House Price Prediction | 아파트 실거래가 예측

서울시 아파트 실거래가 매매 데이터를 기반으로 아파트 가격을 예측하는 대회

ML Team 5. 기지개

목차


LITTERATURE RUSSE
LITTERATURE HISPANIQUE
LITTERATURE PORTUGAISE
LITT. JAPONAISE
ALLEMANDE

1 팀원 소개

팀원 소개

김 기 홍

EDA

데이터 전처리

이 윤 재

데이터 전처리

피처 엔지니어링

이 재 명

피처 엔지니어링

하이퍼파라미터 튜닝

장 은 지

데이터 전처리

하이퍼파라미터 튜닝

최 지 미

퍼실리테이터

피처 엔지니어링


LITTERATURE RUSSE
LITTERATURE HISPANIQUE
LITTERATURE PORTUGAISE

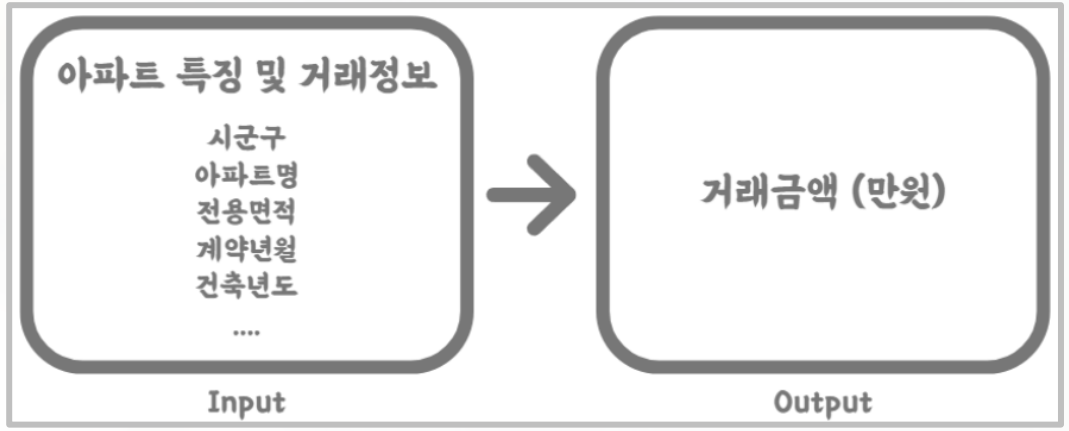
2 대회 소개

2. 대회 소개: 대회 개요

개 요

주제 : 서울시 아파트 실거래가 매매 데이터를 기반으로 아파트 가격 예측

목표 : 2007.01 ~ 2023.06 시점의 데이터를 이용하여 이후 3개월(2023.09)의 부동산 실거래가 예측



input : 9,272개의 아파트 특징 및 거래 정보

ouput : 9,272개의 input에 대한 예상 아파트 거래 금액

결과물 : csv확장자 파일 제출

프로젝트 전체 기간

7월 9일 (화) 10:00 ~ 7월 19일 (금) 19:00 (2주)

2. 대회 소개: 대회 개요

평가지표

RMSE(Root Mean Squared Error)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE는 예측된 값과 실제 값 간의 평균 편차를 측정함

아파트 매매의 맥락에서는 회귀 모델이 실제 거래 가격의 차이를 얼마나 잘 잡아내는지 측정함

2. 대회 소개: 데이터 개요

제공된 데이터 설명

파일명	내용	크기	shape
Train.csv	아파트의 정보에 대한 변수와 거래시점에 대한 변수 (target 포함)	244 MB	1118822 x 52
Test.csv	Train.csv와 동일 (target 제외)	2.45 MB	9272 x 51
sample_submission.csv	제출 데이터 예시	60.1 KB	9272 x 1
bus_feature.csv	추가데이터 - 서울시 버스정류장 위치 정보	924 KB	12584 x 6
subway_feature.csv	추가데이터 - 서울시 지하철역 위치 정보	35.6 KB	768 x 5

예 시

	시군구	번지	본번	부번	아파트 명	전용면적 (㎡)	계약년월	계약 일	층	건축년도	...	건축면적	주차대수	기타/의무/임대/임의=1/2/3/4	단지승인일	사용허가여부	관리비 업로드	좌표X	좌표Y	단지신청일	target
0	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차 우성	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	개포6차 우성	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	개포6차 우성	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	개포6차 우성	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	개포6차 우성	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

Train 데이터

	노드 ID	정류소번호	정류소명	X좌표	Y좌표	정류소 타입
0	100000001	1001	종로2가사거리	126.987752	37.569808	중앙차로
1	100000002	1002	창경궁.서울대학교병원	126.996566	37.579183	중앙차로
2	100000003	1003	명륜3가.성대입구	126.998251	37.582581	중앙차로
3	100000004	1004	종로2가.삼일교	126.987613	37.568579	중앙차로
4	100000005	1005	혜화동로터리.여운형활동터	127.001744	37.586243	중앙차로
...	...	...	...	...	...	...
12579	124000334	25995	우성아파트	127.139338	37.550386	일반차로
12580	124000333	25996	우성아파트	127.140046	37.550643	일반차로
12581	124000332	25997	조일약국	127.123596	37.533630	일반차로
12582	124000331	25998	성내시장	127.125497	37.536155	일반차로
12583	124000330	25999	천호우체국.로데오거리	127.127337	37.540343	일반차로

12584 rows × 6 columns

버스정류장 추가 데이터

	역사_ID	역사명	호선	위도	경도
0	9996	미사	5호선	37.560927	127.193877
1	9995	강일	5호선	37.557490	127.175930
2	4929	김포공항	김포골드라인	37.562360	126.801868
3	4928	고촌	김포골드라인	37.601243	126.770345
4	4927	풍무	김포골드라인	37.612488	126.732387
...	...	...	...	...	...
763	154	종로5가	1호선	37.570926	127.001849
764	153	종로3가	1호선	37.570406	126.991847
765	152	종각	1호선	37.570161	126.982923
766	151	시청	1호선	37.565715	126.977088
767	150	서울역	1호선	37.556228	126.972135

768 rows × 5 columns

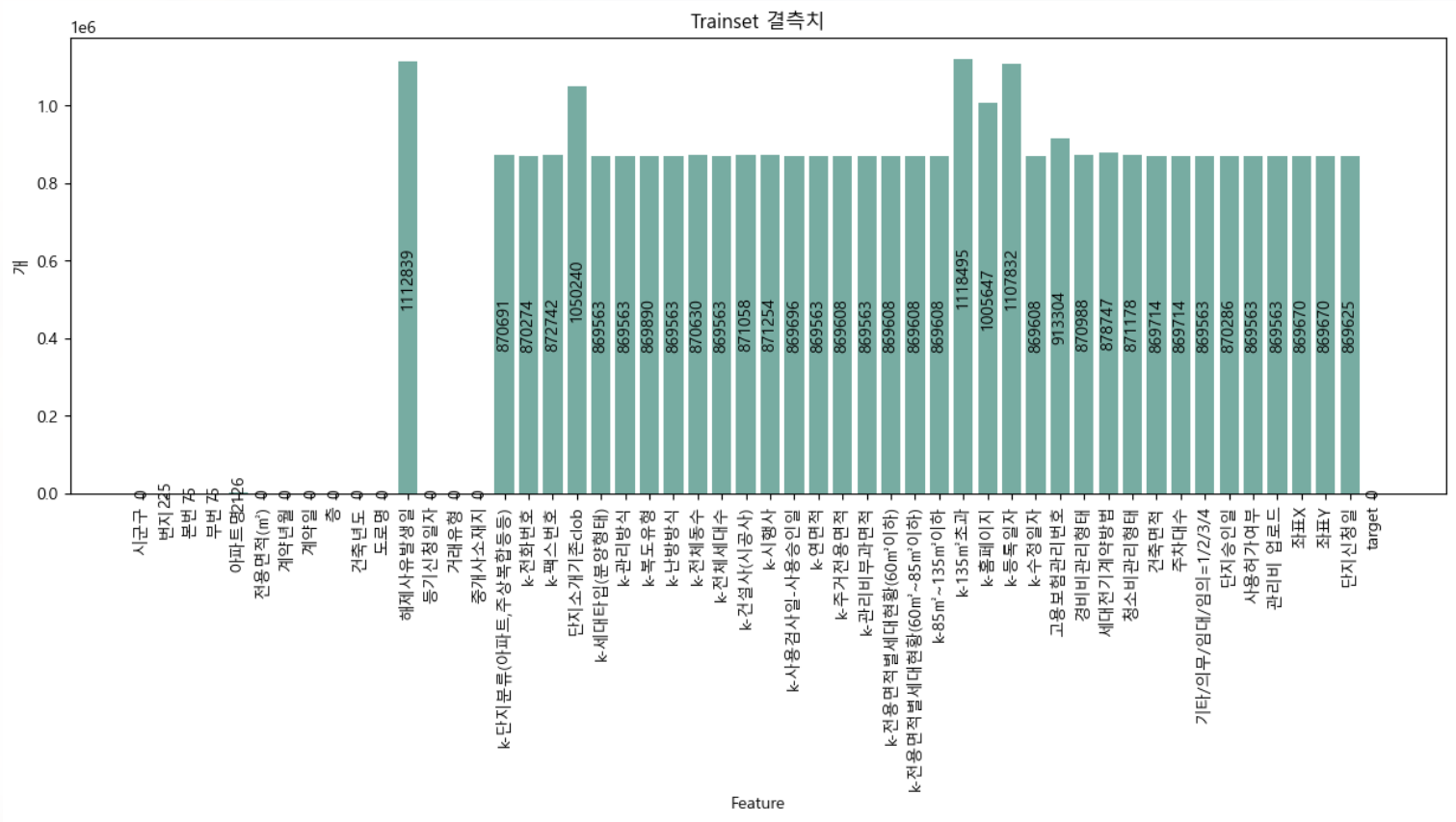
지하철역 추가 데이터

3 Data Description

3. Data Description: 데이터의 기초 통계 및 정보 요약

데이터 기초 통계 및 정보 요약

Train.csv

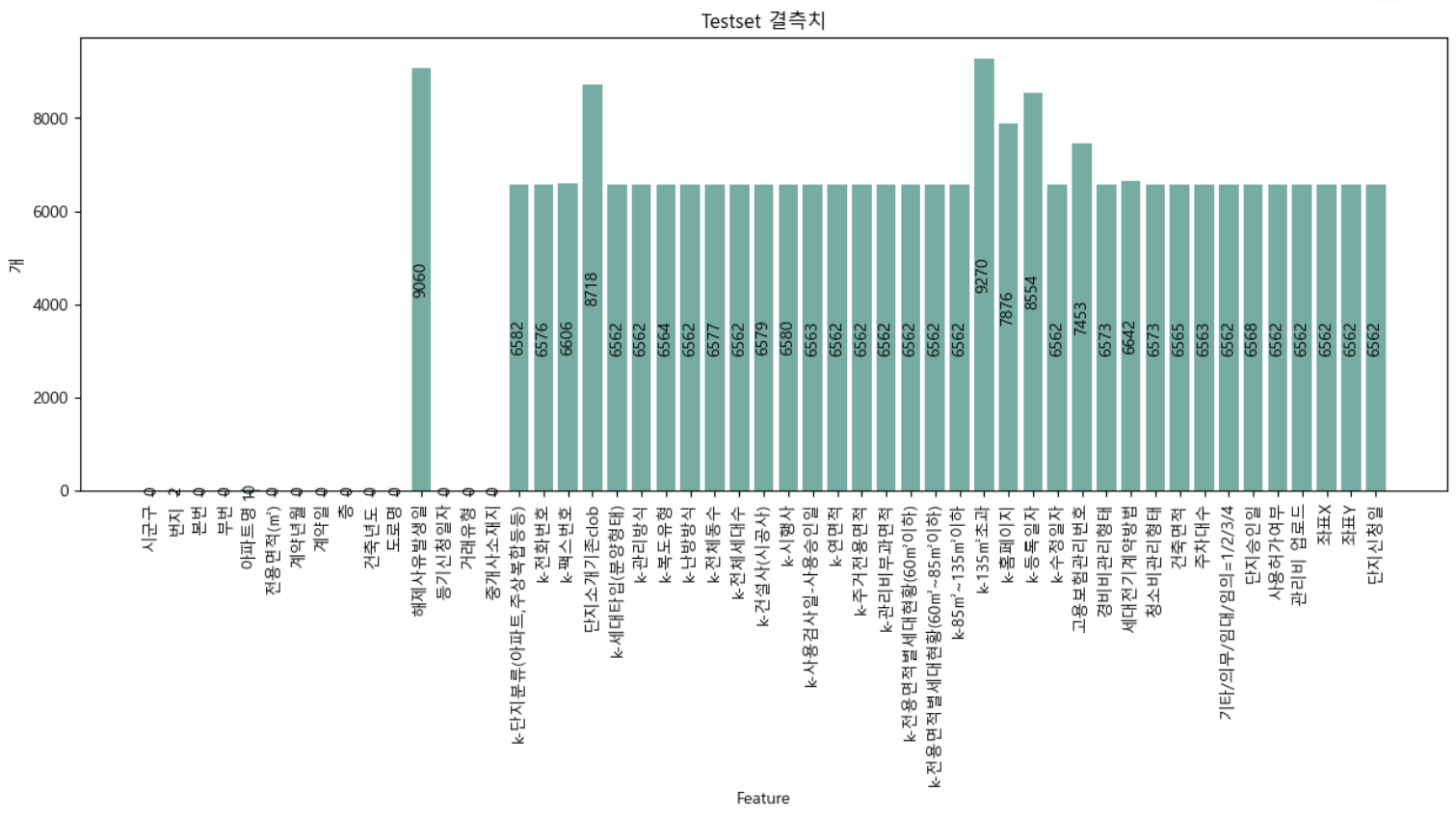


- Shape : 1118822 x 52
- Train.csv의 총 41개 열에서 결측치 발견
- Categorical dtype 29개
- Numerical dtype 23개

3. Data Description: 데이터의 기초 통계 및 정보 요약

데이터 기초 통계 및 정보 요약

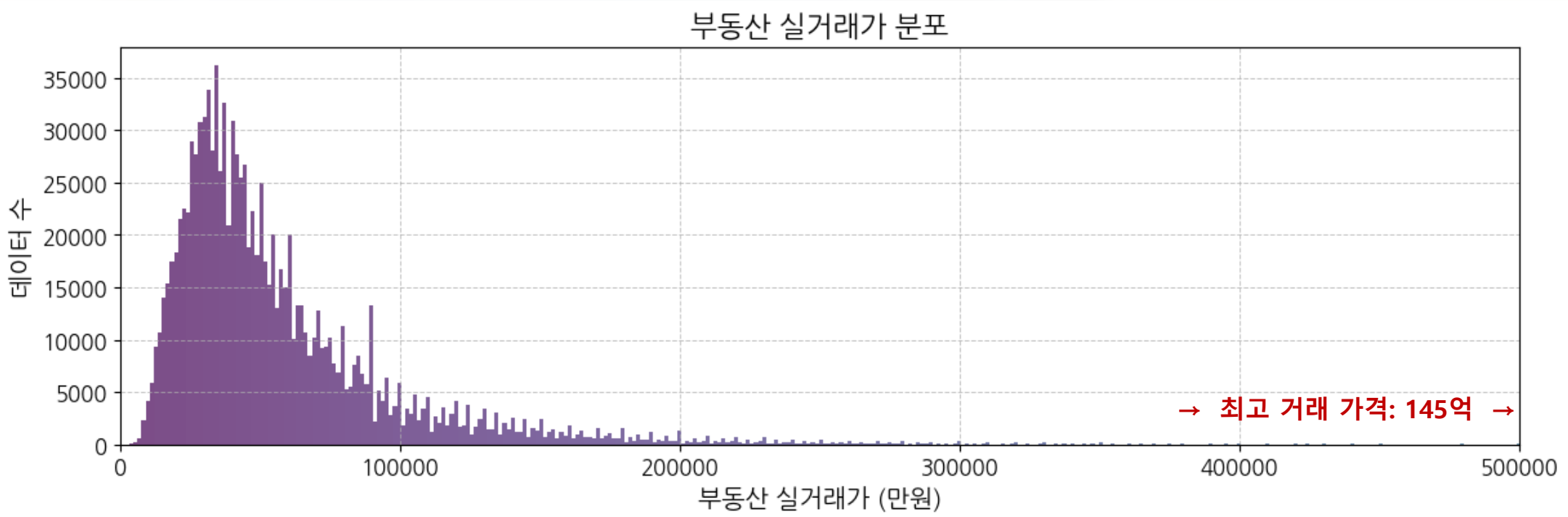
test.csv



- Shape : 9272 x 51
- Test.csv의 총 39개 열에서 결측치 발견
- Categorical dtype 28개
- Numerical dtype 22개

3. Data Description: EDA – target(부동산 실거래가) 분포(1)

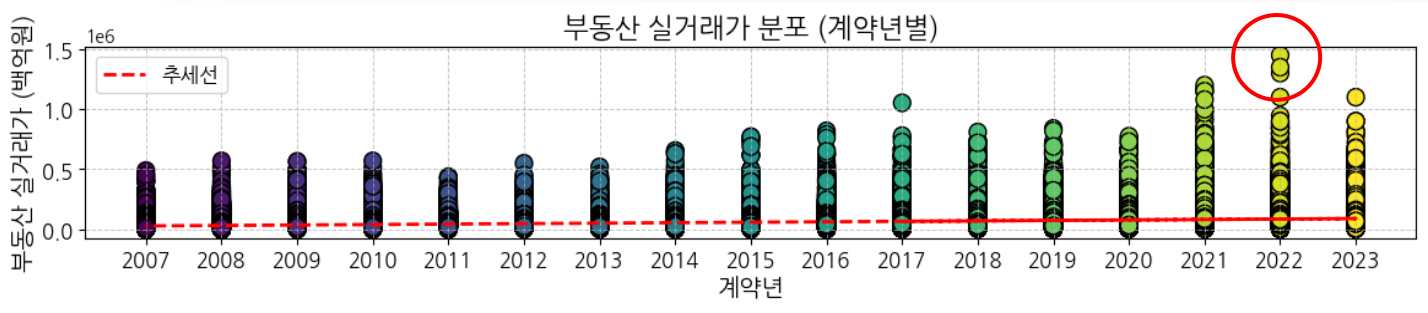
target(부동산 실거래가) 분포



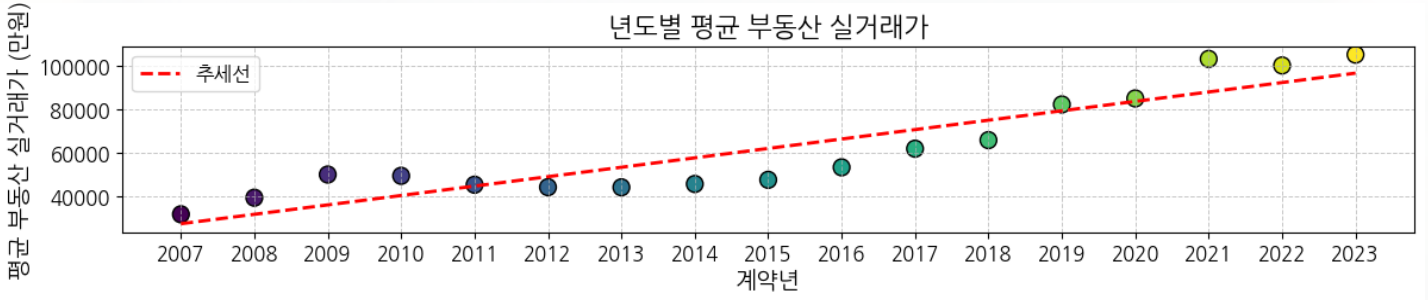
- 양의 왜도(Right skewed) 형태
- 3 ~ 4억 사이의 데이터가 가장 많음
- 20억 이상의 부동산은 데이터 양이 적음 -> 예측이 어려울 것으로 예상

3. Data Description: EDA – target(부동산 실거래가) 분포(2)

target(부동산 실거래가) 분포



- 모든 부동산 거래내역을 년도별로 표시한 차트
- 최고 가격으로 거래된 건 : 2022년도에 분포함

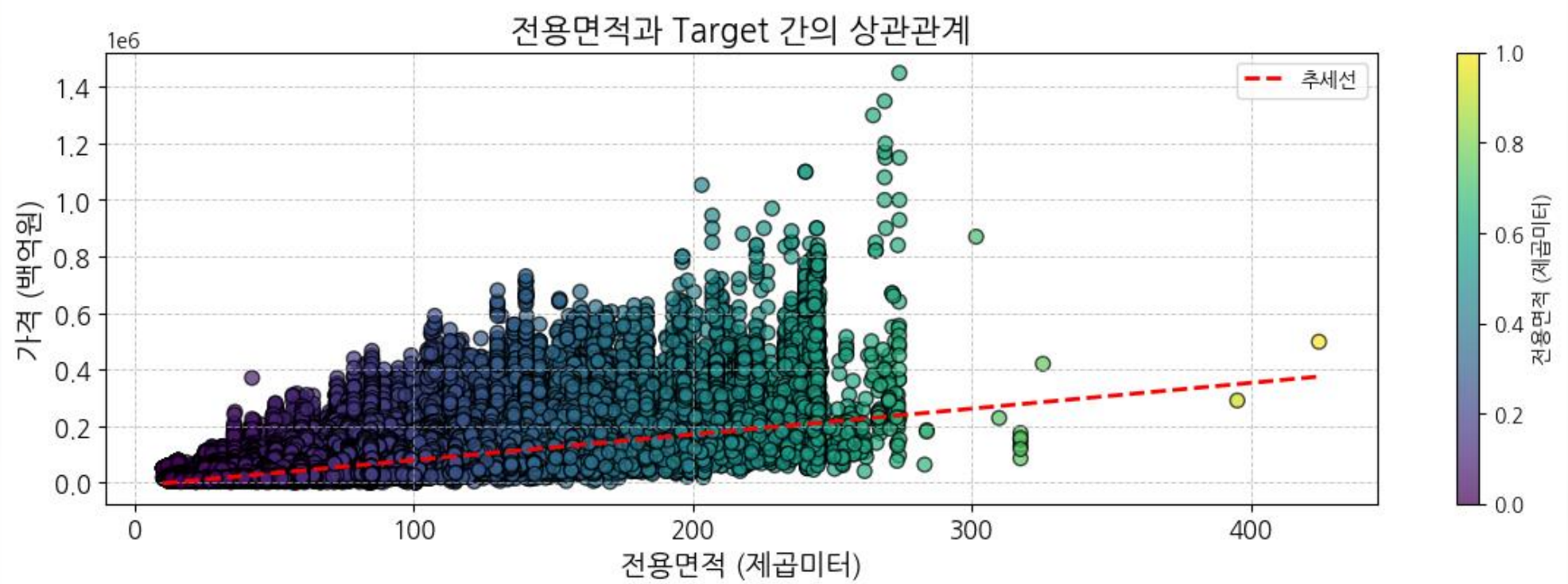


- 년도 별 부동산 실거래가 평균을 구분한 차트
- 국소적으로 하락 하기도 했으나,
  전반적으로 상승 추세

계약 년도와 target(부동산 실거래가)는 양의 상관관계로 보임

3. Data Description: EDA – 전용면적(1)

전용면적

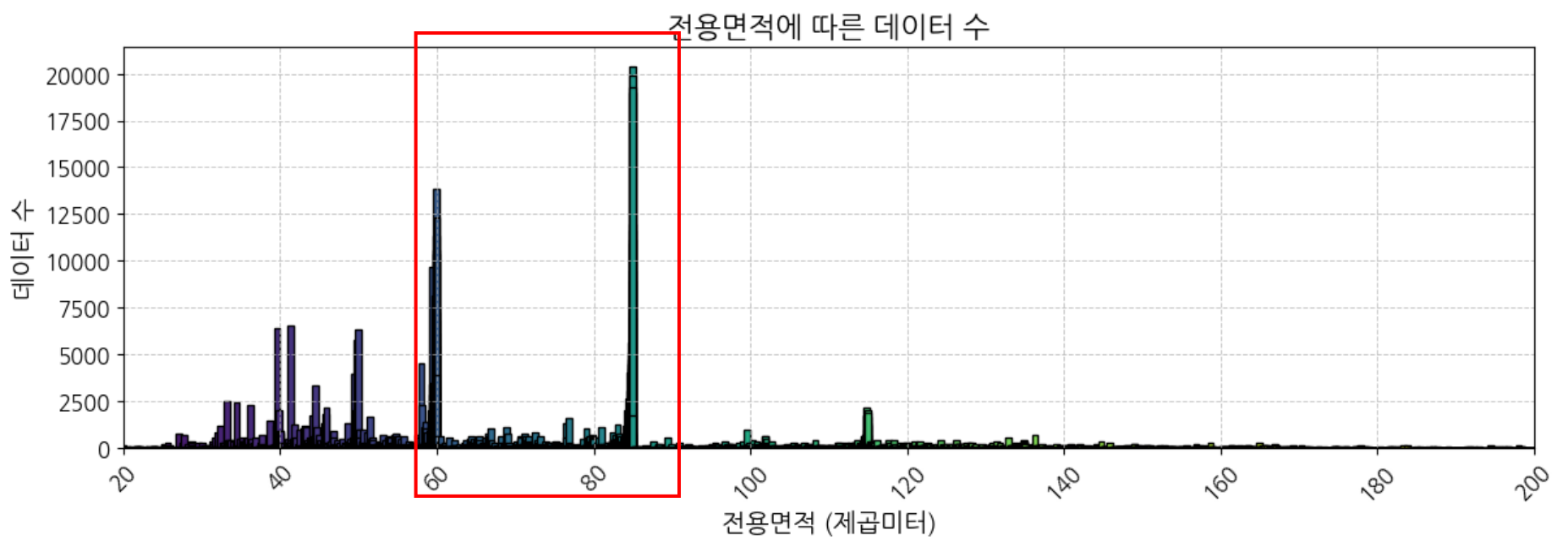


- 일반적으로 전용면적이 클 수록 target(부동산 실거래가) 값이 높음

전용면적과 target 값은 양의 상관관계로 보임

3. Data Description: EDA – 전용면적(2)

전용면적

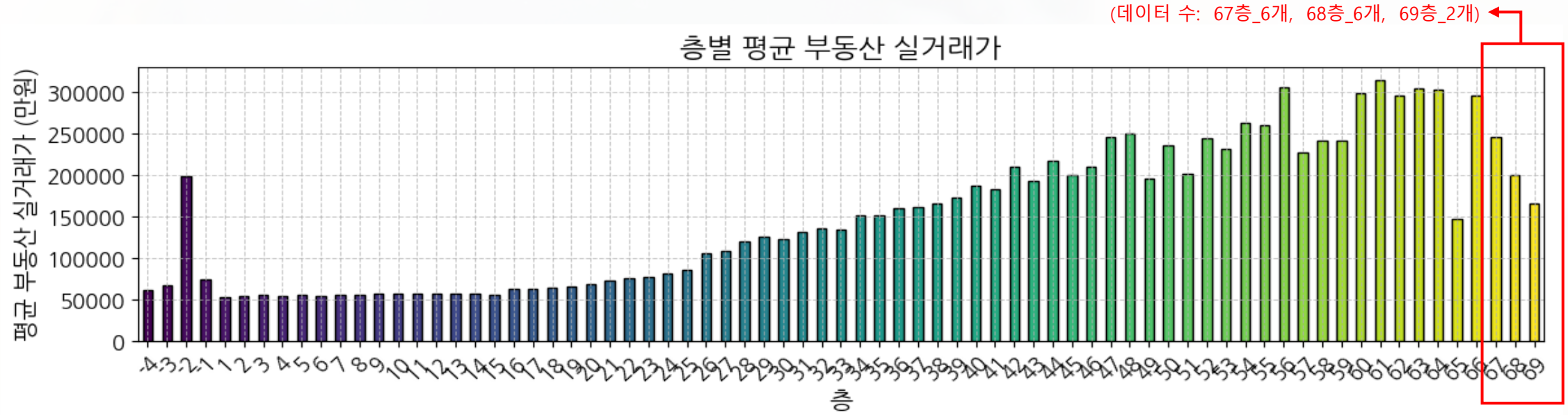


- 국민 평수로 불리우는 **84㎡ 및 59㎡** 부근의 데이터가 가장 많아 보임.
- 59㎡(±1) 및 84㎡(±1) 면적이 전체 데이터에서 차지하는 비율: **약 53%**
- 90㎡ 이하의 면적이 전체 데이터에서 차지하는 비율: **약 83%**

비슷한 값을 그룹화 하여 분석할 필요가 보임

3. Data Description: EDA – 층

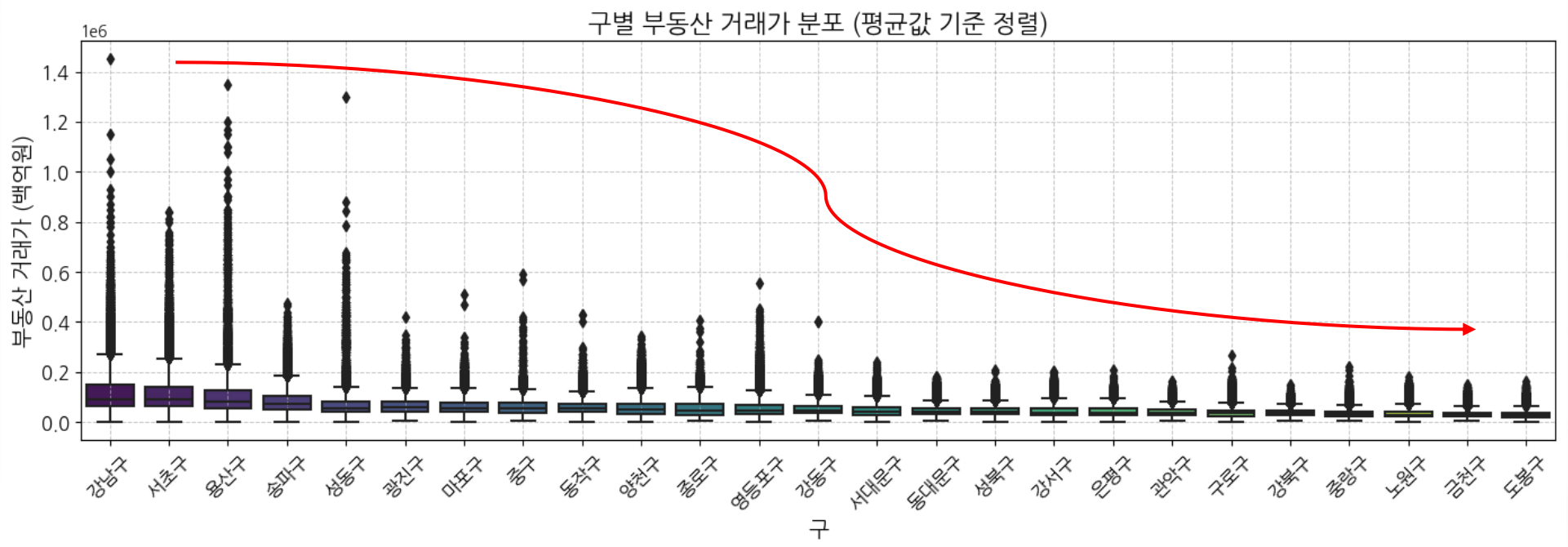
층



- 일반적으로 층이 높을수록 부동산 가격이 높아지는 경향을 보임
- 67층 이상의 고층에서 가격이 다시 하락하는 경향을 보임 ➡ 데이터 개수가 적어 경향성을 함부로 속단하기 어려움

3. Data Description: EDA – 구

구



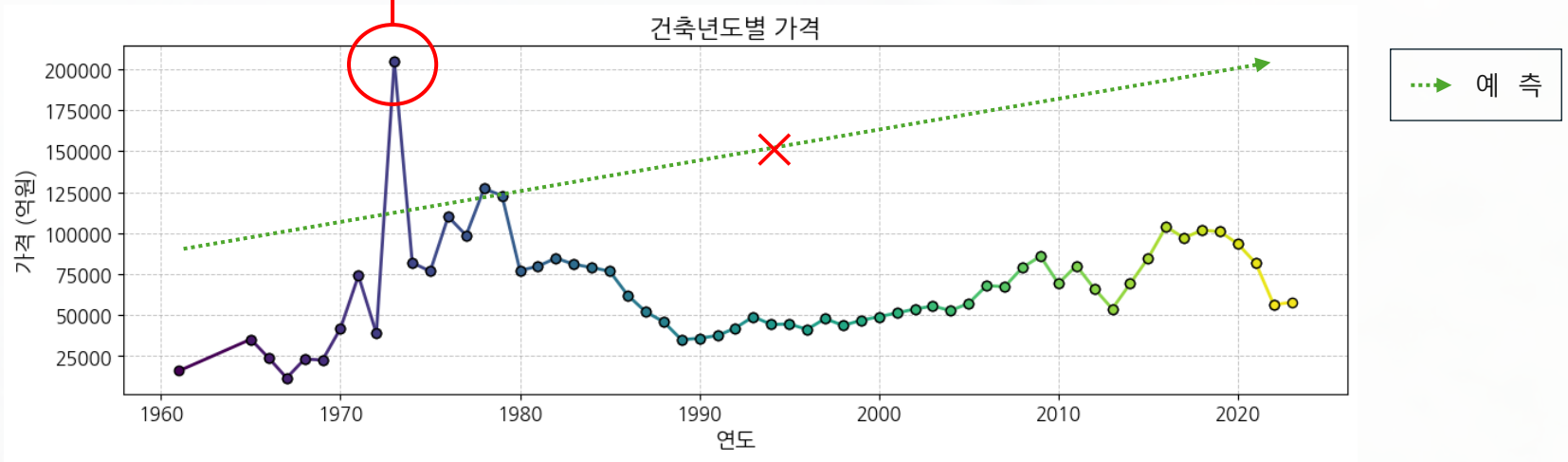
- 강남구, 서초구, 용산구, ..., 순서로 부동산 가격이 높음
- 구 별로 부동산 가격 분포의 차이가 심함

'구', '동'과 같은 주소 및 위치 정보가 target(부동산 실거래가) 값 예측에 중요하게 작용할 것으로 예상됨

3. Data Description: EDA – 건축 년도

건축 년도

1973년 : 서초구 반포주공1단지 등의 부동산이 포함된 것으로 확인



- 1961년 ~ 2023년 사이에 건축된 다양한 부동산이 존재함
- 신축 부동산이 오래된 부동산에 비해 상대적으로 높은 가격일 것으로 예상

  ➡ 건축 년도 데이터만으로는 거래가격과의 경향성을 찾기 어려워 보임
- 오히려, 다른 feature가 target(부동산 실거래가) 값에 더 큰 영향을 줄 것으로 예상됨

3. Data Description: 데이터 전처리

데이터 전처리 도로명주소 이상값 처리

도로명
91517
91518
91519
91520
91521
438860

도로명	
67838	37
67842	37
67845	37
67846	37
67847	37
...	...
539825	43
581039	20
728766	37
728767	37
728768	37

공백 및 주소가 잘못된 데이터 발견

도로명	
91517	NaN
91518	NaN
91519	NaN
91520	NaN
91521	NaN
438860	NaN

도로명	
67838	NaN
67842	NaN
67845	NaN
67846	NaN
67847	NaN
...	...
539825	NaN
581039	NaN
728766	NaN
728767	NaN
728768	NaN

NaN으로 변환

- 도로명 주소가 6글자 이하인 데이터 검색 · 결측치로 변환
  - ➡ 총 2339개 데이터 결측치로 변환
- 결측치로 변환된 주소는 지번주소를 이용하여 채움
  - ➡ 카카오 API를 사용
- 현재 도로명 주소가 존재하지 않으면 변환되지 않음
  - ➡ 991개의 데이터는 결측치는 지번주소로 대체

3. Data Description: 데이터 전처리

데이터 전처리 아파트명 결측값 처리

- 아파트명 결측치 : 2136개
  - ➡ 결측치를 채우기 위해 '건물명 ' 칼럼 생성

```
df2[df2['건물명'].isna()]['번지주소_2'].value_counts()
✓ 0.0s
번지주소_2
서울특별시 영등포구 대림동 1101-1     127
서울특별시 동대문구 장안동 400-3       34
서울특별시 구로구 구로동 745-57        25
서울특별시 구로구 구로동 799-11        24
서울특별시 서초구 서초동 1487-1        23
                                      ...
서울특별시 구로구 구로동 797-9          1
서울특별시 구로구 구로동 792-20         1
서울특별시 구로구 구로동 752-42         1
서울특별시 동대문구 이문동 257-516      1
서울특별시 구로구 구로동 743-14         1
Name: count, Length: 281, dtype: int64
```

- 건물명 결측치를 주소 기준으로 value_counts
  - ➡ 제일 수량이 많은 주소부터 내림차순 정렬

```
# 특정 조건에 맞는 행에서 결측치를 '뜨라네'로 채우기
condition = df2['번지주소_2'].str.contains('서울특별시 강남구 역삼동 828-21')
df2.loc[condition, '건물명'] = df2.loc[condition, '건물명'].fillna('뜨라네')
```

- 결측치에 해당하는 주소를 네이버 주소에서 수동검색
  - ➡ 상가건물이나 교회, 어린이집, 빌라, 주택 등의 건물인 경우가 많음
  - ➡ **빌라, **주택 **아파트 등 실제 건물명을 있으면 값 입력
  - ➡ 건물명이 없거나 상가건물은 '0'으로 채움
- 확인하지 못한 350개 데이터는 동일 주소 묶음이 1~5개인 곳으로 '1'로 처리

3. Data Description: 데이터 전처리

데이터 전처리

X좌표, Y좌표 살리기


```
{
    'status': 'OK',
    'meta': {'totalCount': 1, 'page': 1, 'count': 1},
    'addresses': [
        {
            'roadAddress': '서울특별시 강남구 논현로2길 34 새롬아파트',
            'jibunAddress': '서울특별시 강남구 개포동 1164-12 새롬아파트',
            'englishAddress': '34, Nonhyeon-ro 2-gil, Gangnam-gu, Seoul, Republic of Korea',
            'addressElements': [
                { 'types': ['SIDO'], 'longName': '서울특별시', 'shortName': '서울특별시', 'code': '' },
                { 'types': ['SIGUGUN'], 'longName': '강남구', 'shortName': '강남구', 'code': '' },
                { 'types': ['DONGMYUN'], 'longName': '개포동', 'shortName': '개포동', 'code': '' },
                { 'types': ['RI'], 'longName': '', 'shortName': '', 'code': '' },
                { 'types': ['ROAD_NAME'], 'longName': '논현로2길', 'shortName': '논현로2길', 'code': '' },
                { 'types': ['BUILDING_NUMBER'], 'longName': '34', 'shortName': '34', 'code': '' },
                { 'types': ['BUILDING_NAME'], 'longName': '새롬아파트', 'shortName': '새롬아파트', 'code': '' },
                { 'types': ['LAND_NUMBER'], 'longName': '1164-12', 'shortName': '1164-12', 'code': '' },
                { 'types': ['POSTAL_CODE'], 'longName': '06313', 'shortName': '06313', 'code': '' }
            ],
            'x': '127.0529493',
            'y': '37.4731929',
            'distance': 0.0
        }
    ],
    'errorMessage': ''
}
```


```python
def get_location(address, addrType):
  url = 'https://naveropenapi.apigw.ntruss.com/map-geocode/v2/geocode?query=' + address
  headers = {"X-NCP-APIGW-API-KEY-ID": "~~your-key-id~~", "X-NCP-APIGW-API-KEY": "~~your-api-key~~"}
  api_json = json.loads(str(requests.get(url,headers=headers).text))

  print(f"api_json = {api_json}")

  if api_json['status'] == 'OK' and len(api_json['addresses']) > 0:
    # 2개 이상인 경우도 있을지 확인하기위해..
    is_multi_addr = len(api_json['addresses']) > 1

    x = "0"
    y = "0"
    road_address = ""

    for addrDict in api_json['addresses']:
      if addrType == 'land':
        if getJibunAddr(addrDict) != address: continue
      else:
        if getRoadAddr(addrDict) != address: continue

      x = addrDict['x']
      y = addrDict['y']
      road_address = getRoadAddr(addrDict)
      break

    if x != "0":
      return { "X": x, "Y": y, 'road_address': road_address, 'is_multi_addr': is_multi_addr }

  return {"Y": "0", "X": "0"}
```

- 지번주소 기준으로 네이버 Maps API 로 위도, 경도 값을 채움
- 조회 안되는 지번주소 → 도로명주소를 대신 사용
- 도로명주소로도 조회 안되는 경우 → 수동으로 구글에서 검색해서 가장 비슷한 주소로 좌표 구함

3. Data Description: 외부 데이터 활용

외부 데이터 활용

서울 열린데이터 광장 SEOUL OPEN DATA PLAZA

- 서울시 공동주택아파트정보
- 서울시 가계대출규모

KB 부동산 데이터허브

- KB부동산 매수우위지수

KOSIS 국가통계포털 KOrean Statistical Information Service

- 서울시 주택 담보 대출
- 기대 인플레이션율

NAVER Maps

- X, Y 좌표

DATA 공공데이터포털 .GO.KR

- 서울시 초등학교

kakaomap

- 도로명주소

부동산 빅데이터 플랫폼 Real Estate Big Data Platform

- 아파트 단지 인근 학원 교습소 정보

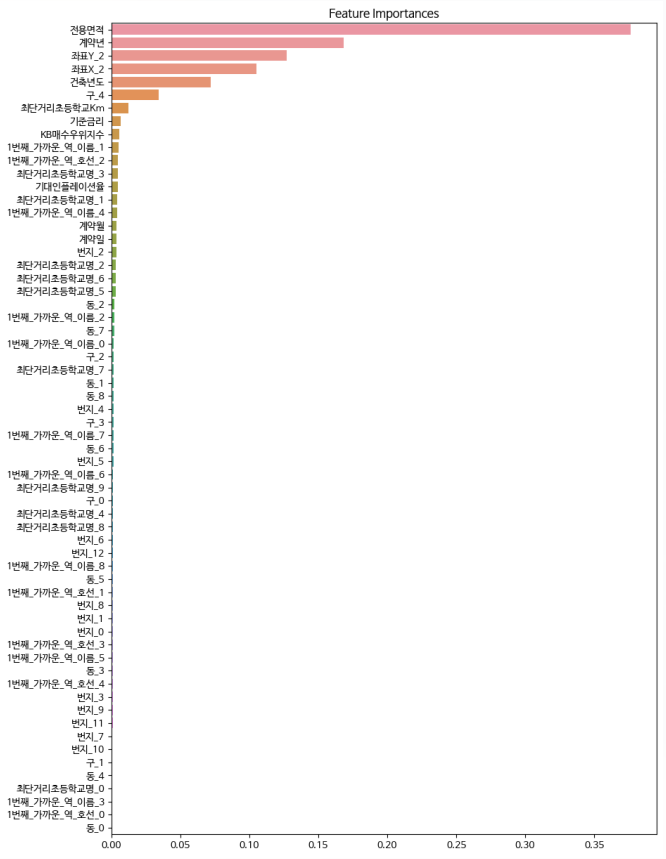
3. Data Description: Feature Engineering

Feature Engineering

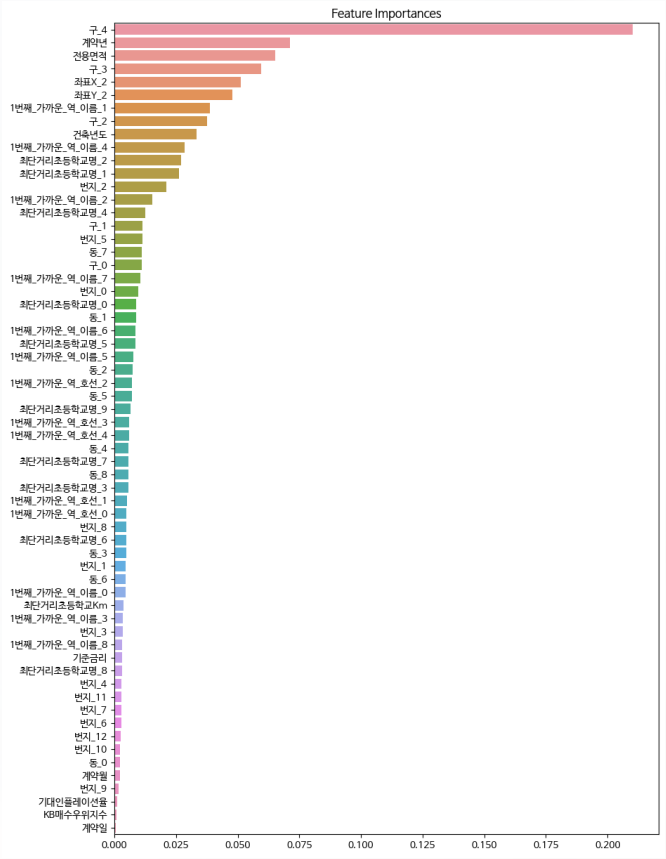
분 류	세부 분류	Columns
아파트 정보	-	건물ID, 세대별 주차대수, 국민 평수
인프라	학 군	최단거리 초등학교 이름, 최단거리 초등학교 거리, 500m 이내 학원 수
	지하철	1/2/3번째 가까운역 이름, 1/2/3번째 가까운 역 호선, 1/2/3번째 가까운 역 거리, 1/2/3번째 가까운 역 도보시간, 5분/5~10분/10~15분 역 개수, 0.5km/1km 내 지하철 수, 0.5km/1km 내 지하철 호선 수, 0.5km/1km 내 가장 가까운 지하철역, 0.5km/1km 내 가장 가까운 호선, 0.5km/1km 내 2호선 여부
	버 스	1/2/3번째 가까운 버스정류장 이름, 1/2/3번째 가까운 버스정류장 거리, 1/2/3번째 가까운 버스정류장 도보시간, 5분/5~10분/10~15분 버스정류장 개수
가 격	-	가장 비싼 아파트, 가격 높은 동, 동별 평균가격, 구별 평균가격, 층 평균가격, 층 거래수, 전용면적 평균가격, 전용면적 거래수, 층 전용면적 평균가격, 층 전용면적 거래수, 건물 평균가격, 건물 거래수, 건물 평균가격 시간가중치, 주변 평균가격, 시간가중치 주변가격, 거리가중치 주변가격
기 타	-	최근 거래 경과일, 주변 거래 수

3. Data Description: Feature Select & Split

Feature Select



RandomForest_FI



XGBoost_FI

- 기존 : 동, 구, 번지, 전용면적, 계약년월일, 건축년도, 좌표X, Y

- 추가 : 최단거리 초등학교 이름, 거리, 최단거리 역 이름, 호선명, KB매수우위지수, 기준금리, 기대인플레이션율

- 기존 10개 + 추가 7개 = 총 17개

- Split : Hold-out

Train : 895,057(80%)

Test : 223,764(20%)


LITTERATURE RUSSE
LITTERATURE HISPANIQUE
LITTERATURE PORTUGAISE
LITT. JAPONAISE
ALLEMANDE

4 Modeling

4. Modeling: Model Select – from Various Models

사용한 모델

Model	모델 특성
Random Forest	훌륭한 성능, 빠른 학습 속도
LightGBM	훌륭한 성능, 빠른 학습 속도
XGBoost	훌륭한 성능, 빠른 학습 속도
CatBoost	훌륭한 성능, 빠른 학습 속도

사용하지 않은 모델

Model	모델 특성
Linear Regression	예측 성능이 낮음
Decision Tree	예측 성능이 낮음
Gradient Boosting	느린 학습 속도

4. Modeling: Hyperparameter Tuning

Hyperparameter Tuning

실험 날짜	실험 시간	작성자	모델	전처리	Train data 수	Valid data	Test data 수	train RMSE	valid RMSE	valid R-squared	valid MAE	valid MedAE	vaild MAPE	test RMSE	Submission 결과	Submission Final	하이퍼파라미터 1	하이퍼파라미터 2	하이퍼파라미터 3
2024. 7. 17	23:30	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2418.9503	6507.7472	0.9805	2877.0491	1317.0000	4.79%		17841.8191		n_estimators=100	-	random_seed=42
2024. 7. 17	20:13	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2460.6588	6584.2471	0.9801					18213.4606		n_estimators=100		
2024. 7. 17	22:50	이재명	앙상블(RandomForest, Gra	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	-	-	-	-	-	-	-	25075.1486	-	n_estimators=100	-	random_seed=42
2024. 7. 17	17:50	이재명	앙상블(RandomForest, Gra	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	-	12806.4665	0.9246	-	-	-	-	32407.6061	-	n_estimators=100	-	-
2024. 7. 18	22:32	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2589.5867	6790.7117	0.9788	3082.1282	-	-	-	17778.0489		n_estimators=100	-	-
2024. 7. 18	23:58	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2613.9835	6863.9498	0.9783	3130.1482	-	-	-	18838.3808	-	n_estimators=100	-	-
2024. 7. 18	20:57	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2390.2657	6426.8927	0.9810	2787.2991	-	-	-	20174.0096	-	n_estimators=100	-	-
2024. 7. 18	16:48	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2404.8734	6469.6503	0.9808	2845.2877	-	-	-	-	-	n_estimators=100	-	-
2024. 7. 18	18:17	이재명	XGB	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	9269.2093	9758.9132	0.9562	5685.6165	-	-	-	-	-	n_estimators=100	-	-
2024. 7. 18		이재명	XGB	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	9269.2093	9758.9132	0.9562	5685.6165	-	-	-	-	-	n_estimators=100	-	-
2024. 7. 19	16:12	이재명	앙상블(RF, LGBM, XGB)	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2978.0940	6014.6606	0.9834	6014.6606	-	-	-	16985.3668	-	RF estimator=300	LGBM estimator=15000,	XGB estimator=3000
2027.7.19	07:15	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2586.4022	6780.4542	0.9789	3103.2865	-	-	-	-	-	n_estimators=100	-	-
2027.7.19	09:10	이재명	Random Forest Regressor	제공 데이터셋에서 특정 피처만 선택 사용 + 최단거리조등학교	895,057	223,765	9,272	2550.6491	6768.0069	0.9790	3070.2096	-	-	-	-	-	n_estimators=200	-	-

실험 날짜	실험 시간	작성자	모델	전처리	Train data 수	Valid data	Test data 수	train RMSE	valid RMSE	valid R-squared	test RMSE	Submission 결과	Submission Final	하이퍼파라미터 1	하이퍼파라미터 2	하이퍼파라미터 3	하이퍼파라미터 4
2024. 7. 11	23:10	장은지	Random Forest Regressor	baseline 코드의 전처리 내용	895,057	9,272			5159.1476	0.9794		46707.2744	-	n_estimators=100	criterion='squared_error'	random_state=1	n_jobs=-1
2024.7.17	0:00	장은지	Random Forest Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883		9,272	2253.7412	6209.4843	0.9824		17106.3368	-	n_estimators=100	-	random_seed=42	
2024.7.17	0:00	장은지	GradientBoosting Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	189.4510	8110.1716	0.9700	-	-	-	n_estimators=100	max_depth=30	random_seed=42	-
2024.7.17	0:00	장은지	GradientBoosting Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	1979.8459	5988.9642	0.9836	-	-	-	n_estimators=100	max_depth=15	random_seed=42	-
2024.7.17	0:00	장은지	AdaBoostr Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	90045.4503	90095.6641	-2.7059	-	-	-	n_estimators=100	-	random_seed=42	-
2024.7.17	0:00	장은지	AdaBoostr Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	90045.4503	90095.6641	-2.7059	-	-	-	n_estimators=100	-	random_seed=42	-
2024.7.17	0:00	장은지	LGBM Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	10237.9423	10551.2044	0.9492	-	-	-	n_estimators=100	-	random_seed=42	-
2024.7.17	0:00	장은지	LGBM Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	12573.0000	12886.0000	0.9242	-	-	-	n_estimators=50	-	random_seed=42	-
2024.7.17	0:00	장은지	XGB Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	8195.0186	8767.8543	0.9649	-	-	-	n_estimators=100	-	random_seed=42	-
2024.7.17	0:00	장은지	CatBoost Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	9464.0435	9826.0936	0.9559	-	-	-	n_estimators=100	-	random_seed=42	-
2024.7.17	0:00	장은지	CatBoost Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	-	-	-	-	-	-	-	-	-	-
2024.7.17	0:00	장은지	Random Forest Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	-	-	-	-	-	-	-	-	-	-
2024.7.18	0:00	장은지	Random Forest Regressor	제공 데이터 셋에서 특정 피처만 선택 사용 + 도로명, 세대수, 주	1,006,939/111883	-	9,272	2287.9534	6208.6108	0.9824	-	-	-	n_estimators=100	max_depth=30	random_seed=42	-

- 팀원들 각자 실험한 내용을 기록하고 공유함
- 기대 효과
  1. 서로 공유하며 중복되는 실험 방지
  2. 다른 팀원의 실험 내용과 본인의 실험 내용 비교하여 성능 개선
  3. 반복되는 실험에 따른 기억 망각 및 왜곡 대비

Model	Hyper parameter
Random Forest Regressor	n_estimators = 300
LightGBM	n_estimators = 15000
XGBoost	n_estimators = 3000

팀 내 best model

4. Modeling: 평가 지표

Test와 Submission의 RMSE 결과가 상이해서 성능 개선이 되었는지 확인이 어려움

➡ **다른 평가 지표를 함께 사용하여 보완**

함께 사용한 평가 지표

R-squared

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

- 회귀 모델의 성능을 평가하는 지표
- 모델이 데이터를 얼마나 잘 설명하는지를 나타냄
- R^2 값이 1에 가까울수록 모델이 데이터를 잘 설명함

MAE

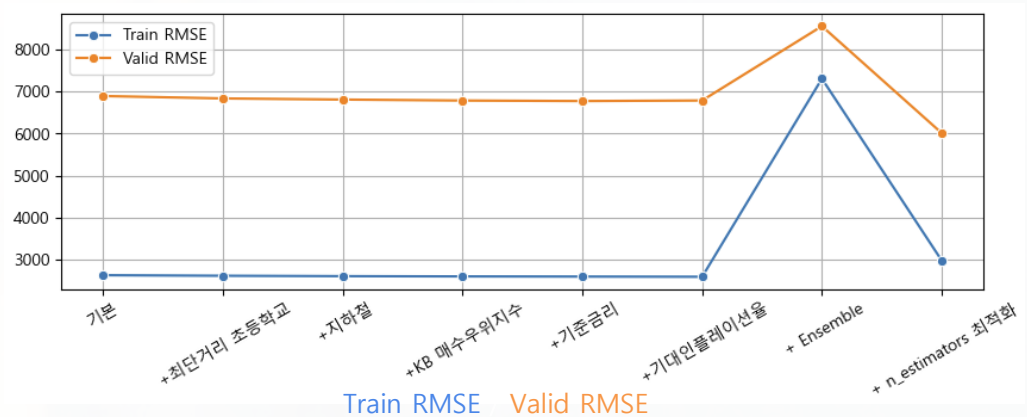
$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- 회귀 모델의 예측 정확도를 평가
- 실제 값과 예측 값 간의 절대 오차의 평균을 계산
- MAE 값이 작을수록 모델의 성능이 좋음

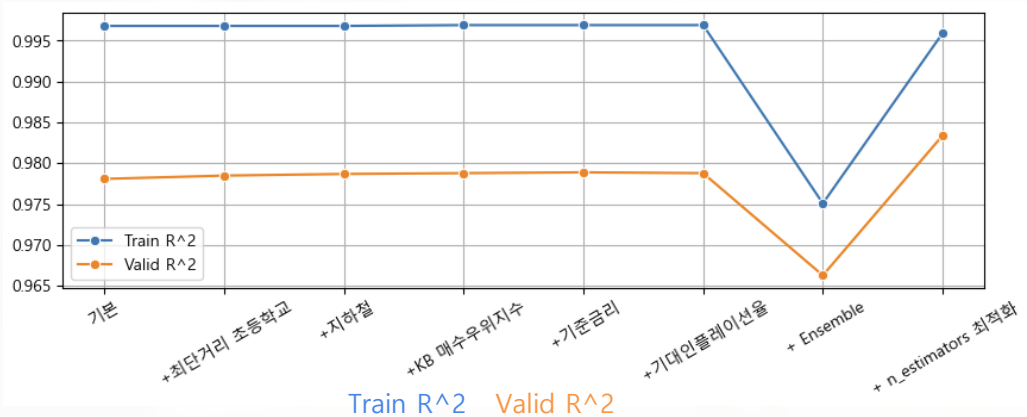
4. Modeling: 평가 지표 별 결과 비교(1)

모델별, 앙상블 평가지표 비교

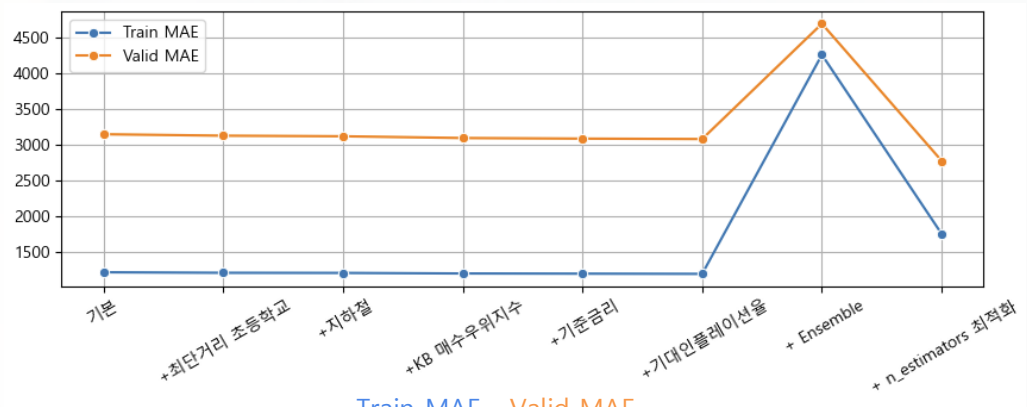
RMSE | R-squared | MAE | submission 비교



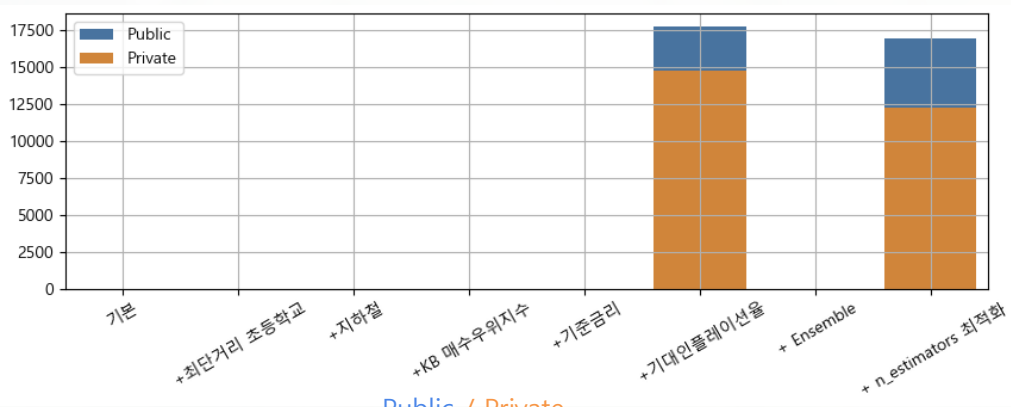
Train RMSE Valid RMSE



Train R^2 Valid R^2



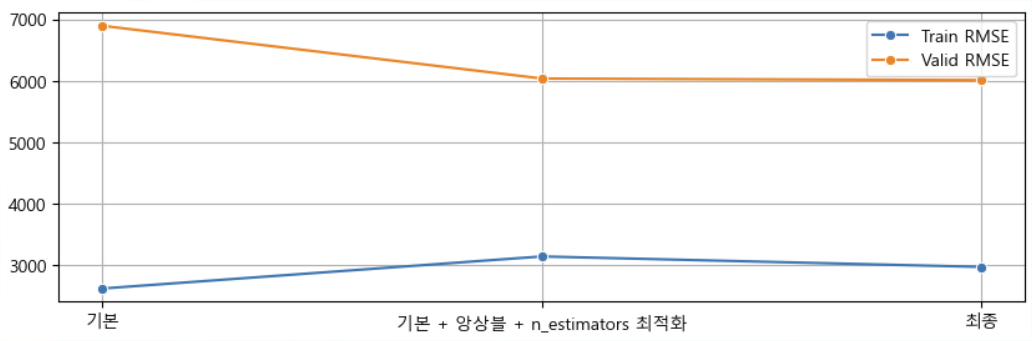
Train MAE Valid MAE



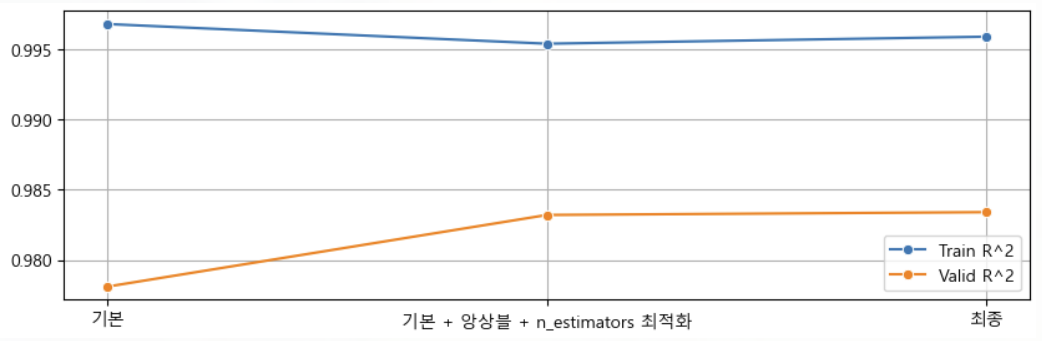
Public / Private

4. Modeling: 평가 지표 별 결과 비교(2)

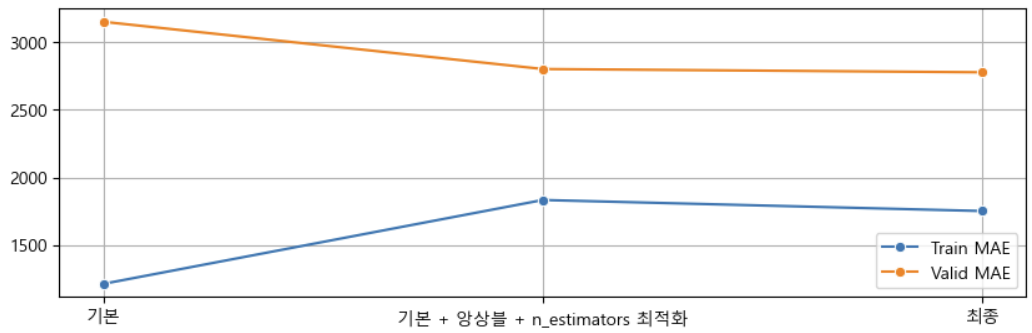
충격적인 결과



Train RMSE Valid RMSE



Train R^2 Valid R^2



Train MAE Valid MAE

4. Modeling: 평가 지표 별 결과 비교(2)

모델별, 앙상블 평가지표 비교

RMSE | R-squared | MAE | submission 비교

피쳐	모델	RMSE (Train / Valid)	R-squared (Train / Valid)	MAE (Train / Valid)	Submission (public / private)
기본 (구, 동, 번지, 전용면적, 계약년, 계약월, 계약일, 건축년도, 좌표X_2, 좌표Y_2)	Random Forest (n_estimators = 100)	2627.0079 / 6896.9111	0.9968 / 0.9781	1213.3631 / 3149.4610	
+ 최단거리 초등학교	Random Forest (n_estimators = 100)	2613.8844(-13.1235) / 6839.7912(-57.1199)	0.9968 / 0.9785(+0.0004)	1205.9546(-7.4085) / 3129.0467(-20.4143)	
+ 지하철	Random Forest (n_estimators = 100)	2604.5450(-9.3394) / 6812.7848(-27.0064)	0.9968(0) / 0.9787(+0.0002)	1204.0203(-1.9343) / 3120.9234(-8.1233)	
+ KB 매수우위지수	Random Forest (n_estimators = 100)	2598.4469(-6.0981) / 6789.7600(-23.0248)	0.9969(+0.0001) / 0.9788(+0.0001)	1196.4952(-7.5251) / 3096.3142(-24.6092)	
+ 기준금리	Random Forest (n_estimators = 100)	2594.1784(-4.2685) / 6777.5173(-12.2427)	0.9969(0) / 0.9789(+0.0001)	1193.6840(-2.8112) / 3087.8560(-8.4582)	
+ 기대인플레이션율	Random Forest (n_estimators = 100)	2589.5867(-4.5917) / 6790.7117(+13.1944)	0.9969(0) / 0.9788(-0.0001)	1190.8269(-2.8571) / 3082.1282(-5.7278)	17778.0489 / 14772.0250

제거했어야

4. Modeling: 평가 지표 별 결과 비교(2)

모델별, 앙상블 평가지표 비교

RMSE | R-squared | MAE | submission 비교

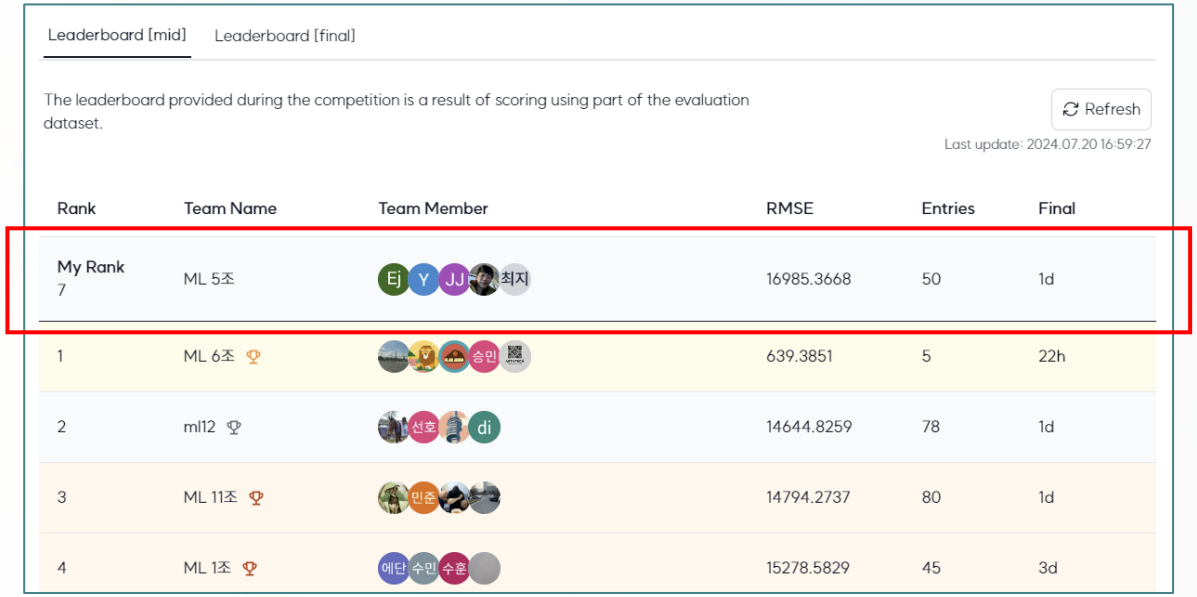
피쳐	모델	RMSE (Train / Valid)	R-squared (Train / Valid)	MAE (Train / Valid)	Submission (public / private)
변동 없음	Random Forest (n_estimators = 100)	2589.5867 / 6790.7117	0.9969 / 0.9788	1190.8269 / 3082.1282	17778.0489 / 14772.0250
	LightGBM (n_estimators = 100)	12032.4852 / 12224.9358	0.9327 / 0.9313	7361.7819 / 7367.3108	
	XGBoost (n_estimators = 100)	9656.7461 / 10096.6722	0.9566 / 0.9532	5733.0738 / 5801.5778	
	Ensemble (n_estimators = 100)	7310.3366(+4720.7499) / 8562.5464(+1771.8347)	0.9751(-0.0218) / 0.9663(-0.0125)	4264.3118(+3073.4849) / 4699.6542(+1617.526)	
변동 없음	Random Forest (n_estimators = 300)	2536.4700(-53.1167) / 6757.5607(-33.151)	0.9970(+0.0001) / 0.9790(+0.0002)	1174.9233(-15.9036) / 3067.1282(-15)	
	LightGBM (n_estimators = 15000)	3385.7394(-8646.7458) / 6190.5937(-6034.3421)	0.9947(+0.062) / 0.9824(+0.0511)	2083.7721(-5278.0098) / 2865.6689(-4501.6419)	
	XGBoost (n_estimators = 3000)	3991.3679(-5665.3782) / 6686.6565(-3410.0157)	0.9926(+0.036) / 0.9795(+0.0263)	2410.3231(-3322.7507) / 3133.3846(-2668.1932)	
	Ensemble	2978.0940(+388.5073) / 6014.6606(-776.0511)	0.9959(-0.001) / 0.9834(+0.0046)	1751.4562(+560.6293) / 2776.2914(-305.8368)	16985.3668(-792.6821) / 12281.7758(-2490.2492)


LITTERATURE RUSSE
LITTERATURE HISPANIQUE
LITTERATURE PORTUGAISE

5 결 과

5. 결과: 최종 순위 및 평가지표 결과

Leaderboard



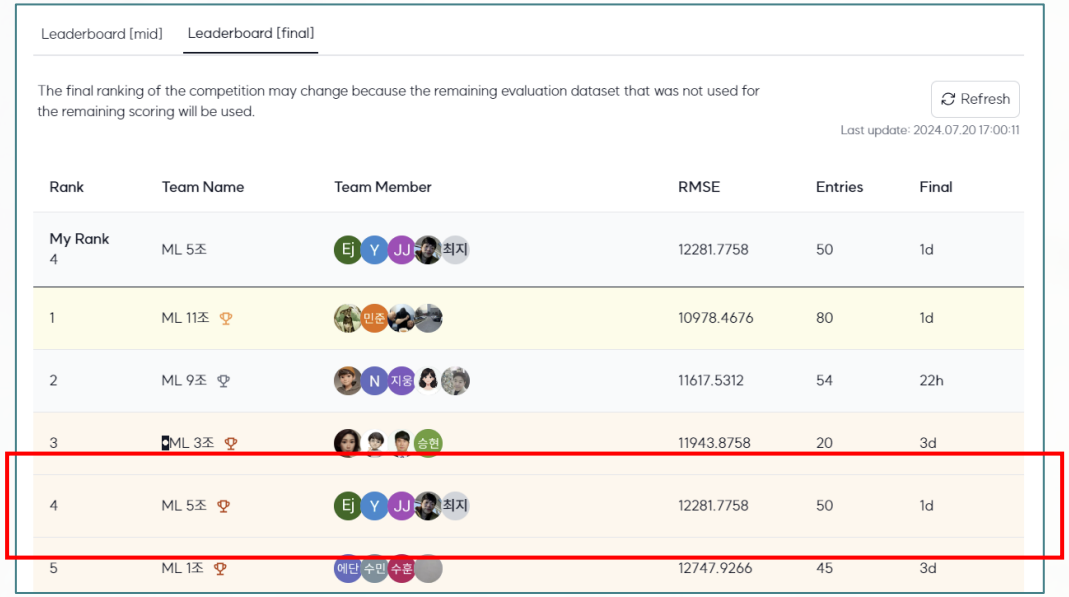
Leaderboard [mid] Leaderboard [final]

The leaderboard provided during the competition is a result of scoring using part of the evaluation dataset.

Refresh

Last update: 2024.07.20 16:59:27

Rank	Team Name	Team Member	RMSE	Entries	Final
My Rank 7	ML 5조	Ej Y JJ 최지	16985.3668	50	1d
1	ML 6조	승민	639.3851	5	22h
2	ml12	선호 di	14644.8259	78	1d
3	ML 11조	민준	14794.2737	80	1d
4	ML 1조	에단 수민 수훈	15278.5829	45	3d



Leaderboard [mid] Leaderboard [final]

The final ranking of the competition may change because the remaining evaluation dataset that was not used for the remaining scoring will be used.

Refresh

Last update: 2024.07.20 17:00:11

Rank	Team Name	Team Member	RMSE	Entries	Final
My Rank 4	ML 5조	Ej Y JJ 최지	12281.7758	50	1d
1	ML 11조	민준	10978.4676	80	1d
2	ML 9조	N 지홍	11617.5312	54	22h
3	ML 3조	승현	11943.8758	20	3d
4	ML 5조	Ej Y JJ 최지	12281.7758	50	1d
5	ML 1조	에단 수민 수훈	12747.9266	45	3d

Public 7위 / Private 4위


LITTERATURE RUSSE
LITTERATURE HISPANIQUE
LITTERATURE PORTUGAISE
LITT. JAPONAISE
ALLEMANDE

6 회 고

6. 회고: 대회 회고

성 능

- 앙상블로 점수 올라감
- But 파라미터 튜닝만으로도 성능을 많이 올릴 수 있음

데이터

- 데이터가 커질수록 관리가 어려움 → 툴을 사용하면 좋겠음
- 외부 데이터 쓸 때 Feature 생성 시간을 줄여야 함
- 기본 제공 데이터에 충실해야 함 → 기본 데이터는 황금과 같다

기 록

- 실험 기록 자동화 필요 (ex. 실험 종료마다 csv 파일에 업데이트 등)
- 점수에 가장 큰 변화를 줬던 포인트를 별도로 기록해서 누락이 없도록 해야 함

평가 지표

- RMSE, R-squared 가 개선되더라도 MAE 가 안 좋으면 Submission 점수가 안 좋아질 수 있음
- 지표 분석을 위한 작업이 필요 (ex. 각 데이터셋/지표 별 추세선 시각화)

병목 부분을 빠르게 개선하여 확보한 시간만큼 최대한 실험을 많이 해보자

6. 회고: 더 적용해 보고 싶은 것들

GPU 사용하는 cuML 을 프로젝트 초반에 셋팅하기

건물명 보다는 건물 형태, 전체세대수보다는 면적 별 세대수가 더 유의미할 것 같음

→ 전체세대수 남기고면적별 세대 수 삭제했을 때는 성능이 떨어짐

아파트명에 몇 동, 몇 호까지 포함된 데이터들도 존재함

→ 이런 부분까지 세밀하게 모두 전처리 해보고 싶음

아파트명을 통일성 있게 전처리 하여, 유니크한 값을 줄여보고 싶음

예측 결과가 음수 값이거나 오차가 큰 일부 값에 대해 아파트별 평균값, 근처 평균값으로 후보정

강의에서 나온 꿀팁 적용하기

팀원분들의 많은 피처를 적용하기

6. 회고: Tip

지우지 마세요!

- Baseline 코드에 학습 시 n_jobs = -1 이 있었는데, 이걸 지워서 학습 시간이 약 5배 더 걸림

나눠 보세요!

- Feature Engineering할 때 chunk로 나눠서 병렬처리 하면 속도 향상됨

관리하세요!

- 메모리 관리를 안 하면 소중한 데이터들이 사라질 수 있음

백업하세요!

- git commit/push 주기적으로 꼭 해야함

복습하세요!

- 강의와 베이스라인 코드를 꼼꼼히 보자. 놓치는 부분만 없어도 성능 향상을 이룰 수 있음

6. 회고: 스터디 & 대회 진행 소감

김기홍

- 좋은 성품의 팀원분들과 함께 할 수 있어서 온전히 학습에 집중할 수 있었음
- 팀원 분들과 함께 나누었던 아이디어와 인사이트를 통해, 성능 개선을 이루어 낼 수 있었음

이윤재

- 팀원들이 서로 보완, 개선해 나가며 프로젝트에 몰입, 좋은 결과를 낼 수 있었음
- 밤을 새며 정신없이 길을 잃는 것보다 충분한 논의를 통해 방향성을 잡는 것이 훨씬 중요하다고 깨달았음

이재명

- 서로 모르는 것을 알려줄 수 있어서 학습 및 프로젝트 진행에 도움을 많이 받을 수 있었음
- 좌절의 순간에 의지가 많이 되었음

장은지

- 모르는 부분을 질문하고, 서로 공유하면서 많이 배울 수 있었음
- 생각했던 것보다 성능개선이 안 되어서 힘들 때 팀원분들이 공유해주신 경험을 참고할 수 있었음
- 실험 결과 기록이 중요함. 기록을 확인하면서 놓쳤던 부분도 되짚어볼 수 있었음

최지미

- 팀원 간 선행 학습 레벨이 크게 차이 나지 않고, 모두의 참여도가 높아서 자연스럽게 협업이 가능했던 것 같음
- 각자의 강점이 서로의 보완점을 채울 수 있었던 좋은 경험이었음

6. 회고: 참고 자료

외부 데이터

- 서울 열린데이터광장 https://data.seoul.go.kr
- 부동산 빅데이터 플랫폼 https://www.bigdata-realestate.kr
- 공공데이터포털 https://www.data.go.kr
- KB부동산 데이터허브 https://data.kbland.kr
- KOSIS 국가통계포털 https://kosis.kr

논 문

- 주정민, 강선미, 최지웅, 한영우, 기계학습을 이용한 아파트 매매가격 예측 연구 : 한국 아파트의 내·외적 데이터 수집과 가격 예측 중심으로(2020.11) https://manuscriptlink-society-file.s3-ap-northeast-1.amazonaws.com/kips/conference/2020fall/presentation/KIPS_C2020B0235.pdf

대 회

- 데이콘 아파트 실거래가 예측 AI 경진대회 https://dacon.io/competitions/official/21265

PPT 디자인

- 사진 https://www.pexels.com/ko-kr/
- color https://coolors.co/palettes/trending

Q & A

경청해 주셔서 감사합니다.

Thank you for listening