

머신러닝, 그게 뭐죠



아파트 가격 형성에 미치는 요인 분석 및 가격 예측





머신러닝, 그게 뭐죠

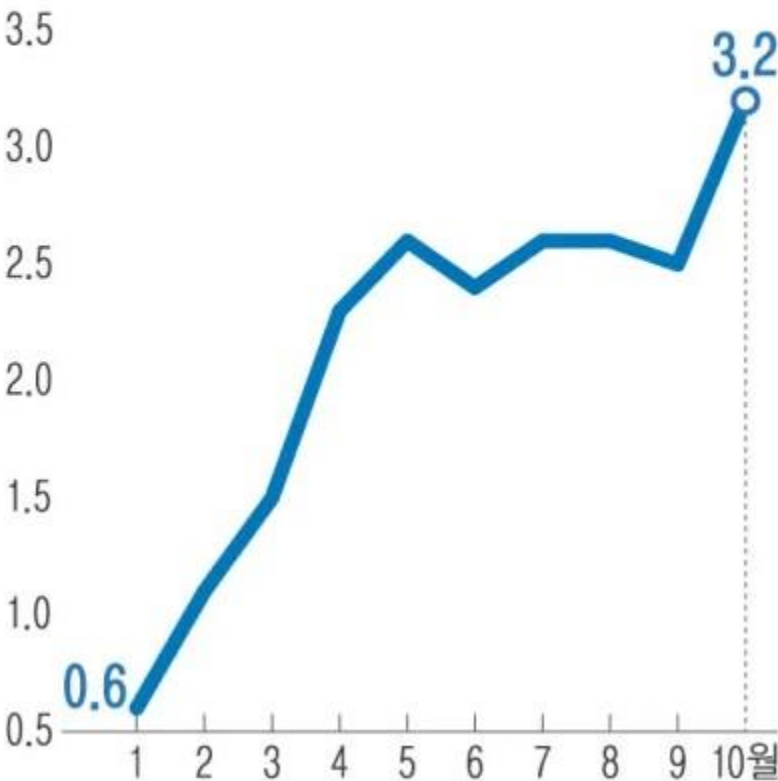
CONTENTS

- 01 주제선정이유
- 02 데이터 수집 과정
- 03 EDA 및 시각화
- 04 모델 구현
- 05 결론 및 아쉬운점



올해 소비자물가 상승률

단위: %, 전년 동월 대비, 자료: 통계청



매매/전세 가격변동

※ 출처: KB부동산

장기적으로 매매, 전세가격이 어떤 추세인지 확인해보세요.





물가 상승률과 집값은 큰 폭으로 상승
하지만 최저임금은 여전히 소폭 상승

지난 5년간 최저임금

(단위= 원)



(자료: 최저임금위원회)



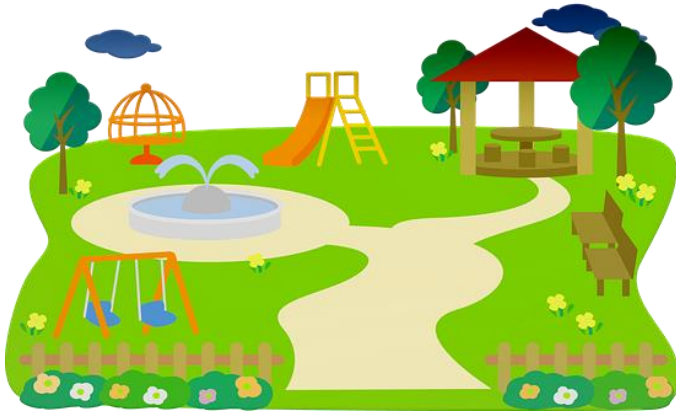
그렇다면 내가 원하는 집을 사려면
얼마정도 있어야 할까?

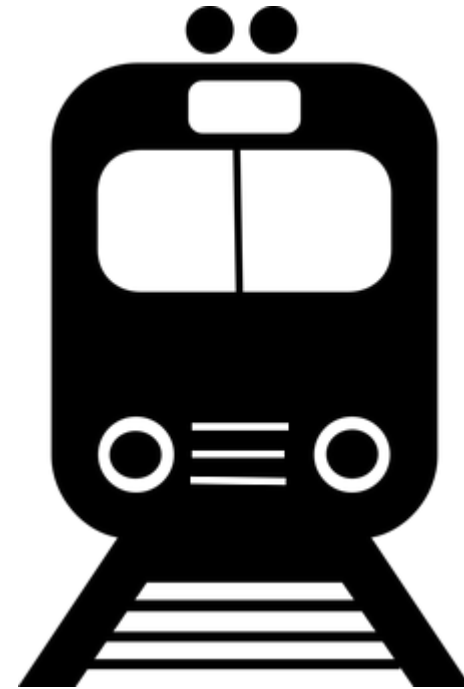


메인 데이터

- 케글에서 다운받은 서울 아파트 실거래가 내역

데이터 수집





EDA 및 시각화



공원 데이터

```
1 # 각 구별 공원 데이터 불러오기
2 park_Gangnam = pd.read_csv('./data/서울특별시_강남구_도시공원_20210709_1625817293166_17363.csv', encoding='utf-8-sig')
3 park_Gangdong = pd.read_csv('./data/서울특별시_강동구_도시공원정보_20210825_1629872998632_15968.csv', encoding='utf-8-sig')
4 park_Gangbuk = pd.read_csv('./data/서울특별시_강북구_도시공원정보_20210824_1629982590816_8180.csv', encoding='utf-8-sig')
5 park_Gangseo = pd.read_csv('./data/서울특별시_강서구_도시공원정보_20210322_1616991304943_20133.csv', encoding='utf-8-sig')
6 park_Gwanak = pd.read_csv('./data/서울특별시_관악구_도시공원정보_20211015_1634617964680_12966.csv', encoding='utf-8-sig')
7 park_Gwangjin = pd.read_csv('./data/서울특별시_광진구_도시공원정보_20201211_1608529999233_8268.csv', encoding='utf-8-sig')
8 park_Guro = pd.read_csv('./data/서울특별시_구로구_도시공원정보_20210218_1613636520082_9690.csv', encoding='utf-8-sig')
9 park_Geumcheon = pd.read_csv('./data/서울특별시_금천구_도시공원정보_20210902_1630927108163_10424.csv', encoding='utf-8-sig')
10 park_Nowon = pd.read_csv('./data/서울특별시_노원구_도시공원정보_20210913_1631586982975_23533.csv', encoding='utf-8-sig')
11 park_Dobong = pd.read_csv('./data/서울특별시_도봉구_도시공원정보_20190101.csv', encoding='utf-8-sig')
12 park_Dongdaemun = pd.read_csv('./data/서울특별시_동대문구_도시공원정보_20210625_1624583959638_8588.csv', encoding='utf-8-sig')
13 park_Dongjak = pd.read_csv('./data/서울특별시_동작구_도시공원정보_20210429_1619684732418_7703.csv', encoding='utf-8-sig')
14 park_Mapo = pd.read_csv('./data/서울특별시_마포구_도시공원정보_20200709_1608170783985_13239.csv', encoding='utf-8-sig')
15 park_Seodaemun = pd.read_csv('./data/서울특별시_서대문구_도시공원정보_20210801_1627885751593_11799.csv', encoding='utf-8-sig')
16 park_Seongdong = pd.read_csv('./data/서울특별시_성동구_도시공원정보_20210930_1632983304518_11837.csv', encoding='utf-8-sig')
17 park_Seongbuk = pd.read_csv('./data/서울특별시_성북구_도시공원정보_20210616_1623821311229_9981.csv', encoding='utf-8-sig')
18 park_Songpa = pd.read_csv('./data/서울특별시_송파구_도시공원정보_20210221_1614749779704_23781.csv', encoding='utf-8-sig')
19 park_Yangcheon = pd.read_csv('./data/서울특별시_양천구_도시공원정보_20210430_1620710785425_20066.csv', encoding='utf-8-sig')
20 park_Yeongdeungpo = pd.read_csv('./data/서울특별시_영등포구_도시공원정보_20210929_1634548360357_10921.csv', encoding='utf-8-sig')
21 park_Yongsan = pd.read_csv('./data/서울특별시_용산구_도시공원정보_20210927_1632788099359_7189.csv', encoding='utf-8-sig')
22 park_Eunpyeong = pd.read_csv('./data/서울특별시_은평구_도시공원정보_20210501_1621212689965_22592.csv', encoding='utf-8-sig')
23 park_Jongno = pd.read_csv('./data/서울특별시_종로구_도시공원정보_20210709_1625794001940_7949.csv', encoding='utf-8-sig')
24 park_Junggu = pd.read_csv('./data/서울특별시_중구_도시공원정보_20211022_1634865794810_7150.csv', encoding='utf-8-sig')
25 park_Jungnang = pd.read_csv('./data/서울특별시_중랑구_도시공원정보_20201112_1605164754016_7760.csv', encoding='utf-8-sig')
26 park_Seochu = pd.read_csv('./data/서울특별시_서초구_도시공원정보_20200125.csv')
```

	위도	경도	구
0	37.692080	127.031204	도봉구
1	37.647426	127.041429	도봉구
2	37.654192	127.029088	도봉구
3	37.647271	127.052319	도봉구
4	37.680396	127.050046	도봉구
...
138	37.460000	127.030000	서초구
139	37.460000	127.020000	서초구
140	37.450000	127.060000	서초구
141	37.450000	127.050000	서초구
142	37.450000	127.040000	서초구

각 구별로 존재하던 데이터



거리 계산 및 지도 시각화에
필요한 칼럼을 제외 후 하나의
데이터 프레임으로 결합



지진 대피소 데이터

	피난처 ID	시도명	시군구명	시설명	상세주소	시설면적	경도	위도	구분	구분명	관리부서
0	0	서울특별시	용산구	용산공업고등학교	서울특별시 용산구 서빙고로 24 (한강로3가)	19129.0	126.967082	37.524409	1	지진실내피소	3
1	0	서울특별시	강남구	개원초등학교 운동장	서울특별시 강남구 개포동 660-10	12032.0	없음	없음	2	지진육외대피소	3
2	0	서울특별시	강북구	상산공원	서울특별시 강북구 수유동 224-4	973.0	127.021805	37.638326	2	지진육외대피소	3
3	0	서울특별시	종로구	중앙고등학교	서울특별시 종로구 계동 1-0	11000.0	126.987035	37.584897	2	지진육외대피소	3
4	0	서울특별시	관악구	신관중학교 운동장	서울특별시 관악구 신림동 87-81	3608.0	126.934573	37.480282	2	지진육외대피소	3
...
2125	99	서울특별시	마포구	마포대진유치원	서울특별시 마포구 성미산로 110 (연남동)	1024.0	126.91939	37.561882	1	지진실내피소	3
2126	99	서울특별시	양천구	양동중학교 운동장	서울특별시 양천구 목동 600-6	1551.0	126.875853	37.54555	2	지진육외대피소	3
2127	99	서울특별시	동대문구	청량고등학교	서울특별시 동대문구 왕산로 303 (청량리동)	3500.0	127.050344	37.589103	1	지진실내피소	3
2128	99	서울특별시	은평구	응암1마을마당	서울특별시 은평구 응암동 97-16	635.0	126.925963	37.600091	2	지진육외대피소	3
2129	99	서울특별시	강동구	전호공원광장	서울특별시 강동구 전호동 360-0	1300.0	127.126471	37.544671	2	지진육외대피소	3

많은 칼럼들이 존재



필요한 칼럼들을 제외하고
나머지 삭제

	시군구명	시설명	상세주소	경도	위도
0	용산구	용산공업고등학교	서울특별시 용산구 서빙고로 24 (한강로3가)	126.967082	37.524409
2	강북구	상산공원	서울특별시 강북구 수유동 224-4	127.021805	37.638326
3	종로구	중앙고등학교	서울특별시 종로구 계동 1-0	126.987035	37.584897
4	관악구	신관중학교 운동장	서울특별시 관악구 신림동 87-81	126.934573	37.480282
5	구로구	개명초등학교운동장	서울특별시 구로구 개봉동 314-165	126.850698	37.483005
...
2125	마포구	마포대진유치원	서울특별시 마포구 성미산로 110 (연남동)	126.91939	37.561882
2126	양천구	양동중학교 운동장	서울특별시 양천구 목동 600-6	126.875853	37.54555
2127	동대문구	청량고등학교	서울특별시 동대문구 왕산로 303 (청량리동)	127.050344	37.589103



소방서 데이터

연번	서 · 센터ID	서 · 센터명	유형구분명	도서지역포함여부	상위서 · 센터ID	일련번호	X좌표	Y좌표
0	1	1116000 관악소방서	소방서		1100000	43	195785.5015	541613.1799
1	2	1114103 영동119안전센터	안전센터/구조대		1114000	44	203265.8115	546962.3998
2	3	1116401 관악119안전센터	안전센터/구조대		1116000	45	195784.8784	541621.2103
3	4	1119106 시흥119안전센터	안전센터/구조대		1119000	46	191196.7211	538802.5714
4	5	1115102 방배119안전센터	안전센터/구조대		1115000	47	198957.0960	541787.0010
...
172	173	1125102 우이119안전센터	안전센터/구조대		1125000	38	201461.9413	560141.8011
173	174	1125103 미아119안전센터	안전센터/구조대		1125000	39	202500.7233	557825.9800
174	175	1111000 노원소방서	소방서		1100000	40	206278.7101	559835.8943
175	176	1110000 도봉소방서	소방서		1100000	41	203797.4045	562711.4224
176	177	1111102 상계119안전센터	안전센터/구조대		1111000	42	206323.8230	561714.8862

많은 칼럼들이 존재



소방서만 추출한 후
X,Y좌표를 위도와 경도로 바꿈

	서 · 센터명	유형구분명	lng	lat	구
0	관악소방서	소방서	126.952355	37.473943	관악구
1	서초소방서	소방서	126.991952	37.498421	서초구
2	구로소방서	소방서	126.865187	37.497799	구로구
3	마포소방서	소방서	126.933181	37.549383	마포구
4	중랑소방서	소방서	127.094855	37.615355	중랑구



대형마트 데이터

관리번호 인허가일자 인허가취소일자 영업상태코드 상세계영업상태코드 상세계영업상태코드 폐업일자 휴업시작일자 휴업종료일자 재개업일자 전화번호 소재지면적 소재지우편번호 지번주소 도로명주소 도로명우편번호 사업장명 최종수정일자 데이터갱신구분 데이터갱신일자 업태구분명



도로명주소 업태구분명 lat lng

```
geolocator = Nominatim(user_agent= 'South Korea')
lat = []
lng = []
for i in range(len(mart_address)):
    try:
        location = geolocator.geocode(mart_address[i])
        lat.append(location.latitude)
        lng.append(location.longitude)
    except:
        lng.append('d')
        lat.append('d')
        print(i, 'no')
```

지오코딩을 통해
위도와 경도 데이터 불러옴



병원 데이터

지오코딩을 통해
위도와 경도 데이터 불러옴

상세영업상태코드
상세영업상태명
폐업일자
휴업시작일자
휴업종료일자
재개업일자
전화번호
소재지면적
소재지우편번호
지번주소
도로명주소
도로명우편번호
사업장명
최종수정일자
데이터갱신구분
데이터갱신일자
업태구분명
좌표정보(X)
좌표정보(Y)
의료기관종별명



사업장명	도로명주소	lat	lng	구
서울적십자병원	서울특별시 종로구 새문안로 9 적십자병원	37.567083	126.967007	종로구
세란병원	서울특별시 종로구 통일로 256	37.566283	126.966250	종로구
경찰병원	서울특별시 송파구 송이로 123 국립경찰병원	37.496561	127.123473	송파구
인제대학교 서울백병원	서울특별시 중구 마른내로 9	37.564920	126.994954	중구
한양대학교병원	서울특별시 성동구 왕십리로 222-1	37.561482	127.035053	성동구

```
lat = []
lng = []
for i in range(len(address)):
    try:
        location = geolocator.geocode(address[i])
        lat.append(location.latitude)
        lng.append(location.longitude)
    except:
        lng.append('d')
        lat.append('d')
        print(i, 'no')
```



도서관 데이터

도서관 일련번 호	도서관명	구 코드	구 명	주소	전화번 호	홈페이지 URL	운영시간	정기 휴 관일	도서 관 구분 명	위도	경도
-----------------	------	---------	--------	----	----------	----------	------	------------	--------------------	----	----



	구	lat	lng
0	강남구	37.520770	127.024565
1	강남구	37.469972	127.091021
2	강남구	37.470543	127.113773
3	강남구	37.471538	127.096582
4	강남구	37.496497	127.032027

거리계산을 위해
위도와 경도를 제외하고 삭제



초등학교 데이터

```
school = school[school['시도교육청명'] == '서울특별시교육청']  
school = school[school['학교급구분'] == '초등학교']  
school = school[school['운영상태'] == '운영']
```



	학교명	위도	경도	구
0	서울양동초등학교	37.525848	126.851240	양천구
1	서울신구초등학교	37.523361	127.023992	강남구
2	서울봉화초등학교	37.613576	127.092409	중랑구
3	서울중화초등학교	37.602922	127.094065	중랑구
4	서울상월초등학교	37.658652	127.056980	노원구
...
599	서울해누리초등학교	37.498412	127.102124	송파구
600	서울항동초등학교	37.477016	126.824110	구로구
601	서울하늘숲초등학교	37.484159	126.841418	구로구
602	서울고일초등학교	37.551543	127.169410	강동구
603	서울가락초등학교	37.500021	127.107642	송파구

전국 초중고 데이터셋



서울시 초등학교만 추출



Cctv와 범죄 발생 데이터

	총계	구
0	1772	종로구
1	2333	중구
2	2383	용산구
3	3602	성동구
4	2588	광진구
5	2497	동대문구
6	3296	중랑구
7	3958	성북구
8	2462	강북구
9	1629	도봉구
10	2415	노원구
11	3791	은평구
12	2940	서대문구
13	2421	마포구
14	3312	양천구
15	2560	강서구
16	4075	구로구
17	2374	금천구

	구	발생
0	종로구	3102
1	중구	3411
2	용산구	2969
3	성동구	2362
4	광진구	3601
5	동대문구	3401
6	중랑구	3726
7	성북구	2567
8	강북구	2770
9	도봉구	2179
10	노원구	3743
11	은평구	3390
12	서대문구	2533
13	마포구	3688
14	양천구	3216
15	강서구	4415
16	구로구	4175
17	금천구	2598

구별로 cctv 수와
범죄 발생 건수를 정리



아파트 거래 데이터

	id	lat	lng	households	buildDate	score	m2	p	min_sales	max_sales	avg_sales
0	2766	37.681604	127.056592	492	200006	4.3	139	42	96160.0	99200.0	97600.0
1	5860	37.679290	127.057021	468	200105	4.1	105	32	77760.0	83520.0	81600.0
2	15564	37.676882	127.058075	57	200502	4.8	86	26	57600.0	73600.0	64800.0
3	3700	37.675277	127.060001	216	199509	4.8	102	31	54400.0	55680.0	55200.0
4	6204	37.676381	127.058361	165	200306	4.8	91	28	44640.0	80480.0	64000.0



아파트 거래 데이터

```
#거리 계산 함수 만들기
from math import cos, asin, sqrt
def distance(lat1, lon1, lat2, lon2):
    p = 0.017453292519943295
    a = 0.5 - cos((lat2 - lat1) * p)/2 + cos(lat1 * p) * cos(lat2 * p) * (1 - cos((lon2 - lon1) * p)) / 2
    return 12742 * asin(sqrt(a)) * 1000
park_first = distance(apt['lat'][0], apt['lng'][0], park['위도'][0], park['경도'][0])
subway_first = distance(apt['lat'][0], apt['lng'][0], subway['위도'][0], subway['경도'][0])
mart_first = distance(apt['lat'][0], apt['lng'][0], mart['lat'][0], mart['lng'][0])
hospital_first = distance(apt['lat'][0], apt['lng'][0], hospital['lat'][0], hospital['lng'][0])
library_first = distance(apt['lat'][0], apt['lng'][0], library['lat'][0], library['lng'][0])
school_first = distance(apt['lat'][0], apt['lng'][0], school['위도'][0], school['경도'][0])
eq_shelter_first = distance(apt['lat'][0], apt['lng'][0], eq_shelter['위도'][0], eq_shelter['경도'][0])
fire_station_first = distance(apt['lat'][0], apt['lng'][0], fire_station['lat'][0], fire_station['lng'][0])
```

```
# 주소를 받아 구만 추가
gu = []
for i in range(len(apt)):
    try:
        location = geolocator.reverse(a[i])
        address = location.raw['address']
        district = address['district']
        gu.append(district)
        print(i, district)
    except:
        gu.append('d')
        print(i, 'no')
```

역지오코딩을 통해서 아파트 데이터에 구 칼럼 생성

거리계산 함수를 통해 위에서 확인한 데이터들과
아파트의 거리를 계산

EDA 및 시각화



```
# 마트와의 거리
for i in range(len(apt)):
    for j in range(len(mart)):
        if apt['구'][i] == mart['구'][j]:
            dist=distance(apt['lat'][i],apt['lng'][i],mart['lat'][j],mart['lng'][j])
            if dist < apt['distance_mart'][i]:
                apt['distance_mart'][i]=dist
```

ted in 2.92s, finished 02:51:29 2021-12-09

```
# 공원과의 거리
for i in range(len(apt)):
    for j in range(len(park)):
        if apt['구'][i] == park['구'][j]:
            dist=distance(apt['lat'][i],apt['lng'][i],park['위도'][j],park['경도'][j])
            if dist < apt['distance_park'][i]:
                apt['distance_park'][i]=dist
```

ted in 57.1s, finished 02:52:26 2021-12-09

```
# 지하철과의 거리
for i in range(len(apt)):
    for j in range(len(subway)):
        if apt['구'][i] == subway['구'][j]:
            dist=distance(apt['lat'][i],apt['lng'][i],subway['위도'][j],subway['경도'][j])
            if dist < apt['distance_subway'][i]:
                apt['distance_subway'][i]=dist
```

ted in 8.19s, finished 02:52:34 2021-12-09

```
# 병원과의 거리
for i in range(len(apt)):
    for j in range(len(hospital)):
        if apt['구'][i] == hospital['구'][j]:
            dist=distance(apt['lat'][i],apt['lng'][i],hospital['lat'][j],hospital['lng'][j])
            if dist < apt['distance_hospital'][i]:
                apt['distance_hospital'][i]=dist
```

EDA 및 시각화



구	distance_park	distance_subway	distance_mart	distance_hospital	distance_library	distance_school	distance_eq_shelter	distance_fire_station
구	103.547943	530.778621	4762.566819	3660.353884	235.631228	345.348255	249.675581	5002.132089
구	111.440575	302.574907	4503.356469	3400.320951	194.775282	601.785605	484.065391	4743.837210
구	185.242267	246.522050	4221.650799	3123.115924	435.601729	367.172413	402.984930	4461.166088
구	90.501611	455.402397	4012.343981	2930.096511	583.258162	231.475023	272.791262	4246.701551
구	211.981883	278.693790	4161.798191	3064.935357	494.939997	310.045767	347.186476	4400.869947
...
구	240.105493	1021.077972	1114.087167	492.318532	285.125398	560.115730	183.791228	528.353127
구	82.641658	3841.982675	4086.454855	2195.184447	117.451394	248.134631	67.652091	3766.316516



crime	cctv
3743	2415
3743	2415
3743	2415
3743	2415
3743	2415
...	...
3216	3312
3216	3312
3216	3312

구별로 범죄 사건 건수와
Cctv개수 추가



지진 대피소 지도 시각화

```
m = folium.Map(
    location=[37.55, 126.98],
    zoom_start=10
)

latlong = apart[['lat', 'lng']]

marker_cluster = MarkerCluster().add_to(m)

for lat, long in zip(latlong.lat, latlong.lng):
    folium.Marker([lat, long], icon = folium.Icon(color="green")).add_to(marker_cluster)

folium.GeoJson(
    geo_str,
    name='지역구'
).add_to(m)

m.choropleth(geo_data=geo_str,
              data=eq_shelter_group,
              fill_color='RdPu',
              fill_opacity=0.7,
              line_opacity=0.2,
              key_on='properties.name',
              legend_name="지역구별 대피소 수"
)

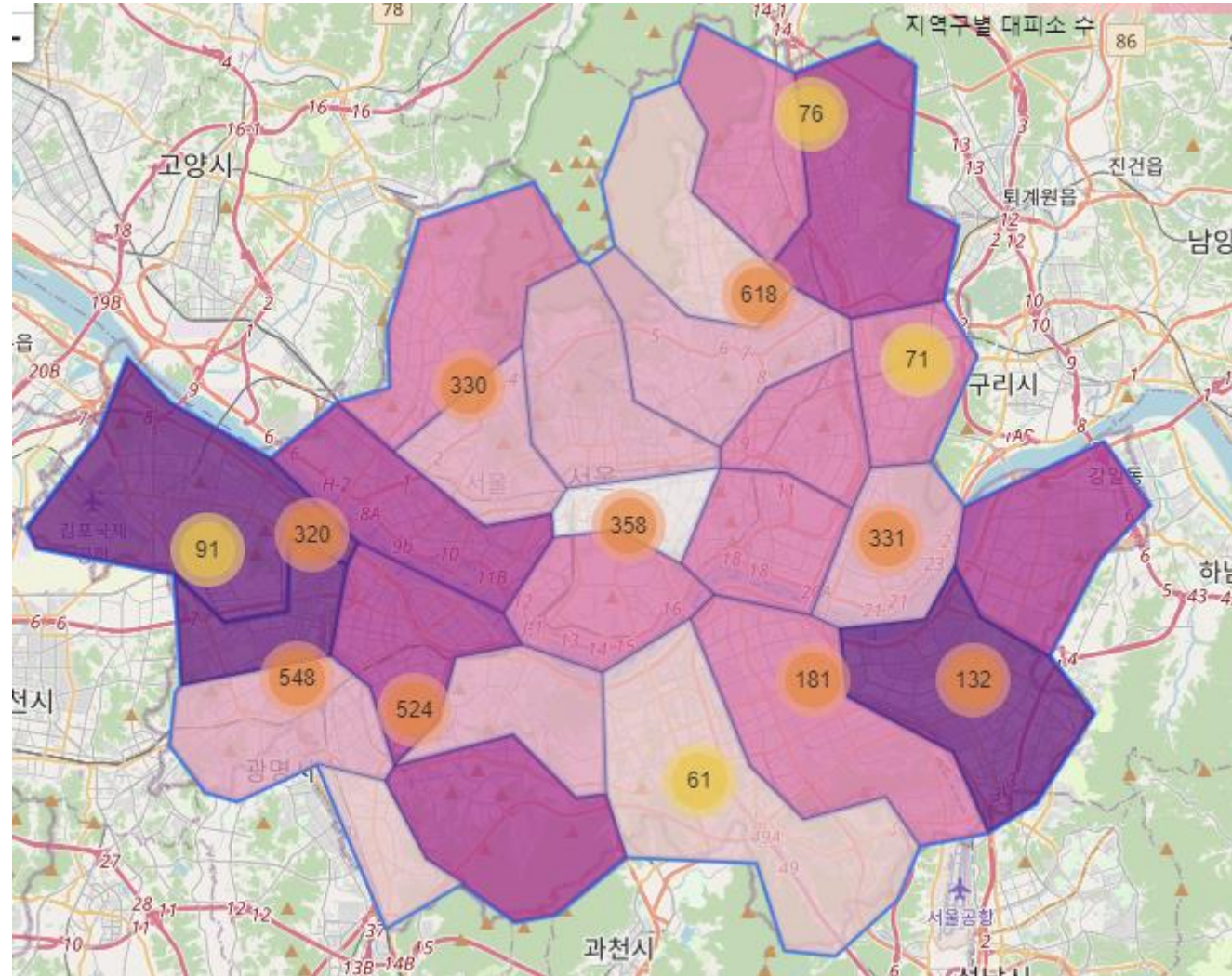
m
```

서울을 구별로 나눈 지도에
지진 대피소의 수에 따라
색을 다르게 나타나게 지정 후
아파트 거래 위치를 지도에 표기

EDA 및 시각화



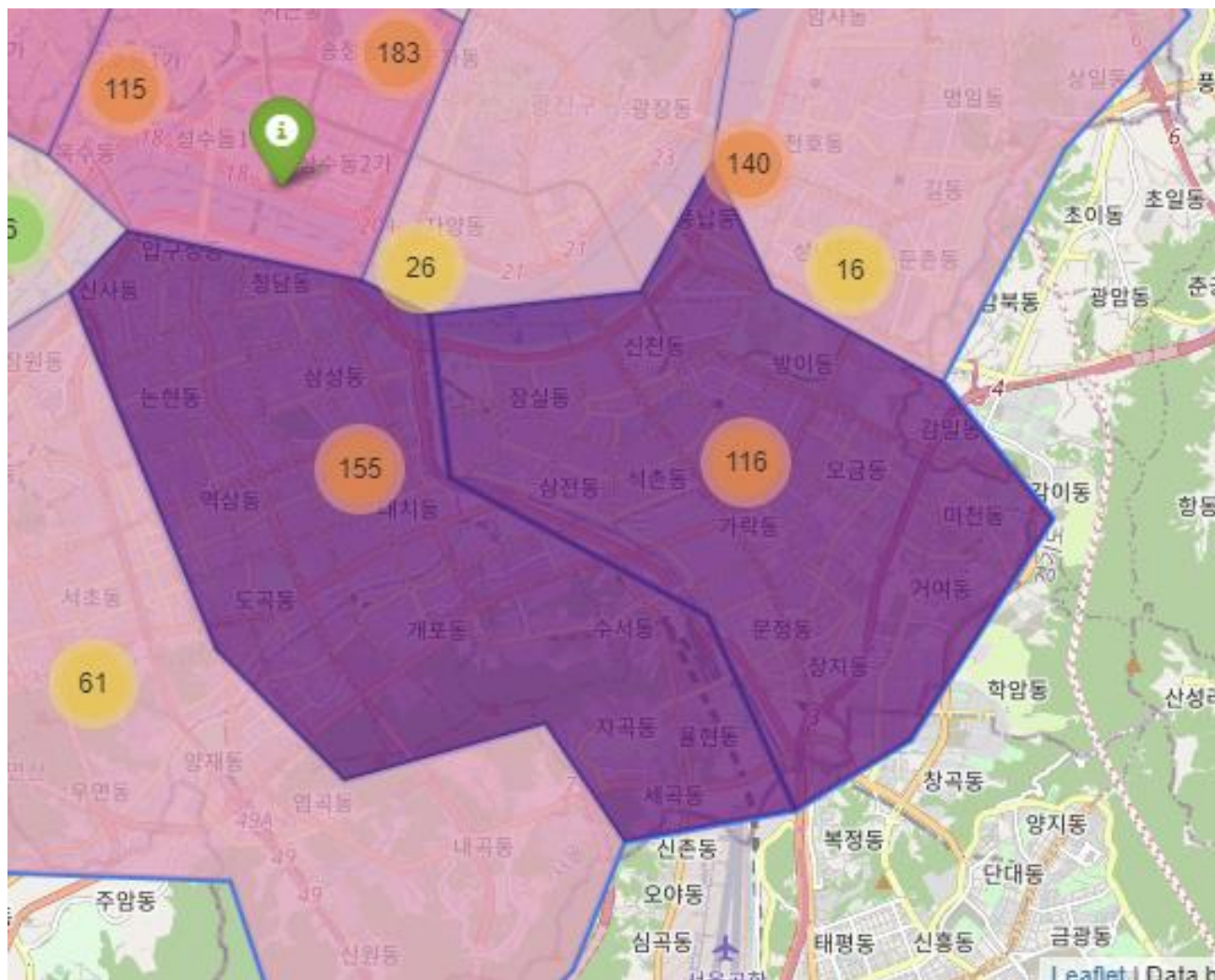
지진 대피소 지도 시각화



EDA 및 시각화



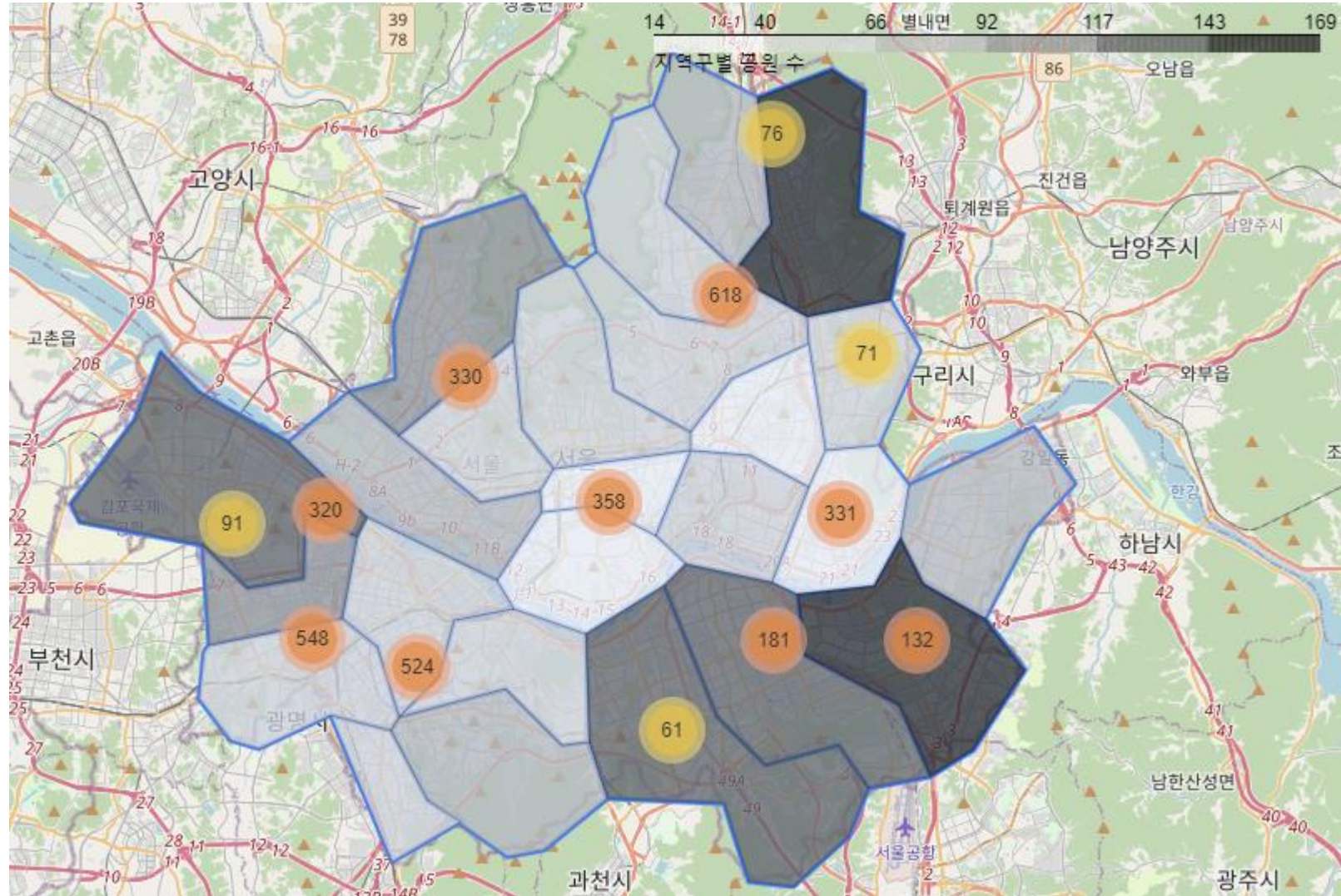
지하철 지도 시각화



EDA 및 시각화



공원 지도 시각화



A stylized illustration of a hand holding a magnifying glass over a clipboard. The clipboard features a bar chart with three bars of increasing height. The background is white with faint, light blue clouds and green horizontal lines, suggesting a clean, professional environment.



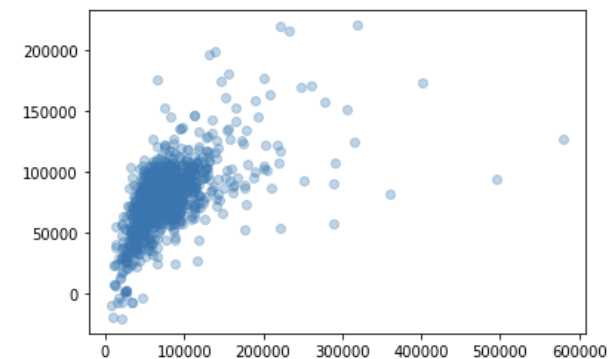
기본 데이터 모델링

```
1 X = apart[['households', 'buildDate', 'score', 'm2']]
2 y = apart[['avg_sales']]
```



MAE : 22906.25
MSE : 1428862908.23
RMSE : 37800.30
R2 : 0.33

기본 데이터로 세대 수, 건축연도, 평수,
평점으로 회귀 모델 구현



실제값과 예측값 그래프



상관관계가 높은 변수만 추가한 모델링

	id	lat	lng	households	buildDate	score	m2	p	min_sales	max_sales	avg_sales	distance_
id	1.000000	-0.003684	-0.064953	-0.120469	0.505803	-0.395558	-0.169551	-0.171069	0.033897	0.078156	0.045576	-0.06
lat	-0.003684	1.000000	0.369260	0.099484	0.013612	0.039301	-0.085408	-0.085162	-0.203279	-0.161693	-0.199920	-0.10
lng	-0.064953	0.369260	1.000000	0.098066	-0.034658	0.042590	0.073411	0.073681	0.164261	0.165113	0.184560	0.04
households	-0.120469	0.099484	0.098066	1.000000	-0.011899	0.177839	-0.092318	-0.091003	0.036566	0.137997	0.088592	-0.00
buildDate	0.505803	0.013612	-0.034658	-0.011899	1.000000	0.000440	0.115492	0.114998	0.310001	0.268545	0.309067	-0.10
score	-0.395558	0.039301	0.042590	0.177839	0.000440	1.000000	0.119829	0.120832	0.107234	0.073768	0.087911	-0.01
m2	-0.169551	-0.085408	0.073411	-0.092318	0.115492	0.119829	1.000000	0.999402	0.689748	0.420438	0.567937	0.06
p	-0.171069	-0.085162	0.073681	-0.091003	0.114998	0.120832	0.999402	1.000000	0.689315	0.419997	0.567553	0.06
min_sales	0.033897	-0.203279	0.164261	0.036566	0.310001	0.107234	0.689748	0.689315	1.000000	0.778970	0.924064	0.03
max_sales	0.078156	-0.161693	0.165113	0.137997	0.268545	0.073768	0.420438	0.419997	0.778970	1.000000	0.916898	0.03
avg_sales	0.045576	-0.199920	0.184560	0.088592	0.309067	0.087911	0.567937	0.567553	0.924064	0.916898	1.000000	0.04
distance_park	-0.064950	-0.107998	0.043314	-0.004073	-0.105688	-0.019063	0.060727	0.060758	0.039456	0.039119	0.044449	1.00
distance_subway	0.003595	-0.215284	-0.398986	-0.048891	0.020488	-0.032718	-0.024279	-0.023469	-0.173839	-0.184393	-0.193340	0.02
distance_mart	0.042163	0.067300	-0.119341	-0.031696	-0.040090	-0.067523	-0.013638	-0.015623	-0.050098	-0.031804	-0.053899	0.03
distance_hospital	0.073745	0.080923	-0.059238	-0.029467	0.085348	-0.031012	-0.016827	-0.016499	0.011885	0.002110	0.009928	-0.01
distance_library	-0.038983	-0.050827	0.049087	-0.025345	-0.106327	-0.053287	0.077484	0.076796	0.094787	0.113544	0.116321	0.15
distance_school	0.067162	-0.047983	0.032957	-0.096157	-0.008288	-0.112829	0.052939	0.052109	0.043471	0.068885	0.054705	0.23
distance_eq_shelter	0.044939	-0.018714	0.108075	0.011254	0.015922	-0.067025	0.059679	0.059169	0.041474	0.044807	0.040409	0.35
distance_fire_station	0.070149	0.029533	0.027296	-0.004055	-0.052051	-0.058617	-0.052427	-0.052301	-0.003913	0.024594	0.012242	0.08
crime	-0.061506	-0.490438	0.032515	-0.020767	-0.061486	-0.005585	0.144390	0.144934	0.322226	0.263252	0.326913	0.04
cctv	-0.046820	-0.414376	-0.068797	0.006629	-0.001143	0.035593	0.116852	0.116858	0.298727	0.260712	0.309652	0.12



상관관계가 높은 변수만 추가한 모델링

distance_subway	distance_mart	distance_hospital	distance_library	distance_school	distance_eq_shelter	distance_fire_station	crime	cctv
0.003595	0.042163	0.073745	-0.038983	0.067162	0.044939	0.070149	-0.061506	-0.046820
-0.215284	0.067300	0.080923	-0.050827	-0.047983	-0.018714	0.029533	-0.490438	-0.414376
-0.398986	-0.119341	-0.059238	0.049087	0.032957	0.108075	0.027296	0.032515	-0.068797
-0.048891	-0.031696	-0.029467	-0.025345	-0.096157	0.011254	-0.004055	-0.020767	0.006629
0.020488	-0.040090	0.085348	-0.106327	-0.008288	0.015922	-0.052051	-0.061486	-0.001143
-0.032718	-0.067523	-0.031012	-0.053287	-0.112829	-0.067025	-0.058617	-0.005585	0.035593
-0.024279	-0.013638	-0.016827	0.077484	0.052939	0.059679	-0.052427	0.144390	0.116852
-0.023469	-0.015623	-0.016499	0.076796	0.052109	0.059169	-0.052301	0.144934	0.116858
-0.173839	-0.050098	0.011885	0.094787	0.043471	0.041474	-0.003913	0.322226	0.298727
-0.184393	-0.031804	0.002110	0.113544	0.068885	0.044807	0.024594	0.263252	0.260712
-0.193340	-0.053899	0.009928	0.116321	0.054705	0.040409	0.012242	0.326913	0.309652
0.021764	0.037007	-0.016027	0.198594	0.230484	0.359230	0.085623	0.042097	0.124364
1.000000	-0.005586	-0.023890	-0.019813	-0.025855	-0.017430	-0.051488	-0.086023	-0.057748
-0.005586	1.000000	-0.048739	0.011652	0.137383	0.090232	-0.022464	-0.273024	-0.195970
-0.023890	-0.048739	1.000000	0.035968	-0.006368	0.001297	0.040814	-0.065954	-0.100090
-0.019813	0.011652	0.035968	1.000000	0.261013	0.257554	-0.017759	0.119412	0.087363
-0.025855	0.137383	-0.006368	0.261013	1.000000	0.553764	0.099121	0.042515	-0.002980
-0.017430	0.090232	0.001297	0.257554	0.553764	1.000000	0.047861	0.029488	0.079995
-0.051488	-0.022464	0.040814	-0.017759	0.099121	0.047861	1.000000	-0.019856	-0.123824
-0.086023	-0.273024	-0.065954	0.119412	0.042515	0.029488	-0.019856	1.000000	0.650653
-0.057748	-0.195970	-0.100090	0.087363	-0.002980	0.079995	-0.123824	0.650653	1.000000



상관관계가 높은 변수만 추가한 모델링

```
1 X = apart[['buildDate', 'm2', 'distance_library', 'crime', 'cctv']]
2 y = apart[['avg_sales']]
```

executed in 14ms, finished 02:55:18 2021-12-09



MAE : 22445.16
MSE : 1293907956.03
RMSE : 35970.93
R2 : 0.39

건축연도, 평수, 도서관과의 거리,
범죄 발생 수, cctv 수 추가 후 회귀 모델 구현



처음 모델보다 모든 수치가
조금이나마 좋아졌다



모든 변수 추가 후 모델 구현

```
1 X = apart.drop(['id', 'lat', 'lng', 'p', 'min_sales', 'max_sales', '구', 'avg_sales'], axis = 1)
2 y = apart[['avg_sales']]
```

executed in 14ms, finished 02:55:17 2021-12-09



Id, 위도, 경도, 가격과 구를 제외한
모든 변수 추가 후 모델 제작

MAE : 21116.95
MSE : 1210401153.89
RMSE : 34790.82
R2 : 0.43



모든 변수가 높은 영향을 끼치는지는
확인할 수 없음

예상한 것 만큼 성능이
좋아지지 않음

시각화에서 확인한 지하철 병원 도서관 중 도서관은
높은 상관관계를 가지고 있던 것을 확인할 수 있음

생각보다 데이터의 오류가
많아 전처리 과정이 어려움

감사합니다

