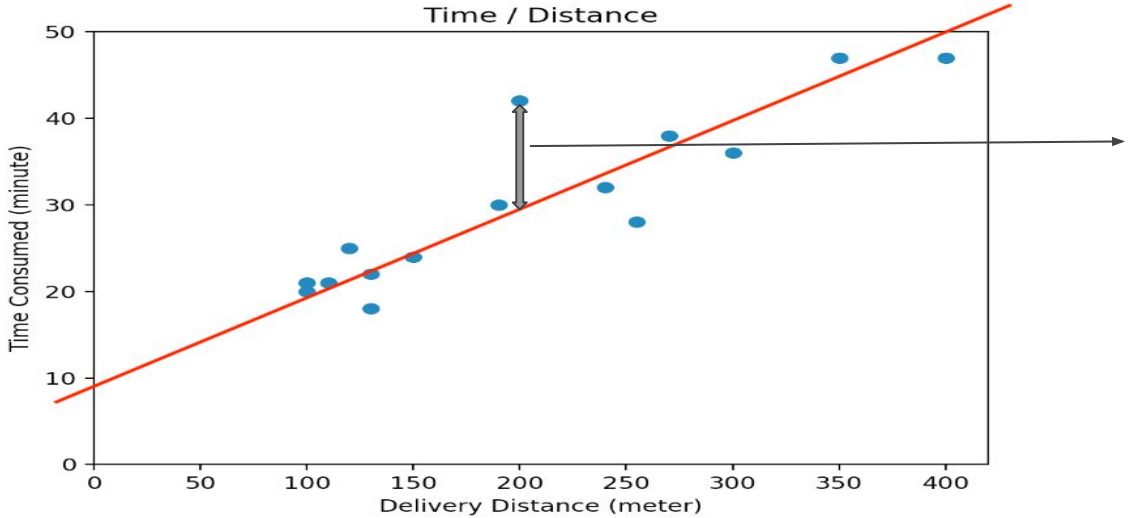
LSTM

- 출력 정보 선택과정
- Output Gate Layer
- 출력 Data 결정
- 다음 노드로 전파



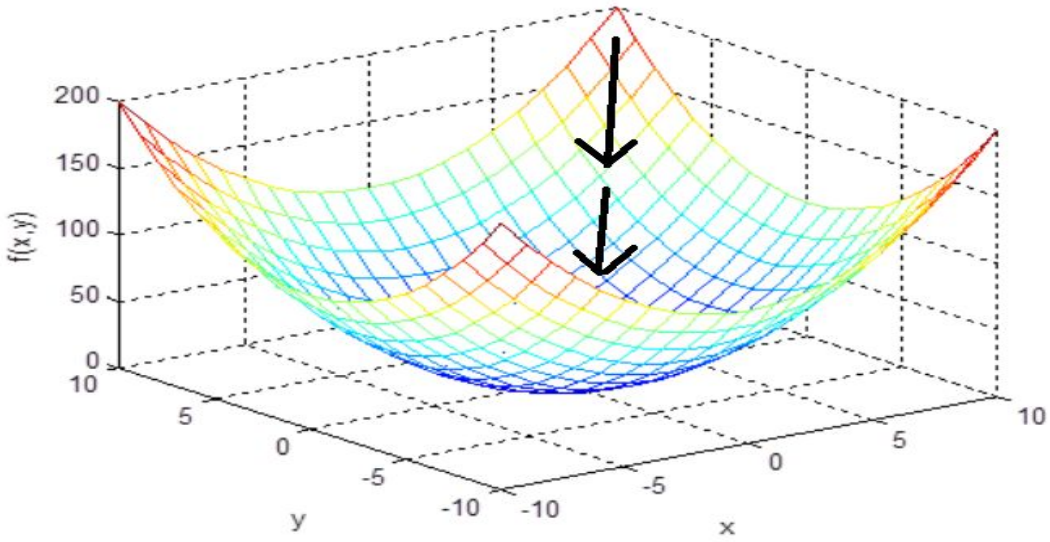
Loss Function(손실함수)

학습(training) 데이터에 어떤 특정 파라미터(parameter/weight)들을 가지고 실제 class와 얼마나 잘 일치하는지에 따라 그 특정 파라미터(parameter/weight)들의 질을 측정하는 **손실함수(loss function)**. 여러 종류의 손실함수(예를 들어, Softmax/SVM)가 있다.



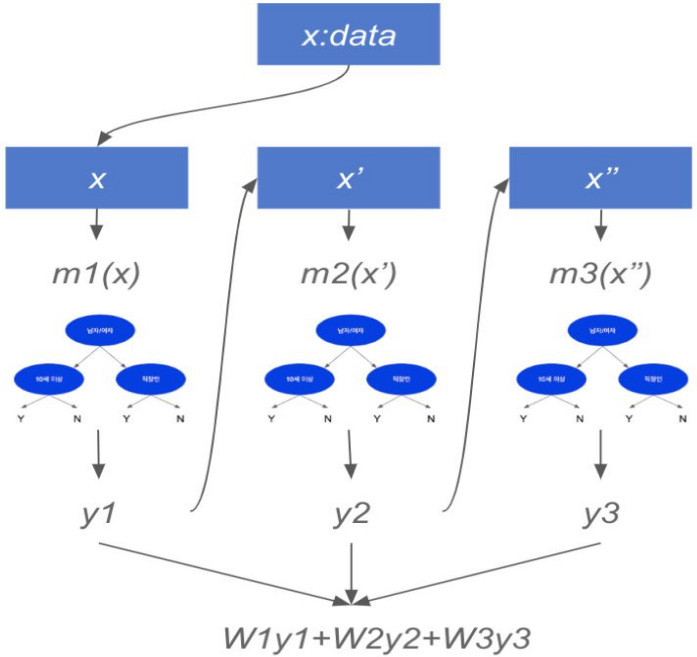
손실함수를 측정하는 방법!!

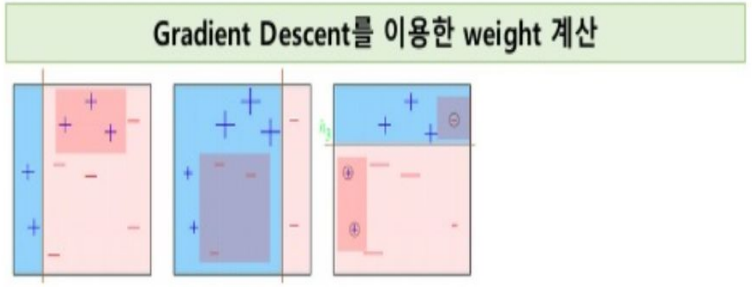
Gradient Descent (경사하강)



직교좌표계에서 거리를 구하는 공식을 활용한 손실함수 그래프

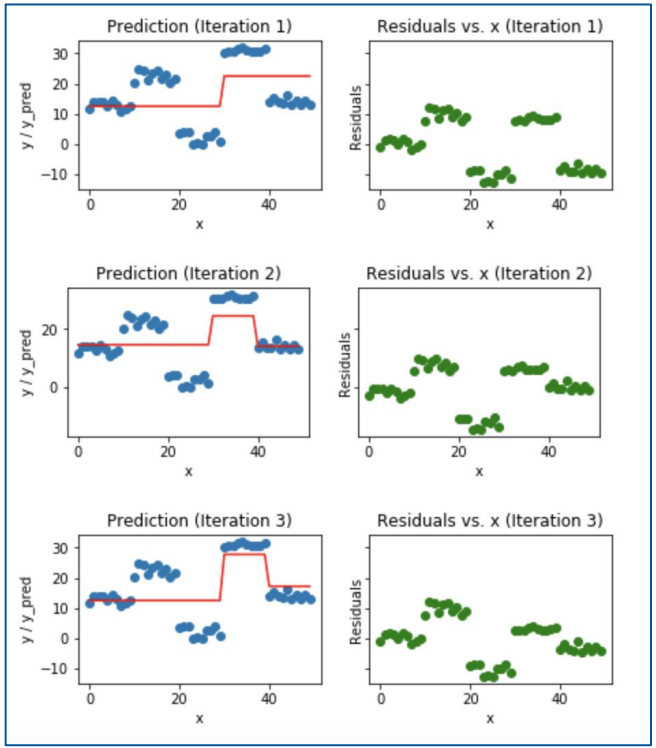
boosting 이란?

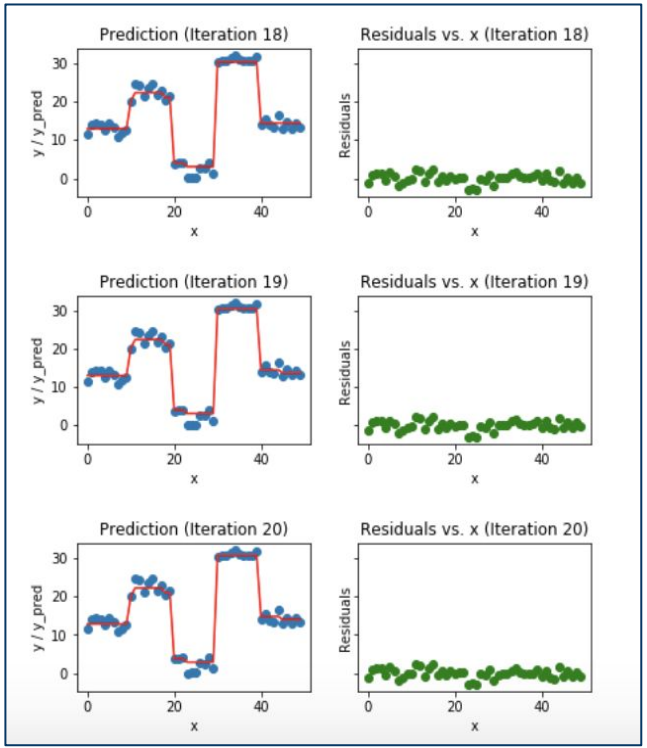




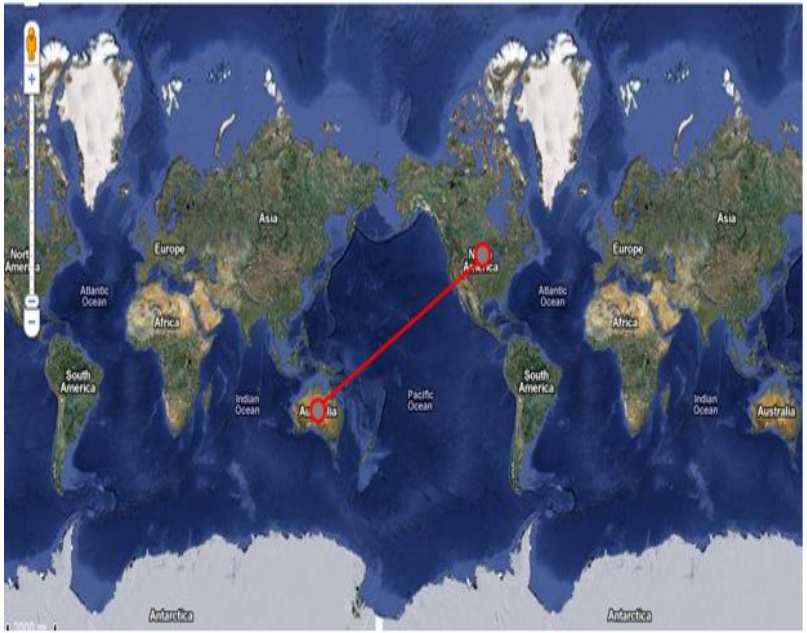
- 1번 Wezk model에서는 3개의 오분류(에러)가 발생
- 2번은 3개 에러를 제대로 분류하기 위해 가중치 부여. (다시 3개 에러 생김)
- 3번은 다시 3개 에러를 해결하기 위한 모델 생성 (다시 3개 에러 발생)
- 최적의 weight(가중치)를 찾을 때 까지 반복

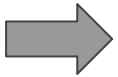
Gradient Boosting 이란?





Natural Gradient 란?





NGBoost

$$\mathbf{w}(k+1) \quad = \quad \mathbf{w}(k) - \mu(k)\mathbf{G}^{-1}(\mathbf{w}(k))\frac{\partial \mathcal{J}(\mathbf{w}(k))}{\partial \mathbf{w}}, (16)$$

Natural Gradient의 값 업데이트 공식

$$\text{유클리드거리 } d(A,B) = \sqrt{(c-a)^2 + (d-b)^2}$$

기본적인 거리를 구하는 공식