

## [ 전력사용량 예측 SI경진대회 ]

✓대회: [전력사용량 예측 SI 경진대회 - DACON](https://dacon.io/competitions/official/235736/overview/description)

<https://dacon.io/competitions/official/235736/overview/description>

✓참고 코드: [j\\_sean팀 | Private 1위\(5.0293\) | XGBoost 단일 모형 - DACON](https://dacon.io/competitions/official/235736/codeshare/2877?page=1&dtype=recent)

<https://dacon.io/competitions/official/235736/codeshare/2877?page=1&dtype=recent>

### 데이터

#### a. [train.csv](#):

60개 건물들의 2020년 6월 1일부터 2020년 8월 24일까지의 데이터  
건물번호(num), date\_time, 전력사용량, 기온, 풍속, 습도, 강수량, 일조, 비전기냉방설비운영, 태양광보유  
1시간 단위로 제공  
전력사용량 포함

#### b. [test.csv](#):


60개 건물들의 2020년 8월 25일부터 2020년 8월 31일까지의 데이터  
3시간 단위로 제공(강수량은 6시간 단위로 제공)  
전력사용량 미포함

#### c. [sample\\_submission.csv](#):

### 코드흐름

#### 1. 라이브러리 로드

#### 2. 데이터 전처리

- 변수들을 영문 명으로 변경
- 시간 Series -> datetime으로 시간 변수들 생성
- 전력소비량의 건물별, 요일별, 시간대별 평균 / 건물별 시간대별 평균/ 건물별 시간대별 표준편차 변수 추가  
(여러 평균, 표준편차 등의 통계량 생성 후 최종적으로 성능 향상에 도움이 된 3개 변수만 추가)
- 공휴일 변수 추가
- 시간(hour) -> cyclical encoding하여 변수 추가 후 삭제 : :  
 (np.sin/cos(2\*np.pi\*train.hour/24))
- CDH(Cooling Degree Hour) & THI(불쾌지수) 변수 추가
- 건물 별 모델 생성 시 무의미한 태양광 발전 시설 / 냉방시설 변수 삭제
- sktime library로 마지막 일주일 validation set(검증셋)으로 설정, plot\_series로 시각화

#### 3. 모델: XGBoost

- XGBoost 선정: 시계열 데이터에 좋은 성능을 보임
- 평가 Metric은 **SMAPE**로 함. 이는 실제값보다 작게 추정할 때 좋지 않으므로 전력사용량을 작게 예측하면 실제로 큰 문제가 될 수 있음을 반영
- 일반 mse를 objective function으로 훈련할 때 과소추정하는 건물들이 있음. 따라서 objective function(목적함수) 재정의  
: residual이 0보다 클 때(실제 값보다 낮게 추정할 때) alpha 만큼의 가중치를 곱해 반영함, gradient(1차 미분함수)/hessian(2차 미분함수) 정의해 return-> XGBoost 훈련
- 기본 목적함수 사용시 SMAPE: 12.7705
- weighted 사용시 SMAPE: 9.9390

#### 4. 모델 튜닝

- 전체 파라미터 튜닝 시 시간이 오래걸리므로 모델 내 파라미터는 gridsearchCV로 튜닝, n\_estimators/weighted\_mse의 alpha값은 early\_stopping으로 튜닝,

#### 5. test Inference 테스트 추정

- train set와 동일하게 test 데이터 전처리
- seed ensemble: seed의 영향을 제거하기 위해 6개의 seed로 훈련, 예측해 예측값의 평균을 구함

#### 6. Post processing 후처리

- 과도한 과소적합을 막기 위해 예측값을 후처리함
- 예측 주로부터 직접 4주의 건물별 요일별 시간대별 전력소비량의 최솟값을 구한 귀 test set의 같은 건물 요일 시간대의 예측값과 비교하여 예측값보다 작으면 최솟값으로 예측값을 대체함
- public score 0.01, private score 0.08의 성능 향상

#### 7. 제출

### 배운점

- 칼럼 별 여러 통계량을 구한 뒤 새로운 변수 추가 시도를 여러 번 해보면 예측에 도움이 되는 피처를 찾을 수 있다.
- 시각화 코드 결과 해석이 조금 어렵지만 성능 향상 결과를 파악하기 좋은 것 같다.
- SMAPE의 평가방법은 과소적합에 취약하다는 특징으로 데이터 또한 과소적합에 불리하다는 인사이트를 얻을 수 있다.
- 과소적합을 막기 위한 전처리, 후처리, 모델 튜닝 과정을 알 수 있었다.
- custom한 Loss function 사용으로 과소추정에 대해 높은 가중치를 부여할 수 있다.
- 전체 데이터 / 모델 별 데이터 각각 모델링을 한 후 성능 좋은 것으로 채택한다.