COMPAS 김해시 화재예측

TEAM-EDA – 김연민, 김현우

Contents

1 Overview

대회 목적

- 소방 및 건물 관련 데이터를 활용하여 건축물의 화재 위험도 분석 및 예측 모델의 개발.

데이터

- 소방 및 건물정보, 방화물 성능 유지여부, 에너지 사용량, 기후정보 등 (총 변수 180개)

소방점검 대상물기준	소방시설 특례5호여부	건물용도	건물 건축면적	전기 및 가스 사용량	방화문, 방화셔터 등의 성능 유지여부(0~5)	풍속	습도
Y	N	단독 주택	423.00	3332	4	1.7	21
N	N	공장	5593.711	51600.0	3	3.6	83

1 Overview

특이사항

① 평가 데이터의 경우 김해시 지역으로 한정

- 훈련 데이터(train.csv) : 경상남도 지역의 데이터
- 검증 데이터(validation.csv) : 김해시 지역으로 한정된 데이터
- 평가 데이터(test.csv) : **김해시 지역으로 한정**된 데이터

② 화재가 나지 않은 경우 시점을 임의로 샘플링

- 불 난 건물들의 시점과 동일한 기간 내에서 추출

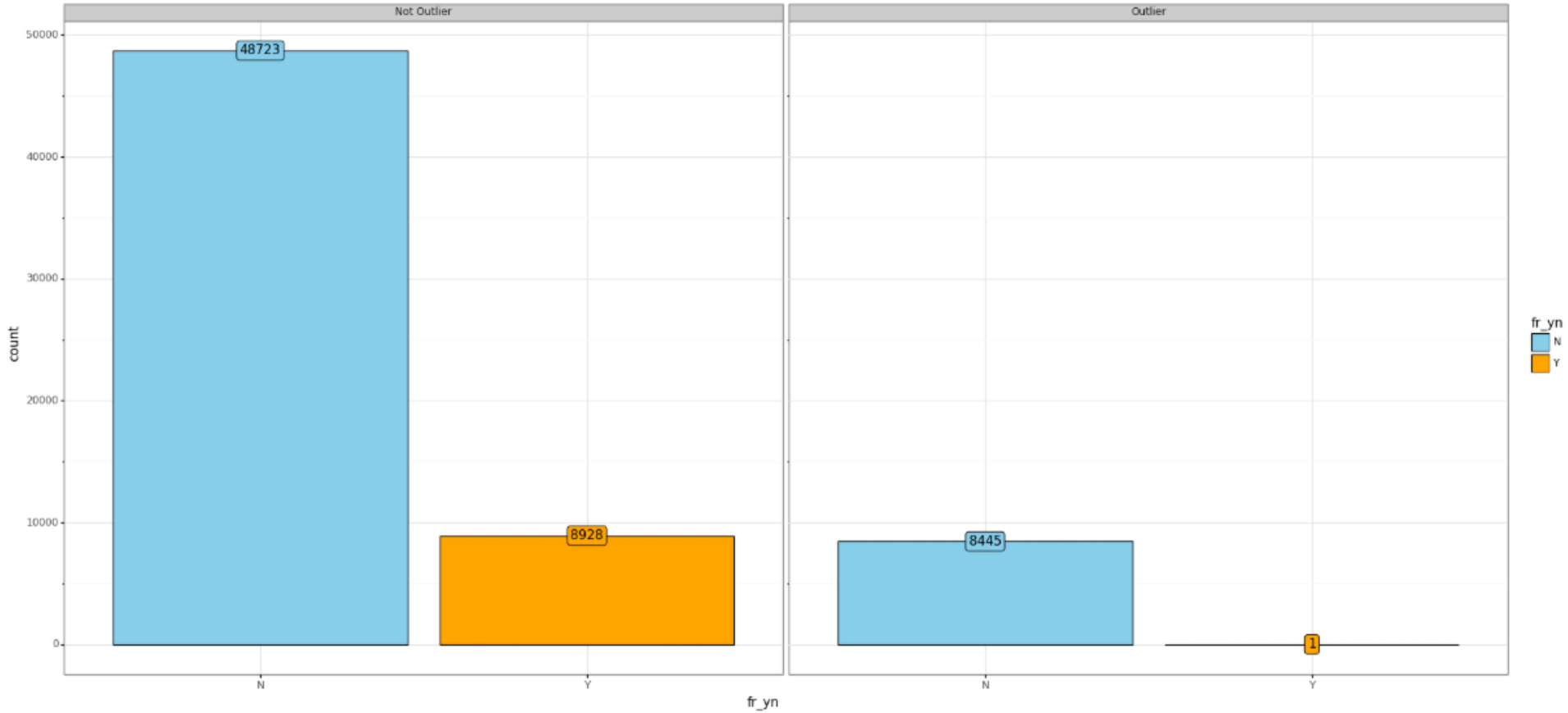
2

Pre-Processing

Outlier 제거

- 화재가 한번도 발생하지 않은 창원시 제거.
- 지역 구분이 되지 않는 4개의 row 제거.

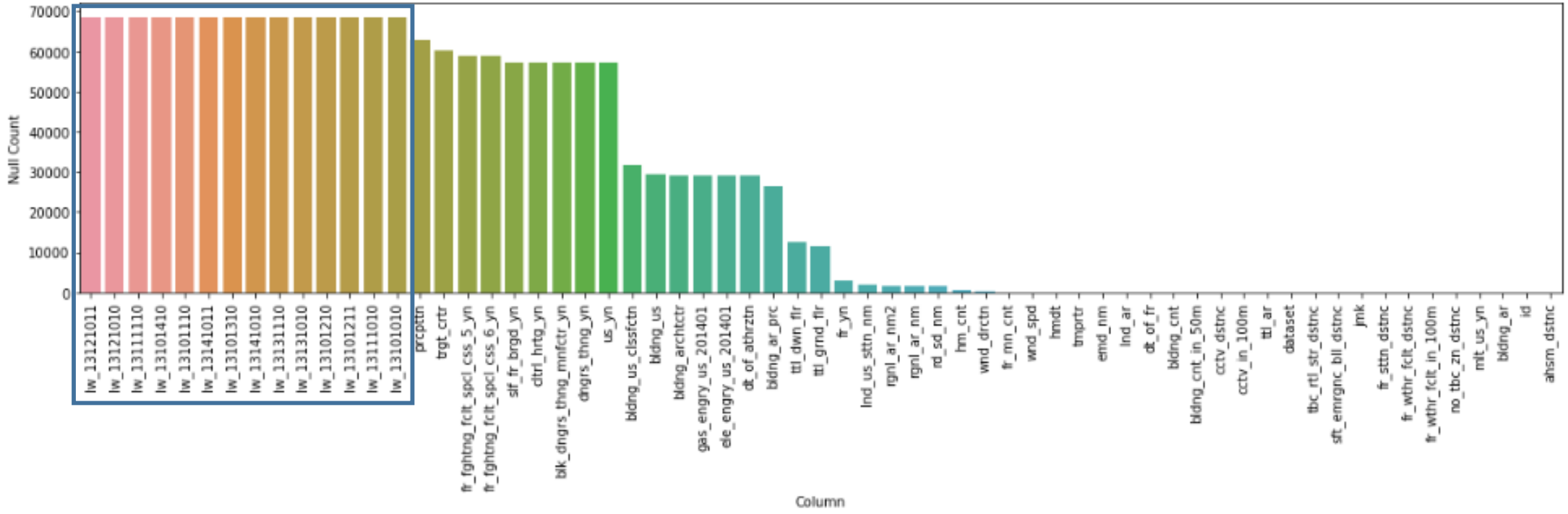
	emd_nm	dt_of_fr	fr_yn	bldng_us	bldng_archtctr	bldng_cnt	bldng_ar	ttl_ar	lnd_ar
17885	NaN	2014-06-20 03:12:49	N	NaN	NaN	4	0.0	0.00	0.0
20869	NaN	2014-03-16 16:17:07	N	단독주택	철근콘크리트구조	1	167.7	515.25	287.3
52035	NaN	2016-05-05 16:55:00	Y	등.식물 관련시설	벽돌구조	7	150.9	150.90	0.0
58625	NaN	2014-12-08 03:52:34	N	NaN	NaN	1	190.0	190.00	2645.0



2 Pre-Processing

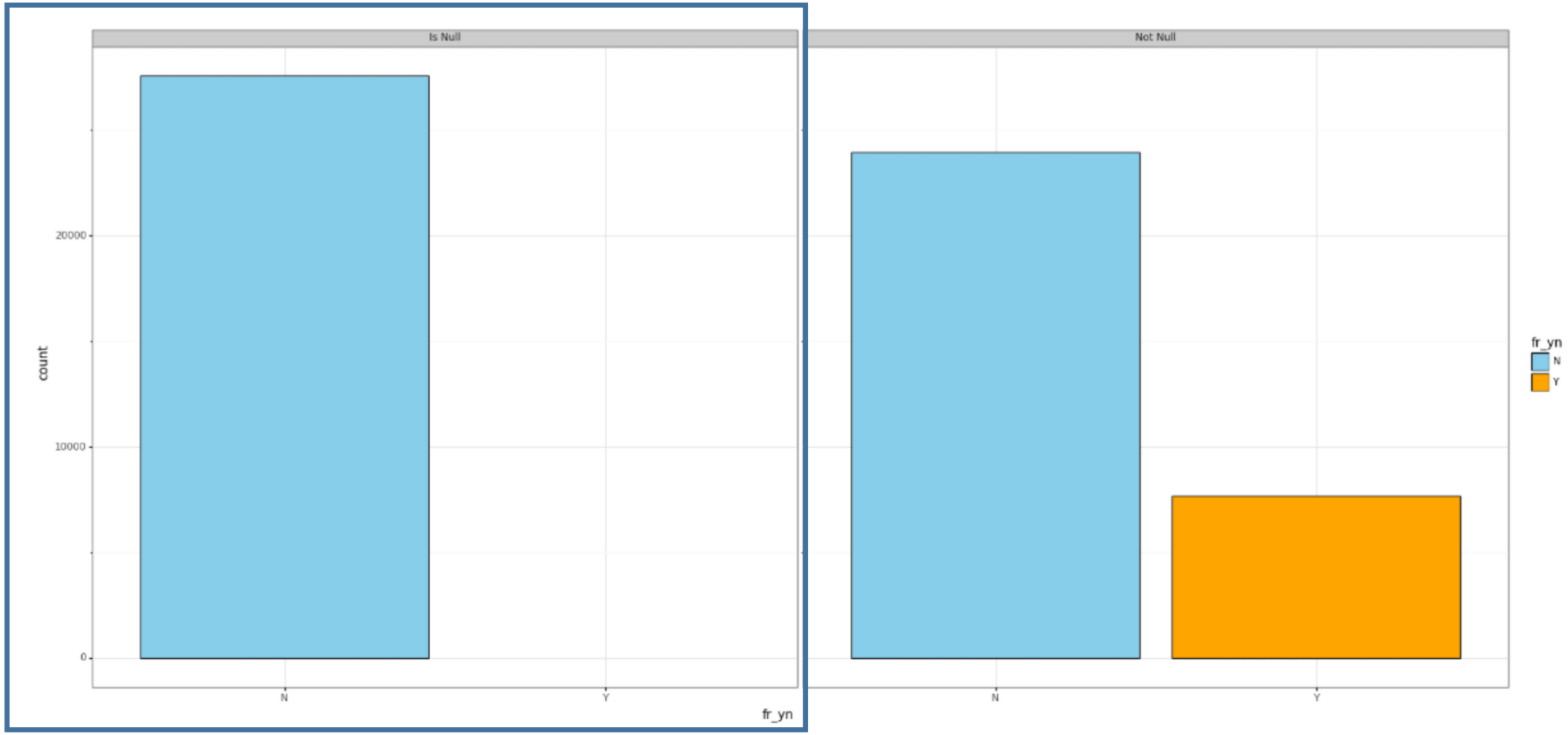
Null 값 전략

- Null값들을 Bar plot으로 표시해보면 계단식으로 구분되어 있는 것을 볼 수 있음.



Null 값 전략

- 건물승인일자(dt_of_athrztn)값이 Null이면 화재가 발생하지 않는 것을 볼 수 있음.

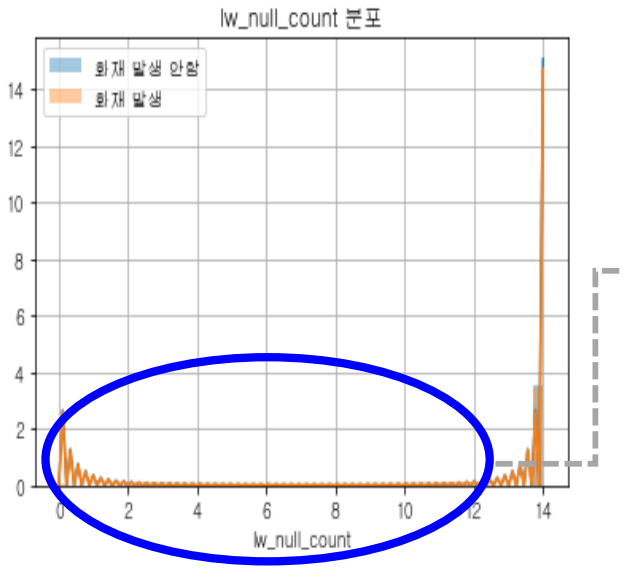


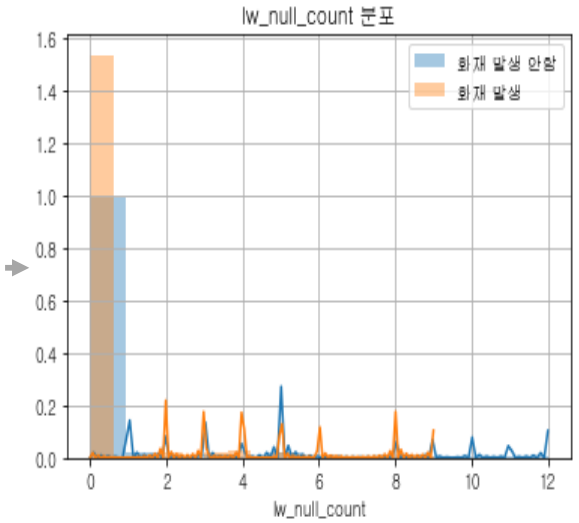
→ Null정보를 Masking할 필요가 있음.

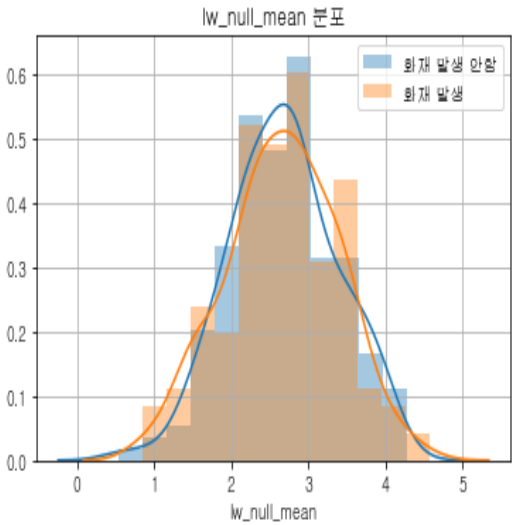
2 Pre-Processing

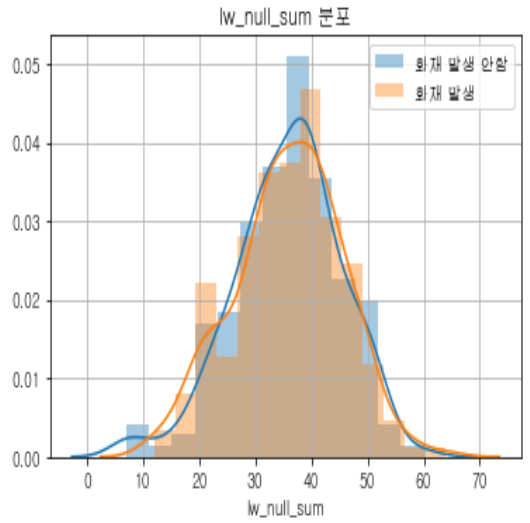
lw null aggregation feature

- 건물의 소방관련 설비 성능 유지(lw feature)변수들은 의미 상으로 화재발생 여부와 매우 중요한 관련이 있어 보이지만 대부분이 Null 값이라 사용하기 무리가 있어 파생변수를 만들고 lw_feature를 제거.



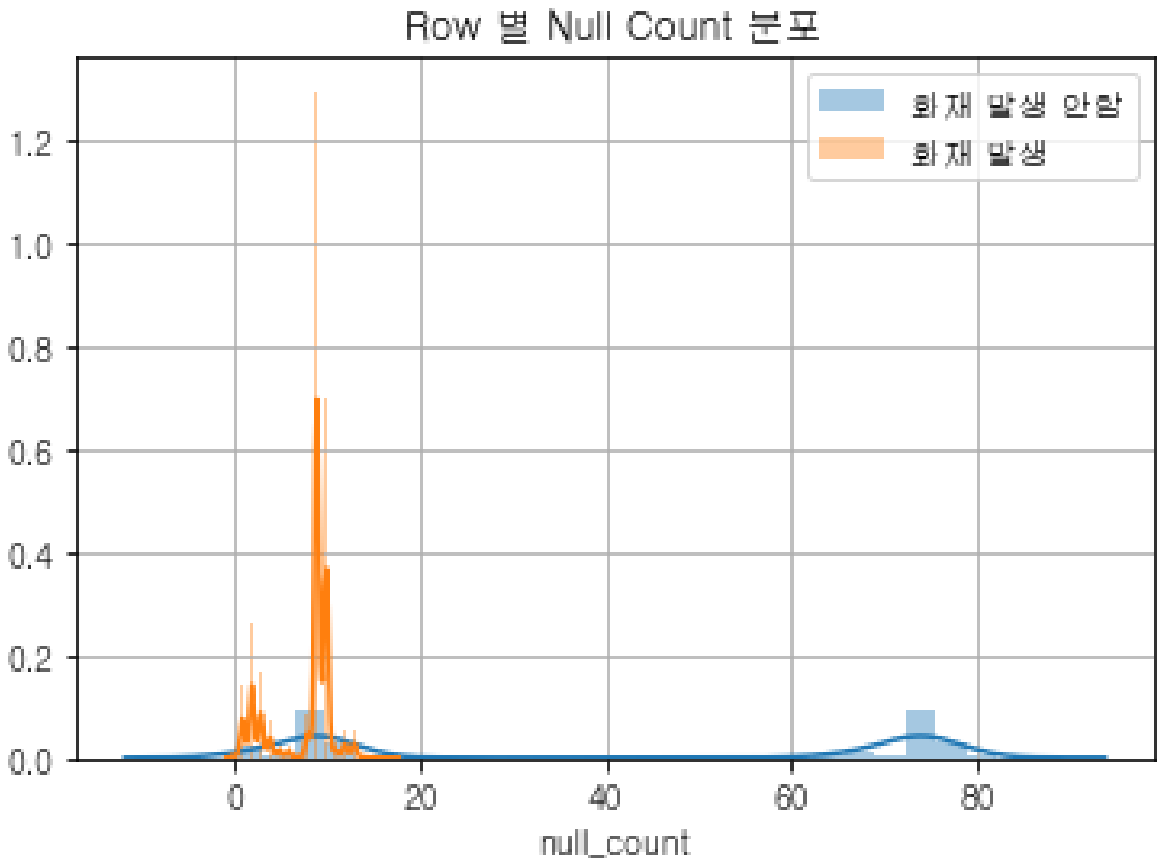






Null 값 전략

- Row 별 Null Count의 개수 : 화재 vs. 화재발생안함 을 구분하는데 중요한 Feature



2 Pre-Processing

건물용도분류명(bldng_us_clssfctn) 결측치

- Category 변수들은 지정상 지목(JMK)가 '공'이더라도 무조건 공업용이 아니었고 bldng_us(건물용도) 공장이 아니었음.
- 일부 채울 수 있는 것도 있었지만 채우더라도 Public Leaderboard 성능이 많이 변하여 Category 변수의 Null값은 채우지 않고 확실한 아래 항목만 채움

```
all_data.loc[(all_data['bldng_us']=='위험물저장및처리시설')&(all_data['lnd_us_sttn_nm']!='공업용'),'bldng_us_clssfctn'] = '상업용'
all_data.loc[(all_data['bldng_us']=='위험물저장및처리시설')&(all_data['lnd_us_sttn_nm']=='공업용'),'bldng_us_clssfctn'] = '공업용
```

강수량(prcpttn) 결측치

- Numerical 변수의 imputation 전략은 합리적으로 생각할 수 있는 가장 좁은 범위부터 넓은 범위로 넓혀가면서 채우기 시작.
- 강수량의 Null값을 채우기 전에 is_null Feature로 Null 특성을 살려줌.
- 강수량 (아래 순서로 범위를 좁혀가면서 Null값 채움을 수행)
    1. 월, 일, 시간, 지역(emd_nm)의 평균값
    2. 월, 일, 지역(emd_nm)의 평균값
    3. 월, 일, 구, 동, 읍, 면(emd_nm_2)의 평균 값
    4. 월, 일, 군, 시(emd_nm_1)의 평균 값
    5. 나머지는 0값

2 Pre-Processing

온도, 바람, 세기, 습도, 바람방향, 관할 소방서 인원 결측치

- 아래 기온 피쳐들은 채워지는 Null값들이 작아 Null을 따로 Masking하지 않음.
- 온도, 바람 세기, 습도, 바람방향 (아래 순서로 범위를 좁혀가면서 Null값 채움을 수행)
    1. 월, 일, 시간, emd_nm(지역)의 평균값으로 채움.
    2. 나머지 null값은 바로 이전 값으로 채움.

- 관할 소방서 인원도 Null Masking 수행.
- 관할 소방서 인원(아래 순서로 범위를 좁혀가면서 Null값 채움을 수행)
    1. 세부 지역의 관할 소방서 인원 평균
    2. 구, 읍, 면, 동 지역의 평균 관할 소방서 인원 평균
    3. 시, 군, 구 단위의 관할 소방서 인원 평균

2

Pre-Processing

행정구역인구 결측치

- 행정구역인구는 Null Masking 수행
- 행정 구역 인구(아래 순서로 범위를 좁혀가면서 Null값 채움을 수행)
  1. 같은 빌딩, 연도
  2. 세부 지역, 연도, 월
  3. 세부 지역, 연도
  4. 세부 지역
  5. 구, 읍, 면, 동 단위 지역, 연도, 월
  6. 구, 읍, 면, 동 단위 지역, 연도
  7. 구, 읍, 면, 동 단위 지역
  8. 시, 군 단위 지역, 연도, 월
  9. 시, 군 단위 지역, 연도
  10. 시, 군 단위 지역

```
############### FILL NA ###############
['same_bld', 'year'] mean -> hm_cnt
Original NA Count:  702
Fill NA Count:  0
Remain Null Count:  702
############### FILL NA ###############
['emd_nm', 'year', 'month'] mean -> hm_cnt
Original NA Count:  702
Fill NA Count:  11
Remain Null Count:  691
############### FILL NA ###############
['emd_nm', 'year'] mean -> hm_cnt
Original NA Count:  691
Fill NA Count:  11
Remain Null Count:  680
############### FILL NA ###############
['emd_nm'] mean -> hm_cnt
Original NA Count:  680
Fill NA Count:  492
Remain Null Count:  188
############### FILL NA ###############
['emd_nm_2', 'year', 'month'] mean -> hm_cnt
Original NA Count:  188
Fill NA Count:  119
Remain Null Count:  69
############### FILL NA ###############
['emd_nm_2', 'year'] mean -> hm_cnt
Original NA Count:  69
Fill NA Count:  1
Remain Null Count:  68
############### FILL NA ###############
['emd_nm_2'] mean -> hm_cnt
Original NA Count:  68
Fill NA Count:  59
Remain Null Count:  9
```

3 Feature Engineering

Feature 추가

- 시간관련 피쳐

① Year: 년도

② Hour: 시간

③ Weekday: 요일

④ Month: 월

⑤ Day: 일

- 지역 Feature 분리

① emd_nm_0: 경상남도

② emd_nm_1: 시, 군

③ emd_nm_2: 구, 읍, 면, 동

④ emd_nm_3: 리, 가, 기타 등등

3 Feature Engineering

Feature 추가

- 같은 빌딩 정보 추가

① 건물건축면적(bldng_ar), 건물연면적(ttl_ar), 토지면적(lnd_ar), 지역(emd_nm) 정보를 조합하여 같은 건물 정보 추가 (단, 지역정보가 없거나 건물건축면적(bldng_ar)이 0인 경우에는 같은 건물로 표시하지 않음)

→ 같은 빌딩 정보에 대한 Label Encoding

② 같은 건물이 몇 번 있는지 빈도 Feature (Frequency Encoding)

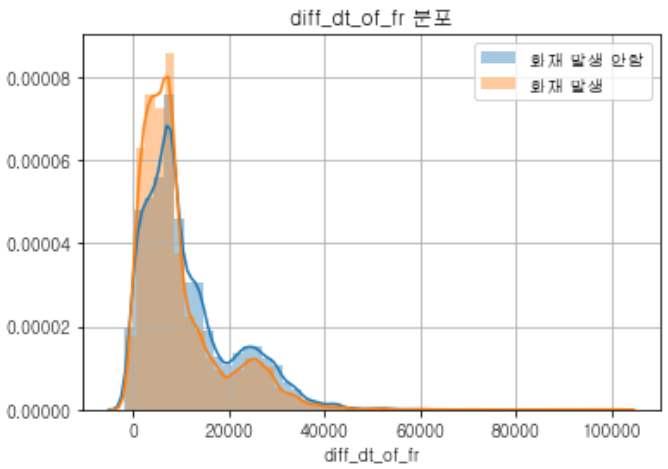
3 Feature Engineering

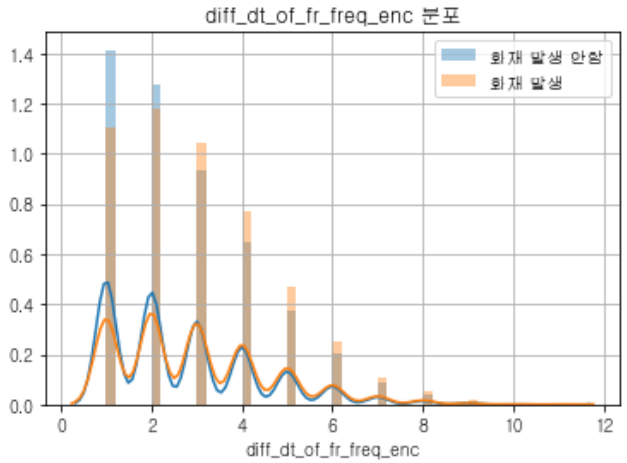
Feature 추가

제 각각의 Format으로 되어 있는 건물 승인 일자 전 처리하여 연도 추출

① 화재발생일부터 건물 승인일자까지의 차이(Days)
② 차이 값의 Frequency Encoding

```
all_data['diff_dt_of_fr'] = all_data[['dt_of_fr','dt_of_athrztn']].apply(lambda row:get_diff_dt_of_fr(row) ,axis=1)
all_data['diff_dt_of_fr_freq_enc'] = all_data['diff_dt_of_fr'].map(all_data['diff_dt_of_fr'].value_counts())
```



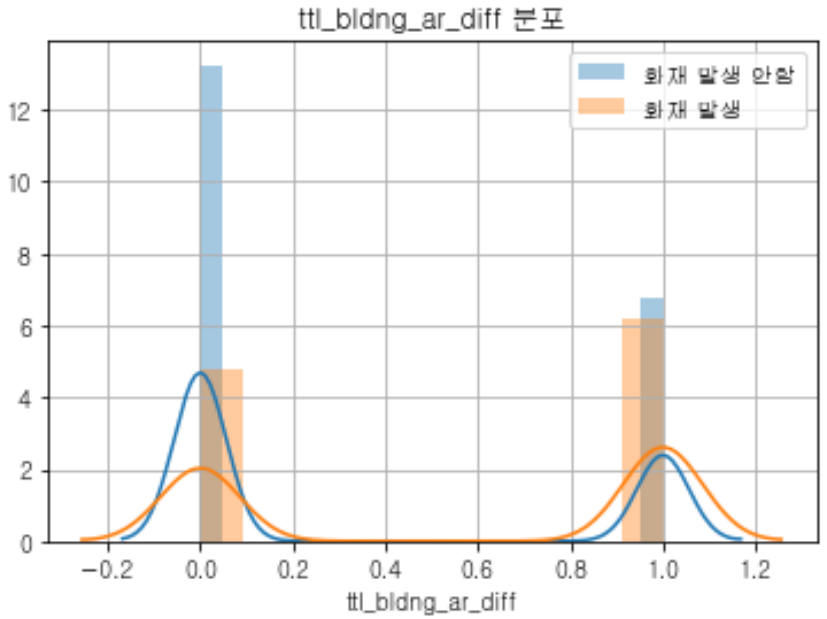


3 Feature Engineering

Feature 추가

건물건축면적(bldng_ar), 건물연면적(ttl_ar)이 같지 않으면 Masking 수행

```
all_data['ttl_bldng_ar_diff'] = 0
all_data.loc[all_data['ttl_ar']!=all_data['bldng_ar'],'ttl_bldng_ar_diff'] = 1
```



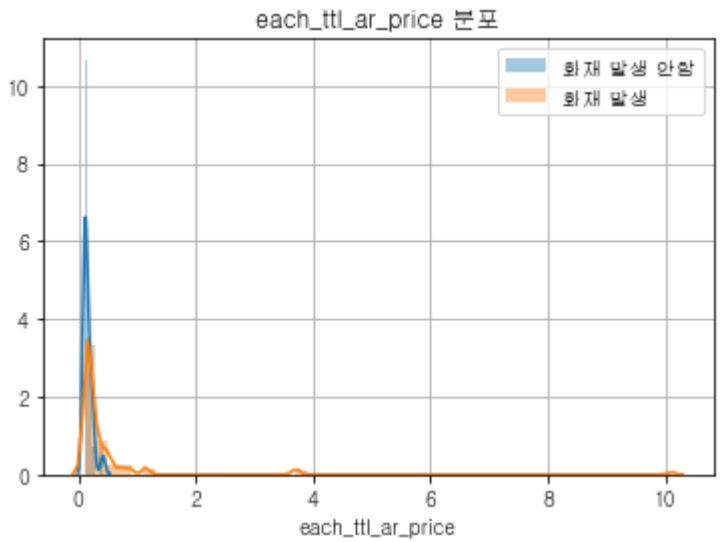
3 Feature Engineering

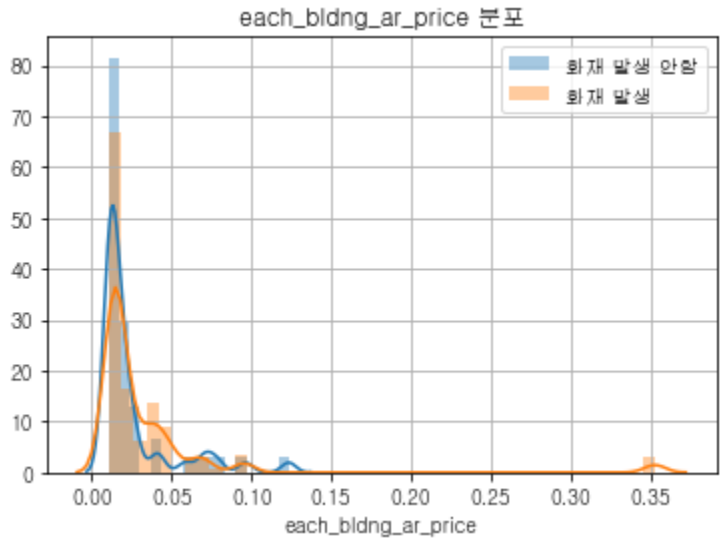
Feature 추가

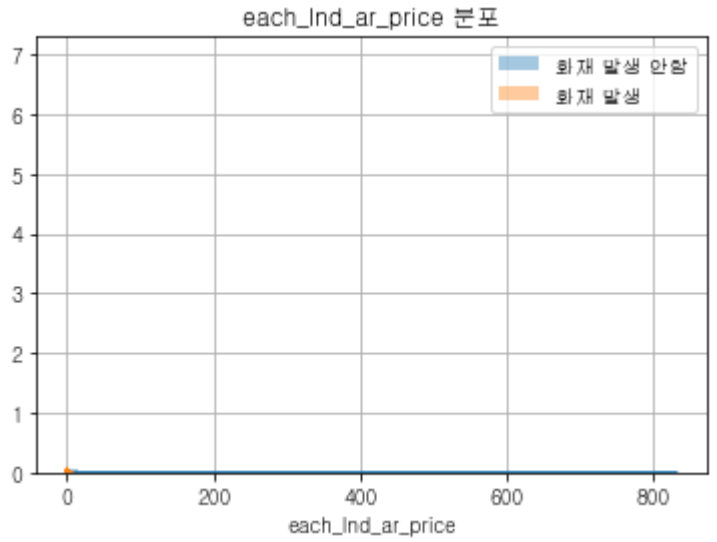
all_data['each_ttl_ar_price'] = all_data['ttl_ar']/all_data['bldng_ar_prc'] # 가격 대비 건물면적

all_data['each_bldng_ar_price'] = all_data['bldng_ar']/all_data['bldng_ar_prc'] # 가격 대비 건축면적

all_data['each_lnd_ar_price'] = all_data['lnd_ar']/all_data['bldng_ar_prc'] # 가격 대비 토지면적







3 Feature Engineering

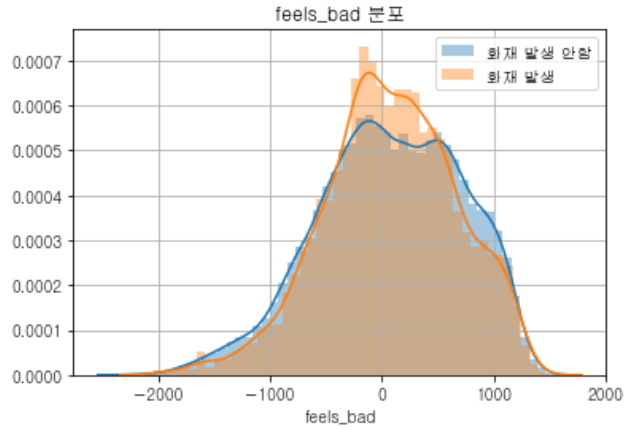
Feature 추가

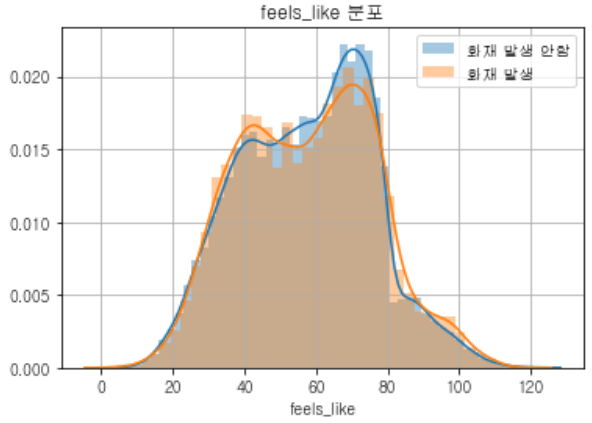
불쾌지수, 체감온도, 열지수 추가

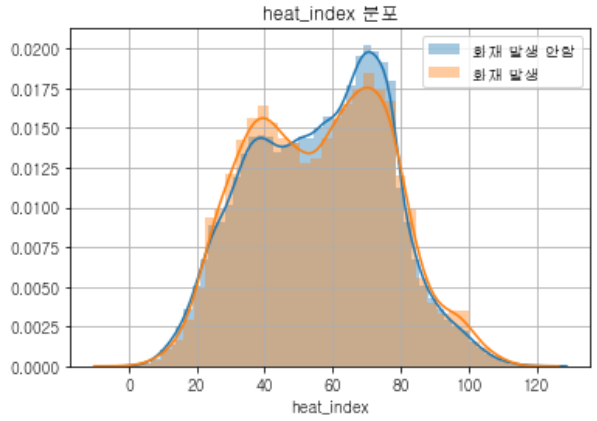
$$\text{불쾌지수} = \frac{9}{5}T - 0.55(1 - RH)(\frac{9}{5}T - 26) + 32 \quad (T: \text{기온}(℃),\ RH: \text{상대습도}(\%))$$

```
# 불쾌지수
all_data['feels_bad'] = (9/5)*all_data['tmprtr'] - 0.55*(1-all_data['hmdt'])*((9/5)*all_data['tmprtr']-26)+32

all_data['tmprtr_f'] = all_data['tmprtr'].apply(lambda x: Temp(x,'c').f)
# meteocalc package를 사용하여 체감온도 계산, 체감온도 계산할 때 화씨 단위로 해야해서 변환 후 계산
all_data['feels_like'] = all_data[['tmprtr_f','hmdt','wnd_spd']].apply(lambda x: feels_like(x['tmprtr_f'],x['hmdt'],x['wnd_spd']),axis=1)
# heat index 계산
all_data['heat_index'] = all_data[['tmprtr_f','hmdt']].apply(lambda x: heat_index(x['tmprtr_f'],x['hmdt']),axis=1)
```







4 Model

LightGBM 사용

- Null Data가 많아도 LightGBM 내부 알고리즘에 의하여 성능이 잘 나옴.
- 빠름.
- 데이터를 조금만 가공해도 평균 이상의 성능이 나오는 것을 볼 수 있음.
- Parameter Tuning이 쉬움.

4 Model

데이터 세팅

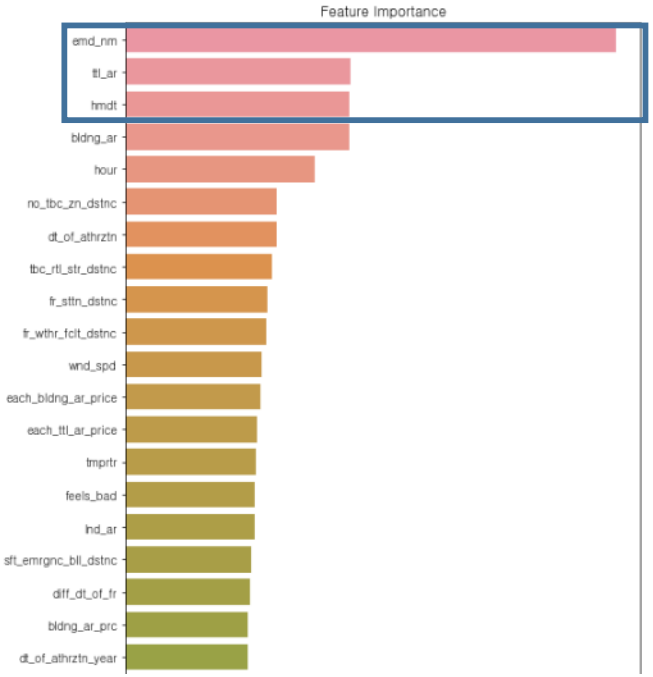
Train과 Validation을 하나로 합쳐서 평가 수행(Train 데이터가 부족했고, 김해시 정보를 더 많이 넣기 위함)

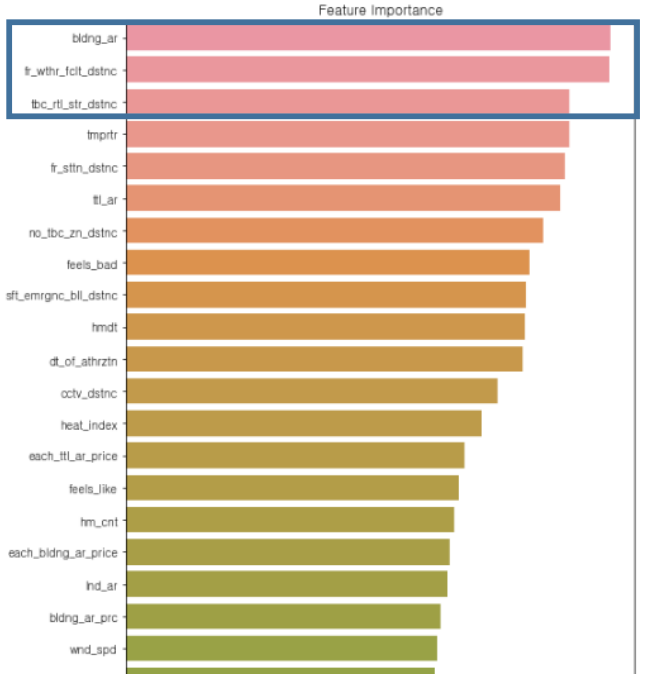
아래 변수들을 LightGBM이 Category 변수로 처리할 수 있도록 함

```
cat_feature = ['same_bld', 'bldng_archtctr','bldng_us','bldng_us_clssfctn',
               'jmk','lnd_us_sttn_nm','rd_sd_nm','rgnl_ar_nm','rgnl_ar_nm2','trgt_crtr','emd_nm']
```

Parameter에 따른 Feature Importance 변화

- 지역관련 Feature가 너무 높게 나오면 LB가 낮게 나오는 경향을 보임
  → 김해시에 대한 정보가 Train에는 적어서 발생하는 문제.





Parameter에 따른 Feature Importance 변화

- Distance, 온/습도, 면적 관련 Feature의 중요도를 높게 보도록 Parameter Tuning
- 또한 제출이 무한이었기 때문에 Parameter와 Seed는 LB기반으로 Tuning 수행

```
space = {
    'num_leaves': hp.quniform('num_leaves', 8, 512, 2),
    'max_depth': hp.quniform('max_depth', 4, 17, 1),
    'min_data_in_leaf': hp.quniform('min_data_in_leaf', 1, 144, 2),
    'feature_fraction': hp.uniform('feature_fraction', 0.3, 1.0),
    'bagging_fraction': hp.uniform('bagging_fraction', 0.5, 1.0),
    'lambda_l1': hp.uniform('lambda_l1', 0, 10.0),
    'lambda_l2': hp.uniform('lambda_l2', 0, 10.0),
    'min_child_weight': hp.uniform('min_child_weight', 0, 50.0),
    'learning_rate': hp.uniform('learning_rate', 0.01, 0.1),
    'min_sum_hessian_in_leaf': hp.quniform('min_sum_hessian_in_leaf', 1, 144, 1),
    #'subsample_for_bin': hp.quniform('subsample_for_bin', 20000, 300000, 20000),
    }

best = fmin(fn=objective,
        space=space,
        algo=tpe.suggest,
        max_evals=50)
hyperopt_dict[c_type] = best
```

5 Post-Processing

특정 카테고리 변수의 조합에 따라 화재가 발생할 확률

- Category 변수를 4개의 조합으로 가능한 모든 경우에 대해 조합.
- 특정 조합의 경우 zero_ratio가 1인 값들이 많음.
- 모델의 오버피팅을 방지하기 위해서 zero_count가 50개 초과이면서 zero_ratio가 0.95 초과인 것들은 0으로 바꿔줌. (해당 수치는 LB를 통해서 테스트)

	cat0	cat1	cat2	cat3	c0u	c1u	c2u	c3u	zero_count	one_count	zero_ratio	one_ratio
104932	bldng_us	jmk	rgnl_ar_nm	rgnl_ar_nm2	단독주택	대	제1종전용주거지역	지정되지않음	497	0	1.0	0.0
65159	bldng_archtctr	jmk	rgnl_ar_nm	rgnl_ar_nm2	벽돌구조	대	제1종전용주거지역	지정되지않음	481	0	1.0	0.0
128928	bldng_us_clssfctn	jmk	rgnl_ar_nm	rgnl_ar_nm2	주거용	대	제1종전용주거지역	지정되지않음	477	0	1.0	0.0
12030	bldng_archtctr	bldng_us	jmk	rgnl_ar_nm	벽돌구조	단독주택	대	제1종전용주거지역	474	0	1.0	0.0
31176	bldng_archtctr	bldng_us	rgnl_ar_nm	rgnl_ar_nm2	벽돌구조	단독주택	제1종전용주거지역	지정되지않음	474	0	1.0	0.0

→ Post-Processing을 통해서 강제로 'N'으로 바꿈.

6 Trial Errors

시행 착오

1. 가스 에너지, 전기 에너지 사용량을 관측 시점에 맞춰서 관측 시점의 전기, 에너지 사용량, 관측 시점 전달 대비 에너지 사용량으로 변수를 추가했는데 나빠짐.
2. 모델의 학습 Metric을 auc, F1 score, binary_logloss 등 다양하게 테스트 해봤지만 auc가 가장 스코어가 좋았음.
3. 세부 지역을 하나씩 빼는 LeaveOneGroupOut 전략을 사용했는데 점수차이가 많지 않았지만 학습시간이 오래 걸려 사용하지 않음.
4. 점수가 LightGBM Seed에 따라 +- 0.015 정도 나타나는 것을 확인하여 LB 기반으로 Seed 최적화 수행, Seed 46이 점수가 좋았음.
5. HyperOpt를 수행하여 CV를 좋게 하였지만 CV가 좋아진다고 LB가 좋아지지 않은 것을 확인.
6. 여러 번 제출 했을 때 평균적으로 Validation Set과 비율이 비슷한 530~560 사이의 개수가 점수가 높은 것을 확인함.
7. Parameter와 Seed를 다르게 하여 Cutoff를 540으로 Voting을 수행하였는데 0.55정도 나오는 것을 확인, 시간이 오래 걸리고 재현하기 힘들어서 포기.
8. Category Combine 변수를 2단계, 3단계, 4단계로 수행하여 변수를 직접 추가했을 때 LB가 좋아지지 않음.

감사합니다

TEAM-EDA