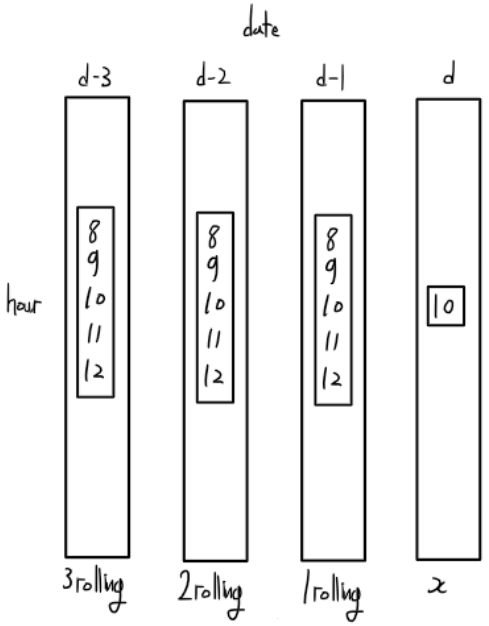
Feature

데이터가 작고, 8월 말과 비슷한 온도의 데이터가 충분히 있다고 판단하여 XGBoost로 훈련하였으나 모든 변수들은 NN을 염두에 두고 생성하였습니다. 추후에 데이터가 충분히 쌓인다면 범주형 변수를 제외한 나머지를 모두 표준화하여 NN으로 훈련을 시도해볼 수 있습니다.

1. whc
2. cos_hour, sin_hour
3. temperature_squared, THI, humidity_squared

Temperature-Humidity Index (THI) 또는 Discomfort Index(DI)

4. temperature_squared_mean, THI_mean



$$i\ rolling_h^d = \frac{x_{h-2}^{d-i} + x_{h-1}^{d-i} + x_h^{d-i} + x_{h+1}^{d-i} + x_{h+2}^{d-i}}{5}$$

$$2\ rolling_{10}^d = \frac{x_8^{d-2} + x_9^{d-2} + x_{10}^{d-2} + x_{11}^{d-2} + x_{12}^{d-2}}{5}$$

$$1\ rolling\_mean_h^d = \frac{x_h^d + 1\ rolling_h^d}{2}$$

$$12\ rolling\_mean_h^d = \frac{x_h^d + 1\ rolling_h^d + 2\ rolling_h^d}{3}$$

$$123\ rolling\_mean_h^d = \frac{x_h^d + 1\ rolling_h^d + 2\ rolling_h^d + 3\ rolling_h^d}{4}$$

$$mean_h^d = \frac{x_h^d + 1 rolling\_mean_h^d + 12 rolling\_mean_h^d + 123 rolling\_mean_h^d}{4}$$

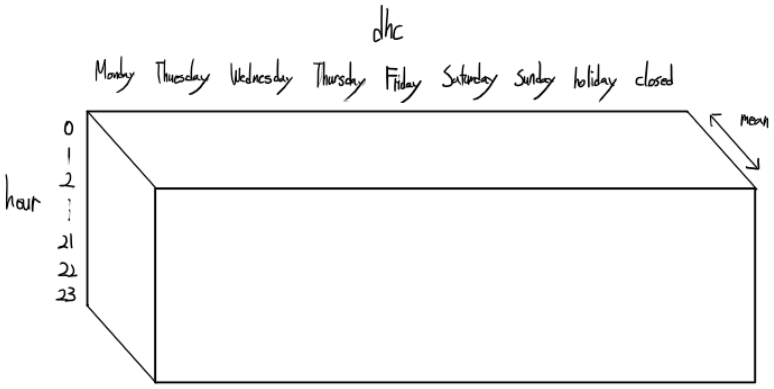
rolling 은 어제 혹은 2, 3 일 전 비슷한 시간대(5window)에 느꼈던 온도, 습도의 정보를 포함합니다. 당일 온도, 습도가 중요한 건물이 있고, 2, 3 일 전의 온도, 습도가 중요한 건물이 있었습니다.
아래 표는 건물별로 rolling 변수들과 power_log1p 간 상관계수를 구한 후, 평균을 낸 값입니다.

	temperature_squared	THI	humidity_squared
vanilla	0.521688	0.487043	-0.275728
1rolling	0.523406	0.483777	-0.273691
2rolling	0.533213	0.487919	-0.279840
3rolling	0.536625	0.490460	-0.268420
1rolling_mean	0.554868	0.505015	-0.314470
12rolling_mean	0.578910	0.517876	-0.338940
123rolling_mean	0.596524	0.527870	-0.350416
mean	0.576859	0.518419	-0.333736
power_log1p	1.000000	1.000000	1.000000

건물에 따라 다르지만 일반적으로 1rolling_mean, 12rolling_mean, 123_rolling_mean, 위의 표에 표시되지 않았지만 1234rolling_mean 까지 상관계수가 점차 증가하는 모습을 보였습니다. feature 마다 1rolling_mean, 12rolling_mean, 123rolling_mean 3 개의 rolling_mean 들을 변수로 추가할 수 있습니다. 하지만 사용하는 rolling_mean 의 수가 증가할수록 전체 CV 는 개선되었으나 건물마다 CV 가 불안정한 모습을 보였습니다. 시간을 고려하여 건물마다 세부적인 변수를 생성하는 것은 고려하지 않았고, 되도록 공통적인 변수를 만들어야 했습니다. 따라서 {feature}_1rolling_mean, {feature}_12rolling_mean, {feature}_123rolling_mean 모든 정보를 포함한 {feature}_mean 변수를 생성하였습니다. {feature}_mean 변수만을 사용할 때, 모든 건물에서 C 가 일관되게 개선되었고, 전체 CV 역시 가장 많이 개선되었습니다.

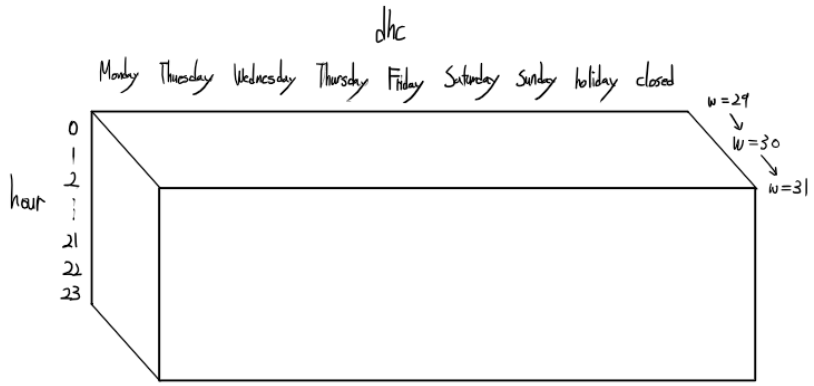
5. power_log1p_stdd_mean

예) 월요일 1 시 전력량들의 평균을 월요일 1 시의 변수로 사용



6. power_log1p_stdd_shift

예) 지난주 월요일 1 시의 전력량을 이번주 월요일 1 시의 변수로 사용

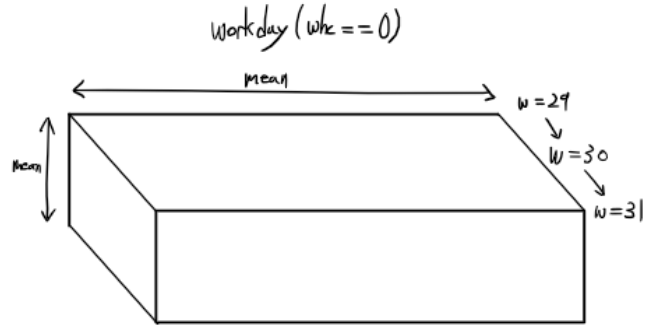


power_log1p_stdd_mean 과 power_log1p_stdd_shift 를 통해 dhc 의 정보를 TS 에 포함시킴으로써 dhc 를 제거할 수 있었습니다.

7. power_log1p_stdd_cumweek_mean_shift

예) 지난주 workday 들의 전력량 평균을 이번주 workday 의 변수로 사용

예) 14 번 건물의 경우 지난주 workday(월, 화, 수, 목, 금)들의 전력량 평균을 이번주 수요일 변수로 사용



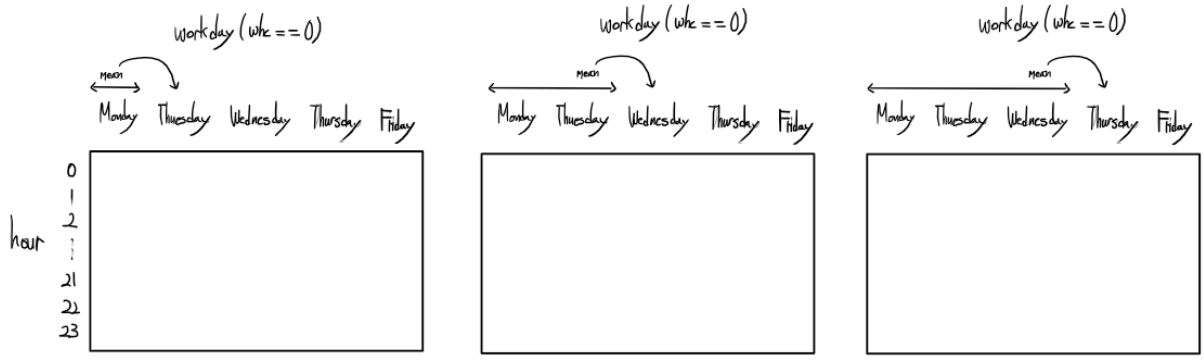
power_log1p_stdd_shift 는 1 주일 간격이고, 데이터가 각각 하나만 존재하므로 변동성이 큽니다. 따라서 간격을 좁히고, 데이터를 묶어 평균을 내서 변동성을 줄일 필요가 있었습니다. workday(whc==0)의 경우 일반적으로 월, 화, 수, 목, 금 5 일의 데이터가 확보되며 지난주 월, 화, 수, 목, 금 전력량의 평균을 이번주 월요일의 변수로 사용하므로 시간 간격이 좁혀지고, 평균값을 사용하므로 변동성을 줄일 수 있습니다. 또한 주간 정보와 workday 의 정보를 포함할 수 있습니다.

8. power_log1p_stdd_thisweek_mean_shift

예) 이번주 workday 인 월요일의 전력량 평균을 이번주 화요일 변수로 사용

예) 이번주 workday 인 월, 화 전력량 평균을 이번주 수요일 변수로 사용

예) 이번주 workday 인 월, 화, 수 전력량 평균을 이번주 목요일 변수로 사용



power_log1p_stdd_cumweek_mean_shift 보다 시간 간격을 좁히기 위해 직전 주가 아닌 해당 주의 데이터를 사용하였고, 안정성을 확보하기 위해 직전 workday 까지의 평균을 구했습니다. 주간 정보와 workday 정보를 포함할 수 있습니다.

세가지 Train Notebook

0. 공통변수:

   whc,

   cos_hour, sin_hour,

   temperature_squared, THI, humidity_squared,

   temperature_mean, THI_mean,

1. EE_TRAIN_0:

   power_log1p_stdd_mean, power_log1p_stdd_shift, power_log1p_stdd_cumweek_mean_shift, power_log1p_stdd_thisweek_mean_shift

	mean
power_log1p_stdd_mean	11.225888
power_log1p_stdd_shift	9.112626
power_log1p_stdd_thisweek_mean_shift	3.538538
cos_hour	2.691489
THI_mean	2.219360
temperature_squared_mean	1.367848
THI	1.304061
sin_hour	1.113261
power_log1p_stdd_cumweek_mean_shift	0.710931
temperature_squared	0.629741
humidity_squared	0.380160

   수치형 변수들의 Feature Importance

2. EE_TRAIN_1:

   power_log1p_stdd_mean, power_log1p_stdd_shift, power_log1p_stdd_cumweek_mean_shift

	mean
power_log1p_stdd_mean	11.323575
power_log1p_stdd_shift	7.429943
cos_hour	4.397443
THI_mean	2.449407
sin_hour	2.018548
temperature_squared_mean	1.713289
THI	1.206053
power_log1p_stdd_cumweek_mean_shift	1.076983
temperature_squared	0.527321
humidity_squared	0.410157

3. EE_TRAIN_2:

   power_log1p_stdd_mean, power_log1p_stdd_shift

	mean
power_log1p_stdd_mean	11.834591
power_log1p_stdd_shift	7.895171
THI_mean	2.056753
temperature_squared_mean	1.577904
sin_hour	1.500297
cos_hour	1.499326
THI	1.268647
temperature_squared	0.618974
humidity_squared	0.498841

하나의 Inference Notebook 에서 두가지 submission

1. submission_1:

   8 월 27 일 이전

   power_log1p_stdd_thisweek_mean_shift 값이 있는 경우 EE_TRAIN_0 적용

   그 외 EE_TRAIN_1 적용

   8 월 27 일 당일, 이후

   EE_TRAIN_2 적용

   Public Score: 5.2707265718

   Private Score: 6.1053683645

2. submission_2:

   8 월 27 일 이전

   EE_TRAIN_1 적용

   8 월 27 일 당일, 이후

   EE_TRAIN_2 적용

   Public Score: 5.2786695632

   Private Score: 6.1087725037

submission_1 과 submission_2 의 점수차이가 크지 않으므로 변수가 적은 submission_2 를 선택하는 것이 바람직해 보입니다.