트리모델을 활용한 서울내 주택 가격 예측

지역적, 주택 특성 분석을 중심으로

20190552손지영, 20201365최세환

요약

서울내의 주택 가격은 지난 2019년부터 급격한 증가세를 보이며, 현재 2022년까지 3년간 계속하여 크게 증가해왔다. 주택 가격의 예측은 주거지 혹은 자산으로서 주택을 구하고자 하는 개인들 외에도 주택공급 시장의 안정화와 도심 인프라 조성을 책임지는 정부차원에서도 그 필요성이 매우 높다. 이번 연구에서는 서울시에서 제공하는 서울시 부동산 실거래가 정보를 중심으로 아파트의 복도유형, 난방방식 등의 아파트별 특징과 아파트의 소재지와 같은 지역적 요인을 추가하여 머신러닝의 예측기법 중 트리 모델들을 활용하여 서울 내 주택 가격 예측 모델을 다수 작성하고 모델 간의 예측치와 실행시간의 비교를 통해 아파트 가격을 98.25%의 정확도로 예측할 수 있는 'XGBoost' 알고리즘을 사용한 모델을 최종적 예측 모델로서 제시하였다. 이를 통하여 서울의 지역적 특성과 아파트의 구조적 특징들을 이용한 가격 예측을 제공하여 추후 부동산 시장 정책 선정 등에 있어 서울 내 아파트 가격 분석에 있어 예측 지표로서 활용할 수 있을 것이다.

서론

서울내 주택의 가격은 지난 2019년에서 2020년 2.67%, 2020년에서 2021년 6.47%[1]로 큰 폭으로 상승하였다. 주택은 인간의 필수적인 3가지 요소 중 하나로서 주택의 가격은 가계소비[2]와 합계출산율[3] 등 우리 사회의 경제, 인구 분야와 같은 전반적인 사회현상과의 강한 상관관계를 가진다. 주택가격의 강한 상관관계성은 정부차원에서도 주택의 가격을 예측하여 적절한 위치와 시기에 주택의 공급을 제어하고자 노력하고 있다.[4]

---

[1] 한국부동산원「전국주택가격동향조사」

[2] 주택가격이 가계소비에 미치는 영향, 전수민, 권선희, 2019

[3] 주택가격과 출산의 시기와 수준: 우리나라 16개 시도의 실증분석, 김민영, 황진영, 보건사회연구, Vol.36, No.1, pp.118-142

[4] 오를까, 내릴까…정부 '안정' 확신에도 전문가들 "글쎄", 서울파이낸스, 노제욱, 2022.01.01

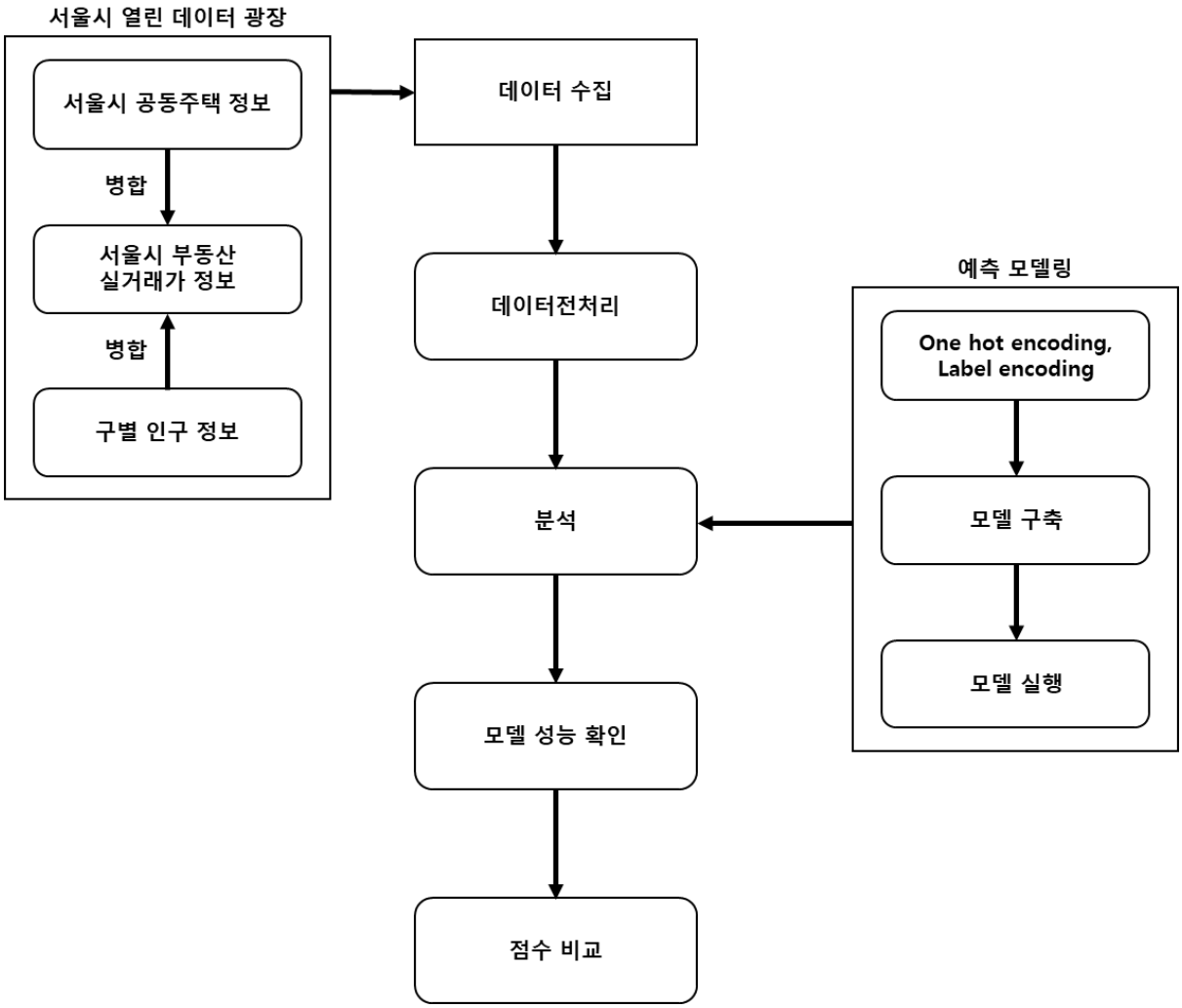
주택공급과정에 있어 행정동, 구와 같은 지역적인 요인들과 난방 방식, 건물용도, 복도 방식 등 아파트의 개별적인 특성을 통한 정확한 가격의 예측이 이루어진다면, 개인의 주택구매 과정과 정부의 주택공급 입지선정에 있어 도움이 될 수 있을 것으로 기대한다,

이에 따라 본연구는 서울내 아파트의 개별 특성과 지역적 요인의 데이터를 이용하여 서울 내 아파트 가격을 예측하도록 한다.

선행연구 검토

남상현, 한태호, 김이주, 이은지(2020)에서는 선형회귀분석을 이용하여 전국의 부동산 가격데이터를 이용하여 부동산 가격의 평균치를 예측하였으며, 원두환, 김형건(2008)은 아파트의 난방방식에 따른 전세가격변화가 존재함을 헤도닉모델을 사용하여 확인하였다. 박운선, 임병준(2011)은 헤도닉 가격모형을 통한 가정과 회귀분석을 통하여 지역에 따라 아파트의 가격요인으로 작용하는 변수에 차이가 존재함을 확인하였다.

연구설계 및 프로세스



데이터 전처리

데이터 수집

서울시 열린 데이터 광장을 이용하여 서울시에서 제공하는 "서울시 부동산 실거래가"를 통해 서울내 부동산 가격데이터를 확보하였으며, "서울시 공동주택 아파트 정보" 데이터를 이용하여, 아파트명과 법정동, 아파트 단지의 경도, 위도 좌표, 연면적, 건물 별 난방방식, 건물유형, 복도방식 등의 정보를 추가하였다. 또한 면적당 인구비 대비 건물의 면적 점유율을 구하기 위하여 서울시에서 제공하는 "서울시 인구밀도 (구별)"를 사용하였다.

각 데이터의 구성은 다음과 같다.

서울시 부동산 실거래가

```
0   접수연도              128580 non-null  int64
1   자치구코드             128580 non-null  int64
2   자치구명              128580 non-null  object
3   법정동코드             128580 non-null  int64
4   법정동명              128580 non-null  object
5   지번구분              118932 non-null  float64
6   지번구분명             118932 non-null  object
7   본번                118932 non-null  object
8   부번                118932 non-null  float64
9   건물명               118940 non-null  object
10  계약일               128580 non-null  int64
11  물건금액(만원)          128580 non-null  int64
12  건물면적(㎡)           128580 non-null  float64
13  토지면적(㎡)           68140 non-null   float64
14  층                 118940 non-null  float64
15  권리구분               0 non-null       object
16  취소일                0 non-null       float64
17  건축년도              128382 non-null  float64
18  건물용도              128580 non-null  object
19  신고구분              0 non-null       string
20  신고한 개업공인중개사 시군구명  0 non-null      object
```

서울시 인구밀도 (구별)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   구별      25 non-null     object
 1   구별인구    25 non-null     int64
 2   구별면적    25 non-null     float64
 3   구별인구밀도  25 non-null     int64
dtypes: float64(1), int64(2), object(1)
memory usage: 928.0+ bytes
```

서울시 공동주택 아파트 정보

```
 0   번호                          2637 non-null   int64
 1   k-아파트코드                    2637 non-null   object
 2   k-아파트명                     2637 non-null   object
 3   k-단지분류(아파트,주상복합등등)       2577 non-null   object
 4   kapt도로명주소                  2461 non-null   object
 5   주소(시도)k-apt주소split        2637 non-null   object
 6   주소(시군구)                    2637 non-null   object
 7   주소(읍면동)                    2637 non-null   object
 8   나머지주소                      2113 non-null   object
 9   주소(도로명)                    2481 non-null   object
 10  주소(도로상세주소)                2474 non-null   object
 11  k-전화번호                     2630 non-null   object
 12  k-팩스번호                     2601 non-null   object
 13  단지소개기존clob                 566 non-null    float64
 14  단지첨부파일                     184 non-null    object
 15  k-세대타입(분양형태)               2614 non-null   object
 16  k-관리방식                     2624 non-null   object
 17  k-복도유형                     2621 non-null   object
 18  k-난방방식                     2630 non-null   object
 19  k-전체동수                     2610 non-null   float64
 20  k-전체세대수                    2633 non-null   float64
 21  k-건설사(시공사)                 2592 non-null   object
 22  k-시행사                      2579 non-null   object
 23  k-사용검사일-사용승인일              2622 non-null   object
 24  k-연면적                      2636 non-null   float64
 25  k-주거전용면적                   2607 non-null   float64
 26  k-관리비부과면적                  2635 non-null   float64
 27  k-전용면적별세대현황(60㎡이하)        2604 non-null   float64
 28  k-전용면적별세대현황(60㎡~85㎡이하)   2604 non-null   float64
 29  k-85㎡~135㎡이하               2604 non-null   float64
 30  k-135㎡초과                   3 non-null      float64
 31  k-홈페이지                     773 non-null    object
 32  k-등록일자                     356 non-null    object
 33  k-수정일자                     2611 non-null   object
 34  고용보험관리번호                    2090 non-null   object
 35  경비비관리형태                    2610 non-null   object
 36  세대전기계약방법                    2489 non-null   object
 37  청소비관리형태                    2613 non-null   object
 38  건축면적                      2624 non-null   float64
 39  주차대수                      2632 non-null   float64
 40  기타/의무/임대/임의=1/2/3/4        2637 non-null   object
 41  단지승인일                     2637 non-null   object
 42  사용허가여부                     2637 non-null   object
 43  관리비 업로드                    2637 non-null   object
 44  좌표X                        2627 non-null   float64
 45  좌표Y                        2627 non-null   float64
 46  단지신청일                      2637 non-null   object
```

데이터 변수 선정

서울내 부동산 실거래가 데이터와 서울시 공동주택 아파트 정보 데이터를 '아파트명'을 기준으로 병합하여 아파트에 대한 실거래가 정보를 취합하였다.

이후, 아파트의 팩스번호, 전화번호, 홈페이지와 같은 분석의 목적과 관련이 없는 변수들을 제거하고, 결측치의 비율이 50%이상인 변수들을 제거하였으며, 분양권과 같은 실물거래가 아닌 데이터와 거래취소일이 존재하는 실제 거래가 일어나지 않은 데이터 또한 삭제하였다. 이후 완전히 중복되는 데이터에 대하여 삭제를 진행하였다.

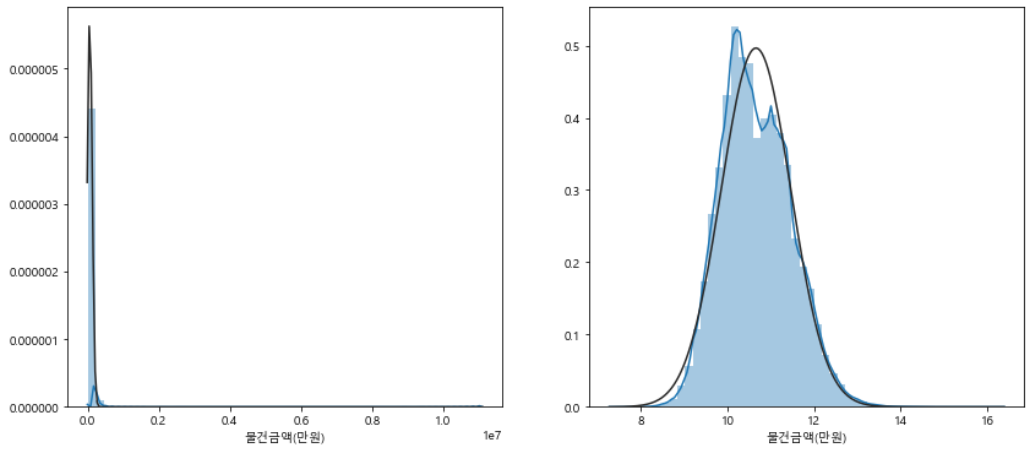
서울시 부동산 실거래가 정보의 경우 2018년도부터 2022년 까지의 데이터가 존재하나, 2020~2022년의 데이터의 경우 코로나로 인한 양적완화와 금리인하로 인하여 같은 년도 내에서도 집값이 급격하게 변화하여 인플레이션의 영향이 비교적 적은 2019년도를 대상으로 가격 분석을 진행하였다.

이에 따라 도출된 데이터의 목록은 다음과 같다.

```
1   자치구코드                    12092 non-null  int64
2   자치구명                     12092 non-null  object
3   법정동코드                    12092 non-null  int64
4   법정동명                     12092 non-null  object
5   지번구분명                    12092 non-null  object
6   건물명                      12092 non-null  object
7   물건금액(만원)                 12092 non-null  int64
8   건물면적(㎡)                  12092 non-null  float64
9   층                         12092 non-null  float64
10  건축년도                     12090 non-null  float64
11  건물용도                     12092 non-null  object
12  번호                       12092 non-null  int64
13  k-아파트코드                   12092 non-null  object
14  k-아파트명                    12092 non-null  object
15  k-단지분류(아파트,주상복합등등)      11991 non-null  object
16  kapt도로명주소                 11666 non-null  object
17  주소(시도)k-apt주소split       12092 non-null  object
18  주소(시군구)                  12092 non-null  object
19  주소(읍면동)                  12092 non-null  object
20  나머지주소                    10626 non-null  object
21  주소(도로명)                  11685 non-null  object
22  주소(도로상세주소)               11685 non-null  object
23  k-세대타입(분양형태)              12011 non-null  object
24  k-관리방식                   12075 non-null  object
25  k-복도유형                   12075 non-null  object
26  k-난방방식                   12083 non-null  object
27  k-전체동수                   12002 non-null  float64
28  k-전체세대수                  12092 non-null  float64
29  k-건설사(시공사)                11949 non-null  object
30  k-시행사                    11947 non-null  object
31  k-사용검사일-사용승인일             12075 non-null  object
32  k-연면적                    12092 non-null  float64
33  k-주거전용면적                  12073 non-null  float64
34  k-관리비부과면적                 12084 non-null  float64
35  k-전용면적별세대현황(60㎡이하)       12073 non-null  float64
36  k-전용면적별세대현황(60㎡~85㎡이하)  12073 non-null  float64
37  k-85㎡~135㎡이하              12073 non-null  float64
38  고용보험관리번호                   9322 non-null   object
39  경비비관리형태                   12081 non-null  object
40  세대전기계약방법                  11591 non-null  object
41  청소비관리형태                   12081 non-null  object
42  건축면적                     12092 non-null  float64
43  주차대수                     12056 non-null  float64
44  기타/의무/임대/임의=1/2/3/4      12092 non-null  object
45  단지승인일                    12092 non-null  object
46  사용허가여부                   12092 non-null  object
47  관리비 업로드                  12092 non-null  object
48  좌표X                      12090 non-null  float64
49  좌표Y                      12090 non-null  float64
50  단지신청일                    12092 non-null  object
```

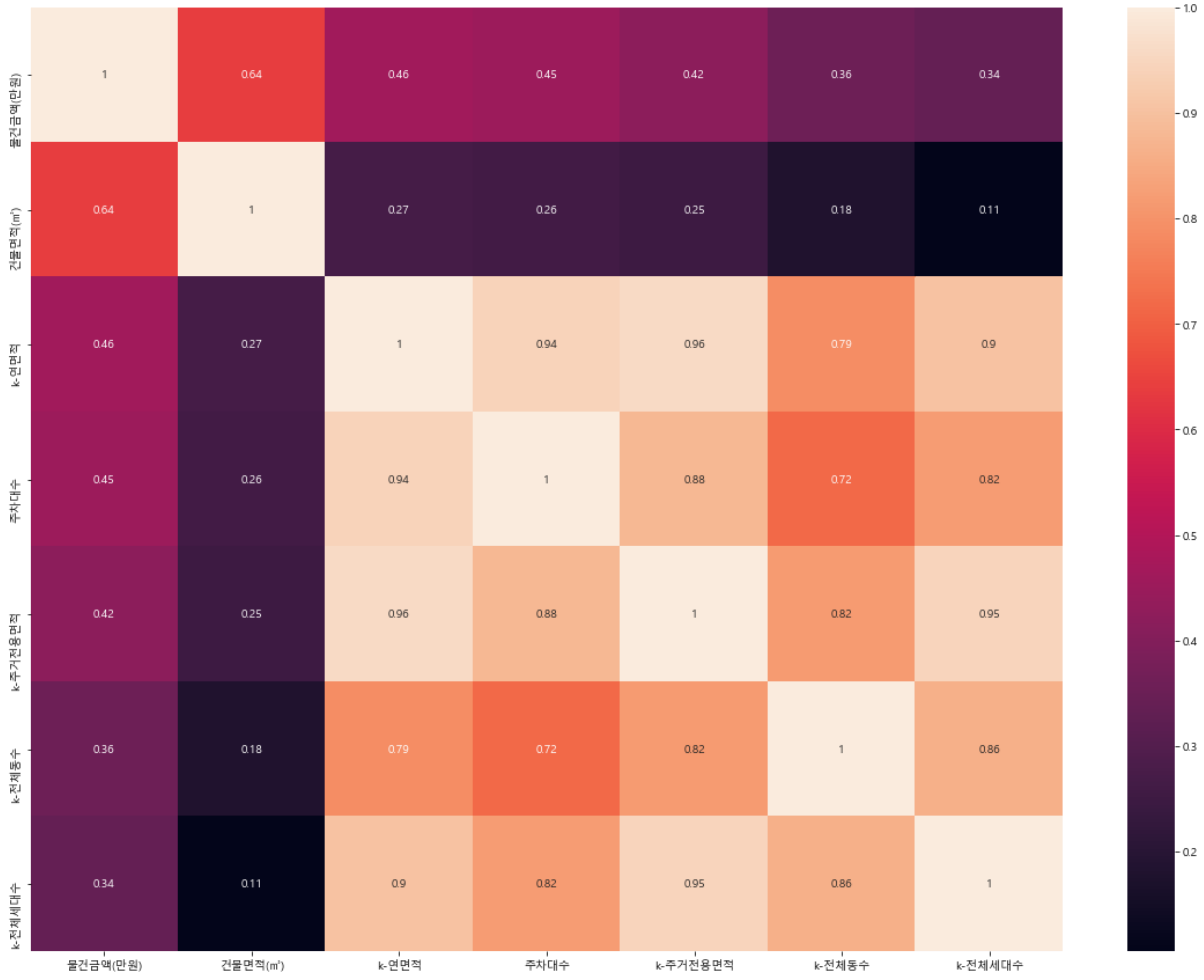
이후 타겟변수인 물건금액의 분포를 확인하고자 히스토그램을 그리고, 왜도를 줄이기 위하여 로그변환을 실시하였다.

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x16e081bfc48>
```

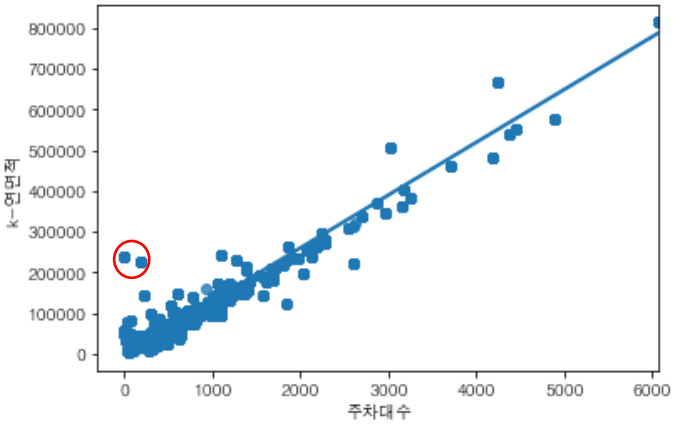


수치형 데이터 처리

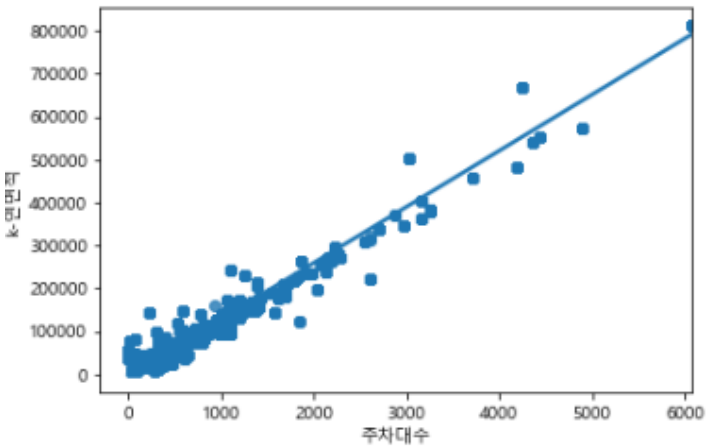
앞서 도출된 데이터 중 범주형 데이터인 아파트명 자치구명 등과 중복될 수 있는 요소인 아파트 코드, 자치구 코드, 경도, 위도 등의 데이터를 삭제하고 수치변수인 "물건금액(만원)","건물면적(㎡)", "k-연면적", "주차대수", "k-주거전용면적", "k-전체동수", "k-전체세대수", "자치구코드", "건축년도", "법정동코드"에 대한 전처리를 진행하고자 해당 변수들과 물건금액의 히트맵을 제작하여 보다 자세히 상관관계를 파악하였다.



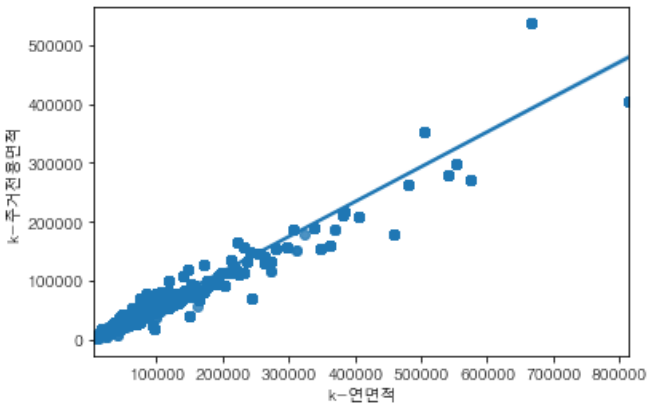
이중 물건금액과 상관계수가 0.4이상인 변수들이 건물면적을 제외하고 연면적과의 상관계수가 0.8을 모두 넘으므로 연면적이 다른 변수들을 모두 대표할 수 있는지를 확인한다.



연면적이 건물의 실면적과 토지면적의 합계라는 것을 토대로 주차대수가 0에 가까운 데이터를 이상치라고 판단, 삭제한다.



이상치를 제외한 residual plot에서 주차대수는 연면적이 대표할 수 있다고 판단, 요인에서 제외한다.



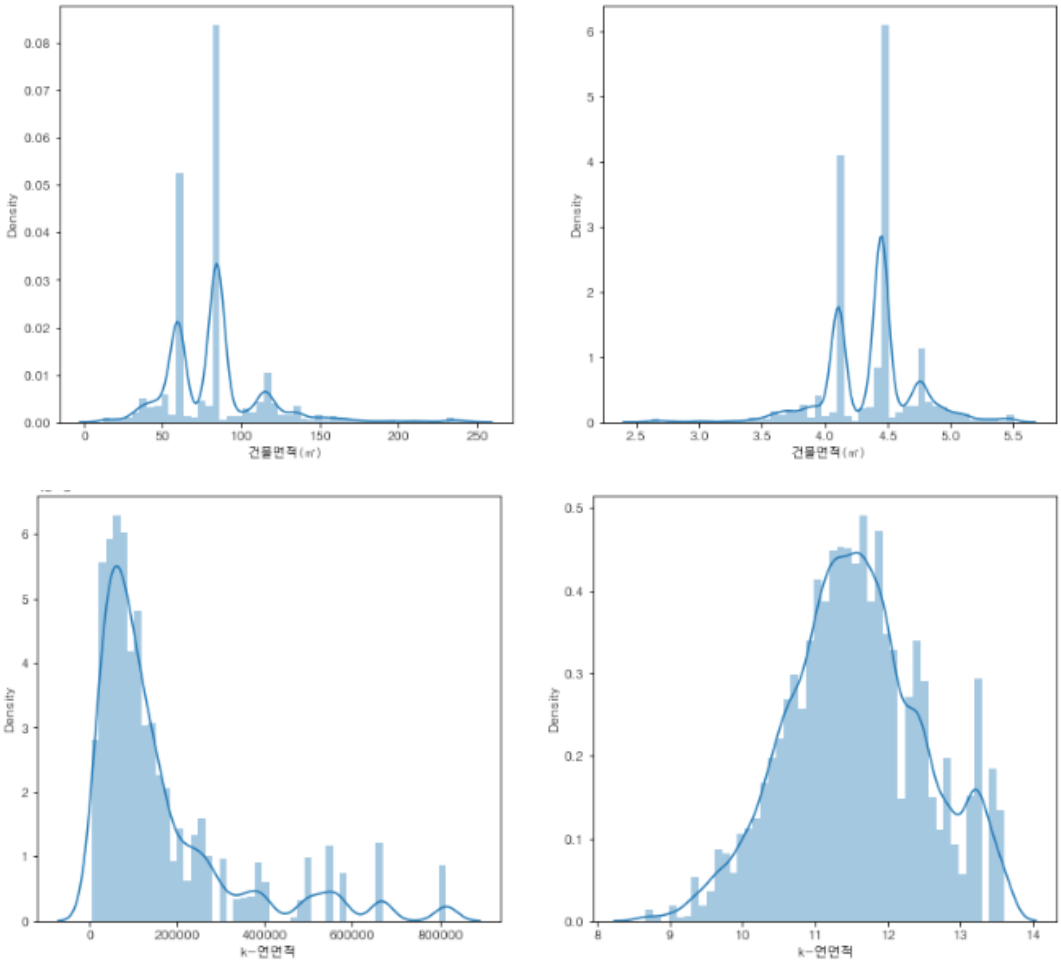
주거전용면적 또한 residual plot의 결과와 히트맵에서의 상관관계를 고려하였을 때 연면적 변수의 특성이 주거전용면적을 대체할 수 있다고 판단하여 분석변수에서 제거한다.

위와 같은 과정을 통해 수치형 데이터에서 모델링에 사용되는 변수는 "k-연면적", "건물면적(m^2)", "k-전체세대수"이다.

이에 따라 수치형 변수에서 남겨진 변수들은 표와 같다.

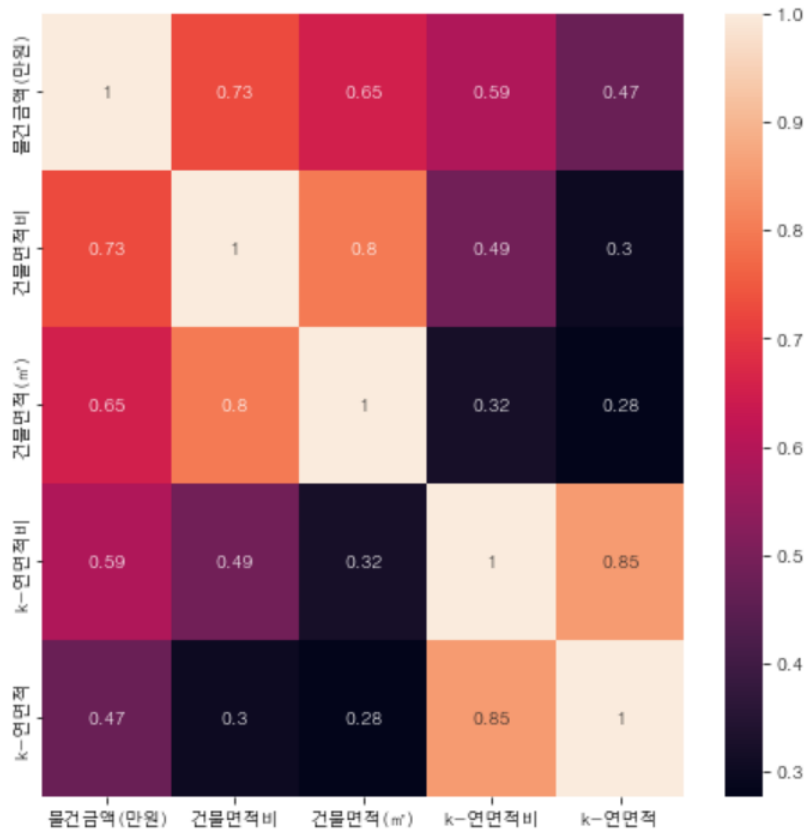
k-연면적	건물면적(m^2)

추가적으로 두 변수에 대한 왜도를 줄이기 위해 로그 변환을 수행하였다.



인구밀도 데이터를 활용한 변수 변환

인구밀도의 증가는 지역내 주택의 가격을 증가시킨다. (O'Sullivan, 2007) 따라서 '서울 열린 데이터 광장'에서 '구별 인구 수', '면적', '인구 밀도' 데이터를 수집하여 기존의 데이터와 구별로 매칭하였다. 그 후 '건물면적'과 'k-연면적' 데이터에 각각의 데이터를 나누면서 타겟 변수인 '물건금액'과의 상관정도를 히트맵을 통해 확인하였다. 그 결과 기존의 데이터에서 '구별 인구밀도'로 나눠주었을 때 상관정도가 가장 높아지는 것을 확인할 수 있었다. 따라서 아래의 그림은 기존의 데이터와 '인구밀도'로 나눈 변수의 비교만을 따로 시각화한 자료이다.



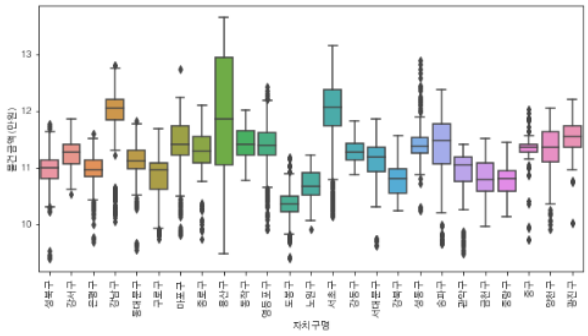
그림의 상관계수의 차이를 살펴보면 건물면적은 0.08이 증가, k-연면적비는 0.12가 증가한 것을 알 수 있다. 따라서 기존의 변수를 '구별 인구밀도'로 나눈 값을 대체하여 분석에 사용하였다.

범주형 데이터 처리

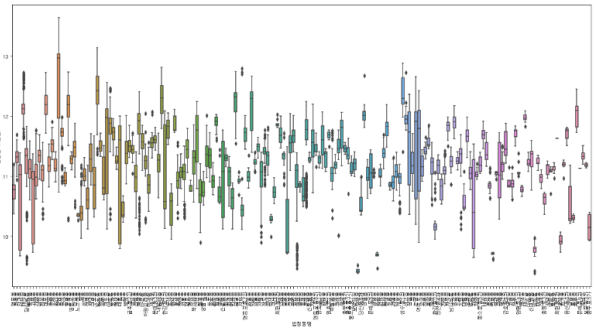
다음은 결측값이 50% 이상 넘는 데이터와 아파트 가격과 관련 없는 범주형 변수들을 1차 제거하고 남은 변수들이다.

자치구명	건축년도	경비비관리형태	k-단지분류
k-복도유형	건물용도	세대전기계약방법	청소비관리형태
k-세대타입	k-세대타입	층	

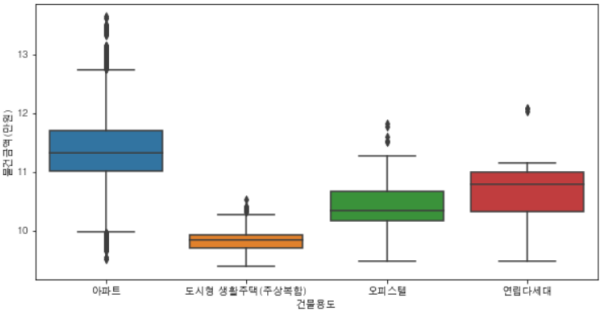
이러한 변수들을 boxplot으로 시각화하였고 같은 의미를 갖거나 각 범주에 따라 평균 차이가 나지 않는 변수들은 제거하고 사용할 변수들을 2차 선별하였다. 아래는 대표적으로 평균차이가 나는 범주형 데이터이다.



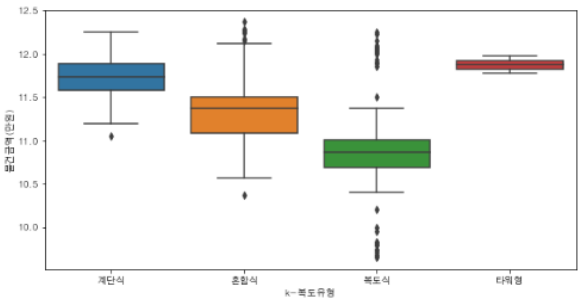
<자치구명: 전체 데이터>



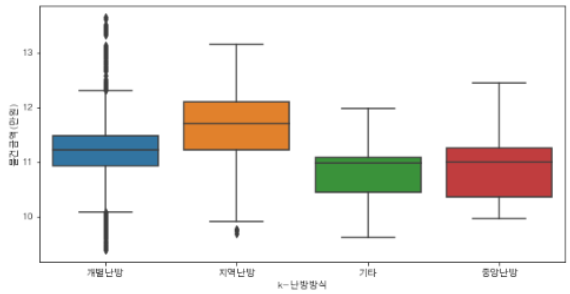
<법정동명: 전체 데이터>



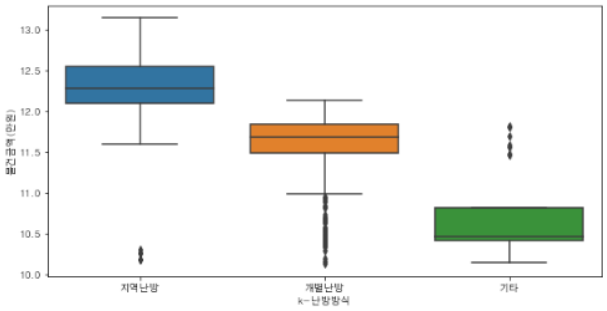
<건물용도: 전체 데이터>



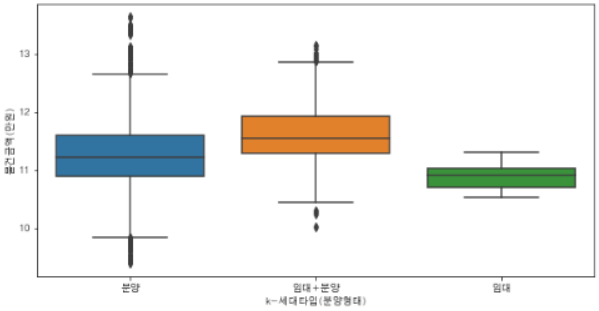
<k-복도유형: 송파구 데이터>



<k-난방방식: 전체 데이터>



<k-난방방식: 서초구 데이터>



<k-세대타입: 전체 데이터>

전체 데이터에서와 구별 데이터를 비교 분석하고 평균차이를 확인하였으며, 이에 따라 최종적으로 모델에 사용되는 범주형 데이터를 선별하였다.

최종 데이터 요약

타겟 변수는 '물건금액(만원)'이며 사용한 변수들은 아래와 같다

변수명	변수 설명	수치형/ 범주형	비고
K-연면적비	건축물의 각 층의 바닥면적합을 구별 인구밀도로 나눈 값	수치형	아파트명이 동일할 시 값이 동일함
건물면적비	건축물의 수직투영한 바닥면적을 구별 인구밀도로 나눈 값	수치형	아파트명이 동일하여도 세대별 특성을 가짐
자치구명	주택이 속한 서울의 구명	범주형	아파트명이 동일할 시 값이 동일함
건물용도	오피스텔, 아파트와 같은 주택의 분류	범주형	아파트명이 동일할 시 값이 동일함
건축년도	아파트가 지어진 해	범주형	년도의 특성상 수치형이 아닌 범주형으로 분류 아파트명이 동일할 시 값이 동일함
K-세대타입	분양 임대와 같은 주택의 양도방식	범주형	아파트명이 동일할 시 값이 동일함
K-난방방식	개별난방, 중앙난방과 같은 주택의 난방작동 방식	범주형	아파트명이 동일할 시 값이 동일함
법정동명	주택이 속한 서울의 동명	범주형	아파트명이 동일할 시 값이 동일함
K-복도유형	계단, 복도, 타워와 같은 주택내 세대가 배치 되어있는 방식	범주형	아파트명이 동일할 시 값이 동일함
층	건물의 층	범주형	년도의 특성상 수치형이 아닌 범주형으로 분류 아파트명이 동일하여도 세대별 특성을 가짐
세대 전기계약방법	단일, 종합과 같이 주택 내 전기공급 계약방법	범주형	아파트명이 동일하여도 세대별 특성을 가짐

모델링

사용하고자 하는 알고리즘

1. K-NN

K-NN 알고리즘은 단순하여 다른 알고리즘에 비해 구현하기가 쉽다. 하지만 모델을 생성하지 않기 때문에 특징과 클래스 간 관계를 이해하는데 제한적이라는 단점이 있다.

2. Decision Tree

decision tree도 간단한 모델로 비교적 빠른 속도로 예측이 가능하다. 또한 if~then 규칙을 사용하여 이해하기 쉽다는 장점이 있다. 하지만 연속형 변수값을 예측할 때 적당하지 않다는 단점이 있다.

3. RandomForest

randomforest는 decision tree 모델 여러 개를 훈련시켜서 그 결과를 종합해 예측하는 앙상블 알고리즘이다. 따라서 일반화 및 성능이 우수하다는 장점을 갖고 있다. 또한 이상치와 결측치에 강건하여 높은 정확도를 나타낸다. 개별 트리 분석이 어렵고 트리 분리가 복잡해지는 경향이 존재한다. 또한 차원이 크고 희소한 데이터에서는 성능이 미흡니다.

4. ExtraTrees

extratrees는 rodomforest와 동일한 원리를 이용하지만 split을 할 때 무작위로 feature을 선정한다. 따라서 랜덤포레스트보다 훈련속도가 빠르고 성능도 미세하게 높다.

5. XGBoost

xgboost는 병렬 처리로 학습, 분류 속도가 빠르다. 또한 xgboost 자체에 과적합 규제 기능으로 강한 내구성을 지닌다. 하지만 작은 데이터에 대해 과적합 가능성이 있다.

6. CatBoost

catboost는 ordered target encoding 방식으로 범주형 데이터를 효과적으로 처리한다. 하지만 데이터 대부분이 수치형 변수이 경우 학습 속도가 느리다는 단점이 있다.

모델링 결과

원핫 인코딩 방식과 레이블 인코딩 방식을 사용하여 두가지 방법으로 모델링을 했을 때 결과를 비교하고자 한다. 분석하기 전 'standardscaler', 'minmaxscaler', 'robustscaler'을 이용하여 간단한 모델링을 돌려본 결과 'standardscaler'을 사용했을 때 성능이 가장 좋았기 때문에 이 방법을 사용하여 스케일링을 하였다.

1. 원핫 인코딩 변한 후 모델링

먼저 범주형 데이터는 원핫 인코딩 방식으로 처리해주었다. 범주형 데이터 중 '자치구명'과 '법정동명'을 모두 인코딩하게 될 경우 타겟 변수를 제외한 변수가 총 222개가 된다. 수행한 알고리즘은 'k-nn', 'decisiontree', 'randomforest', 'extratrees', 'xgboost', 'catboost' 알고리즘으로 총 6가지 모델을 수행하였다.

```
KNeighborsRegressor
        Training time: 0.021s
        Prediction time: 0.860s
        Total time: 0.881s
        MAE: 7049.487290643645
        RMSE: 11798.247704383497
        R2 score: 0.9691059085204098
```

```
DecisionTreeRegressor(random_state=42)
        Training time: 0.479s
        Prediction time: 0.013s
        Total time: 0.492s
        MAE: 7047.171530689977
        RMSE: 12476.538685255751
        R2 score: 0.9717408821209225
```

```
RandomForestRegressor
        Training time: 13.737s
        Prediction time: 0.104s
        Total time: 13.841s
        MAE: 5972.066017254148
        RMSE: 10374.27261362516
        R2 score: 0.9821516562859197
```

```
ExtraTreesRegressor
        Training time: 17.538s
        Prediction time: 0.112s
        Total time: 17.650s
        MAE: 6083.7366610593635
        RMSE: 10175.285896178028
        R2 score: 0.9819654642185772
```

```
XGBRegressor
        Training time: 4.355s
        Prediction time: 0.016s
        Total time: 4.370s
        MAE: 6179.3876865931925
        RMSE: 9962.829380351592
        R2 score: 0.9815123058869758
```

```
CatBoostRegressor
        Training time: 8.823s
        Prediction time: 0.015s
        Total time: 8.838s
        MAE: 6124.355144268549
        RMSE: 9869.169546117368
        R2 score: 0.9823561924950127
```

수행 결과 MAE값이 가장 낮은 모델은 Randomforest, RMSE값이 가장 낮으면서 R2 score가 가장 높은 모델은 'Catboost'로 구할 수 있었다. 또한 모델의 성능 지표는 아니지만 모델을 돌리는데 소요된 시간을 비교해봤을 때, 비교적 가벼운 모델인 'K-NN'과 'Decisiontree'를 사용한 모델이 다른 모델에 비해 수행시간이 짧다는 것을 알 수 있었다.

하지만 위와 같은 경우 222개의 너무 많은 변수를 갖는다. 따라서 원핫인코딩이 아닌 레이블인코딩을 통해 최대한 적은 수의 열로 문제를 해결하였으며, 그 결과는 다음과 같다..

2. 레이블 인코딩 변환 후 모델링

트리 계열의 알고리즘은 레이블 인코딩을 하였을 때 숫자의 순서나 중요도의 영향을 받지 않는다. 따라서 모든 범주형 변수들에 레이블 인코딩을 수행하였다. 'k-nn'알고리즘은 트리계열의 모델은 아니지만 선형 계열과의 모델 성능 비교를 위해 위의 모델들을 동일하게 사용하여 비교하였다.

```
KNeighborsRegressor
        Training time: 0.061s
        Prediction time: 0.195s
        Total time: 0.256s
        MAE: 8516.844416716749
        RMSE: 15336.538490798104
        R2 score: 0.9557081830845495
```

```
DecisionTreeRegressor
        Training time: 0.036s
        Prediction time: 0.016s
        Total time: 0.052s
        MAE: 7481.490773252005
        RMSE: 13458.858832290798
        R2 score: 0.9680318654195414
```

```
RandomForestRegressor
        Training time: 3.365s
        Prediction time: 0.149s
        Total time: 3.514s
        MAE: 6476.412435788055
        RMSE: 10870.44990633121
        R2 score: 0.9779374251236124
```

```
ExtraTreesRegressor
        Training time: 3.350s
        Prediction time: 0.096s
        Total time: 3.446s
        MAE: 6546.813190181937
        RMSE: 11013.428513001445
        R2 score: 0.9798707460520708
```

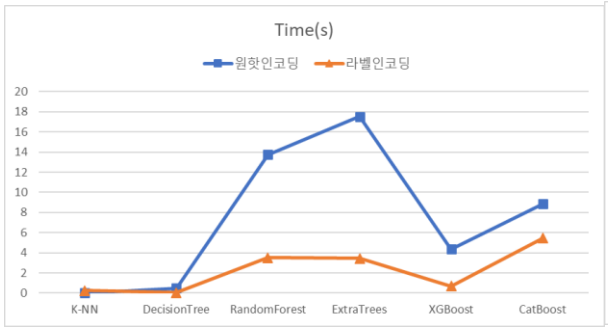
```
XGBRegressor
        Training time: 0.680s
        Prediction time: 0.012s
        Total time: 0.692s
        MAE: 6211.649528544072
        RMSE: 10154.196569148598
        R2 score: 0.9824818887713793
```

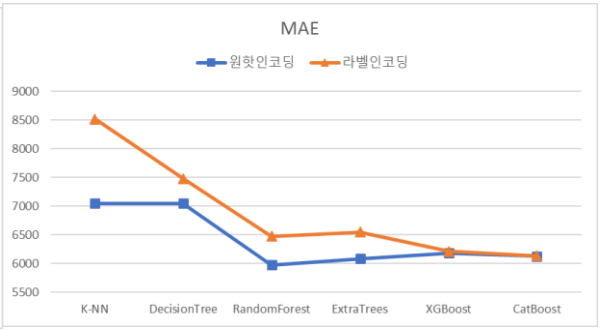
```
CatBoostRegressor
        Training time: 5.461s
        Prediction time: 0.006s
        Total time: 5.467s
        MAE: 6134.224240733634
        RMSE: 9981.688005249121
        R2 score: 0.982748737578792
```

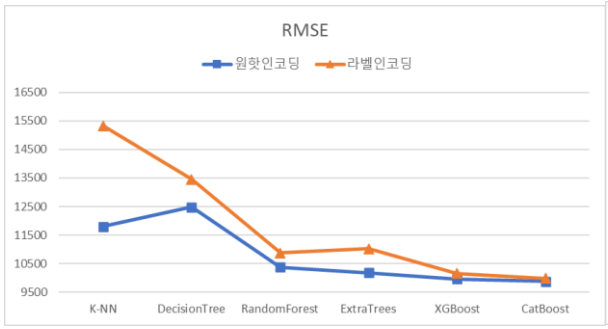
수행 결과 MAE값과 RMSE값이 가장 낮으면서 R2 score이 가장 높은 'Catboost'을 사용한 모델의 성능이 가장 좋다는 것을 확인할 수 있었다. 하지만 높은 성능을 가진만큼 다른 모델에 비해 수행시간이 더 길다는 것도 알 수 있다.

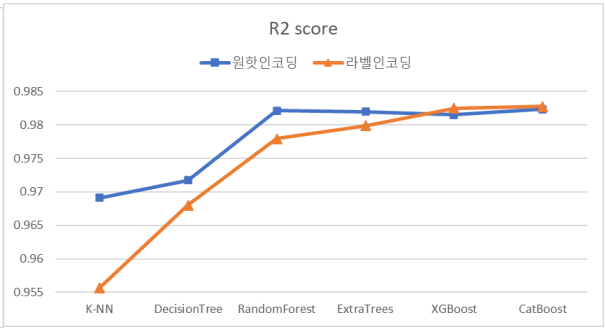
결과 비교 분석

사용한 인코딩의 종류에 따라, 각각의 평가지표에 따라 성능이 좋은 모델들이 다르기 때문에 종합적으로 성능을 비교하고 판단하여 최종 모델을 선택하고자 한다.









먼저 모델의 성능을 비교해봤을 때 원핫 인코딩을 한 모델들이 레이블 인코딩을 한 모델들에 비해 평균적으로 MAE값과 RMSE값은 낮고 R2 score값은 높은 것을 확인할 수 있다. 하지만 Time을 비교해보면 간단한 모델을 제외하고는 원핫 인코딩을 한 모델의 수행시간이 훨씬 길게 걸린다는 것을 알 수 있다. 이는 위에서 언급했듯이 원핫 인코딩을 하는 과정에서 열이 222개까지 늘어났기 때문에 11개의 변수만을 가진 레이블 인코딩 모델들에 비해 오래 걸린 것으로 추측된다.

만약 모델의 성능만을 고려한다면 원핫 인코딩을 사용한 모델을 가장 적합한 모델로 판단였을 것이다. 하지만 현실문제를 해결함에 있어서 데이터의 개수가 늘어나고 고려해야 되는 법정동 개수가 늘어 변수의 개수 역시 더 많아지게 된다면 수행시간은 레이블 인코딩 모델에 비해 현재보다 훨씬 클 것 판단하였다. 따라서 원핫 인코딩을 한 모델과 성능이 가장 비슷하거나 더 좋은 'XGBoost'와 'Catboost'이 가장 적합한 모델이라고 판단하였다. 본 데이터에서 성능적으로 Catboost가 미세하게 우수하지만 추가적으로 뒤에 제시한 한계점을 극복하여 여러 수치형 변수가 추가된다면 오히려 XGBoost가 더 적합할 것이라고 판단하였다. 따라서 'Catboost'에 비해 수행시간이 짧으며 수치형 변수 처리에 더 효과적인 'XGBoost'를 가장 적합한 모델이라 판단하였다.

한계점

위에서 제시한 집값에 대한 자치구별 평균을 비교해보면 구별로 차이가 많이 나는 것을 확인할 수 있다. 하지만 본 데이터에서 구별로 데이터의 개수를 살펴보면 최대 1,205개부터 최소 75개까지 존재한다. 이는 구별로 데이터가 불균형 하다는 의미이며, 모델링을 진행하는 과정에서 구별로 충분히 학습되지 못했음을 알 수 있다. 따라서 만약 오버 샘플링이나 언더 샘플링을 통해 데이터의 불균형을 해소하고 모델링을 진행한다면 보다 좋은 결과를 얻을 수 있을 것으로 예상된다.

본 데이터에는 지역의 특징을 나타내는 '자치구명'과 '법정동명'이 존재한다. 구별로 인구밀도를 결합한 것과 같이 동별로도 결합을 하고 싶었으나 공공 데이터는 '행정동명'을 기준으로 인구밀도를 제공하기 때문에 '법정동명' 변수와 병합할 수 없었다. 따라서 법정동명과 행정동명을 맵핑하기 위해 여러 데이터를 구하였으나 '법정동명'이 연도별로 기준이 바뀌는 경우가 많아 결국 구별 인구밀도만 병합하게 되었다. 만약 동별 인구밀도로 면적비를 나누어 변수를 구한다면 보다 지역별 세부적인 특징을 고려한 변수를 얻을 수 있을 것으로 예상된다.

또한 데이터의 X, Y좌표를 이용하여 학교와의 거리, 지하철 역과의 거리, 공원과의 거리 등 주택의 환경적 요인을 고려하려 했으나, 김화환, 박성필, 송예나(2017)의 연구에서는 광역시에서의 도시철도 접근성에 대한 회귀분석을 진행하여 광주광역시를 제외한 모든 광역시에서 지하철과 같은 도시철도의 접근성이 주택의 가격과 양의상관관계를 가짐을 보였던 것과 다르게 본 연구에서는 물건가격과 교통 인프라, 환경적요인과의 상관관계가 유의미하게 나타나지 못하였다. 이는 박운선, 임병준(2011)의 연구에서 분석된 바와 같이 지역별, 가격대별로 주거환경의 가격형성에서 영향을 주는 요인이 다르게 작용하기 때문인 것으로 예상된다. 따라서 지역내 주택정보에 대한 세부적인 요인을 고려하여 각 지역의 유형별 분류를 이루어 낸 후 가격예측을 진행한다면, 본 연구보다도 세부적이고 정확한 분석이 이루어질 수 있을 것으로 예상된다.

동시에 예측모델의 오버피팅을 방지하고 예측모델의 성능을 끌어올리고자 gridsearchCV를 이용하여 하이퍼파라미터의 분석을 진행하였으나, 가장 예측율이 높은 Xgboost와 Catboost에 대한 하이퍼파라미터 분석이 하드웨어적 한계로 인하여 이루어지지 못하였다. 두 분석방법을 제외하고 각 분석에서 동시의 사용된 Randomforest의 경우 하이퍼파라미터를 통해 분석된 결과가 기존 예측 값과 큰 차이점이 존재하지 않았다[5]. 다만 오류 값의 경우 유의미하다고 볼 수 있을 정도의 감소가 이루어졌으므로, 추후 Xgboost와 Catboost에 대한 분석이 이루어진다면, 보다 정확한 예측 모델링이 가능할 것으로 예상된다.

[5] 표1, 표2 참고

<표1: 원핫 인코딩>

```
Fitting 3 folds for each of 168 candidates, totalling 504 fits

GridSearchCV(cv=3, estimator=RandomForestRegressor(random_state=42), n_jobs=-1,
             param_grid={'max_depth': [32, 33, 34, 35, 36, 37, 38],
                         'min_samples_leaf': [1, 2],
                         'min_samples_split': [4, 5, 6],
                         'n_estimators': [120, 130, 135, 140]},
             return_train_score=True, scoring='r2', verbose=1)

best_score: 0.9765275973353637
best_parameters: {'max_depth': 36, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 130}


RandomForestRegressor(max_depth=36, min_samples_split=5, n_estimators=130,
                      random_state=42)

        Training time: 8.318s
        Prediction time: 0.064s
        Total time: 8.382s
        MAE: 5879.104493572526
        RMSE: 10141.602928876297
        R2 score: 0.9829833498006499
```

<표2: 레이블 인코딩>

```
Fitting 3 folds for each of 168 candidates, totalling 504 fits

GridSearchCV(cv=3, estimator=RandomForestRegressor(random_state=42), n_jobs=-1,
             param_grid={'max_depth': [32, 33, 34, 35, 36, 37, 38],
                         'min_samples_leaf': [1, 2],
                         'min_samples_split': [4, 5, 6],
                         'n_estimators': [120, 130, 135, 140]},
             return_train_score=True, scoring='r2', verbose=1)

best_score: 0.9751120316508096
best_parameters: {'max_depth': 32, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 120}


RandomForestRegressor(max_depth=32, min_samples_split=4, n_estimators=120,
                      random_state=42)

        Training time: 1.801s
        Prediction time: 0.043s
        Total time: 1.844s
        MAE: 6421.8218164967
        RMSE: 10809.50943329871
        R2 score: 0.9779939831906359
```

결론

본연구에서는 서울내 주택가격이 주택단지의 연면적과 건물면적과 같은 면적들, 주택의 위치, 복도유형과 같은 주택의 건축적 특성과 난방방식, 전기계약 방법 등의 내부적 특성이 양의 상관관계를 가짐을 확인하였다. 또한 '자치구', '법정동'과 '구별 인구밀도'와 같은 지역적 특성의 상관관계를 확인할 수 있었다. 따라서 주택의 특성과 지역적 특성을 복합적으로 고려하여 예측 모델링을 구축하였다는 점에서 의의가 있다.

이러한 변수들을 통한 가격 예측에 있어서 최대한 적은 변수만을 사용하기 위해 레이블 인코딩 변환 후 추가로 모델링을 진행하였다. 그 결과 원핫 인코딩을 사용한 모델과 성능은 비슷하면서 보다 짧은 수행시간을 갖는 'XGBoost' 알고리즘을 사용한 최종 모델을 얻을 수 있었다.

다만 환경적요인에 대한 분석이 지역적 특성 고려가 부족하여 이루어지지 못하였던 점, 하드웨어적 한계로 인하여 하이퍼파라미터를 통한 모델의 보정이 이루어지지 않은 점에 있어 후속연구를 통하여 보완하여야 할 부분이 많다.

참고문헌

주택가격이 가계소비에 미치는 영향, 전수민, 권선희, 2019

한국부동산원「전국주택가격동향조사

주택가격과 출산의 시기와 수준: 우리나라 16개 시도의 실증분석, 김민영, 황진영, 보건사회연구, Vol.36, No.1, pp.118-142

오를까, 내릴까…정부 '안정' 확신에도 전문가들 "글쎄", 서울파이낸스, 노제욱, 2022.01.01

머신러닝 기법을 통한 대한민국 부동산 가격 변동 예측 남상현, 한태호, 김이주, 이은지, 2020

난방 방식에 따른 아파트 가격 변화 분석 원두환, 김형건, 2008

지역 및 가격대별 아파트가격결정요인의 차이 분석, 박운선, 임병준, 2011

도시환경이 주거용 토지가격에 미치는 영향에 관한 연구, 노태욱, 강창덕, 2009

Urban economics, New work, Mc-Graw-Hill, O'Sullivan, 2007