

# 여행지 추천 시스템

- 김유리 문지유 임세종 -

A yellow suitcase with a black handle and wheels sits on a sandy beach. In the background, there is a blue sea with three sailboats (one with a red sail, two with white sails) and a blue sky with stylized white clouds. A speech bubble is positioned to the right of the suitcase.

나에게 딱 맞는  
여행지는?



1. 주제 및 선정이유

2. 추천시스템 개요

3. 데이터 설명

4. 데이터 전처리 과정

5. 알고리즘 개발

6. API 개발

7. 모델 실습

8. 의의 및 한계

9. 참고 및 출처

# 1. 주제 및 선정이유



코로나로 인해 해외여행을 가지 못하는 사람들을 위하여 자신에게 어울리는 국내 여행지를 추천하며 추천된 특정 여행지의 맛집을 비롯한 관광명소의 정보를 함께 제공하기 위함



비타민 정규 세션 과정에서 배운 추천시스템 모델을 자세히 공부하고 그 중 협업 필터링 방식을 이용하여 직접 우리만의 추천 알고리즘을 개발해보기 위함

## 2. 추천시스템 개요



### 추천시스템이란?

- 특정 시점에서 어떤 고객이 이용할 만한 서비스나 품목을 추천해주는 시스템
- 추천 시스템은 크게 콘텐츠 기반 필터링과 협업 필터링으로 나뉘어지며, 협업 필터링은 최근접 이웃 협업 필터링과 잠재 요인 협업 필터링으로 나뉘어짐

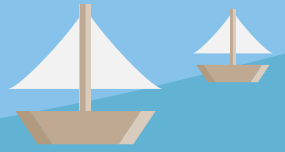
→ 넷플릭스의 추천 시스템 방식으로 잘 알려진 행렬 분해를 이용한 **잠재요인 협업 필터링 기반의** 추천시스템 개발을 목표로 하였음 !





- | 자율 저장 2019_국민여행조사_국내여행 - 제한된 보기 - Excel |          |   |    |         |         |         |         |         |         |    |        |        |        |        |        |        |        |     | 검색 |  |
|-----------------------------------------|----------|---|----|---------|---------|---------|---------|---------|---------|----|--------|--------|--------|--------|--------|--------|--------|-----|----|--|
| 파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 도움말       |          |   |    |         |         |         |         |         |         |    |        |        |        |        |        |        |        |     |    |  |
| V1 MON_AGO_EXP_5                        |          |   |    |         |         |         |         |         |         |    |        |        |        |        |        |        |        |     |    |  |
|                                         | A        | B | C  | D       | E       | F       | G       | H       | I       | J  | K      | L      | M      | N      | O      | P      | Q      | R   |    |  |
| 1                                       | ID       | Q | QQ | MON_EXP | MON_EXP | MON_EXP | MON_EXP | MON_EXP | MON_EXP | S5 | S5A_1  | S5A_2  | S5A_3  | S5A_4  | S5A_5  | S5A_6  | S5A_7  | MON |    |  |
| 2                                       | 11010550 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 3                                       | 11010550 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 4                                       | 11010550 | 2 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 1  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 5                                       | 11010550 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 6                                       | 11010550 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 7                                       | 11010550 | 1 | 0  | 2       | 2       | 2       | 2       | 2       | 1       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 8                                       | 11010550 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 9                                       | 11010550 | 1 | 0  | 2       | 2       | 2       | 1       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 10                                      | 11010550 | 1 | 0  | 2       | 2       | 2       | 2       | 1       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 11                                      | 11010550 | 1 | 0  | 2       | 2       | 2       | 2       | 2       | 2       | 1  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 12                                      | 11010570 | 0 | 0  | 2       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 13                                      | 11010570 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 14                                      | 11010570 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 15                                      | 11010570 | 0 | 0  | 2       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 16                                      | 11010570 | 1 | 0  | 2       | 2       | 2       | 1       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 17                                      | 11010570 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 18                                      | 11010570 | 2 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 19                                      | 11010570 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 20                                      | 11010570 | 1 | 0  | 2       | 2       | 2       | 1       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 21                                      | 11010570 | 0 | 1  | 2       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 22                                      | 11010570 | 2 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! | #NULL! |     |    |  |
| 23                                      | 11010570 | 1 | 0  | 1       | 2       | 2       | 2       | 2       | 2       | 2  | #NULL! | #NULL! |        |        |        |        |        |     |    |  |

### 3. 데이터 설명

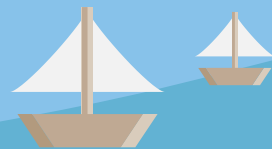


#### SKT data hub의 T-map 검색지 데이터(2020년 기준 월간)

- 매일마다 지역별로 Tmap 어플을 통해 가장 많이 검색된 지명 상위 30개를 추출한 정보로 날짜, 지역(시도), 지역(시군구), 검색지명, 검색지유형1~3, 검색지랭킹 등의 정보를 담고 있으며 총 270만개의 데이터가 존재함

Tmap_2001 - 읽기 전용 - Excel									
파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 도움말									
U12									
	A	B	C	D	E	F	G	H	I
1	일자	지역(시도)	지역(시군구)	검색지명	검색지유형1	검색지유형2	검색지유형3	검색지랭킹(시군구내)	
2	20200101	충남	당진시	왜목마을	AOI	지역	마을	1	
3	20200101	충남	당진시	당진왜목마을	여행/레저	관광명소	축제	2	
4	20200101	충남	당진시	왜목마을	여행/레저	관광명소	해수욕장	3	
5	20200101	충남	당진시	삼교호관광	여행/레저	관광명소	관광단지	4	
6	20200101	충남	당진시	해어름카페	생활편의	카페	커피전문점	5	
7	20200101	충남	당진시	우렁이박사	생활편의	음식점	한식	6	
8	20200101	충남	당진시	행담도휴거	교통편의	교통시설	고속도로휴	7	
9	20200101	충남	당진시	해어름카페	생활편의	카페	카페기타	8	
10	20200101	충남	당진시	현대제철도	기업/단체	업종별기업	제조업	9	
11	20200101	충남	당진시	당진버스터	교통편의	교통시설	버스터미널	10	
12	20200101	충남	당진시	당진시청	공공편의	행정기관	도/시/구/군	11	
13	20200101	충남	당진시	우렁이박사	생활편의	음식점	치킨	12	
14	20200101	충남	당진시	당진IC	교통편의	교통시설	고속도로 I	13	
15	20200101	충남	당진시	모다마을	쇼핑	대형유통점	쇼핑센터	14	
16	20200101	충남	당진시	삼교호합촌	여행/레저	관광명소	공원	15	
17	20200101	충남	당진시	CGV당진	생활편의	문화생활	극장	16	
18	20200101	충남	당진시	아미미술관	생활편의	문화생활	미술관	17	
19	20200101	충남	당진시	롯데마트	쇼핑	대형유통점	할인점	18	
20	20200101	충남	당진시	한진포구	여행/레저	관광명소	포구	19	
21	20200101	충남	당진시	송악IC	교통편의	교통시설	고속도로 I	20	
22	20200101	충남	당진시	당진힐스테	건물/시설	주요건물	아파트브랜	21	
23	20200101	충남	당진시	카페피어	생활편의	카페	커피전문점	22	
24	20200101	충남	당진시	당진1차푸	건물/시설	주요건물	아파트브랜	23	
25	20200101	충남	당진시	당진엠코	건물/시설	주요건물	아파트단지	24	
26	20200101	충남	당진시	독일빵집	생활편의	음식점	분식	25	
27	20200101	충남	당진시	옛날우렁	생활편의	음식점	한식	26	
28	20200101	충남	당진시	창고항	교통편의	교통시설	항구	27	

## 4. 데이터 전처리



### 2019\_국민여행조사\_국내여행

- 행렬 데이터 가공을 위한 열 선정  
: ID, 지역(시군구), 여행 1차 여행유형, 여행활동(21개의 열)
- 여행활동 21가지를 9가지의 테마(ex. 자연관광, 맛집) 로 분류한 후 여행지역별 9가지 테마의 활동비중을 구함

	A	B	C	D	E	F	G	H	I	J
1	지역(시군구)	자연관광	맛집	엔터테인먼트	역사	오락	힐링	쇼핑	문화	체험
2	가평군	0.356335	0.164027	0.03733	0.012443	0.073529	0.281674	0.006787	0.038462	0.029412
3	강릉시	0.316334	0.242627	0.014745	0.038204	0.031501	0.262735	0.021448	0.036193	0.036193
4	거제시	0.345596	0.212932	0.011148	0.049052	0.036789	0.266444	0.020067	0.025641	0.03233
5	경주시	0.288064	0.167639	0.01008	0.181432	0.051459	0.224403	0.011671	0.035013	0.030239
6	공주시	0.34375	0.21875	0.003906	0.136719	0.017578	0.199219	0.019531	0.050781	0.009766
7	광주	0.258197	0.219945	0.008197	0.032787	0.043716	0.217213	0.04235	0.15847	0.019126
8	군산시	0.303473	0.263254	0.010969	0.056673	0.027422	0.206581	0.023766	0.058501	0.04936
9	남해군	0.346479	0.190141	0.012676	0.049296	0.029577	0.283099	0.014085	0.049296	0.025352
10	단양군	0.320346	0.20202	0.027417	0.093795	0.02886	0.227994	0.011544	0.036075	0.051948



지역별 가중치 합이 1이 되게

## 4. 데이터 전처리



- 그 뒤 특정 테마의 비중이 높은 지역이 높은 빈도로 산출되는 문제가 발생하여 각 테마 별 가중치 점수가 높은 순으로 5~1점 부여

### ✓ 완성된 행렬 데이터

	A	B	C	D	E	F	G	H	I	J
1	지역(시군구)	자연관광	맛집	액티비티	역사	오락	힐링	쇼핑	문화	체험
2	가평군	5	1	5	1	5	5	1	2	3
3	강릉시	3	4	4	3	3	5	3	1	4
4	거제시	5	2	2	4	3	5	3	1	4
5	경주시	2	1	2	5	4	3	1	1	4
6	공주시	5	3	1	5	1	1	3	3	1
7	광주	1	3	2	2	4	2	5	5	2
8	군산시	3	5	2	5	2	1	3	3	5
9	남해군	5	1	3	4	2	5	2	2	3
10	단양군	3	1	5	5	2	3	1	1	5
11	담양군	3	4	1	1	2	3	2	5	5



## 4. 데이터 전처리



### SKT data hub의 T-map 검색지 데이터(2020년 기준 월간)

- 개인별 지역 추천시 해당 추천 지역의 가장 많은 검색지를 지닌 명소를 함께 추천하기 위해 사용

```
tourist = tmap[(tmap['지역(시군구)'] == '부산') & (tmap['검색지유형3'].isin(Scene))]  
tourist.head(20)
```

Scene : 자연경관, 관광명소기타, 폭포/계곡, 자연휴양림 등

Unnamed: 0		일자	지역(시도)	지역(시군구)	검색지명	검색지유형1	검색지유형2	검색지유형3	검색지랭킹(시군구내)
894	894	20200105	부산	부산	부산근대역사관	여행/레저	관광명소	관광명소기타	25
1293	1293	20200109	부산	부산	송도해상케이블카베이스테이션	여행/레저	관광명소	관광명소기타	4
5772	5772	20200104	부산	부산	오륙도스카이워크	여행/레저	관광명소	전망대	13
5779	5779	20200104	부산	부산	황령산전망대	여행/레저	관광명소	전망대	20
6303	6303	20200106	부산	부산	태종대전망대	여행/레저	관광명소	전망대	4
6308	6308	20200106	부산	부산	흰여울문화마을	여행/레저	관광명소	관광명소기타	9
8082	8082	20200112	부산	부산	오륙도스카이워크	여행/레저	관광명소	전망대	13
8089	8089	20200112	부산	부산	황령산전망대	여행/레저	관광명소	전망대	20
9751	9751	20200118	부산	부산	송도해상케이블카베이스테이션	여행/레저	관광명소	관광명소기타	2
9771	9771	20200118	부산	부산	송도해상케이블카스카이파크	여행/레저	관광명소	관광명소기타	22
9779	9779	20200118	부산	부산	송도스카이워크	여행/레저	관광명소	관광명소기타	30
10563	10563	20200120	부산	부산	송도해상케이블카베이스테이션	여행/레저	관광명소	관광명소기타	4
12057	12057	20200127	부산	부산	기장해녀촌	여행/레저	관광명소	관광명소기타	28
12256	12256	20200128	부산	부산	오륙도스카이워크	여행/레저	관광명소	전망대	17
12267	12267	20200128	부산	부산	황령산전망대	여행/레저	관광명소	전망대	28

## 5. 알고리즘 개발



$m$ : 총 지역의 개수,  $n$ : 여행 액티비티(여행 가는 이유) 종류의 개수,  $k$ : 사용자가 가본 지역 개수

$A_{n*m}$ : 각 지역의 액티비티 점수

$U_{n*k}$ :  $A$  중 사용자가 가본 지역만 뽑아낸 지역의 액티비티 점수

$R_{k*1}$ : 사용자가 가본 지역의 여행 평점

$M_{n*1} = \frac{U*R}{k}$ 으로서 잠재 요인 행렬들을 곱한 결과의 평균이고,  $N_{n*(m-k)}$ 은 사용자가 가본 지역을 제외한 모든 지역의 액티비티 점수일 때

$$M^T N = [m^t n_1 \quad \cdots \quad m^t n_{m-k}]$$

로  $m-k$ 개의 점수가 최종 결과로 반환이 된다.

이때 가장 높은 점수를 가진 지역 3개를 **추천 지역**으로 반환한다.

이에 대하여 **Tmap 데이터**를 이용하여 각각 자연경관, 맛집, 휴양지, 역사유적, 문화체험지별로 가장 많이 검색된 검색지를 동시에 추천해준다!

# 5. 알고리즘 개발



## 학습 방식 예시

	힐링	액티비티	맛집	평점
서울	①	5	4	②
속초	⑤	2	3	④
여수	⑤	2	3	-

$$(1*2+5*4)/2=11$$

	힐링	액티비티	맛집
서울과 속초의 평균	11	9	10
여수 예측 점수	55	18	30

$$5(\text{여수 힐링}) * 11(\text{사용자 평균 힐링})$$

사용자 기반  
= 액티비티 평균 점수

사용자는 서울과 속초 여행을 가봤다. 서울 여행을 가본 결과 2점이었고, 속초는 4점으로 속초가 더 높았다.

이때, 만족도에는 **여행지 요소**가 큰 작용을 한다고 추측을 할 수 있고 이를 **잠재 요인**으로 설정한다.

사용자 기반 액티비티별 평균을 계산하면 개인이 어떤 활동을 중요하게 생각하는지 점수를 알게 되고, 이를 예측하고 싶은 지역의 액티비티 점수와 함께 곱한다.

따라서 이 사용자는 11점인 **힐링**이 가장 높은 것을 알 수 있으며 따라서 힐링 비율이 높은 **여수**에 높은 점수가 나오게 된다.

## 6. API 개발



- ✓ 추천 시스템의 최종 목표인 실질적인 서비스 이용을 실현하기 위해 API를 개발한다
- ✓ 데이터 분석을 위해 파이썬을 사용하고 이에 맞는 프레임워크 django를 사용한다



### 코드 예시

```
def index(request):  
    return render(request, 'tour/index.html')  
  
@csrf_exempt  
def move(request):  
    time.sleep(2)  
    global location_list, count, ratings  
    location = location_list[count]  
    return render(request, 'tour/question.html', {'  
  
@csrf_exempt  
def q1(request):  
    global location_list, count, ratings, recc_num  
    location = location_list[count]  
    if request.POST['submit_button'] == '네':  
        ans = 'yes'  
        return render(request, 'tour/question.html'  
    elif request.POST['submit_button'] == '아니오':  
        ratings[location] = 0  
        count += 1  
        if count == recc_num:  
            return loading(request)
```

# 7. 모델 실습



## 주피터 창에서 직접 해보기



```
def recommendation():
    data = pd.read_excel('점수표3.xlsx')
    data_real = data.set_index('지역(시군구)')

    print('점수는 0~5점 사이의 정수값으로 체크해주세요!', '\n\n')

    local = list(data['지역(시군구)'].unique())
    travel = random.sample(local, 10)
    my_list=[]
    for i in travel:
        while True:
            Q = input(i+'여행을 해보셨나요?(y/n): ')
            if Q == 'y':
                Score = '0'
                while (Score != '1') and (Score != '2') and (Score != '3') and (Score != '4') and (Score != '5'):
                    Score = input(i+'여행을 평가해주세요!: ')
                    if (Score == '1') or (Score == '2') or (Score == '3') or (Score == '4') or (Score == '5'):
                        Score = int(Score)
                        break
                else:
                    print('제한된 범위 내에서 점수를 입력해주세요!')
            break
        elif Q == 'n':
            Score = 0
            break
        else:
            print('y/n으로 입력하여 주십시오')
    my_list.append(Score)
    print()
    my_score = pd.DataFrame(my_list, index=travel, columns=['점수'])

    total_list=[]
    for i in range(len(my_score)):
        if my_score.iloc[i,0] > 0:
            data_new = data_real[data_real.index==my_score.index[i]]
            data_real = data_real.drop([my_score.index[i]])
            local.remove(my_score.index[i])
            new_list = []
            for j in range(len(data_new.columns)):
```

보령시여행을 해보셨나요?(y/n): n

목포시여행을 해보셨나요?(y/n): n

거제시여행을 해보셨나요?(y/n): n

부산여행을 해보셨나요?(y/n): y

부산여행을 평가해주세요!: 2

태안군여행을 해보셨나요?(y/n): y

태안군여행을 평가해주세요!: 2

대전여행을 해보셨나요?(y/n):

y/n으로 입력하여 주십시오

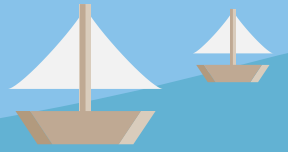
대전여행을 해보셨나요?(y/n): y

대전여행을 평가해주세요!: 3

-----  
당신에게 어울리는 여행지는?

	점수
목포시	280.4
여수시	279.6
제주	276.8

# 7. 모델 실습



HTML 창에서 직접 해보기 ➡

## 추천 장소들

자연경관  
대청호



맛집  
성심당본관



액티비티  
대전오월드



휴양지  
부리공원



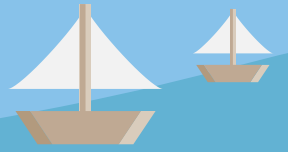
역사유적  
충청남도청구청사







문화체험  
대전아쿠아리움







## 8. 의의 및 한계 Significance



-  동아리 추천시스템 수업에서 배웠던 이론에서 착안하여 나만의 알고리즘을 직접 개발할 수 있어서 뜻 깊었음
-  추천시스템을 구축하는 것은 예측률 및 숫자에 국한되지 않기 때문에 개발자의 의도에 따른 다양한 콘텐츠를 만들어 낼 수 있음
-  다른 분야와 달리 전문 지식 없이도 결과를 확인하고 해석하는 데 있어서 모든 사람이 흥미를 가질 수 있음
-  간단한 API를 만들어 보았지만 조금 더 공부하면서 상품성 있게 개발하는 방법도 연구 가능함

## 8. 의의 및 한계 Limits



-  평점에 대한 데이터가 부족하다 보니 행렬 분해에 있어서 잠재요인을 임의로 지정하여 알고리즘을 개발하였음  
- 가장 큰 문제
-  또한 추천시스템이라는 모델 자체가 평가가 어렵기 때문에 과연 어느 정도의 정확도를 갖는다고 할 수 있는지 애매함
-  잠재요인을 자동으로 설정하는 확률적 경사 하강법을 이용하였더라면 사용자들에게 더 유의미한 정보를 제공할 수 있었으나, 여행지 설문조사의 특성상 여러 여행지에 대한 평가를 하는 데이터를 구하기 어려웠음
-  결국 데이터를 구축하는 것이 가장 어려운 일 중 하나라는 것을 다시 한번 깨달음



## 9. 참고 및 출처



T맵 이용데이터: SKT 빅데이터허브

<https://www.bigdatahub.co.kr/index.do>

2019 국민여행조사 원자료 (국내여행): 관광지식정보시스템

<https://know.tour.go.kr/stat/nRawDataDownloadDis19Re.do>



# 감사합니다

- 김유리 문지유 임세종 -



BITAMIN & TOUR