

# 서울시 범죄 위험을 예측 서비스

캡스톤디자인 중간 발표

F조 : JB조

팀장 : 반수경(221161101)

김민제(219161111)

김성준(219161112)

최원태(219161124)

# 목차

## 1. 팀 소개

## 2. 분석 배경

## 3. 분석 목표

## 4. 추진 전략

Step 1. 탐색

Step 2. 인사이트 탐색

Step 3. EDA

Step 4. 모델링

Step 5. 웹페이지 제작 및 구현

Step 6. 예측 모델 웹 페이지 연동

## 5. 향후 계획

# 1. 팀 소개

반수경

최원태

김성준

김민제

데이터 수집 및 분석  
예측 모델 개발  
웹페이지 연동

데이터 수집 및 전처리  
웹 페이지 제작  
웹페이지 연동

## 2. 분석 배경

- 최근 서울부터 시작하여 전국적으로 묻지마 범죄, 흥기 난동 등이 빈번하게 발생하고 있음
- 사회적으로 큰 이슈가 되고 있어 그 원인을 파악하고 예방하기 위해 "서울시 범죄 위험을 예측 서비스" 를 주제로 선정하게 됨

 뉴스버스 · 2023.07.22.

"남들도 불행하게 만들려고"...신림역 칼부림 30대 진술

서울 신림역 인근 '묻지마 칼부림' 사건 피의자 조모씨가 21일 오후 2시 7분쯤 신림역 인근 상가 골목에서 20대 남성을 흥기로 수 차례 공격한 뒤 골목을 이동하고 있다. (사진=뉴스1 / 골목 CCTV캡처 독자제공) 사건이...



 위키트리 · 2023.07.21.

'신림역 묻지마 칼부림 사건' 그런데 피해자들이 모두...

21일 뉴스1 보도에 따르면 이날 오후 2시 7분쯤 서울 관악구 신림역 인근에서 칼부림 사건이 발생했다. 해당 칼부림 사건으로 인해 피해자 1명이 사망하고 3명이 다쳐 병원 치료 중이다. 경찰 관계자는 뉴스1에 "피의자..."



[단독] 신림역 인근서 칼부림...피해자 4명 병원으로 서울경제 PICK · 2023.07.21. · 네이버뉴스  
신림역 인근서 대낮 '공포의 칼부림'...피해자 4명 ... 한국경제 PICK · 2023.07.21. · 네이버뉴스  
서울 신림역 인근서 무차별 칼부림...다수의 피해자 발생, 긴급 이송 위키트리 · 2023.07.21.  
신림역 인근에서 대낮 칼부림...피해자 4명 병원 이송 서울신문 PICK · 2023.07.21. · 네이버뉴스

관련뉴스 8건 전체보기

 일요신문 · 2023.08.04.

신림 이어 서현역, 고속터미널 미수범까지...'칼부림 전염병' 확...

이런 상황에서 또 한 번 칼부림 사고가 벌어질 위기가 발생했다. 서울 서초구 강남고속버스터미널에서 흥기를 들고 돌아다니던 20대 남성이 경찰에 체포된 것. 8월 4일 오전 10시 39분 즈음 고속버스터미널 경부선 터...



 남도일보 · 2023.08.04.

온라인서 퍼진 '대구 PC방 칼부림·포천터미널 흥기난동' 사실무...

대구경찰청은 "인터넷 커뮤니티에 대구의 한 PC방에서 칼부림이 발생했다는 글이 계속 퍼지고 있지만 사실이 아니다. 발생한 적이 없는 사건이다"고 설명했다. 앞서 이날 서울 강남 고속버스터미널에서 흥기를 들고 ...



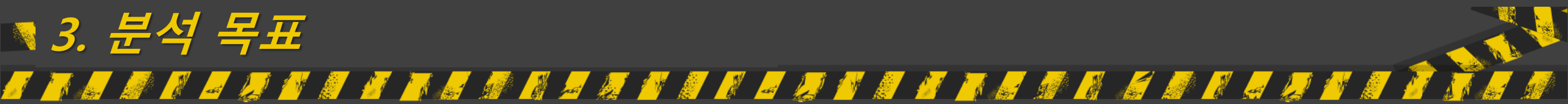
 뉴스1 PICK · 2023.08.04. · 네이버뉴스

이번엔 흥기 들고 자해 시도...고속버스터미널 지하상가서 20대 ...

서상혁 조현기 기자 = 4일 오전 서울 강남 고속터미널 지하 상가에서 흥기를 소지한 남성이 현행범으로 체포됐다. 이 남성은 흥기로 자해를 시도한... 전날(3일) 수인분당선 서현역에서 발생한 불특정 대상을 겨냥한 흥...



고속버스터미널 지하상가서 '흥기' 소지자 체포...불안감 고... 뉴스1 · 2023.08.04. · 네이버뉴스

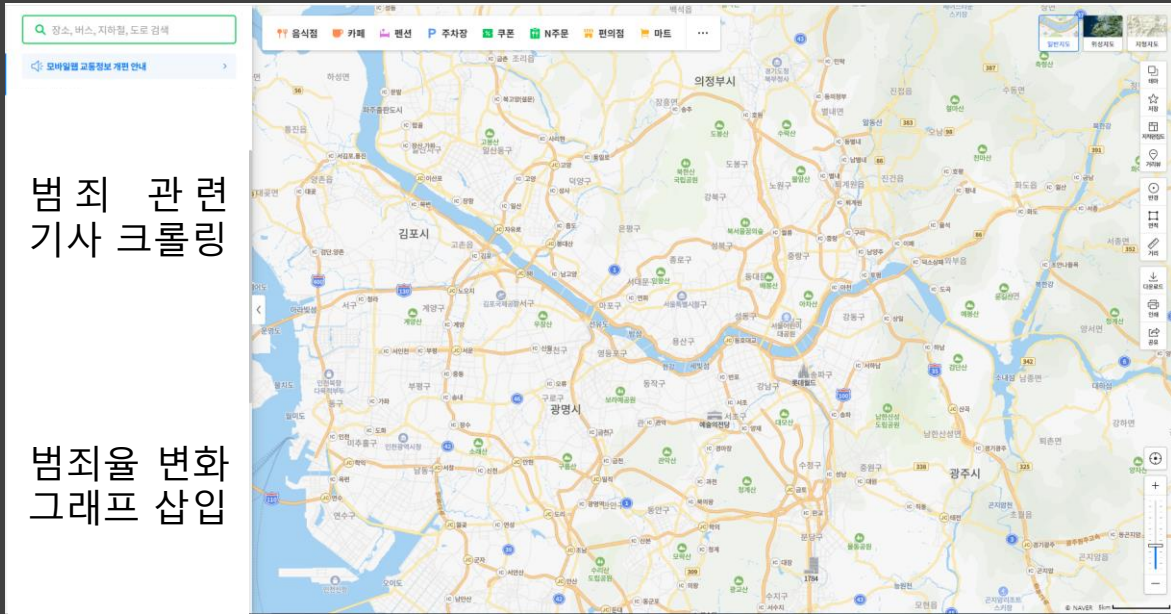


### 3. 분석 목표

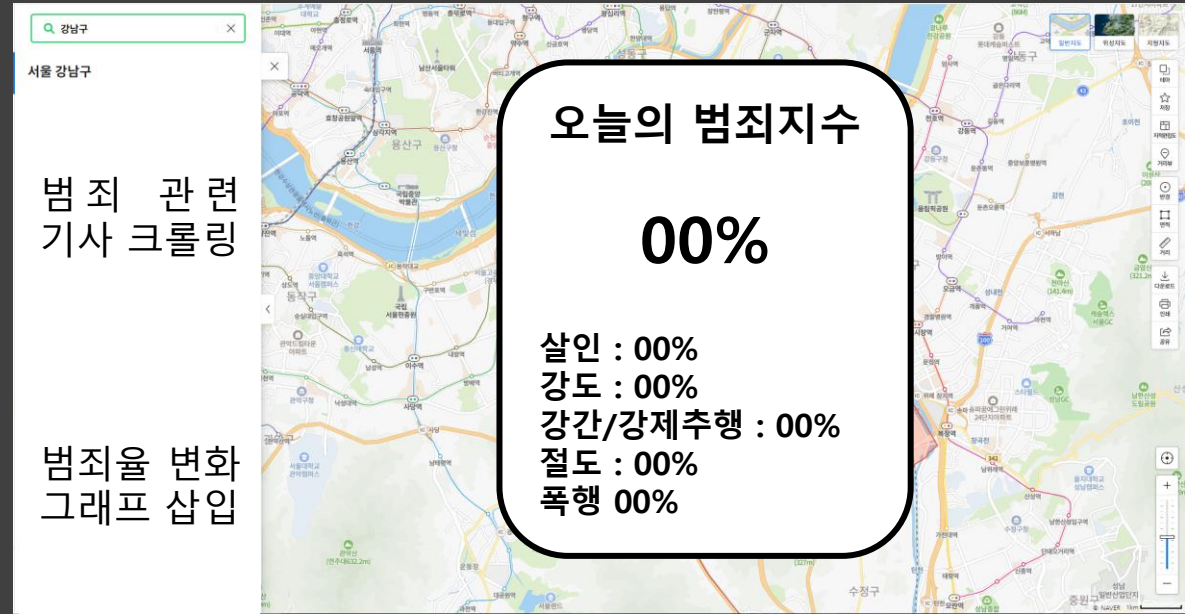
- 정부에서 제공하는 서울시 범죄 정보 데이터에 날씨, 건축물, 인구 데이터 등을 활용하여 범죄율 예측모델 개발
- 데이터들 간 상관관계와 예측 모델을 다양한 방법으로 시각화
- 다양한 머신러닝 프레임워크를 이용해 최적의 예측 모델 개발
- Java Script를 이용해 서울시 범죄율 예측 서비스 웹페이지 구현

# 3. 분석 목표

- 프로젝트 실행 OUTPUT 예시



처음 화면



자치구 검색했을 때 화면

## 4. 추진 전략

### 탐색

- 데이터 소스 및 API 접근 방법 탐구
- 데이터 수집 및 전처리

### 인사이트 발견

- 데이터 시각화를 통해 인사이트 발견

### EDA

- 데이터 요소간 상관관계 파악 및 정규화

### 모델링

- 머신러닝 프레임워크를 이용한 예측 모델 개발

### 웹페이지 제작

- JavaScript를 이용해 웹페이지 제작
- 카카오 맵 API를 이용해 지도 구현

### 예측모델 연동

- Flask를 이용해 예측모델 웹 페이지에 연동



# Step 1. 탐색 - 데이터 수집

- 서울시 범죄 정보, 날씨, 지역, 인구 데이터 수집
- 데이터의 일관성, 정확성, 결측치 등을 확인하고 전처리 진행

데이터 명	데이터 기간	수집 기관
서울시 자치구별 5대 범죄 건수	2020.01 ~ 2022.12	서울특별시 경찰청
날씨 데이터(강수, 풍속, 습도)	2020.01 ~ 2022.12	기상청
주민등록인구	2020.01 ~ 2022.12	서울 열린 데이터 광장
연령대별 1인 가구	2020.01 ~ 2022.12	서울 열린 데이터 광장
건축물	2020.01 ~ 2022.12	건축물 표제부



# Step 1. 탐색 - 데이터 전처리

## • 서울시 자치구별 범죄 데이터

```
# 월별 범죄율 구하기
data['20년_범죄총합'] = data.filter(like='20년').sum(axis=1)
data['21년_범죄총합'] = data.filter(like='21년').sum(axis=1)
data['22년_범죄총합'] = data.filter(like='22년').sum(axis=1)

for month in range(1, 13):
    data[f'20년{month}월_범죄건수'] = (data[f'20년{month}월_살인'] + data[f'20년{month}월_강도'] + data[f'20년{month}월_도둑'] + data[f'20년{month}월_성폭행'] + data[f'20년{month}월_강간'])
    data[f'21년{month}월_범죄건수'] = (data[f'21년{month}월_살인'] + data[f'21년{month}월_강도'] + data[f'21년{month}월_도둑'] + data[f'21년{month}월_성폭행'] + data[f'21년{month}월_강간'])
    data[f'22년{month}월_범죄건수'] = (data[f'22년{month}월_살인'] + data[f'22년{month}월_강도'] + data[f'22년{month}월_도둑'] + data[f'22년{month}월_성폭행'] + data[f'22년{month}월_강간'])

for month in range(1, 13):
    data[f'20년{month}월_범죄율'] = (data[f'20년{month}월_범죄건수'] / data[f'20년_범죄총합']) * 100
    data[f'21년{month}월_범죄율'] = (data[f'21년{month}월_범죄건수'] / data[f'21년_범죄총합']) * 100
    data[f'22년{month}월_범죄율'] = (data[f'22년{month}월_범죄건수'] / data[f'22년_범죄총합']) * 100

# 년도별 범죄율
data['총범죄건수'] = data['20년_범죄총합'] + data['21년_범죄총합'] + data['22년_범죄총합']

data['20년_범죄율'] = (data['20년_범죄총합'] / data['총범죄건수']) * 100
data['21년_범죄율'] = (data['21년_범죄총합'] / data['총범죄건수']) * 100
data['22년_범죄율'] = (data['22년_범죄총합'] / data['총범죄건수']) * 100
```

실행 코드

- **Filter**를 통해 연도별 범죄건수를 더하여 20년~22년 까지의 범죄건수 총합을 구함
- 1월부터 12월까지의 5대 범죄 데이터를 더하여 월별 범죄건수 총합을 구함
- **(월별 범죄건수 / 연도별 범죄총합) \* 100** 식을 이용하여 월별 범죄율을 구함
- 연도별 범죄총합을 더해 3년간의 총범죄건수 구하고, **(연도별 범죄총합 / 총범죄건수) \* 100** 식을 이용해 연도별 범죄율 구함

# Step 1. 탐색 - 데이터 전처리

## • 서울시 날씨 데이터

```
# 데이터에서 필요한 칼럼만 가져오기
new_columns = ['일시', '강수량(mm)']
rain = rain[new_columns]

temp = temp['평균기온(°C)']
wind = wind['평균풍속(m/s)']
humidity = humidity['평균습도(%rh)']

# 행 기준으로 데이터프레임 합치기
df = pd.concat([rain, temp, wind, humidity], axis=1)

# NaN 값 삭제
df = df.dropna()
# 데이터 저장
df.to_csv('날씨데이터총합_20202022.csv', encoding='cp949', index=False)
```

실행 코드

- 강수량, 기온, 풍속, 습도 데이터를 읽어와 필요한 column 만 추출하여 하나의 파일로 합쳐줌
- 새롭게 생성된 파일의 NaN 값을 삭제해 주고 파일 저장
- 상관관계 분석 후 점수가 높지 않다면, 최저·최고 기온, 풍속, 습도 등 다른 칼럼을 사용하여 다시 분석 예정



# Step 1. 탐색 - 데이터 전처리

- 서울시 인구 관련 데이터 (1인 가구)

```
# 1인가구 데이터 남여 구분하여 새로운 데이터프레임 생성
pop_female = pop[pop['성별(1)'] == '여자']
pop_male = pop[pop['성별(1)'] == '남자']

pop_female.rename(columns={'2020_20대': '2020_20대_여자', '2020_30대': '2020_30대_여자',
                             '2021_20대': '2021_20대_여자', '2021_30대': '2021_30대_여자',
                             '2022_20대': '2022_20대_여자', '2022_30대': '2022_30대_여자'})
pop_male.rename(columns={'2020_20대': '2020_20대_남자', '2020_30대': '2020_30대_남자',
                          '2021_20대': '2021_20대_남자', '2021_30대': '2021_30대_남자',
                          '2022_20대': '2022_20대_남자', '2022_30대': '2022_30대_남자'})

del pop_female['성별(1)']
del pop_male['성별(1)']

pop_female.to_csv('1인가구_여자.csv', encoding='cp949', index=False)
pop_male.to_csv('1인가구_남자.csv', encoding='cp949', index=False)
```

- 1인 가구 데이터에서 남/여로 구분하여 새로운 데이터프레임을 생성하고 columns을 원하는 이름으로 변경해줌
- 필요 없는 칼럼은 삭제하고 새로운 파일 생성

실행 코드



# Step 1. 탐색 - 데이터 전처리

- 서울시 인구 관련 데이터 (주민등록인구)

```
pop1 = pd.read_csv('주민등록인구.csv', encoding='utf-8')  
  
pop1 = pop1.drop(0) # 0 번째 행 삭제  
  
pop1.rename(columns={'동별(2)': '자치구', '2020': '2020_세대수', '2020.1': '2020_한국인', '2020.2': '2020_등록외국인',  
                    '2021': '2021_세대수', '2021.1': '2021_한국인', '2021.2': '2021_등록외국인',  
                    '2022': '2022_세대수', '2022.1': '2022_한국인', '2022.2': '2022_등록외국인'})  
  
pop1.to_csv('자치구별 주민등록인구.csv', encoding='cp949', index=False)  
pop1
```

자치구	2020_세대수	2020_한국인	2020_등록외국인	2020_고령자	2021_세대수	2021_한국인	2021_등록외국인
강남구	234872	539231	4824	75096	234233	533042	4824
강동구	196499	459970	4028	70019	201613	462664	4028
강북구	145896	308055	3514	63422	144536	299182	3514
강서구	266982	580185	5716	88984	271175	574315	5716
관악구	274811	495060	14743	78875	276597	485699	14743
광진구	167427	346682	13427	50560	167949	339996	13427
구로구	180531	404408	28080	70198	180929	396754	28080
금천구	114715	231733	16102	39842	117595	230811	16102

- drop() 함수를 이용해 필요 없는 행 삭제
- rename() 함수를 이용해 column을 필요에 맞게 변경하고 새로운 파일로 저장

실행 코드

## Step 2. 인사이트 발견

- Matplotlib 라이브러리를 이용하여 시각화를 진행하고 인사이트 파악

```
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'Malgun Gothic'
# 그래프의 크기를 설정합니다.
plt.figure(figsize=(15, 6))

# 자치구를 x축으로, 20년 범죄총합을 y축으로 하는 선 그래프를 그립니다.
plt.plot(df['자치구'], df['20년_범죄총합'], marker='o', linestyle='-')

# x축 레이블 회전
plt.xticks(rotation=45)

# 그래프의 타이틀과 축 레이블을 추가합니다.
plt.title('자치구별 20년 범죄총합', fontsize=16)
plt.xlabel('자치구', fontsize=14)
plt.ylabel('20년 범죄총합', fontsize=14)

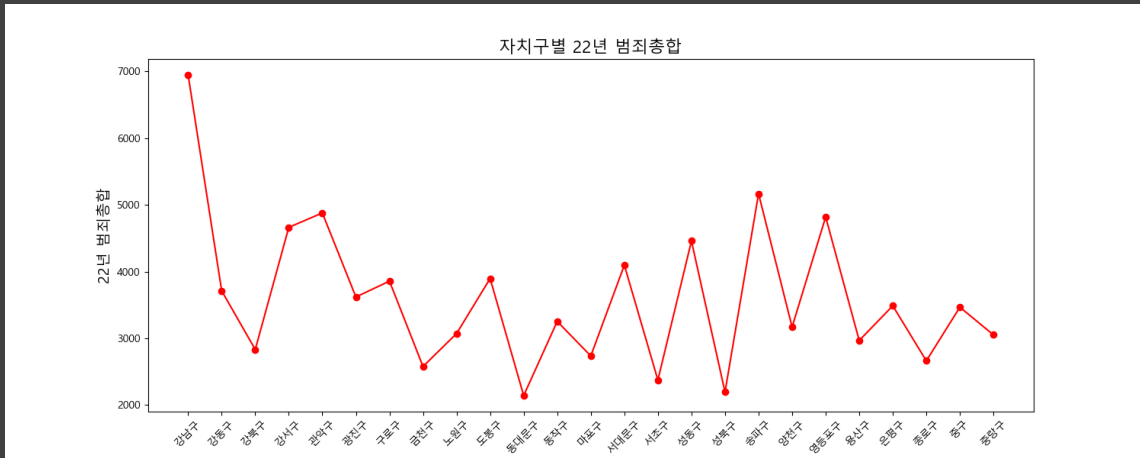
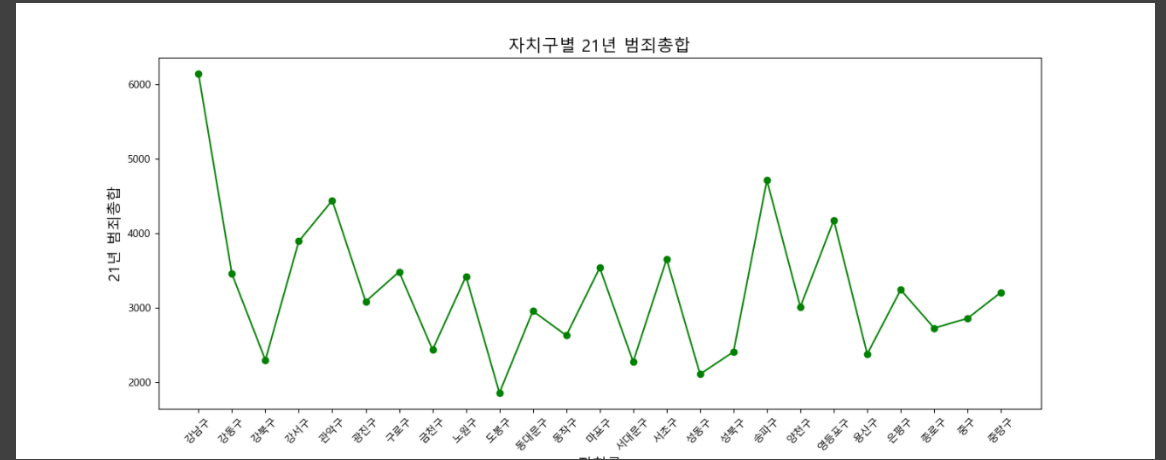
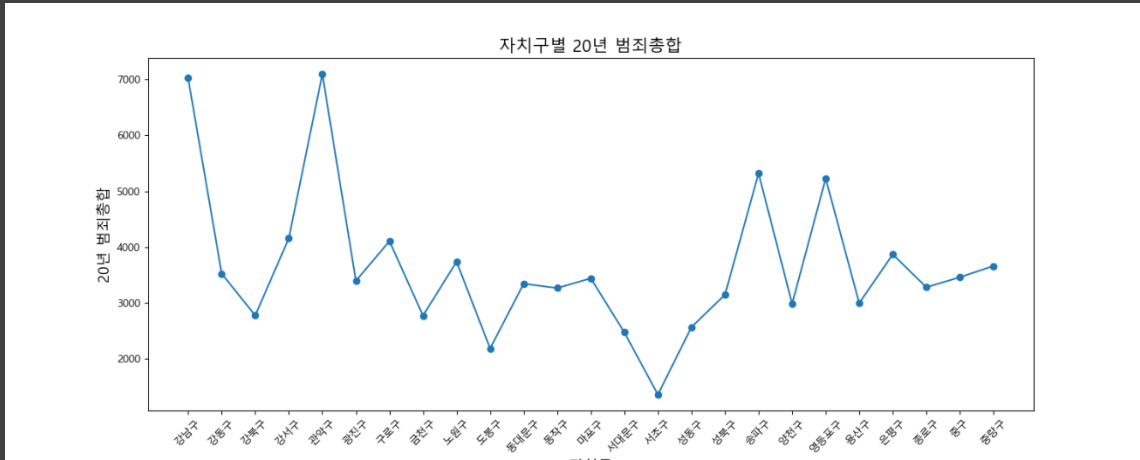
plt.savefig("20년도 자치구별 범죄총합 시각화.png")
# 그래프를 표시합니다.
plt.show()
```

실행 코드

- 2020년 ~ 2022년의 연도별 자치구 범죄 총합 데이터를 시각화하여 어느 지역에 범죄가 많이 일어나는지 파악해볼 수 있음

# Step 2. 인사이트 발견

## • 연도별 자치구 범죄 총합 시각화 결과



- 전체적으로 강남구, 송파구, 영등포구의 범죄수가 높게 나타남
- 2020년 이후 관악구의 범죄수가 많이 떨어졌음

## Step 2. 인사이트 발견

- Matplotlib 라이브러리를 이용하여 시각화를 진행하고 인사이트 파악

```
plt.rcParams['font.family'] = 'Malgun Gothic'
plt.figure(figsize=(15, 6))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.2, hspace=0.4)

# 강남구
plt.subplot(231)
plt.plot(crime_1['Unnamed: 0'], crime_1['강남구'], marker='o', linestyle='-')
plt.title('강남구', fontsize=16)

# 강동구
plt.subplot(232)
plt.plot(crime_1['Unnamed: 0'], crime_1['강동구'], marker='o', linestyle='-')
plt.title('강동구', fontsize=16)

# 강북구
plt.subplot(233)
plt.plot(crime_1['Unnamed: 0'], crime_1['강북구'], marker='o', linestyle='-')
plt.title('강북구', fontsize=16)

# 강서구
plt.subplot(234)
plt.plot(crime_1['Unnamed: 0'], crime_1['강서구'], marker='o', linestyle='-')
plt.title('강서구', fontsize=16)
```

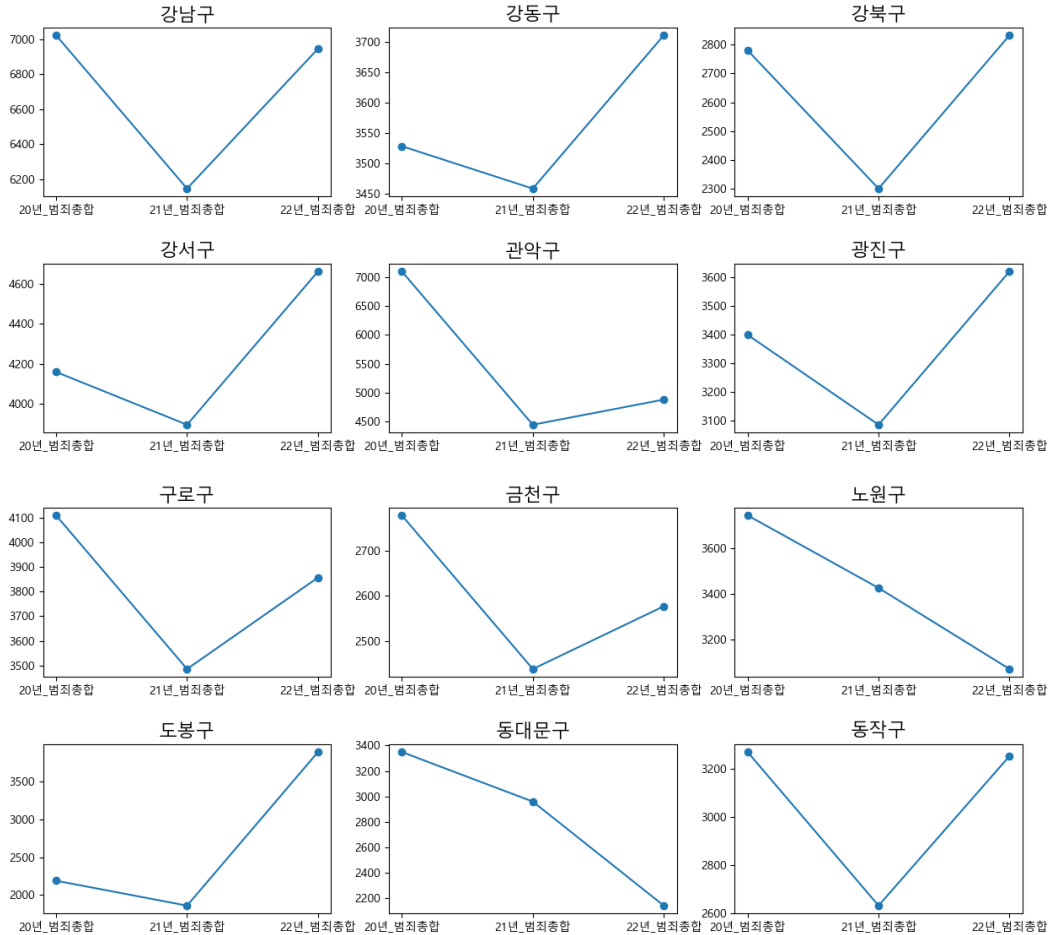
- Subplot을 이용해 자치구별 범죄건수 변화 파악

실행 코드



# Step 2. 인사이트 발견

## • 자치구별 범죄 수 변화



- 대체로 2021년에 범죄가 많이 줄어들었음
- 관악구, 노원구, 동대문구, 성북구, 은평구, 종로구, 중랑구 는 연도가 지날수록 범죄수가 줄어들고 있음
- 반면, 강동구, 강서구, 도봉구, 서대문구, 성동구, 양천구 는 연도가 지날수록 범죄수가 증가하고 있음을 알 수 있음

## Step 3. EDA

- Scikit-learn의 linear model을 통해 범주율과의 상관관계 파악

```
linear_regression = linear_model.LinearRegression()
linear_regression.fit(X = pd.DataFrame(df['2020_20대_남자']), y = df['총범죄건수'])
prediction = linear_regression.predict(X = pd.DataFrame(df['2020_20대_남자']))
print('절편', linear_regression.intercept_)
print('계수', linear_regression.coef_)
print('점수', linear_regression.score(X = pd.DataFrame(df['2020_20대_남자']), y = df['총범죄건수']))
print('-----')
linear_regression.fit(X = pd.DataFrame(df['2020_20대_여자']), y = df['총범죄건수'])
prediction = linear_regression.predict(X = pd.DataFrame(df['2020_20대_여자']))
print('절편', linear_regression.intercept_)
print('계수', linear_regression.coef_)
print('점수', linear_regression.score(X = pd.DataFrame(df['2020_20대_여자']), y = df['총범죄건수']))

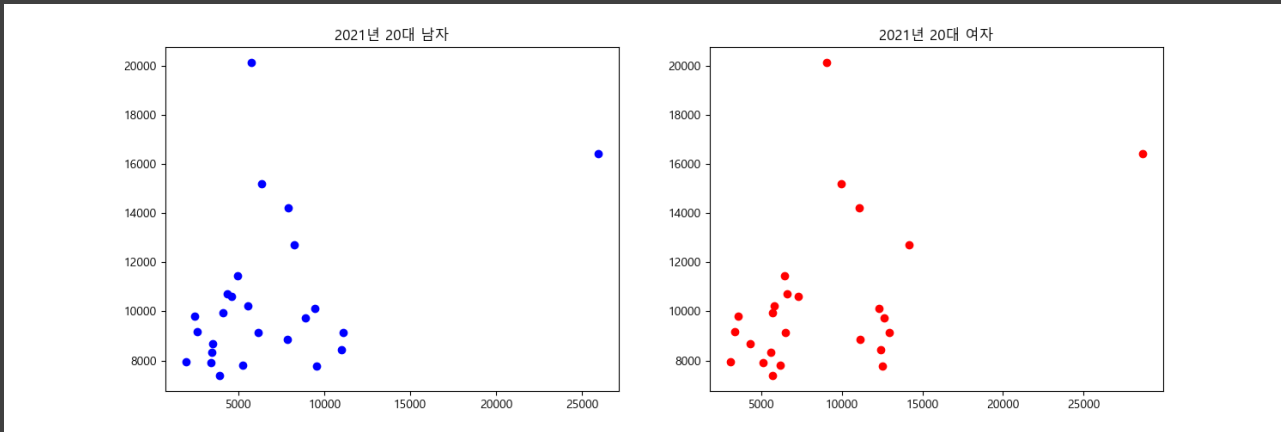
# 시각화
plt.subplot(121)
plt.scatter(df['2020_20대_남자'], df['총범죄건수'], color='b')
plt.title('2020년 20대 남자')

plt.subplot(122)
plt.scatter(df['2020_20대_여자'], df['총범죄건수'], color='r')
plt.title('2020년 20대 여자')
```

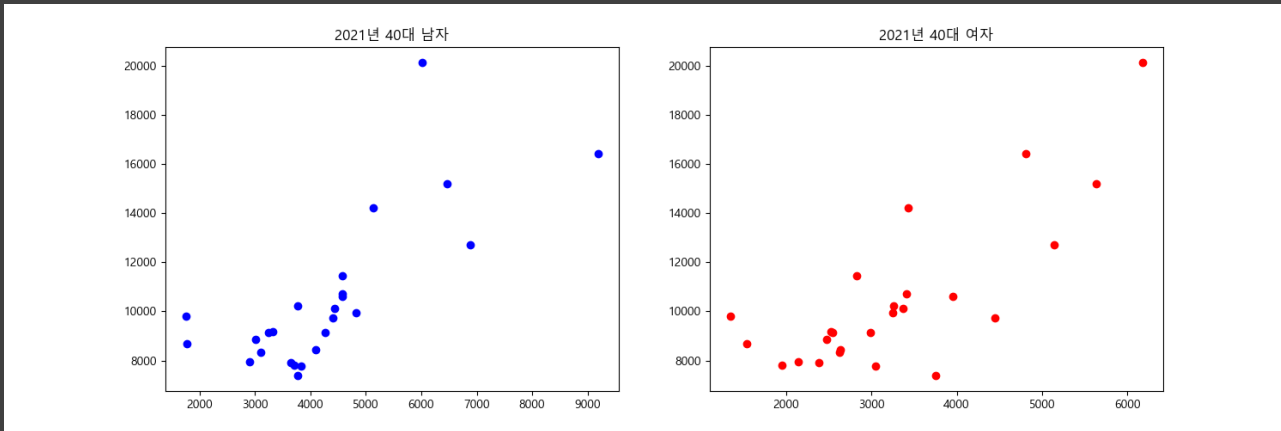
- 20년 ~ 22년, 20대에서 50대까지의 연령대별 남녀의 상관관계를 분석함

# Step 3. EDA

## • 1인 가구 20대에서 50대까지의 연령대별 상관관계 분석



21년 20대 남녀



21년 40대 남녀

## • 1인 가구 연령대별 평균 상관관계 점수

- 20대 남자 - 0.15, 여자 - 0.21
  - 30대 남자 - 0.50, 여자 - 0.63
  - 40대 남자 - 0.56, 여자 - 0.61
  - 50대 남자 - 0.20, 여자 - 0.25
- 
- 30대 남자와 40대 여자가 가장 높은 상관관계를 가지고 있음
  - 20대, 50대 남녀는 상관관계가 높지 않음을 알 수 있음
  - 30대 남자는 가해자, 40대 여자는 피해자가 많을 것으로 추정

## Step 3. EDA

- Scikit-learn의 linear model을 통해 범주율과의 상관관계 파악

```
linear_regression = linear_model.LinearRegression()
linear_regression.fit(X = pd.DataFrame(df['2022_등록외국인']), y = df['총범죄건수'])
prediction = linear_regression.predict(X = pd.DataFrame(df['2022_등록외국인']))
print('절편', linear_regression.intercept_)
print('계수', linear_regression.coef_)
print('점수', linear_regression.score(X = pd.DataFrame(df['2022_등록외국인']), y = df['총범죄건수']))

plt.rcParams['font.family'] = 'Malgun Gothic'
plt.figure(figsize=(15, 5))

plt.scatter(df['2022_등록외국인'], df['총범죄건수'], color='b')
plt.title('2022년 등록외국인')

plt.savefig("2022년 등록외국인 상관관계.png")
```

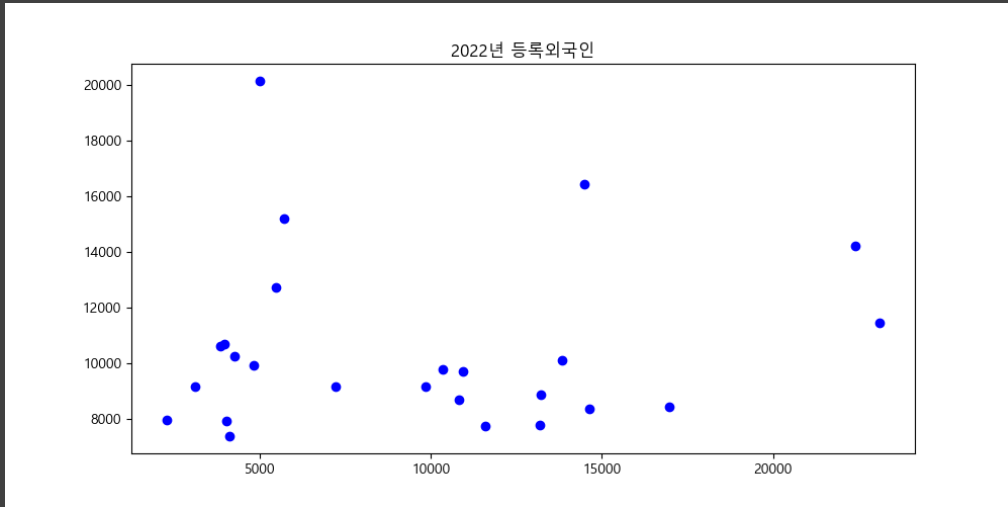
✓ 0.4s

절편 10136.855326087545  
계수 [0.03511376]  
점수 0.004478467918637596

- 20년부터 22년까지 등록외국인, 고령자, 한국인, 세대수 상관관계를 분석함

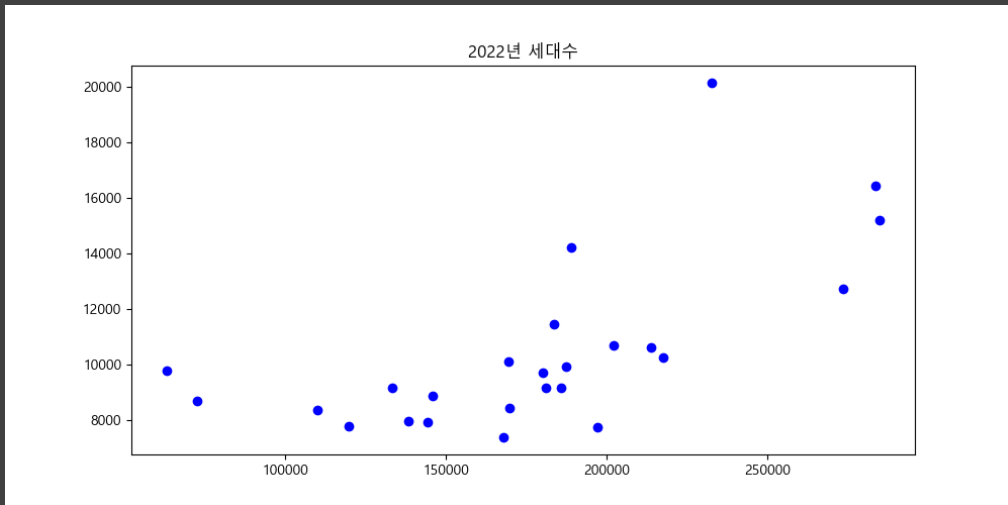
# Step 3. EDA

## • 주민등록인구 상관관계 분석



### • 주민등록인구 평균 상관관계 점수

- 등록외국인수 - 0.01
- 고령자 - 0.23
- 세대수 - 0.43
- 한국인 - 0.35



- 등록외국인수는 상관관계가 전혀 없으므로 분석 대상에서 제외
- 세대수와 한국인수는 고령자에 비해 높은 상관관계를 가짐
- 전체적으로 높은 상관관계를 가지지 않음

## Step 4. 모델링

- SVM, Random Forest, Decision Tree와 같은 머신러닝 프레임워크를 사용하여 예측 모델 개발
- 모델을 학습 및 검증한 후, 정확성 평가를 통해 예측 모델 최적화

## Step 5. 웹 페이지 제작 및 구현

```
<!-- 지도 불러오기 -->
<div id="map" style="width:100%;height:100vh;"></div>

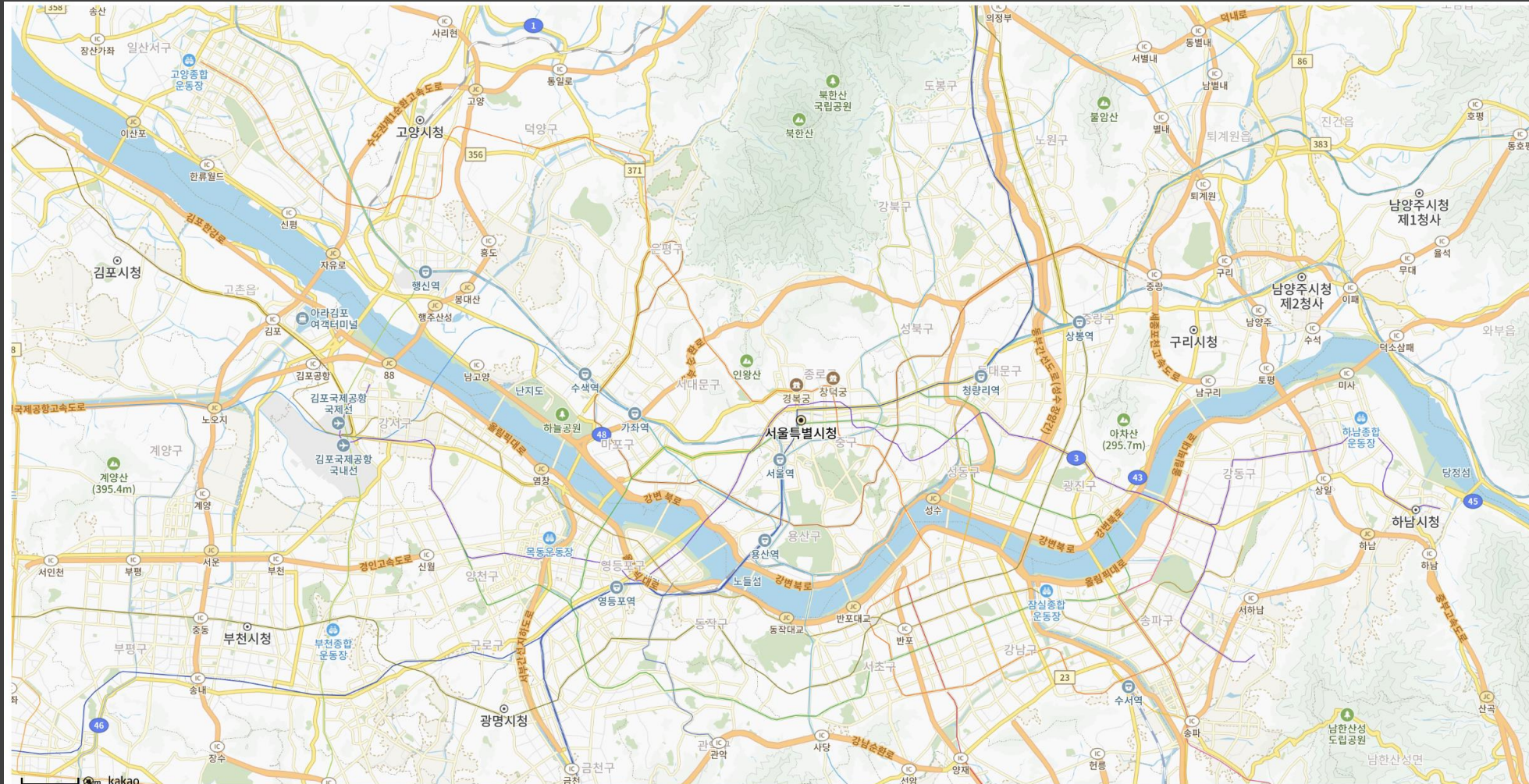
<script src="카카오맵 지도 api 카카오맵 지도 api 카카오맵 지도 api 카카오맵 지도 api 카카오맵 지도 api 카카오맵 지도 api" ></script>
<script>
  var mapContainer = document.getElementById('map'), // 지도를 표시할 div
      mapOption = {
        center: new kakao.maps.LatLng(37.57346, 126.9790), // 지도의 중심좌표
        level: 8, // 지도의 확대 레벨
        mapTypeId : kakao.maps.MapTypeId.ROADMAP // 지도종류
      };

  // 지도를 생성한다
  var map = new kakao.maps.Map(mapContainer, mapOption);
```

- Java Script에서 Kakao Maps API를 사용해 지도를 나타낸다
- 서울의 중심지인 종로를 지도의 중심좌표로 위치
- 25개의 구가 모두 보이도록 지도의 확대 레벨 설정



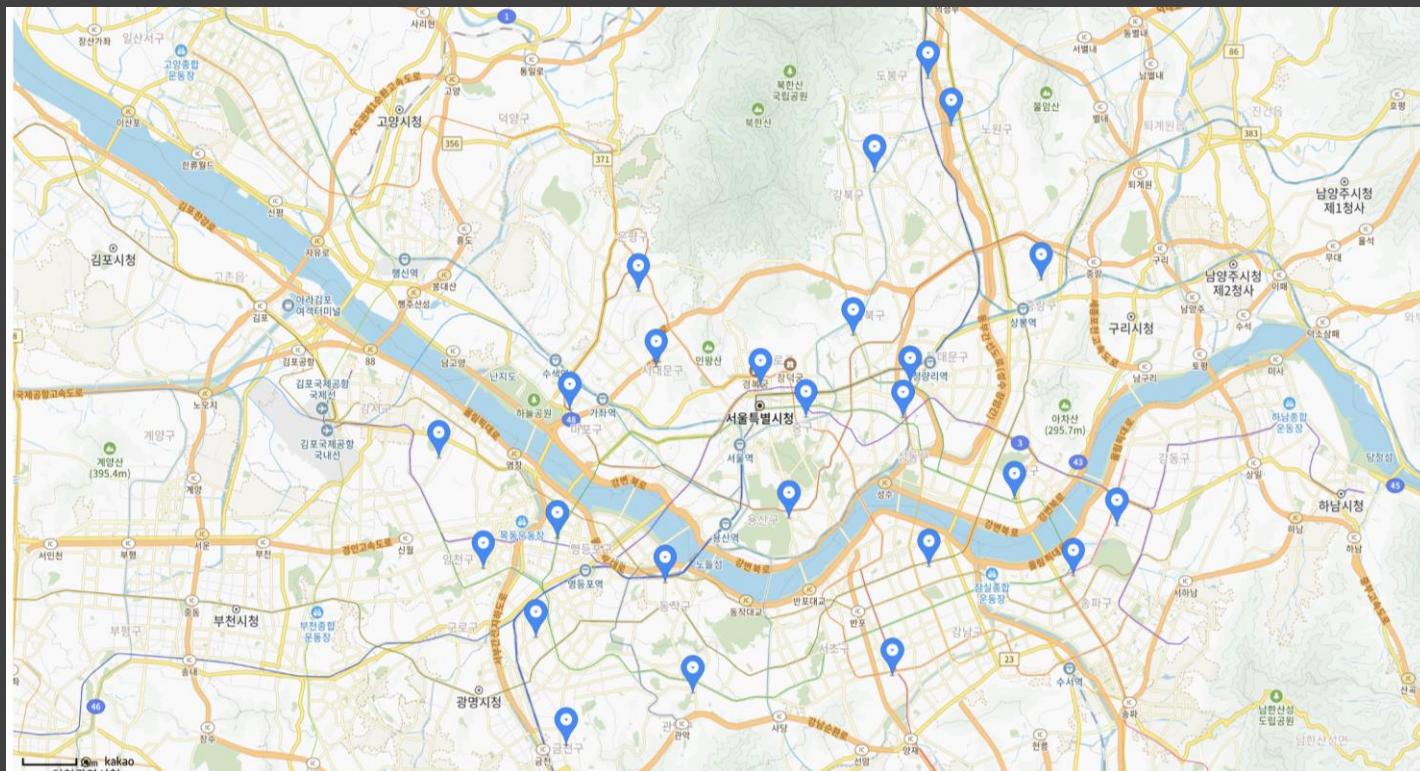
# Step 5. 웹 페이지 제작 및 구현





# Step 5. 웹 페이지 제작 및 구현

- 구현한 지도상에 서울시 자치구별 마크업



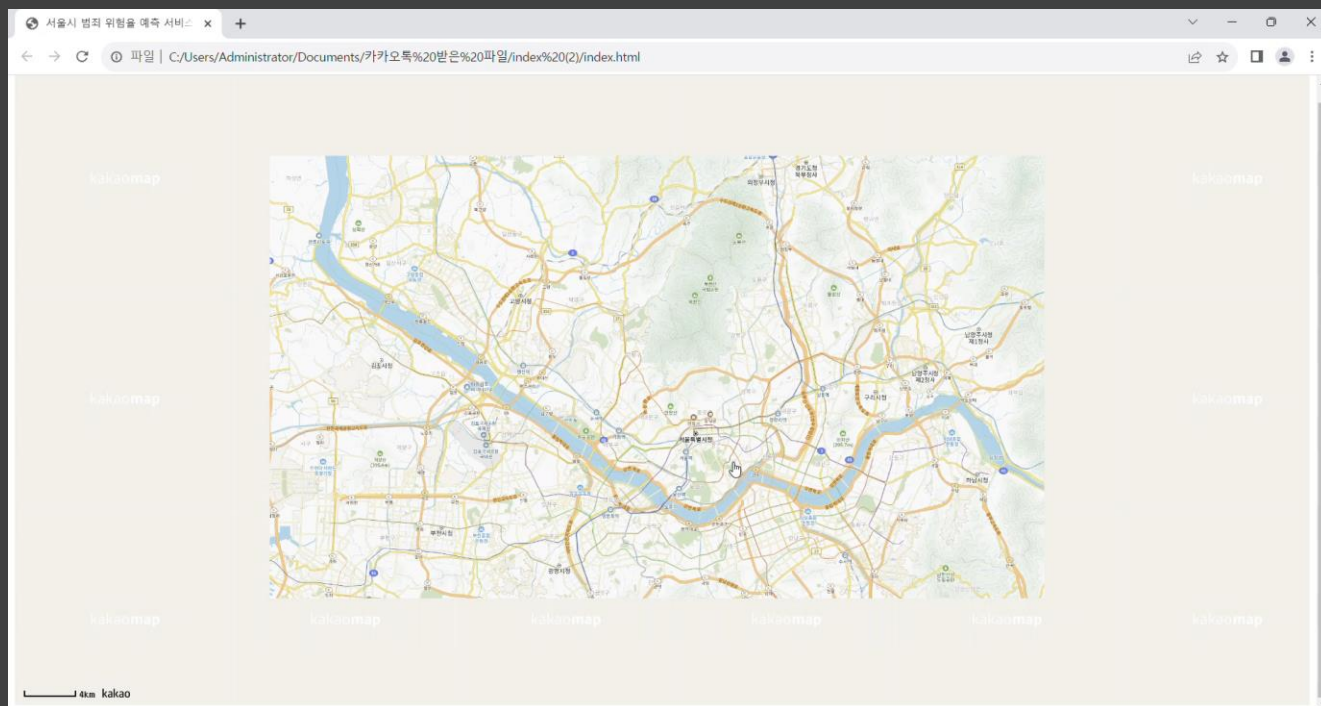
```
var 데이터 = [  
  [37.51757, 127.0474, '<div style="padding: 5px">강남구</div>'],  
  [37.53020, 127.1237, '<div style="padding: 5px">강동구</div>'],  
  [37.63990, 127.0255, '<div style="padding: 5px">강북구</div>'],  
  [37.55115, 126.8484, '<div style="padding: 5px">관서구</div>'],  
  [37.47819, 126.9515, '<div style="padding: 5px">관악구</div>'],  
  [37.53855, 127.0822, '<div style="padding: 5px">광진구</div>'],  
  [37.49562, 126.8878, '<div style="padding: 5px">구로구</div>'],  
  [37.46211, 126.9003, '<div style="padding: 5px">금천구</div>'],  
  [37.65425, 127.0566, '<div style="padding: 5px">노원구</div>'],  
  [37.66891, 127.0470, '<div style="padding: 5px">도봉구</div>'],  
  [37.57433, 127.0398, '<div style="padding: 5px">동대문구</div>'],  
  [37.51256, 126.9401, '<div style="padding: 5px">동작구</div>'],  
  [37.56622, 126.9015, '<div style="padding: 5px">마포구</div>'],  
  [37.57941, 126.9365, '<div style="padding: 5px">서대문구</div>'],  
  [37.48368, 127.0327, '<div style="padding: 5px">서초구</div>'],  
  [37.56355, 127.0369, '<div style="padding: 5px">성동구</div>'],  
  [37.58937, 127.0167, '<div style="padding: 5px">성북구</div>'],  
  [37.51451, 127.1059, '<div style="padding: 5px">송파구</div>'],  
  [37.51699, 126.8666, '<div style="padding: 5px">양천구</div>'],  
  [37.52631, 126.8963, '<div style="padding: 5px">영등포구</div>'],  
  [37.53239, 126.9907, '<div style="padding: 5px">용산구</div>'],  
  [37.60275, 126.9293, '<div style="padding: 5px">은평구</div>'],  
  [37.57346, 126.9790, '<div style="padding: 5px">종로구</div>'],  
  [37.56387, 126.9976, '<div style="padding: 5px">중구</div>'],  
  [37.60632, 127.0932, '<div style="padding: 5px">중랑구</div>'],  
];
```





# Step 5. 웹 페이지 제작 및 구현

- 마커 클러스터러 생성



- 인포윈도우 겹침 현상 예방

```
// 마커 클러스터러를 생성합니다
var clusterer = new kakao.maps.MarkerClusterer({
  map: map, // 마커들을 클러스터로 관리하고 표시할 지도 객체
  averageCenter: true, //마커들의 평균 위치를 클러스터 마커 위치로 설정
  minLevel: 10 // 클러스터 할 최소 지도 레벨
});
```

# Step 5. 웹 페이지 제작 및 구현

- 왼쪽 상단에 사이드바 및 검색기능 구현

```
<style>
.map_wrap, .map_wrap * {margin:0;padding:0;font-family:'Malgun Gothic',dotum,'돋움',sans-serif;font-size:1em;}
.map_wrap a, .map_wrap a:hover, .map_wrap a:active{color:□#000;text-decoration:none;}
.map_wrap {position:relative;width:100%;height:1000px;}
#menu_wrap {position:absolute;top:0;left:0;bottom:0;width:250px;margin:10px 0 30px 10px;padding:5px 0 0 10px;}
.bg_white {background:□#fff;}
#menu_wrap hr {display:block;height:1px;border:0;border-top:2px solid □#5F5F5F;margin:3px 0 0 10px;}
#menu_wrap .option{text-align:center;}
#menu_wrap .option p {margin:10px 0;}
#menu_wrap .option button {margin-left:5px;}
#placesList li {list-style:none;}
#placesList .item {position:relative;border-bottom:1px solid □#888;overflow:hidden;cursor:pointer;}
#placesList .item span {display:block;margin-top:4px;}
#placesList .item h5, #placesList .item .info {text-overflow:ellipsis;overflow:hidden;white-space:nowrap;}
#placesList .item .info{padding:10px 0 10px 55px;}
#placesList .info .gray {color:□#8a8a8a;}
#placesList .info .jibun {padding-left:26px;background:url(https://t1.daumcdn.net/localimg/localization/2013/201302/2013020010.png);width:1em;}
#placesList .info .tel {color:□#009900;}
#placesList .item .markerbg {float:left;position:absolute;width:36px;height:37px;margin:10px 0 0 10px;}
#pagination {margin:10px auto;text-align:center;}
#pagination a {display:inline-block;margin-right:10px;}
#pagination .on {font-weight:bold;cursor:default;color:□#777;}
</style>
```

```
<!--사이드바-->
<div class="map_wrap">
  <div id="map" style="width:100%;height:100vh;position:relative;overflow:hidden;"></div>
  <div id="menu_wrap" class="bg_white">
    <div class="option"></div>
  </div>
</div>
```

- 사이드바 생성

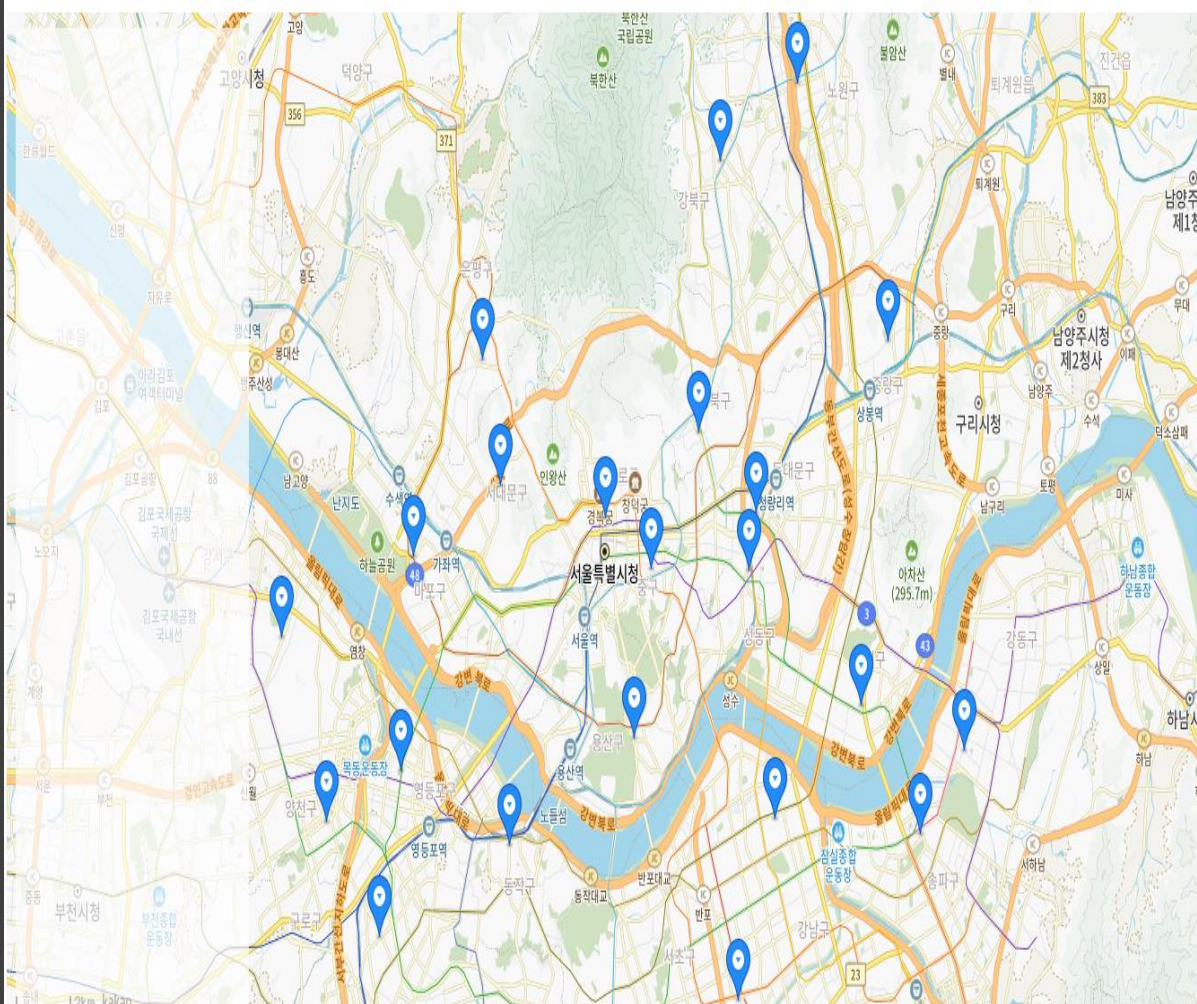
```
<div>
  <form onsubmit="searchPlaces(); return false;">
    키워드 : <input type="text" value="" id="keyword" size="15">
    <button type="submit">검색하기</button>
  </form>
</div>
```

- 검색기능 구현

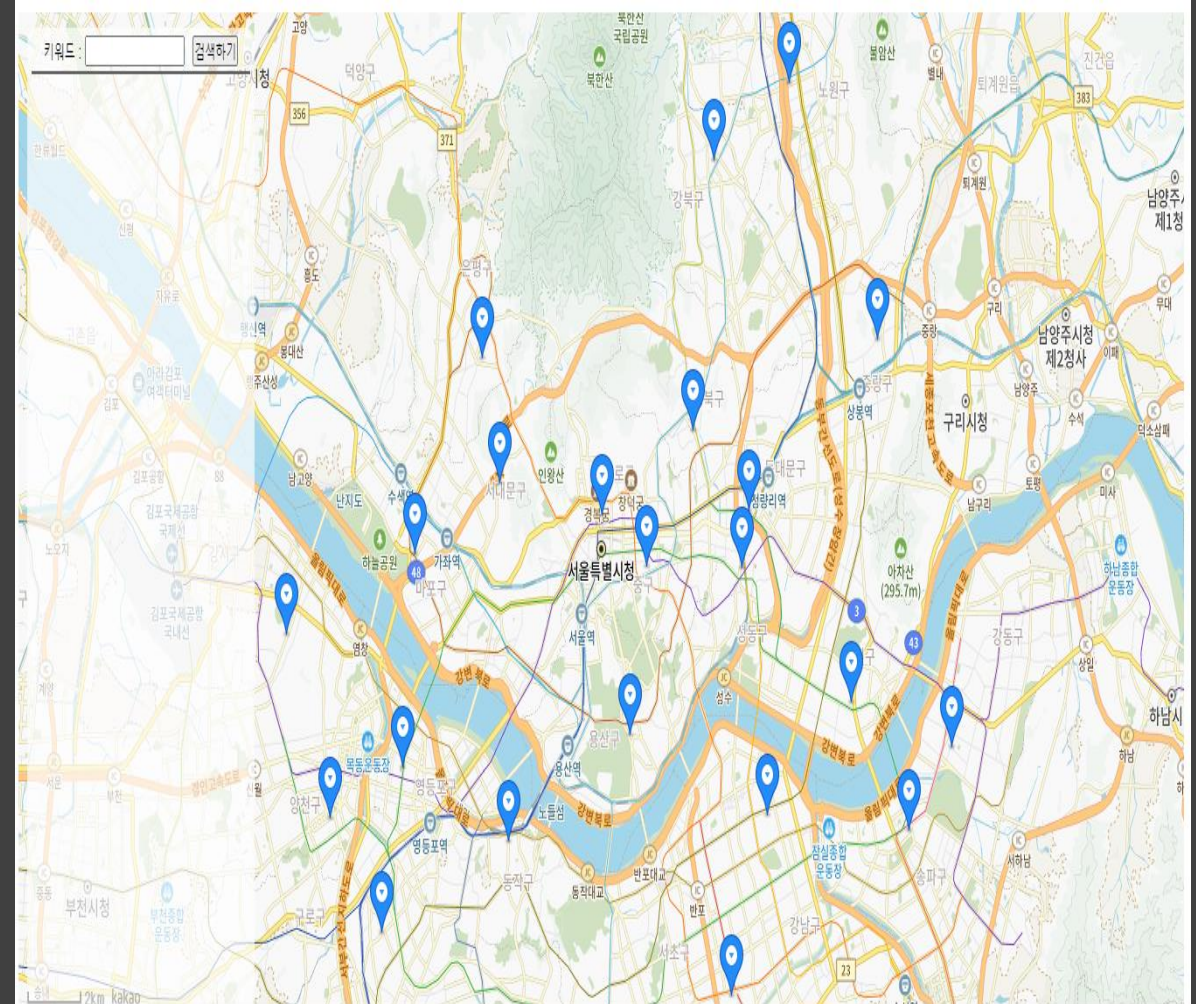


# Step 5. 웹 페이지 제작 및 구현

## • 사이드바 구현



## • 검색기능 구현



## Step 6. 예측 모델 웹 페이지 연동

- Python 프레임워크인 Flask를 사용하여 예측모델을 웹 페이지에 연동



## 5. 향후 계획

- 탐색
  - 건축물 관련 데이터 전처리
- EDA
  - 날씨, 건축물 데이터 상관관계 분석
- 모델링
- 웹페이지 구현
  - 웹 크롤링 : 범죄관련 기사
- 예측 모델 연동



감사합니다.