Upstage AI Lab 7기

ML 경진대회 : House Price Prediction

서울시 아파트 실거래가 매매 데이터를 기반으로 아파트 가격을 예측하는 대회

정 혜린

Data info

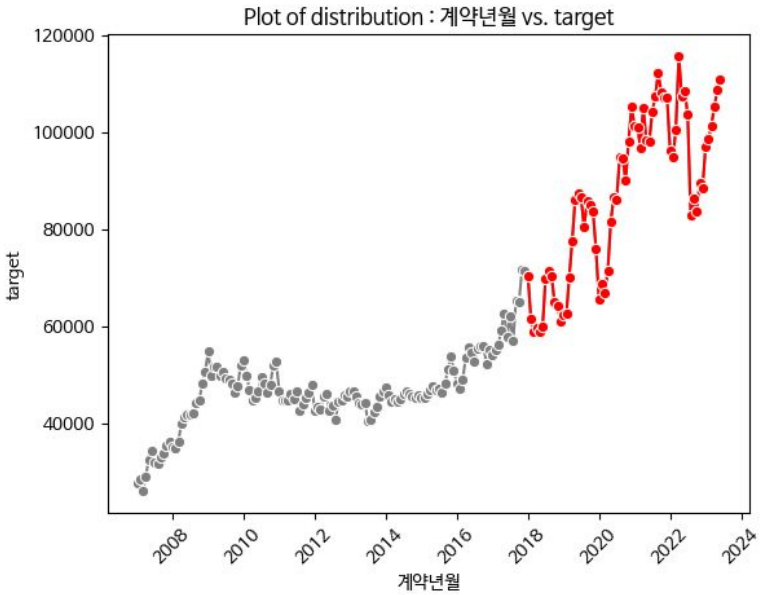
❑ 데이터셋 설명

- Target : 각 거래 일자에 따른 서울시 아파트의 거래금액(만원)
- Raw dataset은 확장자 csv 형태로 주어짐
- Dataset : (1,118,822, 52)

	시군구	번지	본번	부번	아파트명	전용면적(㎡)	계약년월	계약일	층	건축년도	...	건축면적	주차대수	기타/의무/임대/임의=1/2/3/4	단지승인일	사용허가여부	관리비업로드	좌표X	좌표Y	단지신청일	target
0	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성	79.97	201712	8	3	1987	...	4858.0	262.0	임의	2022-11-17 13:00:29.0	Y	N	127.057210	37.476763	2022-11-17 10:19:06.0	124000
1	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성	79.97	201712	22	4	1987	...	4858.0	262.0	임의	2022-11-17 13:00:29.0	Y	N	127.057210	37.476763	2022-11-17 10:19:06.0	123500
2	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성	54.98	201712	28	5	1987	...	4858.0	262.0	임의	2022-11-17 13:00:29.0	Y	N	127.057210	37.476763	2022-11-17 10:19:06.0	91500
3	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성	79.97	201801	3	4	1987	...	4858.0	262.0	임의	2022-11-17 13:00:29.0	Y	N	127.057210	37.476763	2022-11-17 10:19:06.0	130000
4	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성	79.97	201801	8	2	1987	...	4858.0	262.0	임의	2022-11-17 13:00:29.0	Y	N	127.057210	37.476763	2022-11-17 10:19:06.0	117000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1118817	서울특별시 은평구 구산동	382	382.0	0.0	갈현현대	59.94	200707	12	11	1998	...	0.0	366.0	의무	2013-06-04 16:18:51.0	Y	N	126.905638	37.612962	2013-03-07 09:46:27.0	20000
1118818	서울특별시 은평구 구산동	382	382.0	0.0	갈현현대	59.94	200708	25	10	1998	...	0.0	366.0	의무	2013-06-04 16:18:51.0	Y	N	126.905638	37.612962	2013-03-07 09:46:27.0	20000
1118819	서울특별시 은평구 구산동	382	382.0	0.0	갈현현대	84.83	200708	31	20	1998	...	0.0	366.0	의무	2013-06-04 16:18:51.0	Y	N	126.905638	37.612962	2013-03-07 09:46:27.0	28000
1118820	서울특별시 은평구 구산동	382	382.0	0.0	갈현현대	84.83	200709	15	8	1998	...	0.0	366.0	의무	2013-06-04 16:18:51.0	Y	N	126.905638	37.612962	2013-03-07 09:46:27.0	29000
1118821	서울특별시 중구 묵정동	11-67	11.0	67.0	묵정	52.46	200701	10	5	1981	...	7354.0	45.0	임의	2020-07-10 00:00:00.0	Y	Y	127.000071	37.560706	2017-09-05 20:06:39.0	13250

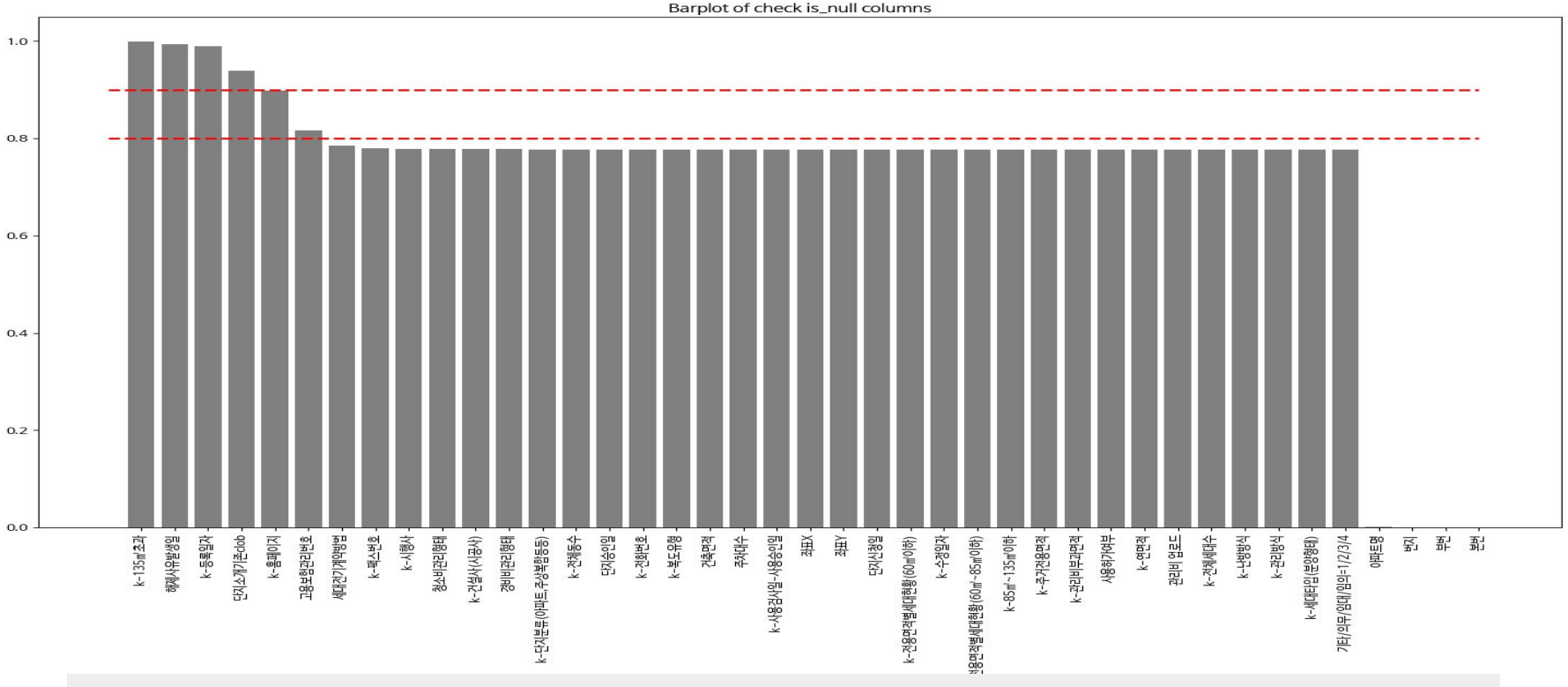
1118822 rows × 52 columns

EDA : target(y) distribution



➔ 2018년 이후로 급격히 증가하는 경향을 보임
➔ 특히, 2020년은 급감하는 이유는 코로나의 영향 때문이라 유추 가능

EDA : Check nan ratio



➔ Target을 제외한 대부분의 설명변수들의 결측치 비율이 **80%**이상임을 알 수 있음.
➔ (좌표X, 좌표Y)를 제외하고 결측값 비율이 **80%** 이상인 변수들은 과감히 제거

EDA : 좌표X, 좌표Y 채우기

```python
#restapi 활용하기
import json
import requests

api_key = '3565d8117dbf5e77aebe3710ba824c91'

def addr_to_lat_lon(addr):
    url = f'https://dapi.kakao.com/v2/local/search/address.json?query={addr}'
    headers = {"Authorization": f"KakaoAK {api_key}"}
    response = requests.get(url, headers=headers)

    # 상태 코드 확인
    if response.status_code != 200:
        return None, None, response.status_code

    result = response.json()
    # 결과값이 없는 경우 처리
    if not result['documents']:
        return None, None, response.status_code

    match_first = result['documents'][0]['address']
    return float(match_first['x']), float(match_first['y']), response.status_code

# 주소 리스트를 처리하는 함수
def process_addresses(address_list):
    success_data = []  # 성공적으로 처리된 데이터 저장
    failed_addresses = []  # 실패한 주소 저장

    for addr in address_list:
        x, y, status = addr_to_lat_lon(addr)
        if status == 200 and x is not None and y is not None:
            success_data.append({"road_address": addr, "latitude": y, "longitude": x})
```

B	C	D
address	status_code	road_address
서울특별시 강남구 도곡동 893-2	200	서울시 강남구 남부순환로 363길 30
서울특별시 강남구 삼성동 22	200	서울시 강남구 삼성로 629
서울특별시 노원구 상계동 109-5	200	서울시 노원구 덕릉로123길 53
서울특별시 서초구 잠원동 52-2	200	서울시 서초구 잠원로 14길 42
서울특별시 서초구 신원동 690	200	서울시 서초구 청계산로7길 43
서울특별시 서초구 신원동 nan	200	서울시 서초구 헌릉로 8길 10-12
서울특별시 서초구 서초동 1335	200	서울시 서초구 효령로 403
서울특별시 강남구 일원동 690-1	200	서울특별시 강남구 개포로 110길 38
서울특별시 강남구 대치동 977	200	서울특별시 강남구 도곡로 93길 23
서울특별시 강남구 역삼동 763	200	서울특별시 강남구 도곡로43길 20
서울특별시 강남구 개포동 138	200	서울특별시 강남구 삼성로 11
서울특별시 강남구 대치동 633	200	서울특별시 강남구 삼성로 51길 37
서울특별시 강남구 대치동 610	200	서울특별시 강남구 삼성로 51길 37
서울특별시 강남구 삼성동 19-4	200	서울특별시 강남구 삼성로 651
서울특별시 강남구 논현동 276	200	서울특별시 강남구 선릉로 111길 27
서울특별시 강남구 청담동 106-21	200	서울특별시 강남구 선릉로 190길 124
서울특별시 강남구 역삼동 709	200	서울특별시 강남구 선릉로 85길 18
서울특별시 강남구 청담동 106-11	200	서울특별시 강남구 선릉로190길 120
서울특별시 강남구 역삼동 712	200	서울특별시 강남구 역삼로 307-1
서울특별시 강남구 역삼동 716-1	200	서울특별시 강남구 역삼로 315-1

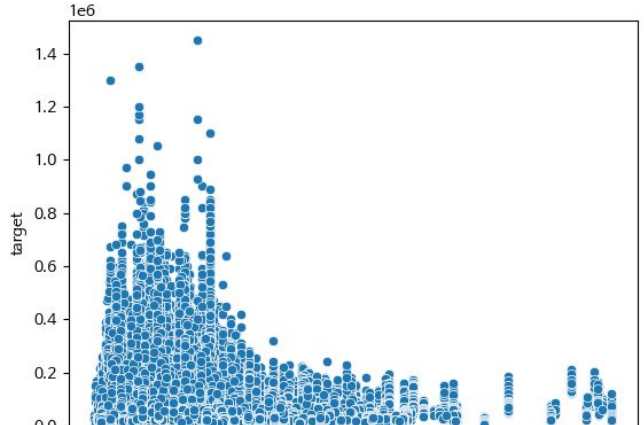
- ➔ 카카오 RestAPI로 최대한 결측값을 채움
- ➔ 그래도 채우지 못한 주소들 -> 도로명 주소로 일일이 변환
- ➔ (도로명 주소, 기존 아파트명)을 활용하여 추가로 해당 변수의 값을 채움
- ➔ 결측치 개수는 876232개 -> 12093개로 감소
- ➔ Dataset shape : (1116001, 17)

EDA : 역세권_파생변수 생성

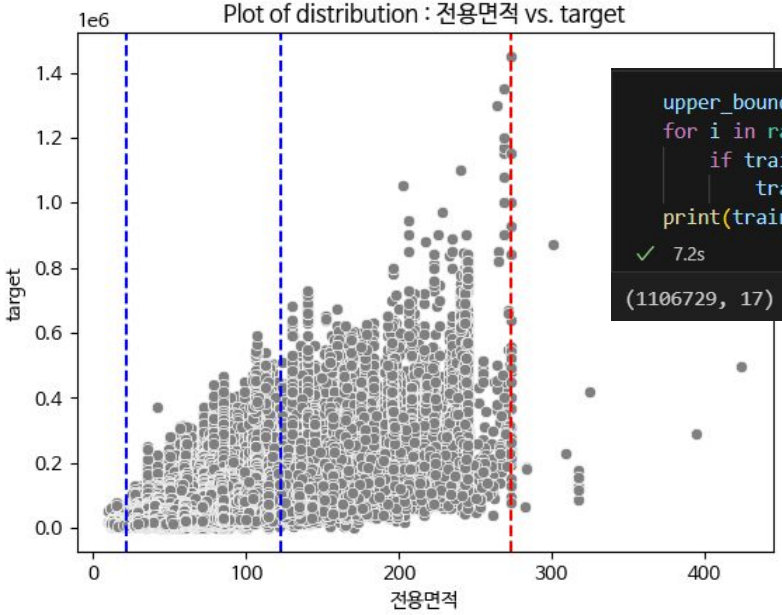
```python
# 지하철역과의 최단거리변수
sub_results = []
for _, place in sucess_df_v5.iterrows():
    place_y, place_x = place["좌표Y"], place["좌표X"]
    min_distance = float("inf")
    closest_stop = None

    for _, stop in sub_df.iterrows():
        stop_y, stop_x = stop["좌표Y"], stop["좌표X"]
        distance = haversine(place_y, place_x, stop_y, stop_x)

        if distance < min_distance:
            min_distance = distance
            closest_stop = stop["역사_ID"]

    sub_results.append({
        "Place": place["add_to_xy"],
        "Closest_역사 ID": closest_stop,
        "sub_km": min_distance
    })
sub_results_df = pd.DataFrame(sub_results)
```

- ➔ '역세권'을 고려
- ➔ Harversine 공식을 기반으로 고유한 아파트 ID마다 지하철역과의 최단거리 변수 생성
- ➔ "sub_shortest(km)

Target vs. sub_shortest(km) 산점도 분포를 통해 지하철역과의 거리가 가까울수록 집값이 높은 경향이 있다고 유추할 수 있음.



EDA : 전용면적 변수




```
upper_bound = train_tmp_[train_tmp_['target'] == train_tmp_.max()['target']]['전용면적'].values #273.96
for i in range(0,train_tmp_.shape[0]):
    if train_tmp_.loc[i,'전용면적'] > upper_bound:
        train_tmp_.loc[i,'전용면적'] == upper_bound
print(train_tmp_.shape)
```

✓ 7.2s

```
(1106729, 17)
```

```
'''
@@ 2. 전용면적 -> 전용면적to평수 -> 평수to그룹
'''
def m2_to_pyeonsoo(df):
    df['전용면적to평수'] = df['전용면적'] / 3.3058
    df['전용면적to평수'] = df['전용면적to평수'].astype(int)
    return df
```

➔ Target값에 대한 전용면적 변수의 분포를 미루어보아,
➔ 273.96m^2의 값들로 upper bound를 수렴하였음
➔ 전용면적 변수를 평수로 변환하여 '전용면적to평수'라는 변수 추가

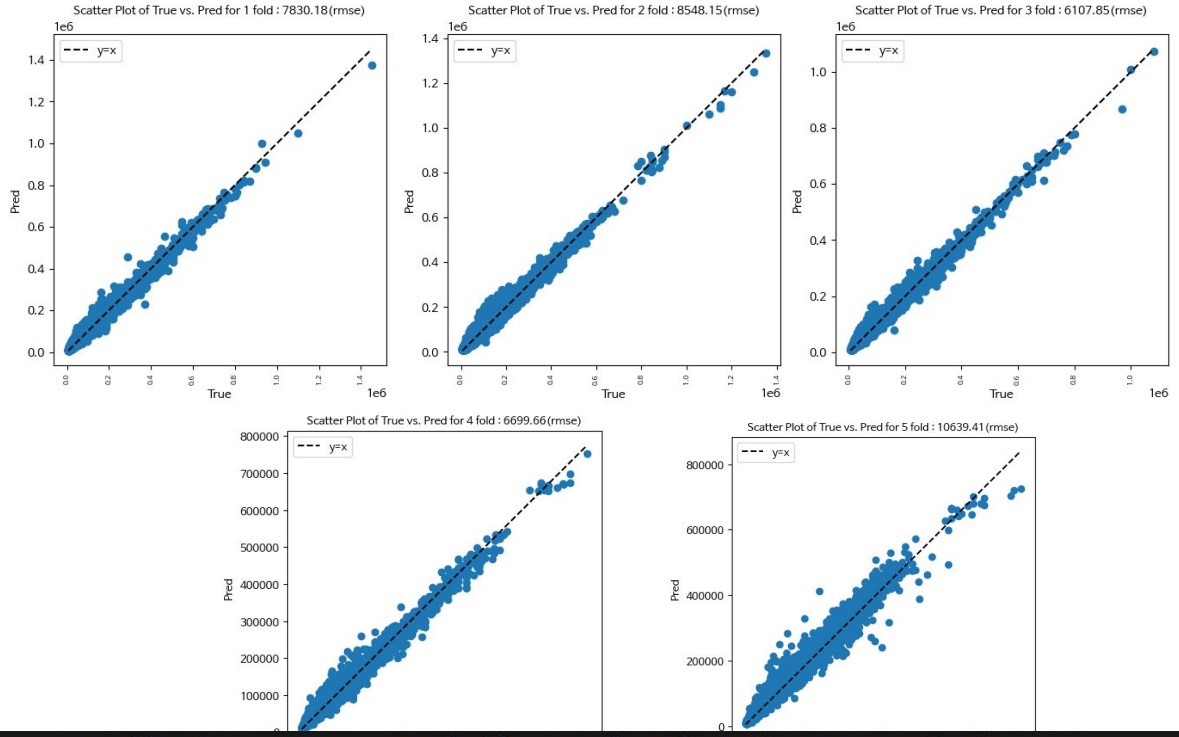
Model Training

```
dt_train_ = dt_train[dt_train['계약년월'] >= 201801]
```

```
n_splits = 5
tscv = TimeSeriesSplit(n_splits=n_splits)

num_leaves_list = [7,100,300]  # 트리의 리프 노드 수
max_depth_list = [7,10] # 트리의 최대 깊이
lr_list = [0.1]     # 학습률
n_estimators_list = [1000,2000] #부스팅 라운드 수
min_data_in_leaf_list = [10]
feature_fraction_list = [0.8,1.0]

all_ = [num_leaves_list, max_depth_list, lr_list, n_estimators_list,
        min_data_in_leaf_list, feature_fraction_list]
hy_space = [s for s in itertools.product(*all_)]

results_lgbm = []
for hy in tqdm(hy_space):
    print(hy)
    mdllgbm = LGBMRegressor(num_leaves = hy[0],
                            max_depth = hy[1],
                            learning_rate = hy[2],
                            n_estimators = hy[3],
                            min_data_in_leaf =hy[4],
                            feature_fraction = hy[5] ,
                            boosting_type = 'dart',
                            objective='regression', random_state=777, n_jobs=-1)

    fold_lgbm_rmse = []
    for fold, (train_idx, val_idx) in enumerate(tscv.split(X_train)):
        print(f"Fold {fold + 1}")
        X_train_, X_val_ = X_train.iloc[train_idx], X_train.iloc[val_idx]
        y_train_, y_val_ = y_train.iloc[train_idx], y_train.iloc[val_idx]
        # 모델 학습
        mdllgbm.fit(X_train_, y_train_,
                    eval_set=[(X_val_, y_val_)],
```

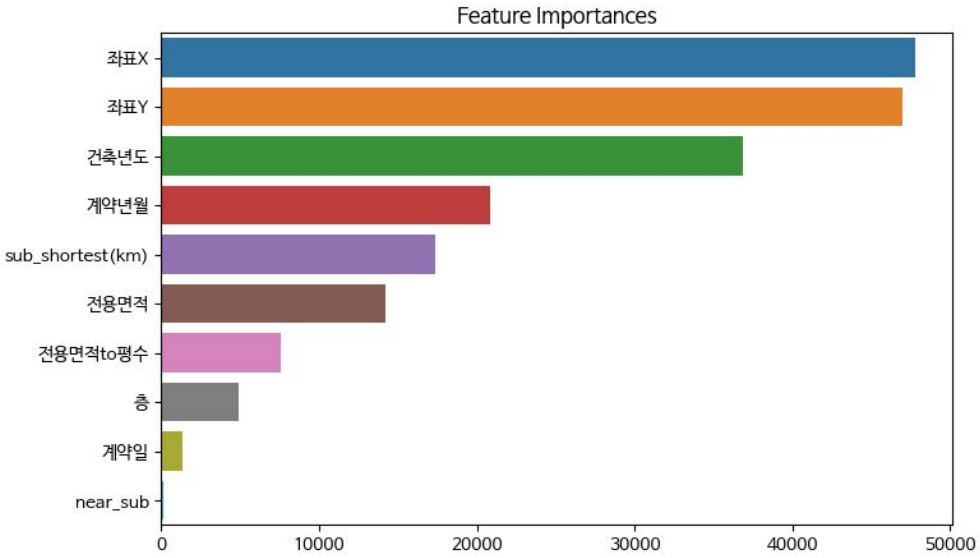
- ➔ 고유한 아파트 ID 마다 시계열 특성을 지닌 데이터
- ➔ target (y)의 분포를 통해, 2018년 기준으로 가격이 급격히 증가했음을 고려함
- ➔ train_set을 2018년 이후로 지정
- ➔ For문을 통해, validation_set에 대한 최소 5-fold 평균 rmse를 도출하는 lgbmregressor의 주요 하이퍼파라미터 set 찾기
- ➔ Avg rmse ; 18937.04

Model Validation : Scatterplot of True vs. pred for each fold




```
lgbm_mdl = LGBMRegressor(num_leaves = 100, max_depth = 10, learning_rate=0.1, n_estimators=2000,
                         min_samles_leaf = 10, feature_fraction = 1.0,
                         boosting_type = 'dart',objective='regression', random_state=777, n_jobs=-1)
```

Inferences : Feature Importances



1. 해당 모델은 좌표(x,y) / 건축년도 / 계약년월 / sub_shortest(km) (역 최단거리) 순서로 변수의 중요도를 도출

2. 위 결과를 통해, 아파트 매매가의 변동은 위치 / 건축년도 / 역세권 변수가 많은 영향을 끼침을 알 수 있다.

3. 역의 최단거리를 뜻하는 연속형 변수가 중요함에도 500을 기준으로 2개 범주로 나눈 near_sub 변수의 중요도가 낮게 학습된 이유를 무엇인가.

Inferences : 회고 및 소감

1. 시간이 더 많았다면 앞의 결과를 바탕으로 연관된 파생변수를 더 추가해보면 좋았을 것 같다.
   a. '층' , '아파트명'도 target과 상관관계가 높았음
   b. 특히 아파트마다 고층이 다름을 고려해볼 것

2. 추가로, 더 많고 다양한 범위의 하이퍼파라미터를 적용해보고 싶다.

3. Validation 과정의 scatterplot 결과에서도 알 수 있듯이, 상대적으로 고액의 매매가는 true 값보다 낮은 값을 예측하는 경향이 나타남
   a. target 분포를 바탕으로 값의 기준을 정한 뒤, 고가 예측 모델 / 저가 예측 모델로 나누어 예측값의 앙상블

소감 : rmse 값을 비롯하여 결과가 아쉽지만, 내/외부 데이터 수집부터 내가 다뤄보지않았던 시계열 데이터를 활용한 예측 경진대회를 경험할 수 있어서 매우 유익했다!