

미친 집값... 어떤 숨겨진 원인이 있을까?

2016147588 김신희



(인스타 그램 집값 크롤링 결과 워드클라우드.png)

*크롤링 코드 확인 희망 시 *selenium, beautifulsoup* 설치 후 *crwaling/인스타크롤링.ipynb* 확인(5000 개 크롤링이므로 약 4~5 시간 소요)

추가된 데이터 (출처 마지막 페이지에 명시됨):

- 서울시 동별 주차장 개수 및 면적
- 서울시 동별 평균 연령
- 서울시 세대원수별 세대수
- 서울시 동별 인구밀도
- 서울시 동별 주택종류별 통계
- 서울시 동별 의료기관 통계

삭제된 데이터

대규모점포, 병원, 의원, 관광숙박업, 유흥주점 인허가 데이터
서울시 버스정류장 데이터

환경설정 (주의사항 있음):

- 개발 환경:
 - Jupyter Notebook
- 사용 라이브러리:
 - Pandas (pip install pandas)
 - Numpy (pip install numpy)
 - Re
 - BeautifulSoup (pip install beautifulsoup4)
 - Matplotlib (pip install matplotlib)
 - Datetime
 - Seaborn (pip install seaborn)
 - Plotly, Plotly.express, Plotly.graph_objects (pip install plotly)
 - Missingno (pip install missingno)
 - Warnings (pip install --upgrade warnings)
 - Json
 - Scipy (python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose)
 - Sklearn (pip install scikit-learn)
 - Folium (pip install folium)
 - LightGBM (pip install lightgbm)
 - Shap (pip install shap)
 - Geopandas
 - Pyproj

*주의 **Geopandas** 와 **Pyproj** 는 설치가 매우 까다롭습니다. 설치 후에도 개인 PC 에 맞춰 경로를 지정해줘야 제대로 동작하는 경우도 있으므로 올려 드린 링크를 참고하시거나 추가로 geopandas 와 Pyproj 설치 Document 혹은 블로그들을 참고하셔야 합니다(설치 시 document 에 있는 requirements 에 맞게 설치해야 합니다! 또한 설치 시 오류가 발생하면 uninstall 후 아래 첨부한 내용을 거친 후 jupyter 를 재시작 해야 합니다)

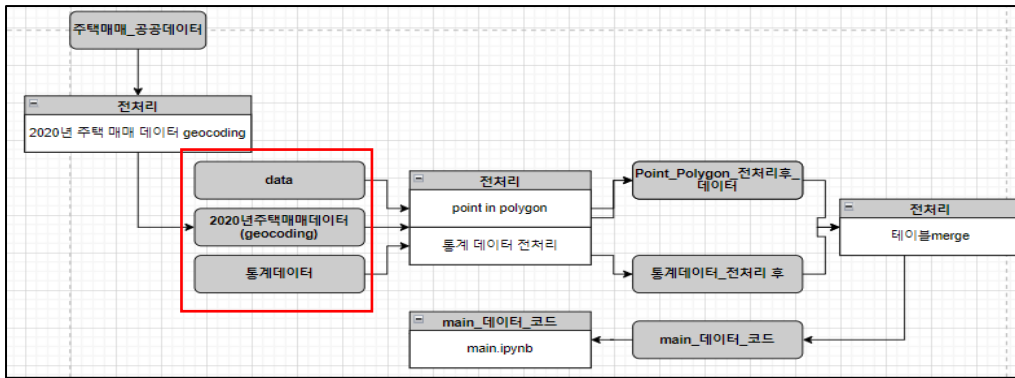
참고 블로그: 1. <https://blog.daum.net/geoscience/1659>

2. <https://pypi.org/project/Shapely/>

****geopandas & shapely 설치 오류 발생 시**** (geopandas 설치 오류 발생 시 python 버전을 낮춰야 합니다. 또한 python 3.6 이면 cp36 을 다운)

1. <https://blog.naver.com/PostView.nhn?blogId=kokoyou7620&logNo=222175644751>
2. <https://byeon-sg.tistory.com/entry/geopandas-%EC%84%A4%EC%B9%98-%EC%97%90%EB%9F%AC-python-setuppy-egginfo-Check-the-logs-for-full-command-output-failed-with-initial-frozen-solve-Retrying-with-flexible-solve>

Code Structure:



* 회색 블록들은 directory 를 의미

* 흰색 블록들은 ipynb 파일 의미

- 전처리 파일: ① “2020 년 주택 매매 데이터 geocoding”, ② “point in polygon”, ③ “통계 데이터 전처리”, ④ “테이블 merge”
- 메인 데이터 분석 파일: ⑤ main.ipynb
- **실행 방법:** zip 파일 압축 해제 후 “전처리”와 “main_데이터_코드” 폴더 안에 있는 jupyter 파일들을 아래에 기재한 순서대로 실행

**주의: 디렉토리 구조는 절대 바뀌면 안됩니다.*

- **실행 순서: 위에 기재한 ①②③④⑤ 순서대로 실행.**

1. 가장 왼쪽 상단에 위치한 “주택매매_공공데이터” 안에 있는 주택매매 raw data 를 “전처리”폴더 안에 있는 “2020 년 주택매매데이터 geocoding.ipynb” 전처리를 거친 후 “2020 년 주택매매데이터(geocoding)”폴더 안에 저장한다.
2. “data”, “통계데이터” directory 에는 주택 매매데이터 외의 raw_data 들이 들어있다
3. 빨간색 테두리 안에 있는 데이터들을 “전처리” 폴더에 있는 전처리 code 들을 거친 후 “Point_polygon_전처리후_데이터”, “2020 년주택매매데이터(geocoding)” 및 “통계데이터_전처리 후” 폴더에 저장한다
4. 전처리를 거친 데이터들을 “전처리” 폴더에 있는 “테이블 merge.ipynb”을 통해 각각 행정구, 행정동, 분기를 기준으로 merge 한 후 “main_데이터_코드”폴더에 “main_data.csv”로 하나의 테이블로 저장된다.
5. 마지막으로 “main_데이터_코드”폴더의 “main_data.csv”를 read 하여 main.ipynb 파일에서 data analysis 을 시작한다.

Preprocessing(EDA):

배경 설명:

해당 프로젝트에서 추구하는 목표는 어떤 요인들이 “주택거래가”에 영향을 미치고 어떤 영향을 주느냐를 찾기 위함이다. 그러므로 본인은 주택매매 데이터들을 “행정동”별로 groupby 하고 행정동별로 통계된 데이터를 수집하여 분석했다.

1. 2020 년 주택 매매데이터 geocoding.ipynb

- 수집한 데이터들을 모두 “동별”로 groupby 해주어야 한다, 해당 데이터를 확인해보면 “시군구” 컬럼에 “동” 관련 내용이 있지만 어떤 데이터는 “행정동”으로 표시되어 있고 어떤 데이터들은 “법정동”으로 표시되어 있다.
- 따라서 “법정동”으로 표시된 데이터들만 따로 추출하여 “행정동”으로 매핑하는 것이 기술적으로 어렵기 때문에 우선 “시군구”와 “도로명” 컬럼을 연결하여 “검색키워드 1” 컬럼을 만들었고 “시군구”와 “단지명”컬럼을 연결하여 “검색키워드 2” 컬럼을 생성했다.
- 1 차적으로 “검색키워드 1”컬럼을 KakaoMap API 의 입력으로 넣어 해당 주소의 좌표데이터를 얻어 “lon”, “lat” 컬럼에 넣어주었다, 1 차 검색으로 좌표검색이 안된 주소들은 2 차적으로 “검색키워드 2” 컬럼을 KakaoMap API 의 입력으로 주어 좌표를 얻은 후 최종적으로 “2020 년주택매매데이터(geocoding)”폴더에 저장하였다.

*주의: KakaoMap API 를 사용하기 위해

자신의 api key 를 등록해야 사용할 수

있다. 코드에 오른쪽과 같이 주석으로

명시해 두었다. 해당 ipynb 파일은 약

40~50 분의 실행 시간이 소요된다. (우선 제

API 넣었으니 사용하시면 됩니다. 다만

하루에 최대 10~30 만번 사용제한이 있어

cannot found 'documents'라는 예러가 뜨면 다른 api 키를 사용하시거나 다음 날에 실행시켜야 합니다.

```
headers = {  
    #아래 줄 ----- 안에 자신의 카카오맵 API key 입력  
    "Authorization": "-----"  
}  
n_th_api = 0  
def KakaoMap_get_lat_lon(address):  
    url = 'https://dapi.kakao.com/v2/local/search/keyword.json?query={}'.format(address)  
    places = requests.get(url, headers = headers).json()['documents']
```

2. Point_in_polygon.ipynb

- “2020 년 주택매매데이터 geocoding.ipynb”와 같이 주택매매데이터 외 geocoding 이 필요한 다른 데이터들에게도 좌표 컬럼 “lon”, “lat”을 추가해준다.
- “data”폴더 안에 있는 “Seoul_geo.geojson”를 읽어 각 행정동 별 Multi-Polygon 데이터를 정의한다.
- 주택매매데이터와 다른 데이터 “행정동”컬럼이 없는 데이터들 모두 “lon”, “lat”컬럼을 입력으로 주어 “행정동”의 length 만큼 for 문을 돌아 point.within(gdf.geometry[i].buffer(0))가 True 이면 “행정동”을 string 으로 return 해준다. Return 받은 데이터들을 “행정동” 컬럼으로 받은 후 “Point_polygon_전처리후_데이터” 폴더에 저장한다.

```
서울주택매매20년도_geocoding = pd.read_csv('../2020년주택매매데이터(geocoding)/2020년주택매매데이터.csv').drop('Unnamed: 0',axis=1)  
서울주택매매20년도_geocoding['lon'] = 서울주택매매20년도_geocoding['lon'].astype(float)  
서울주택매매20년도_geocoding['lat'] = 서울주택매매20년도_geocoding['lat'].astype(float)  
서울주택매매20년도_geocoding['geometry'] = 서울주택매매20년도_geocoding.apply(lambda row : Point([row['lon'], row['lat']]), axis=1)  
서울주택매매20년도_geocoding = gpd.GeoDataFrame(서울주택매매20년도_geocoding, geometry='geometry')  
서울주택매매20년도_geocoding.crs = {'init':'epsg:4326'}  
서울주택매매20년도_geocoding = 서울주택매매20년도_geocoding.to_crs({'init':'epsg:5179'})  
  
서울주택매매20년도_geocoding['행정동'] = 서울주택매매20년도_geocoding['geometry'].apply(return_office_dong)
```

```
#주의!!-----  
#C:/Users/Anaconda3/Library/share  
os.environ['PROJ_LIB']='C:/Users/Anaconda3/Library/share'  
#주의!!-----  
  
gdf.crs = {'init':'epsg:4326'}  
gdf = gdf.to_crs({'init':'epsg:5179'})  
  
#소속된 경계구역을 반환한다.  
def return_office_dong(point):  
    dong_df = pd.DataFrame()  
    dong_df['dong'] = gdf['adm_nm'].str.split().str[2]  
    dong_list = dong_df.dong.tolist()  
    for i,dong in enumerate(dong_list):  
        if point.within(gdf.geometry[i].buffer(0)) == True:  
            print(str(dong_list[i]))  
            return str(dong_list[i])  
    else:  
        return None  
# Point(126.989574, 37.569006).within(gdf.geometry[0].buffer(0))
```

*주의: 해당 파트에서는 **Geopandas, shapely** 라이브러리를 사용한다. 해당 라이브러리들은 설치가 까다롭고 또한 특정 PC 들에서는 왼쪽 그림과 같이 경로를 직접 설정해줘야 동작하기 때문에 주의해서 사용해야 한다.

*추가로 shapely 의 Point 메서드를 사용하기 위해서는 해당 라이브러리에서 지원하는 좌표계로 바꿔줘야 하므로{'init': 'epsg:4326'} 부분과 같은 작업을 해줘야 한다.

* 해당 코드는 apply 와 for 문으로 돌아가기 때문에 **전체 실행시간이 약 40~50 분 소요된다.**

3. 통계데이터_전처리.ipynb

- Geocoding 이 필요 없는 통계 데이터들을 전체적으로 전처리하는 파트이다.
- 대표적으로 아래와 같은 전처리 기법들을 사용하였다.

```
외국인데이터['년도'] = 외국인데이터['시점'].str.split(' ').str[0].astype('int')  
외국인데이터['분기'] = 외국인데이터['시점'].apply(lambda x: int(x[6:7]))
```

```
동별대학교갯수 = 서울대학교데이터.pivot_table(values='키값',aggfunc='count',index=['자치구','행정동'])
```

4. 테이블_merge.ipynb

- 전처리를 거친 데이터들을 하나의 테이블을 기준으로 merge 한다
- merge 는 “주택매매데이터”를 기준으로 merge 한다, 예를 들면 아래 코드와 같이 주택매매데이터가 왼쪽에 있으므로 pandas 의 merge 옵션에 ‘left’를 입력하여 left merge 를 실행한다.

```
data = pd.merge(주택매매데이터,구별학생수,how='left',on=['년도','행정구'])
```

- 또한 데이터마다 통계 기준이 다르다, 예를 들면 위 코드에서의 “구별 학생수” 데이터는 “년도”, “행정구”를 기준으로 통계가 되었고 아래 코드에서의 “동별 평균연령”데이터는 “년도”, “행정구”, “행정동”, “분기” 별로 통계가 되어있기 때문에 상황에 맞게 merge 할 키 값을 다르게 주었다.

```
data = pd.merge(data,동별평균연령,how='inner',left_on = ['행정구','행정동','년도','분기'],right_on = ['자치구','행정동','년도','분기'])
```

Data Analysis (CDA):

1. 데이터 테이블 설명:

- index : 년도, 월, 분기, 행정구, 행정동
- row: 83498 column: 43 결측치: 0

년도	월	분기	행정구	행정동	거래금액(만원)	전용면적(㎡)	층	건축년도	유치원생수	유치원학급수	초등학교수	초등학교학급수	중학교수	중학교학급수	고등학교수	고등학교학급수	대학수	대학교학급수	유년부양자	노년부양자	노년복지수	인구수	면적	인구밀도	전체종류주택
2020	2	6	강남구	개포4동	149500	77.75	4	1988	3260	181	25905	990	15602	555	...	166	0.0	441	15.6	17.4	111.5	22765	1.49	15279	7409
				개포4동	162750	79.97	4	1987	3260	181	25905	990	15602	555	...	166	0.0	441	15.6	17.4	111.5	22765	1.49	15279	7409
				개포4동	160000	79.97	5	1987	3260	181	25905	990	15602	555	...	166	0.0	441	15.6	17.4	111.5	22765	1.49	15279	7409
				개포4동	150000	79.97	2	1987	3260	181	25905	990	15602	555	...	166	0.0	441	15.6	17.4	111.5	22765	1.49	15279	7409

9	강남구	개포1동	40000	77.71	5	2007	2700	
		개포1동	40000	77.71	5	2007	2700	
2	6	강남구	개포4동	100000	134.66	5	2002	1180

83489 rows x 43 columns

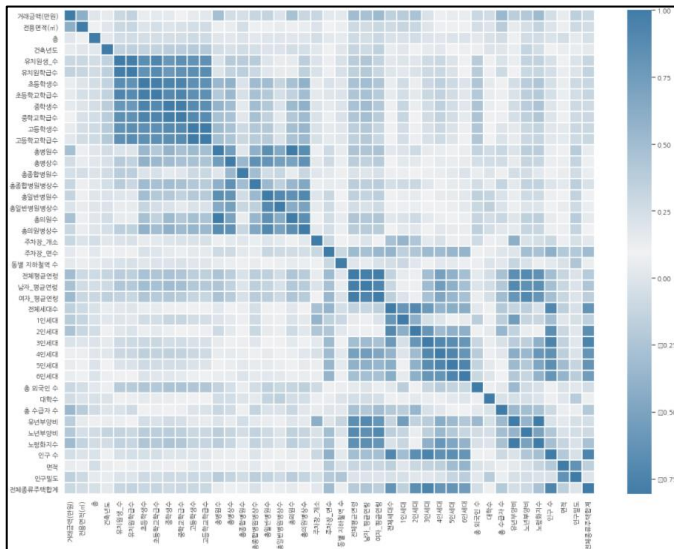
83489 rows × 43 columns

2. 상관계수(correlation) 분석

- Seaborn 의 heatmap 을 사용하였다, correlation 에 음수도 포함되어 있기 때문에 더욱 직관적인 시각화를 위해 아래와 같이 cmap 을 설정하여 적용하여 절대값을 기준으로 색상의 명암을 나타냈다

```
corr = data.corr(method='pearson')
plt.rcParams['font.family'] = 'NanumGothic'
plt.figure(figsize = (20,15))
cmap = sns.diverging_palette(240,240, as_cmap=True)
sns.heatmap(corr,cmap=cmap,linewidths=1)
```

- 추가로 scipy 라이브러리에 내장된 pearsonr 함수를 통해 각 feature 끼리의 correlation 과 p-value 를 확인한 결과 아래와 같다



```
data_columns = data.drop('거래금액(만원)',axis=1).columns
data_pvalue = pd.DataFrame()
for i in range(len(data_columns)):
    temp = pearsonr(data[data_columns[i]],data['거래금액(만원)'])
    if(temp[1] < 0.05):
        temp_ser = pd.DataFrame(data=np.array(pearsonr(data[data_columns[i]],data_pvalue)),axis=0)
        data_pvalue = pd.concat([data_pvalue,temp_ser],axis=0)
print(len(data_pvalue), len(data.drop('거래금액(만원)',axis=1).columns))
data_pvalue
```

	corr	p-value
전용면적(㎡)	0.606307	0.000000e+00
층	0.184188	0.000000e+00
건축년도	0.043089	1.299573e-35
유치원생수	-0.102484	1.034763e-193
유치원학급수	-0.145707	0.000000e+00
초등학교수	0.132650	0.000000e+00
초등학교학급수	0.037719	1.121840e-27
중학교수	0.160060	0.000000e+00
중학교학급수	0.089810	4.665028e-149

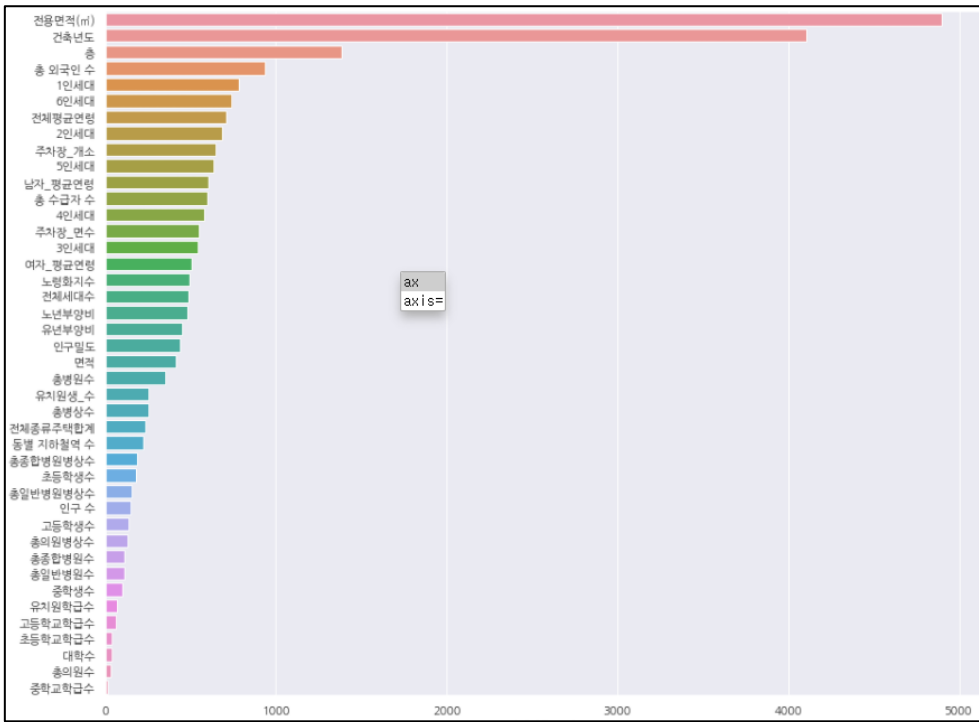
- 오른쪽 그림과 같이 각 feature 들의 label 인 “거래금액(만원)” 컬럼과의 P-value 가 유의확률 0.05 보다 큰 경우에만 Pearsonr 데이터프레임으로 생성되게끔 설정 하였다, 확인 결과 42 개의 컬럼 모두 label 과 상관관계가 존재하는 것을 의미한다. 또한 “corr”컬럼을 통해 label 과 양, 혹은 음의 상관관계를 가지는지 파악할 수 있다.

3. 회귀모델을 사용한 feature importance 분석 *본 프로젝트의 목적은 예측모델 개발이 아님을 인지하고 있습니다

- Feature 와 label 간의 어떤 관계가 있는지 더욱 자세하게 파악하기 위해 충분한 설명력을 지닌 모델의 feature importance 를 파악하는 것도 좋은 방법이다.
- 최근에 핫한 lightGBM 의 regression 모델을 사용했다, 또한 충분히 설명력이 있는 모델을 만들기 위해 KFold, GridSearch 기법을 사용했다 (PCA 는 해석이 어려워 적용하지 않았음)
- R2_score 를 확인해본 결과 아래와 같이 나왔다. 0.964 정도의 점수를 보여주고 있고 이를 통해 해당 모델이 충분히 설명력이 있다고 볼 수 있다.

```
r2_score(test_LGB_y2, y_LGB_pred2)
0.9650771769543856
```

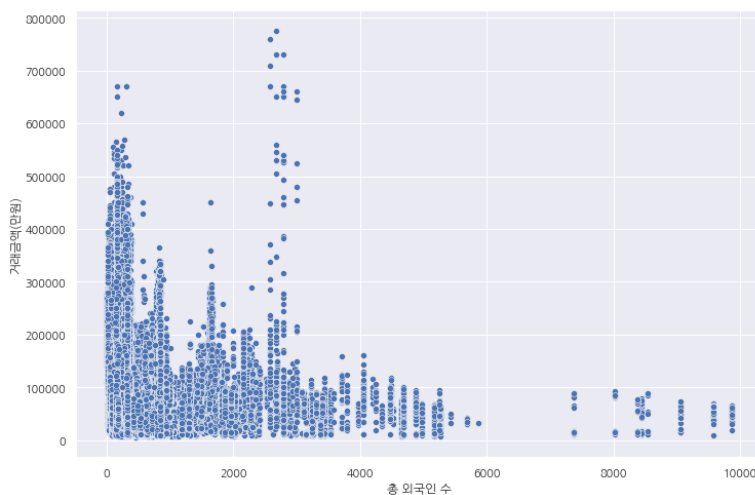

➤ 해당 모델의 feature importance 를 시각화한 결과 아래와 같이 나왔다.



“전용면적(m²)”, “건축년도”, “층” 컬럼의 feature importance 가 가장 높게 나왔다. 그 다음으로 “총 외국인 수”, “1인세대 수”, “6인세대 수”, “전체 평균연령” 등이 높은 importance 를 보였다.

해당 plot 을 통해 각 feature 들이 주택 거래금액에 얼마나 많은 영향을 미쳤는지 확인할 수 있었다.

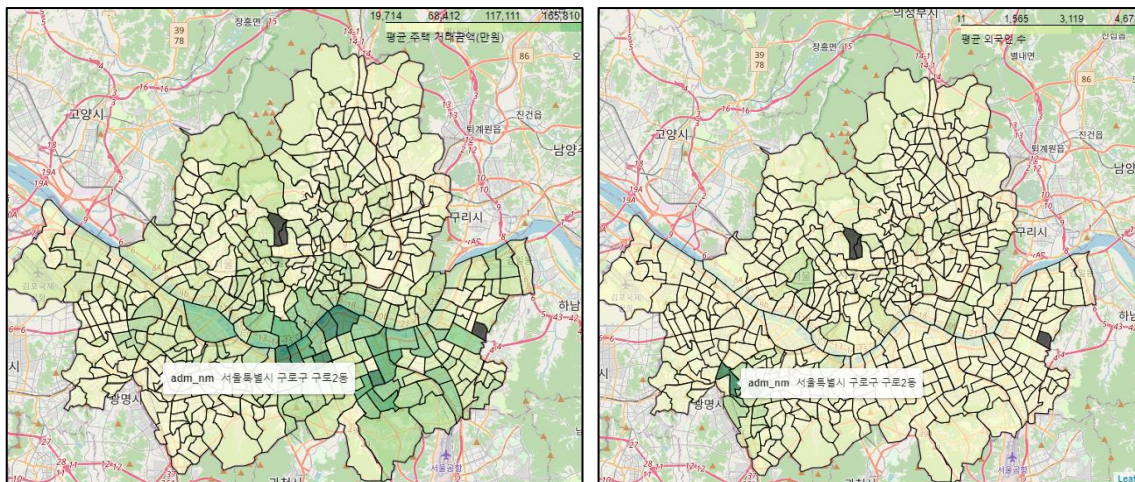
➤ 위 feature importance 를확인해보면 “총 외국인 수”가 4 번째로 높은 중요도를 가지고 있으므로 해당 컬럼을 시각화 해보았다.



seaborn 의 scatterplot 을 통해 확인한 결과 외국인 수가 적을 수록 주택 거래금액이 평균적으로 높게 나왔다, 반대로 외국인 수가 많아질수록 특정 시점, 즉 외국인수가 약 2000~3000 사이인 지점에 거래금액이 굉장히 높게 나온 기록이 있다, 외국인 수가 3000 이상인 지역들은 주택 거래금액이 낮음을 확인할 수 있다.

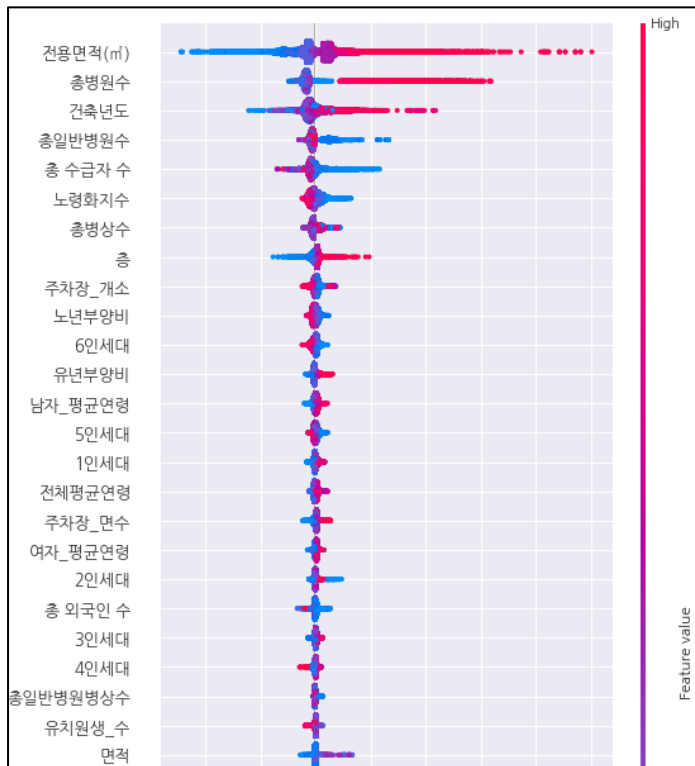
정리하자면 외국인 수가 적은 것은 주택 거래 금액에 큰 영향을 미치지 않지만 외국인 수가 많으면 주택 거래 금액이 평균적으로 낮아지는 것을 알 수 있다.

➤ “총 외국인 수”와 “거래금액(만원)”의 관계를 한 단계 더 파악하기 위해 folium 을 통해 지도시각화를 해보았다.



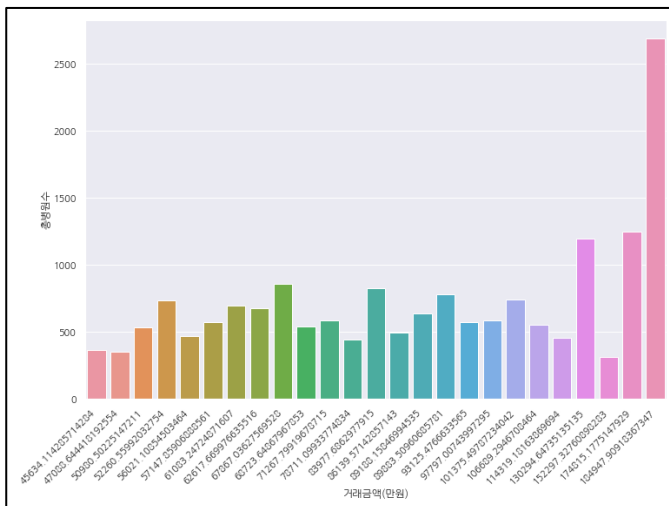
확인 결과 외국인 수가 적은 것은 주택 거래금액에 영향이 없지만 외국인이 많은 지역들은 대체로 주택 거래 금액이 낮다.

- 다음으로 SHAP 를 이용해 feature importance 를 심층 분석해보았다



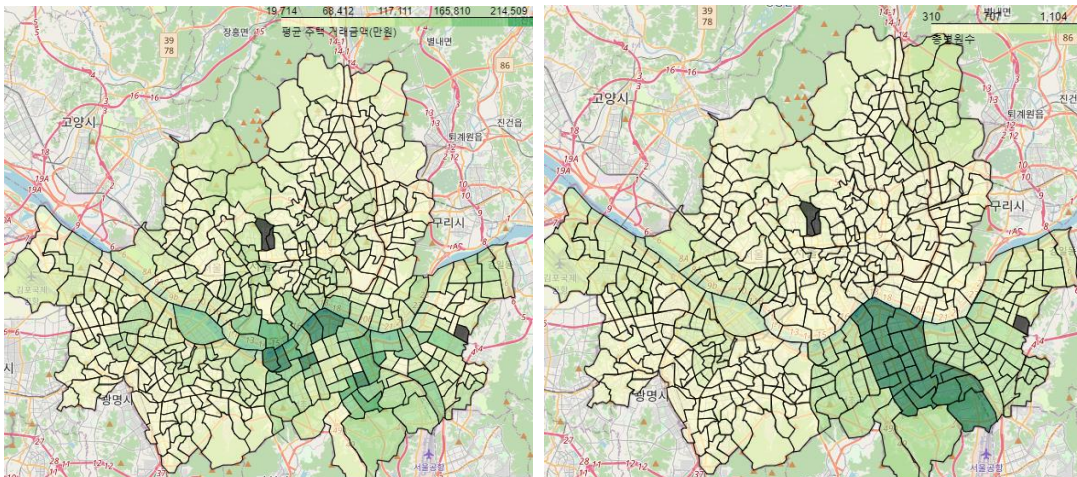
해당 plot 은 바로 위에서 시각화한 feature importance 보다 한 층 더 깊게 분석할 수 있었다. 예로 “총 병원수”를 분석해보자면 병원 수가 많을 수록 주택 거래 가격이 오른다, 하지만 반대로 병원수가 적은 것은 주택 거래가격에 큰 영향을 끼치지 않는다고 해석할 수 있다

- “총 병원 수” 컬럼과 “거래금액(만원)”의 관계를 시각화 해보았다.



“거래금액(만원)” 이 낮은 지역들은 “총 병원 수”가 크지 않고 비슷하게 나왔다, 하지만 거래금액이 매우 높은 지역들은 총 병원 수가 다른 지역에 비해 압도적으로 높게 나왔다.

- Folium 을 통해 총 병원 수와 거래금액의 관계를 확인해보았다



확인 결과 총 병원 수가 적은 것은 주택 거래 금액에 영향이 없지만 병원 수가 매우 많으면 주택 거래 가격 또한 매우 높아지는 것을 확인할 수 있다.

정리

데이터 분석의 가장 중요한 것은 충분한 데이터와 정확한 전처리라고 생각하여 데이터 수집과 전처리에 비중을 크게 뒀습니다. 또한 해당 프로젝트에서 모델개발은 요구사항이 아닌 것을 충분히 인지하고 있지만 본인이 생각하기에 통계적 지식만을 통해 데이터를 분석하는 것보단 좋은 모델을 만들어 해당 모델을 내부를 들여다보는 것도 분석의 질을 높이는 좋은 방법이라 생각했습니다.

Feature 는 굉장히 많지만 보고서의 길이제한과 발표시간에 한계가 있으므로 “총 외국인 수”와 “총 병원 수” 두 컬럼을 선택하여 주택 거래 가격 간의 관계를 여러가지 시각화 라이브러리를 사용해 분석해봤습니다. 나머지 컬럼들도 위에서 설명한 방식대로 correlation, p-value, SHAP, folium,seaborn 등등을 연계하여 분석할 수 있습니다.

전체 데이터(main_data.csv) 구조 (빨간 테두리는 추가된 데이터를 의미)

유치원학생수	유치원학급수	유치원생_수	분기	매월평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균	구평균
--------	--------	--------	----	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

추가 데이터 출처

- 서울시 동별 주차장 개수 및 면적 <https://data.seoul.go.kr/dataList/10152/S/2/datasetView.do>
- 서울시 동별 평균 연령 <https://data.seoul.go.kr/dataList/10591/S/2/datasetView.do>
- 서울시 세대원수별 세대수 <https://data.seoul.go.kr/dataList/10592/S/2/datasetView.do>
- 서울시 동별 인구밀도 <https://data.seoul.go.kr/dataList/10584/S/2/datasetView.do>
- 서울시 동별 주택종류별 통계 <https://data.seoul.go.kr/dataList/10585/S/2/datasetView.do>
- 서울시 동별 의료기관 통계 <https://data.seoul.go.kr/dataList/10123/S/2/datasetView.do>