6팀

오종민, 장대청, 김효진, 박규리

1. 데이터셋 설명

데이터 선정

전국도서관표준데이터 : DATA 공공데이터포털 .GO.KR

데이터 품질(Data Quality)

최신성	정확성	친숙함
마지막 업데이트 : 12.29.20	객관적 정보	처리 과정을 이해하기 쉬운 코드 누구나 발표를 들으며 추가하고 싶은 기능 구상 가능

df.columns
Index(
['도서관명', '시도명', '시군구명', '도서관유형',

'휴관일', '평일운영시작시각', '평일운영종료시각',
'토요일운영시작시각', '토요일운영종료시각',
'공휴일운영시작시각', '공휴일운영종료시각',

'열람좌석수', '자료수(도서)', '자료수(연속간행물)',
'자료수(비도서)',

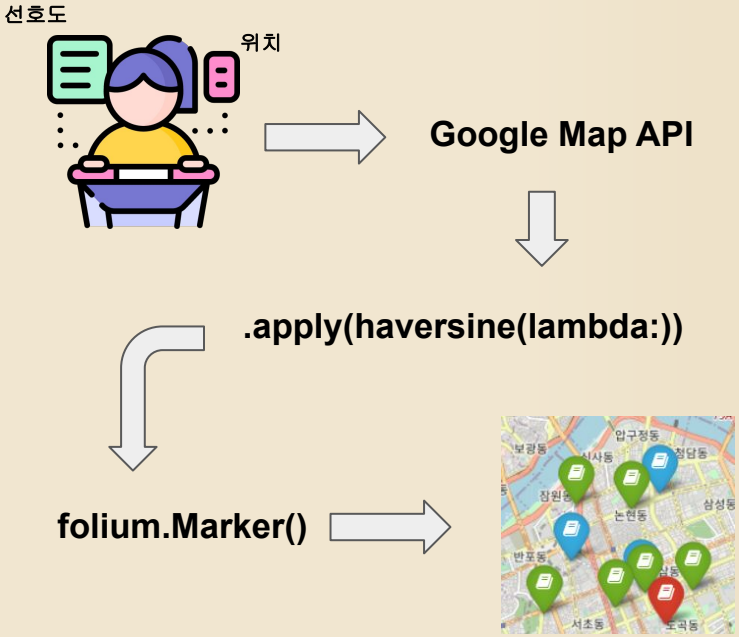
'대출가능권수', '대출가능일수',
'소재지도로명주소', '운영기관명', '도서관전화번호',
'부지면적', '건물면적', '홈페이지주소', '위도', '경도',
'데이터기준일자', '제공기관코드', '제공기관명'],
dtype='object')

df_top_norm.columns
Index(
['열람좌석수', '자료수', '건물면적', '거리', 'score',
'위도', '경도', '자료수(원본)', '열람좌석수(원본)'],
dtype='object')

1. 데이터셋 설명

만들고자 하는 서비스

서울시내 도서관 추천



가장 추천하는 도서관 : 등빛도서관

서비스 기획 이유

서비스 사용 대상자 **多**

: 코로나 바이러스가 종식된다면 모든 연령층에서 사용할 수 있는 서비스

서비스 사용?

입력 : 사용자위치, 선호도

결과 : 가장 추천하는 도서관 출력
사용자 선호도 기반 **Top 10** 도서관 맵표시

2. 전처리

1. 전체 데이터 중 **필요한 열**만 골라냄

'도서관명'
'시도명'
'시군구명'
'열람좌석수'
'자료수(도서)'
'자료수(연속간행물)'
'자료수(비도서)'
'위도'
'경도'
'건물면적'
'도서관유형'

2. **비슷한 유형**의 자료끼리 묶음

자료수(도서), 자료수(연속간행물), 자료수(비도서)

→ '자료수'로 합친 후 삭제

3.'**서울 내 도서관**'만 남기기

→ 서울 대상 서비스 위함

2. 전처리

- **결측치발생**

  위도 36개

  경도 35개

  건물면적 188개

<건물면적 결측치 채우기>

1. 건물면적 값이 null인 데이터프레임 생성
2. '작은도서관' 유형

   (건물면적/열람좌석수)의 중위값 * 열람좌석수

   *188개 중 '작은도서관'이 185개
3. 나머지 3개 → 구글링
4. 기존 데이터프레임에 삽입

```
null_df = df_lib[(df_lib["건물면적"].isnull()) & (df_lib["도서관유형"] == "작은도서관")]
notNull_df = df_lib[df_lib["건물면적"].notnull() & (df_lib['도서관유형'] == "작은도서관")]
buildmean = np.median(notNull_df["건물면적"] / notNull_df["열람좌석수"])
df_lib["건물면적"].fillna(round(null_df["열람좌석수"] * buildmean, 2), inplace = True)
buildmean
```

2. 전처리

<위도, 경도 결측치 채우기>

1. 위도와 경도가 null인 데이터프레임 생성 (loc_null_df)
2. 구글맵 API 접근
→ 도서관 이름으로 좌표 도출
3. 받아온 좌표들을 새 리스트에 append
4. 검색이 불가한 도서관은 error_list로 묶어줌
5. 좌표 열 새로 만들기
→ 유저와 도서관 사이 거리 계산시 편하게 좌표를 보내기 위함
6. 좌표 열에 받아온 리스트 삽입
7. error_list 좌표를 구글링으로 추가
8. 필요없어진 위도, 경도 열 삭제

```
#결측치 채우기 2. 위도와 경도(3)

loc = [] #좌표열에 넣어줄 좌표값
error_list = [] #구글맵내에서 검색이 되지않는 도서관

for lib_name in loc_null_df.index:
    tmp = gmaps.geocode(lib_name) #주소로 검색해 좌표로 돌려주는 API기능

    if bool(tmp): #검색이 된다면
        tmp_loc = tmp[0].get('geometry')
        lat = tmp_loc['location']['lat']
        lng = tmp_loc['location']['lng']
        if ((lat > 0) & (lng > 0)): #좌표가 -값이면 검색이 잘못된것이므로 er
            loc.append(str(lat) + ", " + str(lng))
        else:
            loc.append("0, 0")
            error_list.append(lib_name)
    else: #주소로 검색이 안 된다면
        loc.append("0, 0")
        error_list.append(lib_name)
        pass

#구글맵 API 활용 코드
```

2. 전처리

- ***정규화***

  Min-Max 정규화

  → 열람좌석수, 건물면적, 자료수 정규화

  sklearn 사용

→ 문제점

각 값들 사이에 편차가 심함

최솟값이 0이 아닌 열 존재

거리는 수치가 작을수록 좋으나

나머지는 수치가 클수록 좋음

```
# Min-Max 정규화
scaler = preprocessing.MinMaxScaler()
scaler.fit(df_top_nodist)
df_top_norm = scaler.transform(df_top_nodist)

# 정규화값 결정하기
df_top_norm = pd.DataFrame(df_top_norm)
df_top_norm.index = df_top_nodist.index
df_top_norm.columns = df_top_nodist.columns
df_top_norm['거리'] = df_top_dist
df_top_norm
```

feat. 강사님

3. 순위도출

내 위치 가져오기

구글맵 API를 이용하여
주소를 입력 받아 json을 반환
json 형태에서 위경도만 저장

에러처리

검색이 안될경우
위경도가 음수(-)가 나올경우
다시 한번 묻기

```
loc = input("위치 입력 : ")
#'거리' 구하기 밑작업
df['거리'] = 0.00

#Google Map API 접속
gmaps_key = "Mykey"
gmaps = googlemaps.Client(key = gmaps_key)

#좌표값 받는 함수
def user_location(user_loc):
    while True :
        tmp = gmaps.geocode(user_loc) #주소로 검색해 좌표로 돌려주는 API기능
        if bool(tmp) : #검색이 된다면
            tmp_loc = tmp[0].get('geometry')
            lat = tmp_loc['location']['lat']
            lng = tmp_loc['location']['lng']
            if ((lat > 0) & (lng > 0)): #검색성공
                return lat, lng
        else:
            print('정확한 도로명 주소를 남겨주세요')
            user_loc = input("위치 입력 : ")
            continue
```

3. 순위도출

거리 계산

사용자가 입력한 좌표와 도서관 좌표 비교
→ 내 주소와 도서관 사이 거리 계산
→ 새로운 열에 추가.

거리계산 패키지 “haversine” 사용
- 두 개의 위경도 값을 계산하여 거리를 튜플(tuple)로 반환해주는 패키지

* haversine 함수가 처리시간이 오래걸리는 문제

apply , lambda 를 사용 → 처리시간이 대폭 줄어듬.

```
loc = input("위치 입력 : ")
```

```
#입력받은 위치 기반 '거리' 열 구하기
location = loc
loc1 = user_location(location)
df['거리'] = df.apply(lambda row: haversine(loc1, (row['위도'], row['경도'])), axis = 1)
#haversine 함수 재사용 #처리 시간을 줄이기 위한 apply 함수 활용
```

3. 순위도출

거리 계산

사용자 위치와 가장 가까운 도서관 추려내기.

거리 계산 결과를 통해 가까운 순으로 정렬하고
상위 10개만 남기기.

```
#가까운 도서관 상위 10개 뽑기
df_top = df.sort_values(by='거리', axis=0).head(10)
df_top
```

	도서관명	거리
414	즐거운도서관	0.946042
401	역삼2동작은도서관	0.980440
415	행복한도서관	1.365206
400	대치1작은도서관	1.427621
413	대치도서관	1.591153
389	삼성도서관	1.629165
411	정다운도서관	1.639776
417	역삼도서관	1.714317
416	역삼푸른솔도서관	1.737463
410	논현도서관	1.759980

3. 순위도출

선호도 입력

선호도 종류: **열람좌석수 , 자료수 , 건물면적**

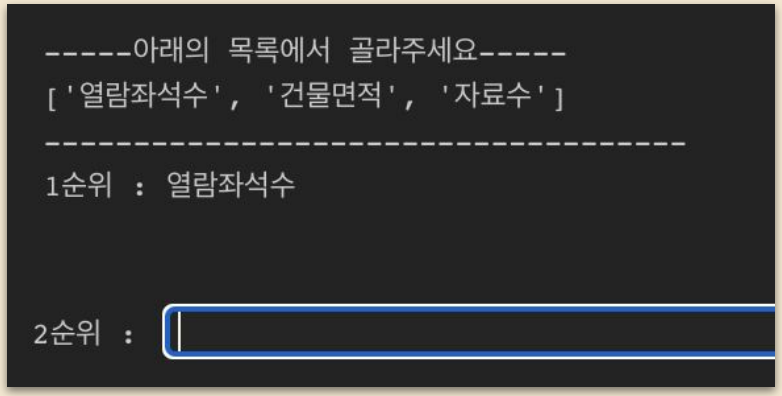
이 중 선호도 순위를 입력 받아

6:3:1 비중으로 최종점수 계산식에 반영.

 : 난 도서관 가서 공부를 할꺼야!

 : 난 도서관에서 많은 책을 읽고 싶어!

 : 난 쾌적한 도서관을 이용하고 싶어!



3. 순위도출

점수계산

정규화 데이터프레임 선호도 반영

최종점수(Score) 계산 후 높은 순 정렬.

이용자 선호도 반영 도서관 랭킹

도서관명	열람좌석수	자료수	건물면적	거리	score
논현도서관	1.000000	0.768423	0.743590	1.759980	8.999192
대치도서관	0.726744	1.000000	1.000000	1.591153	8.360465
역삼푸른솔도서관	0.674419	0.588951	0.976331	1.737463	7.564457
행복한도서관	0.720930	0.723092	0.712032	1.365206	7.184768
역삼도서관	0.563953	0.508344	0.637081	1.714317	5.803308
즐거운도서관	0.441860	0.565301	0.497041	0.946042	4.707588
정다운도서관	0.279070	0.780072	0.461538	1.639776	3.839106
삼성도서관	0.267442	0.243578	0.045365	1.629165	1.984324
대치1작은도서관	0.046512	0.191145	0.000000	1.427621	0.470215
역삼2동작은도서관	0.000000	0.000000	0.096489	0.980440	0.289467

위치 : 선릉역

1순위: 열람좌석수
2순위: 자료수
3순위: 건물면적

4. 시각화

데이터 성질

! 장소 중심 데이터

질문

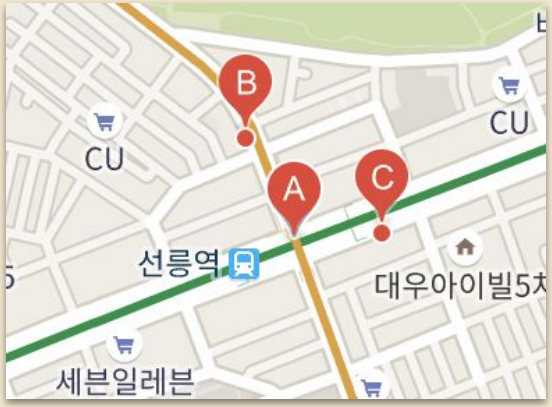
01 좌표값이 있나?

02 지도 하나에서 표현이 다 될까?

03 보기에 좋은가?

→ *다 괜찮다.*

그럼 지도로 만들자



도구 : **folium library**

4. 시각화

1차 시각화

샘플 데이터 구현

map, marker 기능 parameter 이해,

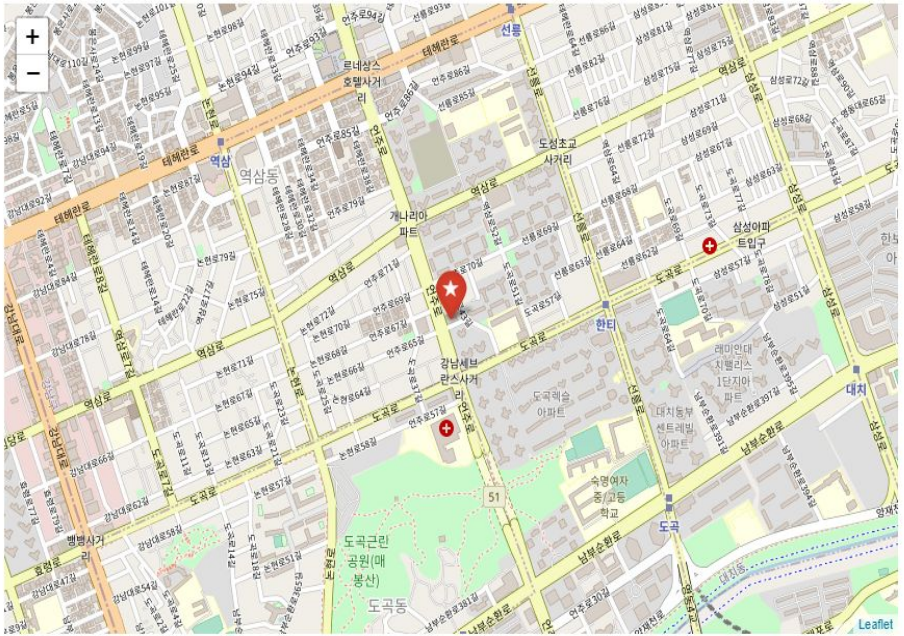
지도 시각화 형태, popup, icon 확인

```
In [15]: import folium

map = folium.Map(location=([df_top_norm['위도'][n], df_top_norm['경도'][n]]), zoom_start=15)

folium.Marker([df_top_norm['위도'][n], df_top_norm['경도'][n]],
              popup = "library info",
              icon= (folium.Icon(icon="star", color="red"))).add_to(map)


map
```



4. 시각화

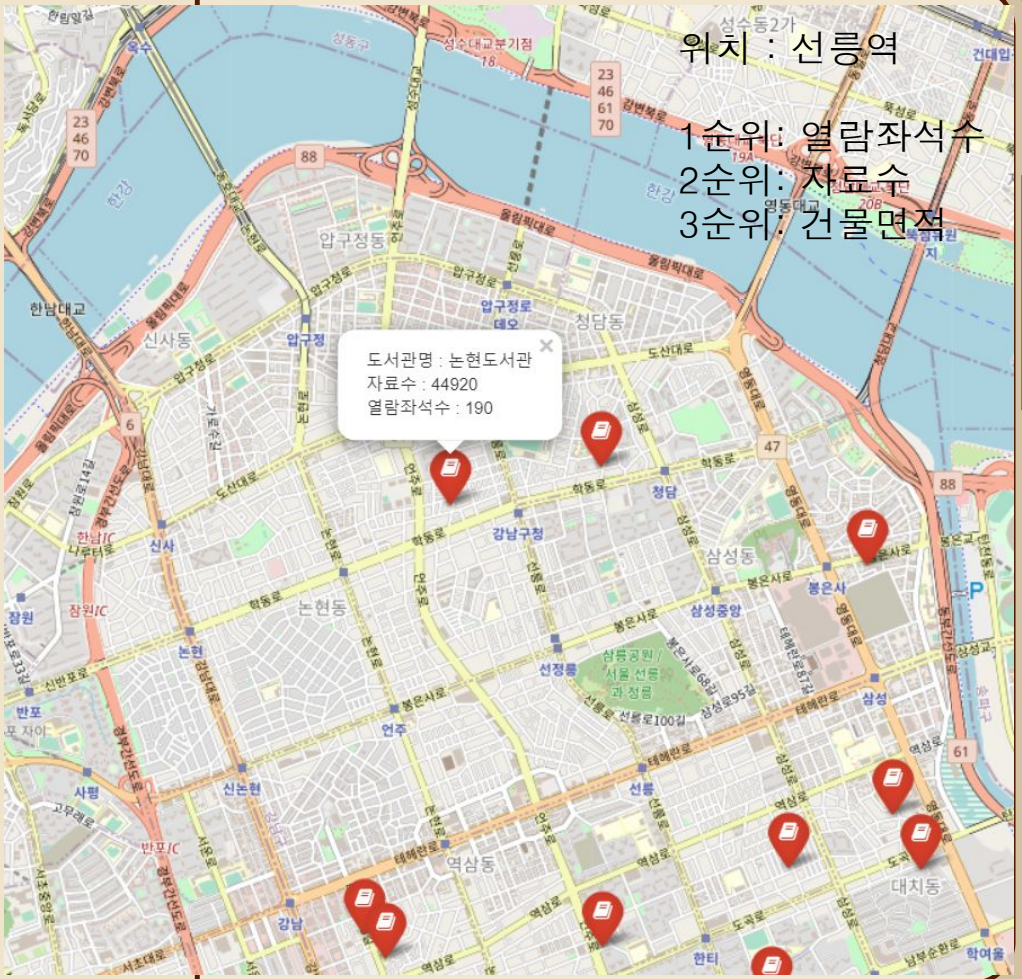
2차 시각화

샘플 데이터가 아닌
데이터 전부를 표현

- popup에 세부 정보 구현

- icon 바꾸기

- popup **한글 깨짐** 현상을
  피하기 위해 html로 피신



4. 시각화

3차 시각화

추천 시스템 강조

- 순위별 색깔 조정

1위 : red
2~4위 : blue
5~10위 : green

- popup에 순위 추가

-- **html** 바로 열기

-- 이미지 **정리**

5. 총평

아쉬운 점

1.지도에 추천순위를 바로 숫자 아이콘으로 표시하지 못한 점

2. 선호도에 따른 가중치를 임의로 제시한 점(이미 가지고 있는 데이터와 연계하는 방법은 없었을까?)

3.코드를 좀 더 간결하게 표현하지 못한 점

4. 시각화 과정에서 folium.Marker를 사용할 때 더 많은 것을 표현하지 못한 점

추가할 점

1. 시각화에서 선발된 도서관 정보를 좀 더 자세하게 제공.

2. 반복적인 부분들은 함수로 묶어서 표현

3. 서비스를 전국 도서관 대상으로 확대

4. 사용자가 이용을 원하는 날짜&시간을 기준으로 갈 수 있는 도서관인지 알려주는 기능 (상세검색)