

# 졸업작품개발보고서

딥러닝 기반 울산 맛집/명소 개인화 추천 시스템 플랫폼



담당교수

윤석훈 교수님

IT융합학과 20201730 박영준

팀원

IT융합학과 20172079 박상원

산업경영공학과 20181794 김창영



## [목 차]

1. 작품명 .....	1
2. 작품개요 .....	1
3. 개발 배경 및 목적 .....	2
4. 졸업작품 내용 .....	4
4-1. 추천 시스템 연구 설계 및 프로세스 .....	5
4-1-1. [1단계] 데이터 수집 .....	5
4-1-2. [2단계] 데이터 전처리 .....	5
4-1-3. [3단계] 데이터 분석 및 EDA .....	5
4-1-4. [4단계] 콘텐츠 기반 추천기 .....	7
4-1-5. [5단계] 협업 추천기 .....	11
4-1-6. [6단계] 하이브리드 추천기 .....	13
4-1-7. 분석 결과 .....	14
4-2. 시스템 구성 .....	10
4-2-1. 개발환경 구성(구성도 포함) .....	15
4-2-2. 소프트웨어 구성(구성도 포함) .....	15
4-3. 기능 .....	18
4-3-1. 기존 유저 추천 .....	18
4-3-2. 신규 유저 추천 .....	18
4-3-3. 로그인/회원가입 .....	19
4-3-4. 장소 마커 표시 .....	20
4-3-5. 추천 장소 정보 제공 .....	20
4-3-6. 추천 장소 리뷰 등록 .....	21

4-4. 사용법 .....	22
4-4-1. 설치방법 .....	22
4-4-2. 회원가입 및 로그인 .....	22
4-4-3. 카테고리 선택 .....	23
4-4-4. 사전 설문 조사 .....	23
4-4-5. 추천 장소 표시 .....	24
4-4-6. 장소 리뷰 등록 .....	24
4-5. 개발환경 .....	25
4-6. 졸업작품 설명 .....	26
4-6-1. 파일/모듈 구성 .....	26
4-6-2. 주요 함수/모듈 별 기능 .....	27
4-6-3. 데이터베이스 구조 .....	31
4-6-4. 주요 기능 흐름도 .....	32
5. 제작 일정 .....	33
6. 향후 연구와 기대효과 .....	34

## 1. 작품명

**국문:** 딥러닝 기반 울산 맛집/명소 개인화 추천 시스템 플랫폼

**영문:** Deep Learning-based Personalized Recommender System  
Platform for Ulsan Restaurants and Attractions

**작품분류:** 웹개발, 인공지능, 추천 시스템, 울산, 여행

**Keyword:** 웹개발, 인공지능, 추천 시스템, 울산, 맛집, 명소, 딥러닝, 개인화, 데이터 수집, 데이터 분석, 사용자 취향 분석, 카카오맵 API, 로그인/회원가입, 마커 표시, 정보 제공

## 2. 작품개요

이 작품은 딥러닝을 기반으로 하여 사용자의 취향과 기호를 분석하여 울산 지역의 맛집과 명소를 개인에게 맞춤형 추천을 통해 제공한다. 사용자는 로그인/회원가입 기능을 통해 개인 계정을 생성하고, 이를 통해 개인화된 추천을 받을 수 있다. 시스템은 네이버, 구글, 카카오 페이지 등을 웹 크롤링하여 얻은 맛집과 명소에 대한 다양한 정보 데이터를 통해 추천을 진행한다.

수집한 데이터는 전처리와 모델링 과정을 거쳐 딥러닝 모델을 구축한다. 이 모델은 사용자의 이전 평가 및 리뷰 데이터를 기반으로 사용자 취향을 학습하고, 새로운 입력에 대해 개인화된 추천을 수행한다. 카카오맵 API를 활용하여 추천된 맛집과 명소를 지도상에 마커로 표시하고, 마커 클릭 시 해당 장소의 정보를 상세히 제공한다.

이 작품은 울산의 문화 및 관광 산업에 도움을 주는 것을 목적으로 하며, 향후에는 시스템의 고도화와 수익형 모델의 구축을 계획하고 있습니다. MLOps를 통해 데이터 수집, 모델 학습, 배포 등의 ML lifecycle을 효과적으로 관리하고, 기업과의 협력과 광고를 통해 수익을 창출할 수 있는 체계를 구축할 예정이다. 이를 통해 지속적인 개선과 발전을 이루어내며 울산의 문화 및 관광 산업에 긍정적인 영향을 미칠 것으로 기대된다.

### 3. 개발 배경 및 목적

울산은 산업과 경제의 중심지로 알려져 있으며, 이로 인해 관광 명소나 맛집으로 유명하지 않다는 인식이 있다. 이에 따라 울산은 관광산업홍보 부족으로 인해 많은 관광객들이 울산을 관광 목적지로 선택하지 않는 문제를 겪고 있다. 또한, 울산은 청년들 사이에서 일자리와 미래에 대한 가능성이 제한적으로 인식되어 청년 유실 문제가 발생하고 있다.

동남지방통계청이 울산·부산·경남지역 청년인구 현황과 이동 등을 분석한 결과 2020년 기준 울산의 15세 이상 39세 이하 청년 인구는 36만 3천 명으로 7년 전인 2013년보다 17.1% 감소했습니다.

이같은 감소율은 부산의 14.2%, 경남의 15.6%에 비해 동남권에서 가장 높은 것입니다.

연령대별로는 10대 인구의 감소율이 36.3%로 가장 높았고 20대 5.2%, 30대 17.8% 등이었습니다.

지출산으로 인구 자체가 줄어든 점을 감안하더라도 이같은 가파른 청년인구 감소의 원인은 타지역 전출때문입니다.

동남권 청년인구의 최다 전출지역은 역시 수도권으로 전체의 60.3%를 차지했고 이어 대구·경북, 충청 등의 순이었습니다.

전출 사유로는 직업 문제가 전체의 63.6%로 가장 많았고 교육,주택 등의 순이었습니다.

-이현진, “울산 청년인구 계속 줄어…60%가 수도권 전출”, KBS NEWS, 2022.02.23

이러한 배경을 고려하여 울산 맛집 명소 추천 시스템을 개발하게 된 것이다. 이 시스템은 울산의 다양한 맛집과 명소를 소개하고 홍보함으로써 울산의 관광 객층을 확대하고 관광산업을 활성화시키기 위한 목적을 가지고 있다. 또한, 개발된 시스템을 통해 울산의 매력적인 장소와 일자리 가능성을 청년들에게 알리고, 청년들이 울산에 머무르고 일할 의향을 가질 수 있도록 도와줌으로써 청년 유실 문제를 해결하고자 한다.

첫째로, 울산은 주로 산업과 관련된 이미지로 알려져 있어서 관광 명소가 부족하다는 평가를 받고 있다. 이는 많은 사람들이 울산을 관광 목적지로 선택하지 않거나 울산에서의 체류 기간을 단축시키는 원인 중 하나이다. 따라서, 우리는 울산의 다양한 맛집과 명소를 추천하고 홍보할 수 있는 시스템을 개발하여 관광객들이 울산을 더욱 흥미롭게 여길 수 있도록 하고자 한다.

둘째로, 많은 사람들이 맛집과 명소를 찾을 때 인터넷 검색 엔진이나 소셜 미디어

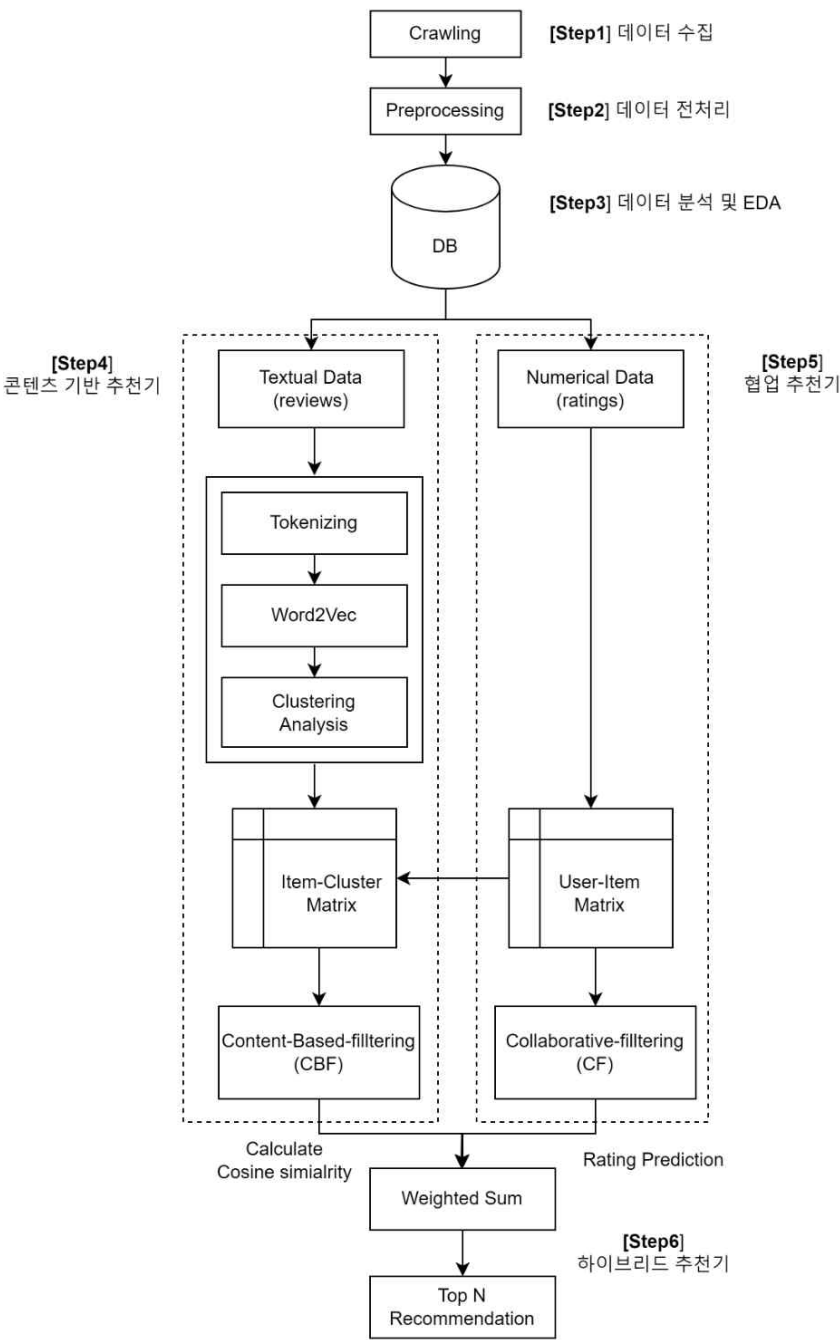
를 이용한다. 그러나 울산에 대한 맛집과 명소 정보는 분산되어 있거나 충분하지 않아서 원하는 정보를 얻기 어려운 경우가 많다. 이에 따라, 울산 맛집 명소 추천 시스템은 사용자들에게 정확하고 다양한 정보를 제공하여 울산의 매력을 소개하고 방문자들이 편리하게 맛집과 명소를 찾을 수 있도록 도와줄 것이다.

마지막으로, 울산 맛집 명소 추천 시스템은 지역 경제의 활성화를 목표로 한다. 울산은 많은 사람들이 찾아오는 주요 산업 도시이지만, 관광 산업은 그에 비해 상대적으로 발전이 미흡하다. 따라서, 울산의 맛집과 명소를 추천하고 홍보하여 관광객 수를 늘리고 지역 경제에 긍정적인 영향을 미치고자 한다. 이를 통해 지역 상권 활성화와 일자리 창출에 기여할 수 있을 것으로 기대된다.

이 작품의 개발 목적은 다음과 같다. 울산 지역의 다양한 맛집과 명소에 대한 정보를 수집하고, 사용자의 취향과 기호를 분석하여 맞춤형 추천을 제공함으로써 사용자들에게 만족스러운 여행 경험을 제공하는 것이다. 개인화된 추천 시스템은 사용자의 다양한 요구와 선호도를 고려하여 최적의 선택지를 제시함으로써 사용자들이 시간과 노력을 절약하고, 더욱 풍부하고 특별한 경험을 할 수 있도록 도와준다. 또한, 울산의 문화 및 관광 산업에 활기를 더하고, 관광객 유치와 마케팅 전략 수립에 기여할 수 있다. 따라서 이 작품은 울산 지역의 발전과 문화/관광 산업 육성을 위한 중요한 역할을 수행할 것으로 기대된다.

4. 졸업작품 내용

4-1. 추천시스템 연구 설계 및 프로세스



[Fig 1] 추천 시스템 프로세스

본 프로젝트는 사용자별 장소에 대한 평점데이터를 기반으로 평점을 예측하는 협업 추천기와 장소별 사용자의 리뷰 데이터를 활용하여 추천해주는 콘텐츠 기반 추천을 조합한 하이브리드 모델을 구현하였다. [Fig 1]은 본 프로젝트의 추천 시스템의 프로세스이다. 각 단계별 프로세스에 대해 순차적으로 설명을 하겠다.

#### 4-1-1. [1단계] 데이터 수집

Kakao map과 Google map에서 울산 음식점/울산 명소를 검색한 결과를 기반으로 각 장소에 대한 기본 정보, 평균평점, url정보와 사용자별 리뷰데이터를 수집한다.

[Table 3] 맛집 데이터 개요

	사용자 수	장소 수	평점 수	리뷰 수
계	24,545명	492곳	54,500개	40,225개

[Table 4] 명소 데이터 개요

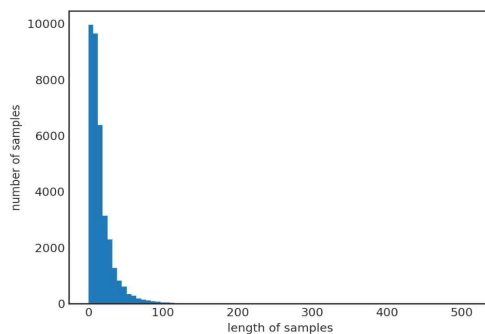
	사용자 수	장소 수	평점 수	리뷰 수
계	9,153명	495곳	15,405개	10,548개

#### 4-1-2. [2단계] 데이터 전처리

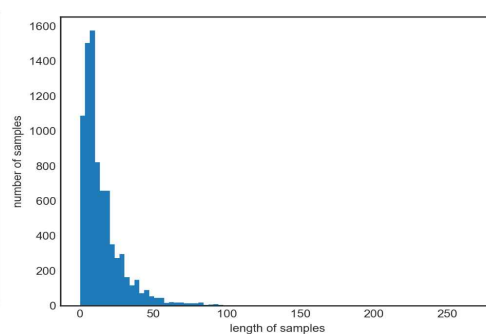
##### 1) 정규표현식 및 불용어 제거

수집한 사용자 리뷰 데이터에는 이모티콘과 특수문자로 이루어진 문구들이 존재했다. 이러한 데이터들을 텍스트분석하기에는 부적합한 형태이므로 다음과 같은 정규방정식 `replace("[^ㄱ-하-ㅣ가-힐]", "")`을 통해 한국어가 아닌 다른 특수문자를 제거한다.

불용어는 텍스트에 일반적으로 자주 나타나는 단어들로, “아”, “휴”, “아이고”, “아이구”, “어”와 같이 문자의 의미를 파악하는 데는 크게 기여하지 않는 단어들을 말한다. 불용어 제거 시 문서 크기 감소, 메모리 절약, 노이즈 제거, 분석 정확성 향상의 효과를 얻을 수 있다. 따라서 676개의 불용어 사전을 정의하여 불용어를 제거하였다.



[Fig 3] 맛집 리뷰 길이



[Fig 4] 명소 리뷰 길이

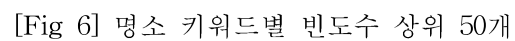
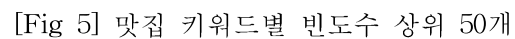
#### 4-1-3. [3단계] 데이터 분석 및 EDA

##### 1) 형태소 분석

형태소 분석은 한국어 문장을 형태소라는 최소 의미 단위로 분리하는 작업이다. 본 프로젝트에서는 한국어 정보처리를 위한 패키지인 KoNLPy의 Mecab을 사용하여 형태소 분석을



## 2) 키워드 분석

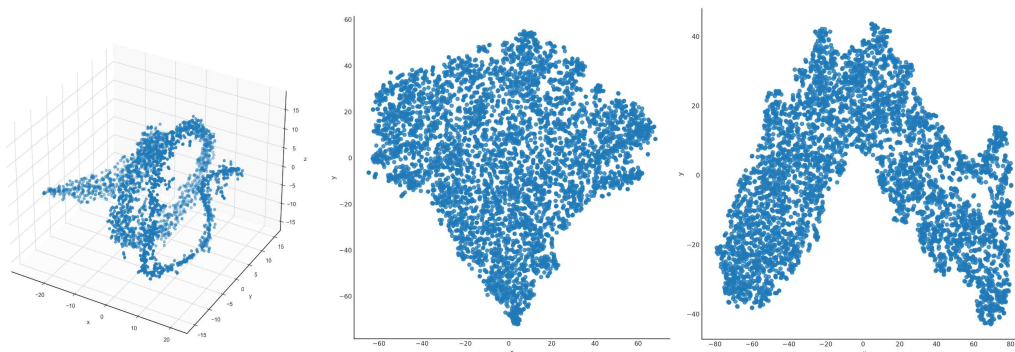


- 7 -

으로 사용하여 중심 단어를 예측하는 과정에서 단어 간의 관계를 학습한다.

Skip-gram은 중심 단어를 입력으로 사용하여 주변 단어들을 예측하는 모델이다. 중심 단어로부터 주변 단어들을 예측하는 과정에서 단어 간의 관계를 학습한다.

맛집 데이터의 임베딩 결과

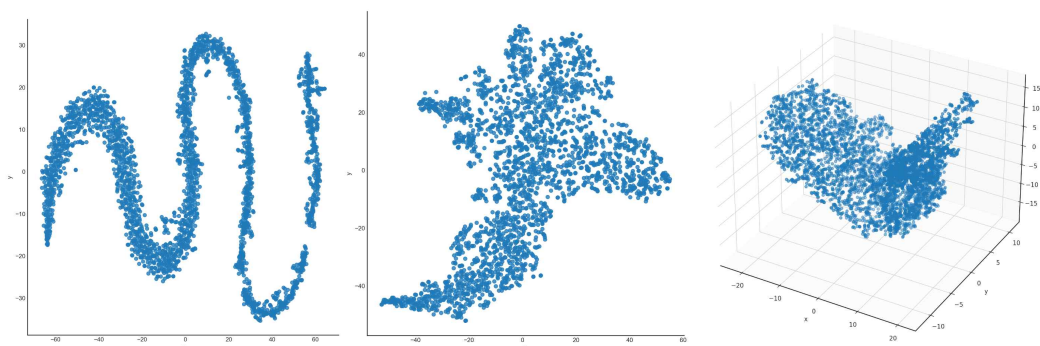


[Fig 12] skip-gram 3차원

[Fig 10] skip-gram 2차원

[Fig 11] CBOW 2차원

명소 데이터 임베딩 결과



[Fig 14] CBOW 2차원

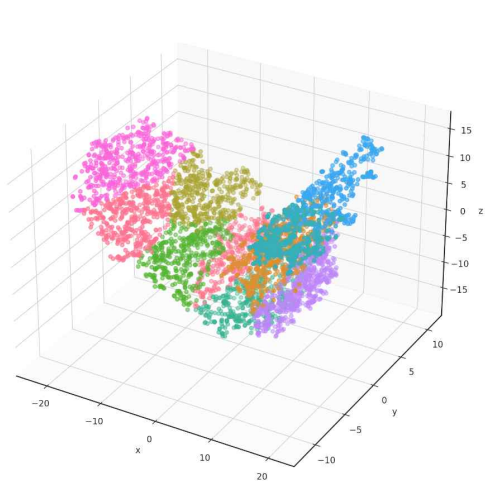
[Fig 13] skip-gram 2차원

[Fig 15] skip-gram 3차원

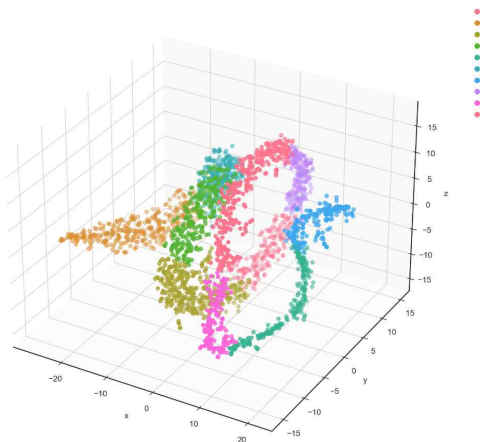
### 3) 군집 분석

앞서 수행한 워드 임베딩 결과를 바탕으로 군집 분석을 수행했다. 특히 계층적 군집화(WARD)와 비계층적 군집화(K-means) 알고리즘을 활용하여 군집 분석을 진행하였다.

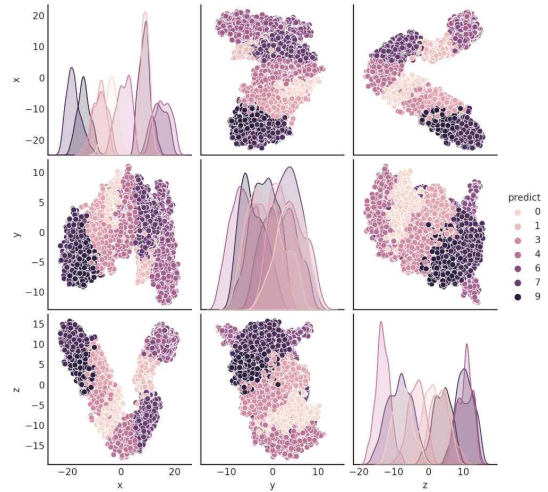
WARD는 대표적인 계층적 군집화 알고리즘 중 하나로, 클러스터 간의 유사성을 계층 구조로 표현하는 방법이다. WARD 알고리즘은 각 데이터 포인트를 개별적인 클러스터로 시작하여 유사성에 기반하여 클러스터를 병합해 나가는 과정 거친다. 해당 알고리즘은 클러스터 간의 거리 측정 방식으로 유사도의 증가량을 기준으로 한다. 처음에는 각 데이터 포인트가 개별 클러스터로 간주되며, 이후에는 가장 가까운 클러스터를 병합하여 새로운 클러스터를 형성한다. 이 과정은 모든 데이터 포인트가 하나의 클러스터로 묶일 때까지 반복된다.



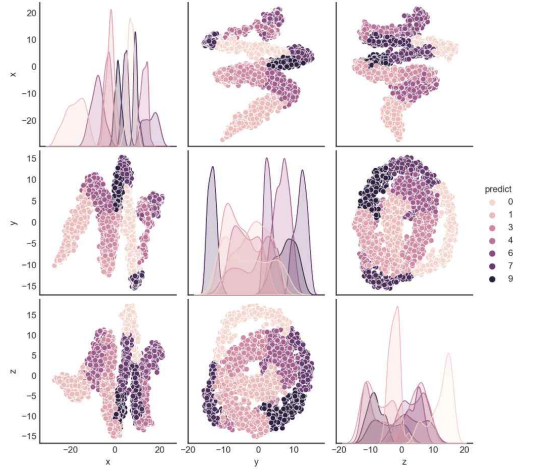
[Fig 16] 맛집 데이터 Kmeans 군집화



[Fig 18] 명소 데이터 WARD 군집화

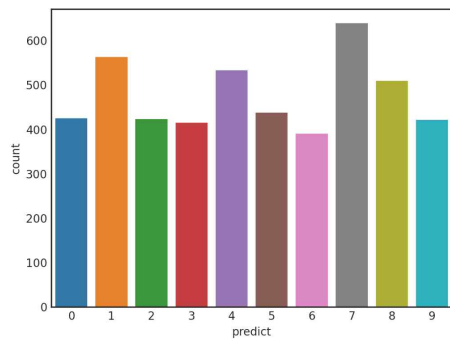


[Fig 17] 군집별 pairplot

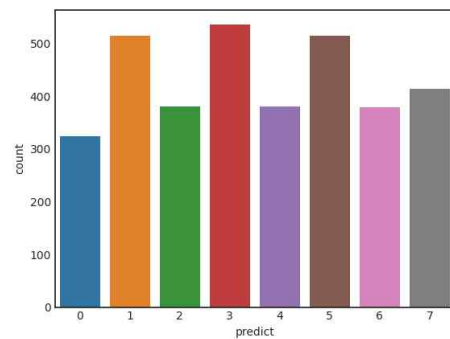


[Fig 19] 군집별 pairplot

K-means는 비계층적 군집화 알고리즘 중 가장 널리 사용되는 방법 중 하나이다. 해당 알고리즘은 사전에 지정한 클러스터의 개수(K)에 따라 데이터를 그룹으로 분류하는 방법이며, 초기에 K개의 중심점을 무작위로 선택 후, 각 데이터 포인트를 가장 가까운 중심점에 할당한다. 그리고 할당된 데이터 포인트들의 평균값을 계산하여 새로운 중심점을 업데이트한다. 이 과정을 반복하여 중심점의 위치가 변하지 않을 때까지 반복한다. 이렇게 최종적으로 각 데이터 포인트가 할당된 클러스터에 속하게 된다



[Fig 20] 맛집 군집 빈도 수



[Fig 21] 명소 군집 빈도 수

[Table 5] 맛집 군집별 형태소

군집 번호	형태소
cluster 0	순천, 노후, 경협, 최적, 접근, 커피숍, 성인, 교육, 벗, ...
cluster 1	곳, 계절, 공기, 연인, 분위기, 명소, 겨울, 조성, 전망, 은하수, ...
cluster 2	바비큐, 자전거길, 도서관, 폭포수, 도시락, 봄날, 어려움, 불법, ...
cluster 3	변화, 고향, 그늘막, 도중, 남녀노소, 음악, 신선, 작년, 대중교통, ...
cluster 4	보람, 아무것, 레이저, 스팟, 화원, 납량, 제주, 명절, 하와이, ...
cluster 5	아기, 대표, 일출, 가보세요, 부산, 중구, 짱, 감상, 동구, 단점, ...
cluster 6	출입구, 트래킹, 간식, 시골, 선바위, 일몰, 장미꽃, ...
cluster 7	곳, 수, 울산, 산책, 길, 물, 공원, 장소, 정원, 시설, ...
cluster 8	산채, 사회, 애견, 속도, 만평, 내어, 섹션, 봄단장, ...
cluster 9	사색, 꽃구경, 친화, 볼, 동반, 낙후, 드라이브, 바닷바람, ...

[Table 6] 명소 군집별 형태소 예시

군집 번호	형태소
cluster 0	적당, 분위기, 수다, 칸막이, 체인, 울산역, 짬, 돈값, ...
cluster 1	흙, 공짜, 닥, 별개, 감, 번창, 특색, 햄버거, 이하, ...
cluster 2	진하, 한가위, 대왕, 농협, 줍니다. 모기, 더러움, ...
cluster 3	디쉬, 로컬, 회정, 혜자, 개인차, 뚜껑, 삼, 주먹밥, 성거, 호주, ...
cluster 4	요인, 산채, 냉, 금, 무거동, 지난번, 걱정, 성의, 우린, ...
cluster 5	맛, 음식, 가격, 고기, 집, 맛집, 울산, 듯, 최고, ...
cluster 6	곳, 친절, 수, 커피, 분위기, 직원, 사람, 주차, 뷰, 주차장, ...
cluster 7	소소, 보장, 초심, 우수, 세팅, 베리, 밥맛, 의외, 갈일, ...
cluster 8	바람, 난무, 군대, 월빙, 픽, 상관, 어버이날, 타워, ...
cluster 9	가자미회, 환장, 효, 용서, 신, 가십, 춘, ...

#### 4) 콘텐츠 기반 필터링(CBF)

콘텐츠 기반 필터링은 아이템 자체의 특성과 사용자의 선호도를 분석하여 유사한 아이템을 추천하는 방법이다. 위 방법은 아이템의 내용, 속성, 특성 등의 데이터를 활용하여 사용자에게 관련성 높은 아이템을 추천해줄 수 있다.

본 프로젝트에서는 아이템 간의 유사성을 측정하기 위한 평가 방법으로 코사인 유사도를 사용한다. 코사인 유사도는 두 벡터 사이의 코사인 각도를 이용하여 유사도를 측정하는 방식이다. 코사인 유사도는 다른 유사도 계산 방식과는 다르게 벡터의 크기에 영향을 받지 않고 정규화 효과가 있기에 불필요한 크기 정보를 제거하고 방향성에 집중할 수 있다는 장점이 있다.

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad \text{----- 식 (1)}$$

식 (1)은 A와 B는 각 두 벡터를 나타내며,  $A \cdot B$ 는 두 벡터의 내적을 의미한다.  $\|A\|$ ,  $\|B\|$ 는 각 벡터 A와 B의 크기를 나타낸다.

[Fig 22]은 장소별 리뷰들을 형태소별로 분리한 토큰들을 군집별로 평균을 집계하여 만든 Item-Cluster Matrix이다. Item-Cluster Matrix를 기반으로 코사인 유사도를 계산한다.

	$c_0$	$c_1$	$c_2$	...	$c_9$
$i_0$	0	0	0	...	1
$i_1$	0	1	...		4
$i_2$	1	16	...	1	0
$i_3$	...				
...					
$i_n$					

[Fig 22] Item-Cluster Matrix

입력으로 넣은 아이템과 코사인 유사도 계산으로 얻은, 즉 가장 유사한 아이템을 상위 K개를 추천한다.

#### 4-1-5. [5단계] 협업 추천기

##### 1) 협업 필터링(CF)

협업 필터링은 사용자들의 과거 행동 정보(Profile)를 기반으로 아이템에 대한 예측과 추천을 수행하는 방법이다. 사용자들 간의 상호작용 데이터를 분석하여 사용자의 취향이나 관심

사를 파악하고, 이를 기반으로 다른 사용자와 비슷한 취향을 가진 사용자들이 좋아할 만한 아이템을 추천하는 방식이다.

	$i_0$	$i_1$	$i_2$	...	$i_n$
$u_0$	4		3	...	
$u_1$		1.5			
$u_2$	1	2	...		
$u_4$	...				
...					
$u_k$				...	

[Fig 23] User-Item Matrix

[Fig 23]은 사용자와 아이템간의 평점을 나타낸 행렬인 User-Item Matrix이며 위 행렬을 데이터로 사용하여 협업 필터링을 통해 사용자가 이용하지 않은 아이템에 대한 평점 예측을 진행한다. 본 프로젝트에서는 KNNwithmeans, SVD, SVD++, NMF, NCF 방법으로 실험을 진행했다.

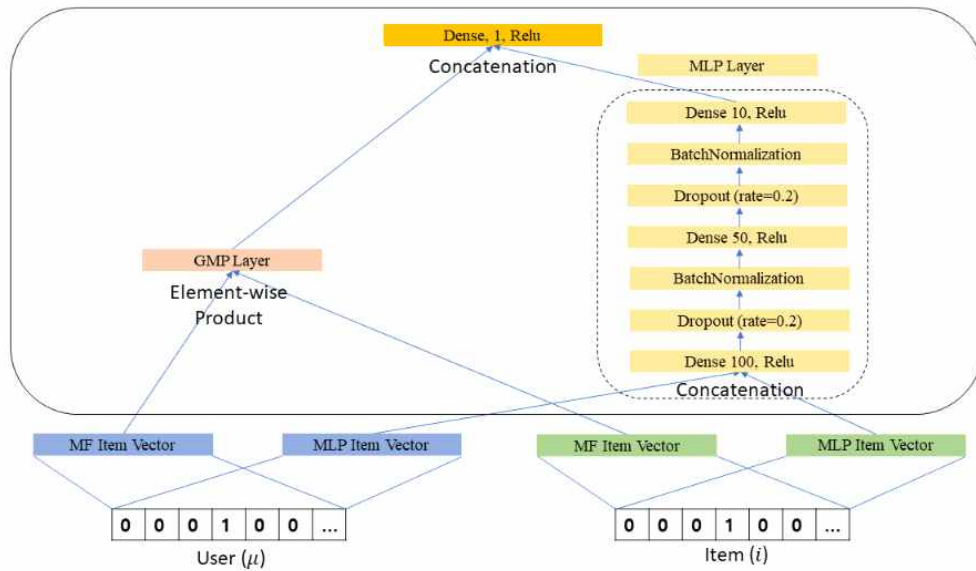
KNNwithmeans는 KNN알고리즘을 사용하면서 평균값을 고려하여 예측하는 방법이다. 다른 방법에 비해 비교적 간단하고 직관적인 방법으로, 구현이 쉽다.

SVD(Singular Value Decomposition)는 행렬분해기반 방법이다. 사용자-아이템 평점 행렬을 저차원의 사용자 특성 행렬과 아이템 특성 행렬로 분해하고, 이를 통해 사용자의 선호도와 아이템의 특성을 추출한다. 추출된 특성을 사용하여 사용자가 아직 평가하지 않은 아이템을 추천한다.

SVD++(Singular Value Decomposition++)은 SVD의 확장 버전이며, 추가적인 암시적 rating이 더해진 방식이다.

NMF(Non-Negative Matrix Factorization)은 SVD와 유사한 행렬 분해 방법이다. 그러나 NMF는 음수가 아닌 행렬에 대해서만 분해를 수행하며, 분해된 행렬들은 음수가 아닌 값을 가진다.

NCF(Neural Collaborative Filtering)은 딥러닝을 기반으로 한 협업 필터링 모델이다. Multi-Layer Perceptron(MLP)와 Generalized Matrix Factorization(GMF)를 결합한 모델이다. GMF는 행렬 분해 방법 중 하나로, 사용자와 아이템 간의 상호작용 행렬을 낮은 차원의 잠재 요인으로 분해한다. MLP는 다층 퍼셉트론으로, 사용자와 아이템 간의 상호작용 예측하기 위해 신경망의 가중치를 학습한다. [Fig 4]는 본 프로젝트의 NCF의 아키텍처이다. 수집한 데이터에 적합하게끔 활성화 함수로 ReLU를 사용하는 Dense layer를 생성하였으며, 추가적으로 Dropout layer와 BatchNormalization layer를 추가하여 MLP부분을 구성하였다.

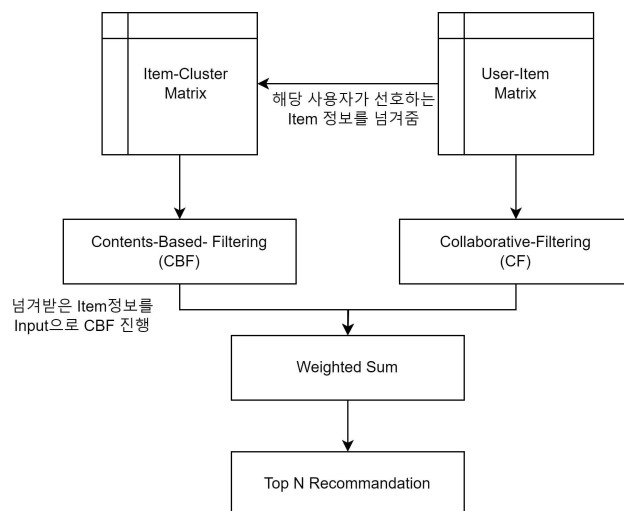


[Fig 24] NCF 모델 아키텍처

#### 4-1-6. [6단계] 하이브리드 추천기

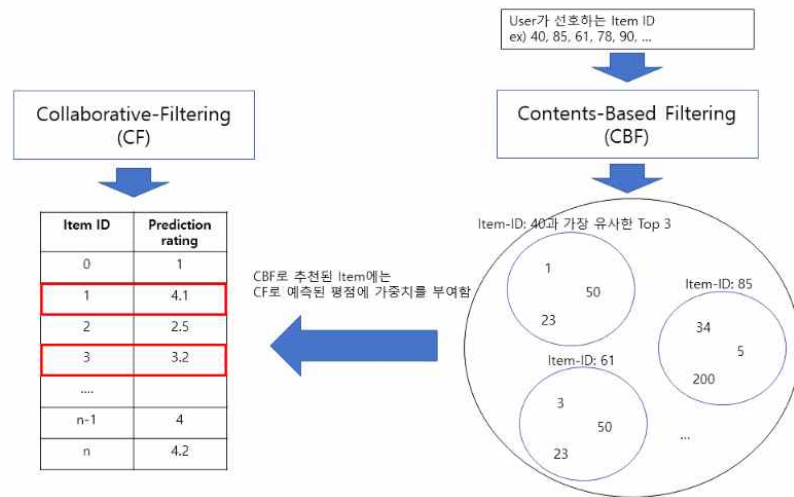
##### 1) 하이브리드 추천

콘텐츠 기반 필터링은 아이템 자체의 정보에 의존한다는 점이다. 아이템 자체의 특성에만 초점을 맞추기 때문에 다양한 관점이나 새로운 추천 요소를 고려하기 어렵다. 또한 사용자의 과거 행동 데이터를 고려하지 않고 아이템의 내용만을 기반으로 추천을 수행하기에 사용자의 동적인 취향 변화와 같이 개인화 요소를 고려하기 어렵다. 협업 필터링의 경우 사용자의 행동 데이터에 의존하기 때문에 새로운 사용자나 새로운 아이템에 대한 추천에는 한계점이 있다. 따라서 두 방식의 보완점을 상쇄할 수 있는, 콘텐츠 기반 필터링(CBF)과 협업 필터링(CF)을 조합하는 하이브리드 추천 시스템을 도입했다.



[Fig 25] CBF와 CF 조합





[Fig 26] 가중치 부여 과정

해당 시스템의 사용자의 선호했던 장소를 콘텐츠기반필터링(CBF)의 입력으로 넣어 Item-Cluster Matrix 각 장소에 대해 유사한 Top3를 도출한다. 협업필터링(CF)에서 해당 사용자에게 대한 예측된 결과에 CBF에서 도출된 장소에 대해서 가중치를 부여한다.

#### 4-1-7. 분석 결과

##### 1) 평가 방법

실제 평점과 예측 평점간의 차이를 평가하는 지표인 RMSE와 사용자 관점에서 사용자가 선호하는 아이템을 정확하게 추천해주는 평가하는 지표인 Hit Rate을 사용한다.

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (pred_i - target_i)^2} \quad \text{----- 식 (2)}$$

RMSE는 예측 모델의 성능을 평가하는 지표로 널리 사용된다. 하지만 RMSE는 추천 시스템에서 사용자의 관점에 직접적으로 부합하지는 않습니다. RMSE는 예측값과 실제값 간의 오차를 평가하는 지표이지만, 추천 시스템에서는 사용자가 실제로 관심을 가지고 선택할 아이템을 정확하게 추천하는 것이 더 중요하다.

$$HitRate@K = \frac{\text{사용자가 실제로 선호한 아이템 개수}}{\text{추천한 아이템 개수 (K개)}} \quad \text{----- 식 (3)}$$

반면, Hit Rate은 사용자가 실제로 선호하는 아이템을 몇 개나 정확하게 추천했는지를 측정하는 지표이다. Hit Rate은 추천 시스템의 목표에 가장 부합하며, 사용자의 관점과 목적에

직접적으로 연결된다. 따라서 RMSE와 Hit Rate을 평가 지표로 사용하되 최종 모델 성능 평가로는 Hit Rate을 선정했다.

## 2) 추천시스템 성능 실험 결과

우선 협업필터링을 통한 평점 예측을 수행한 결과, [Table 5], [Table 6]과 같이 결과가 산출되었다. 맛집 추천의 경우 NCF모델의 RMSE가 1.103으로 가장 작은 것을 확인할 수 있으며, 명소 추천의 경우 SVD모델의 RMSE가 0.887으로 가장 작은 것을 확인했다.

[Table 5] 맛집 협업 필터링 RMSE 결과

	KNNwith means	SVD	SVD++	NMF	NCF
RMSE	1.144	1.026	1.048	1.120	1.103

[Table 6] 명소 협업 필터링 RMSE 결과

	KNNwith means	SVD	SVD++	NMF	NCF
RMSE	0.975	0.887	0.899	0.971	0.913

다음은 기존 협업 필터링 모델과 협업필터링과 콘텐츠기반필터링을 조합한 하이브리드 추천 모델의 Hit Rate을 비교한 결과이다.

[Table 7] 맛집 추천 결과

	KNNwith means	SVD	SVD++	NMF	NCF	Hybird (CBF+NCF)
Hit Rate@10	0.39	0.49	0.48	0.41	0.51	0.53
Hit Rate@15	0.58	0.633	0.627	0.607	0.633	0.633
Hit Rate@20	0.73	0.745	0.76	0.76	0.77	0.775

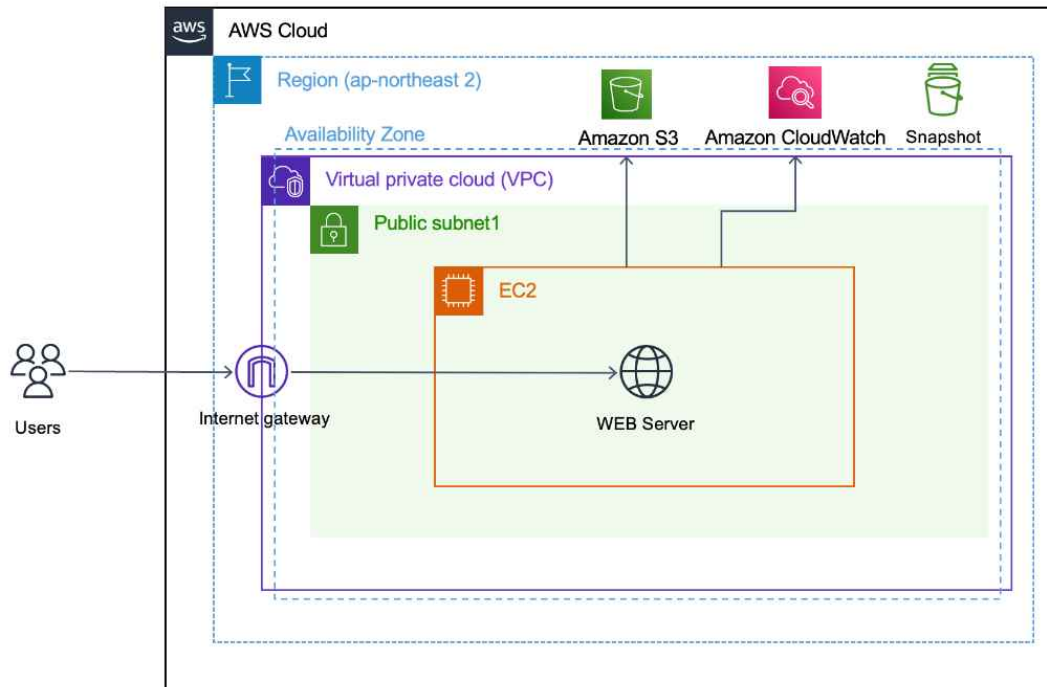
[Table 8] 명소 추천 결과

	KNNwith means	SVD	SVD++	NMF	NCF	Hybird (CBF+NCF)
Hit Rate@10	0.68	0.63	0.66	0.62	0.7	0.71
Hit Rate@15	0.846	0.826	0.84	0.84	0.862	0.862

맛집 추천과 명소 추천에서 CBF와 NCF를 조합한 하이브리드(Hybrid) 모델의 성능이 전반적으로 좋게 나왔다. 따라서 본 프로젝트의 추천시스템에서의 핵심 모델로 하이브리드 모델을 사용하기로 결정하였다.

## 4-2. 시스템 구성

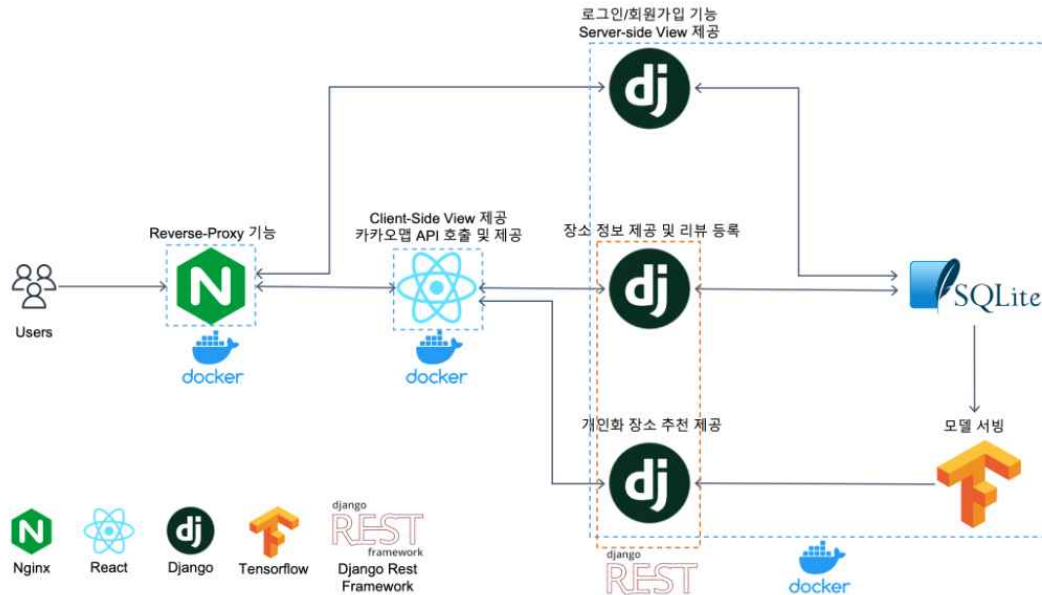
### 4-2-1. 개발환경 구성(구성도 포함)



[Fig 27] 개발 환경 구성도

울산 맛집/명소 개인화 추천 시스템 플랫폼은 다음과 같이 AWS 클라우드 환경에서 배포되고 있다. 구성에는 EC2 인스턴스, Amazon S3, Amazon CloudWatch, Snapshot으로 구성되어 있고 사용자는 Availability Zone 구역에 접근할 수 있다. EC2 인스턴스는 클라우드 컴퓨팅 기본 요소 중 하나로, Amazon Elastic Compute Cloud에서 실행되는 가상 서버이다. EC2 인스턴스 내에서는 웹 서버가 실행되고 있다. 다음으로 Amazon S3는 Amazon Simple Storage Service로 웹 서비스 클라우드 스토리지 서비스이다. 예기치 못한 EC2 인스턴스 다운을 예방하여 백업 파일을 보관하는 용도로 사용되고 있다. 다음으로 Amazon CloudWatch는 모니터링 및 관찰 서비스로 EC2 인스턴스와 상호작용하여 EC2 인스턴스의 모니터링, 로그 수집, 경고 및 자동 조치 등을 가능하게 한다. 현재 프로젝트에서는 성능 모니터링, 과금 방지 관찰 용도로 사용되고 있다.

## 4-2-2. 소프트웨어 구성(구성도 포함)

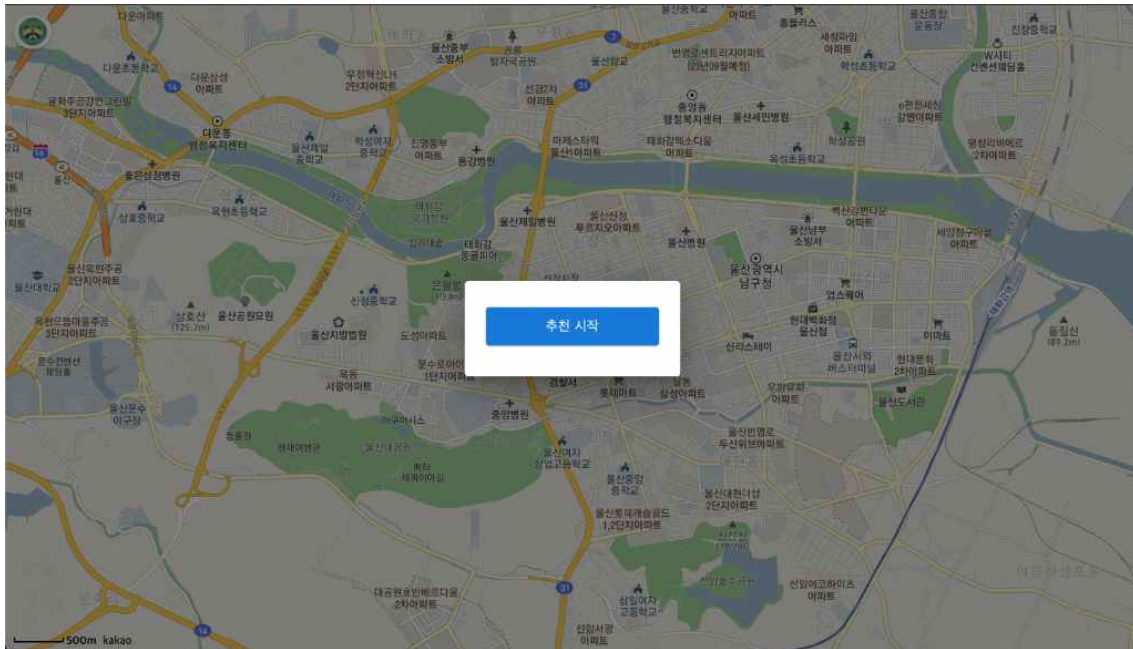


[Fig 28] 소프트웨어 구성도

본 소프트웨어는 Nginx, React, Django 별로 도커 컨테이너를 띄워 구성하였다. django의 각각 기능들은 하나의 서버 내에서 관리되고 있지만 주요 기능 별로 나눠서 구성도에 나타난 상태이다. Nginx는 클라이언트와 웹 서버 사이에서 동작하는 리버스 프록시의 역할로 웹 서버의 앞 단에 배치되어 클라이언트의 요청을 받아들이고, 이를 내부의 웹 서버로 전달하는 역할을 한다. 다음으로 React는 프론트엔드 역할을 수행하며 카카오톡 API를 호출하여 Client-Side View를 제공한다. 다음으로 Django는 백엔드 서버로 로그인/회원가입 기능과 Server-Side View 제공, 장소 정보 제공 및 리뷰 등록 기능을 수행하고 Tensorflow 프레임워크를 통해 개인화 장소 추천을 제공하고 있다. 장소 정보 제공 및 리뷰 등록과 개인화 장소 추천 제공 기능의 응답 데이터는 Django Rest Framework를 이용하여 json 메시지 형태로 통신한다. 마지막으로 데이터베이스는 단일 파일 형태로 제공되는 경량 데이터베이스 엔진인 SQLite를 사용한다.

## 4-3. 기능

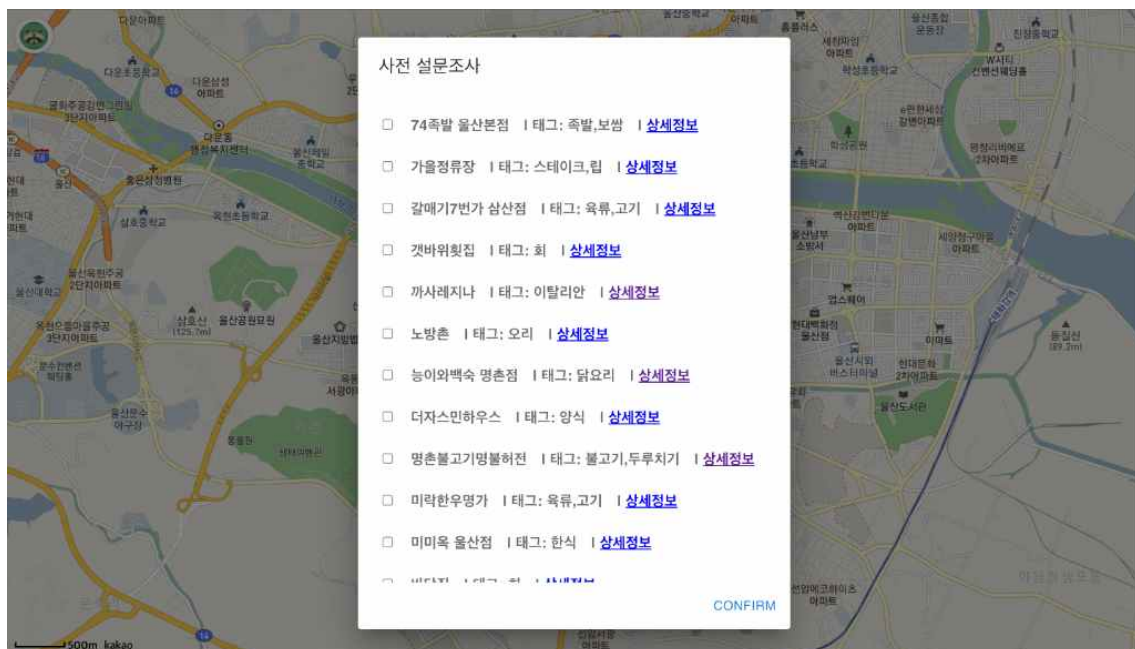
### 4-3-1. 기존 유저 추천



[Fig 29] 기존 유저 추천 시작 다이얼로그

기존 유저는 음식점/명소 카테고리를 선택 시 다음과 같이 추천 시작 버튼의 다이얼로그가 생성된다. 추천 시작 버튼을 클릭 시 django 서버에 API 요청을 보내 하이브리드 기반 개인화 추천이 실행되며 결과적으로 추천 장소들을 응답 받는다.

### 4-3-2. 신규 유저 추천



[Fig 30] 신규 유저 사전 설문조사 다이얼로그

신규 유저는 음식점/명소 카테고리를 선택 시 다음과 같이 사전 설문조사 다이얼로그가 생성된다. 사용자는 장소에 대한 상세정보를 눌러 관심있는 장소들에 대해서 선택을 하고 CONFIRM을 수행하면 django 서버에 API 요청을 보내 컨텐츠 기반 추천이 실행되며 결과적으로 추천 장소들을 응답 받는다.

### 4-3-3. 로그인/회원가입



[Fig 31] 로그인 페이지



[Fig 32] 회원가입 페이지

사용자는 email, 아이디, 비밀번호를 통해 회원가입을 진행할 수 있다. 이때 비밀번호는 테이블에 PBKDF2 암호화 알고리즘을 기반으로 해시 함수를 만들어 저장하기 때문에 brute-force attack에 대한 보호를 제공한다. 사용자는 회원가입 후 아이디, 비밀번호를 통해 로그인을 진행할 수 있다.



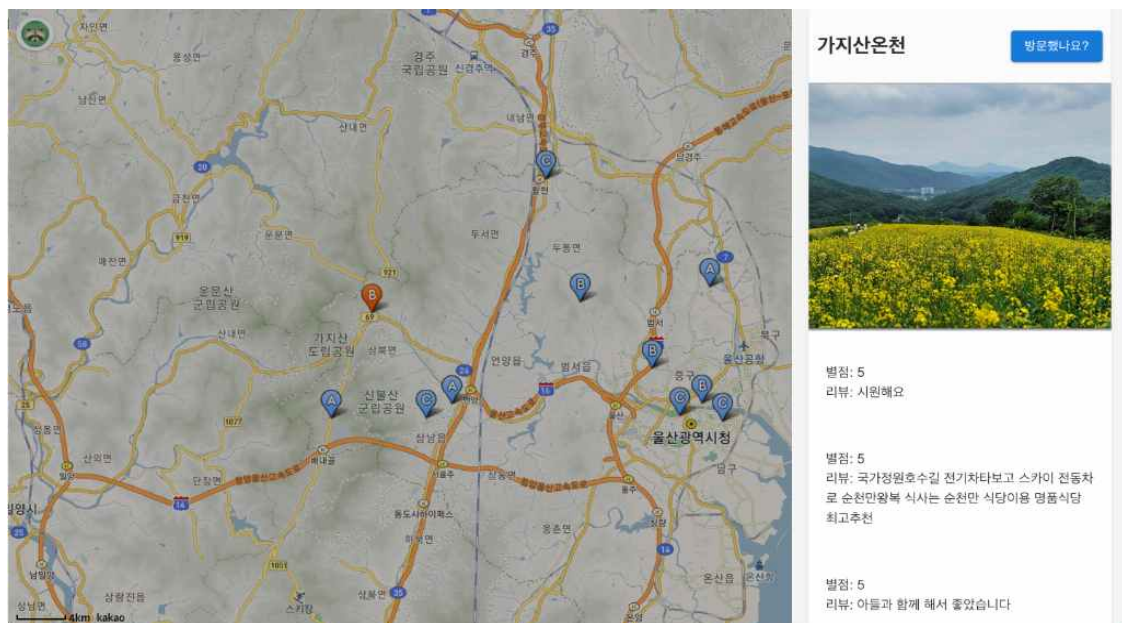
#### 4-3-4. 장소 마커 표시



[Fig 33] 추천 장소 지도 마커 표시

신규 유저 또는 기존 유저가 django 서버에 API 요청 후 추천 모델을 돌려 추천 장소들을 응답 받으면 카카오맵에서 제공하는 마커 표시 API를 활용하여 다음과 같이 추천된 장소들을 마커로 표시해준다.

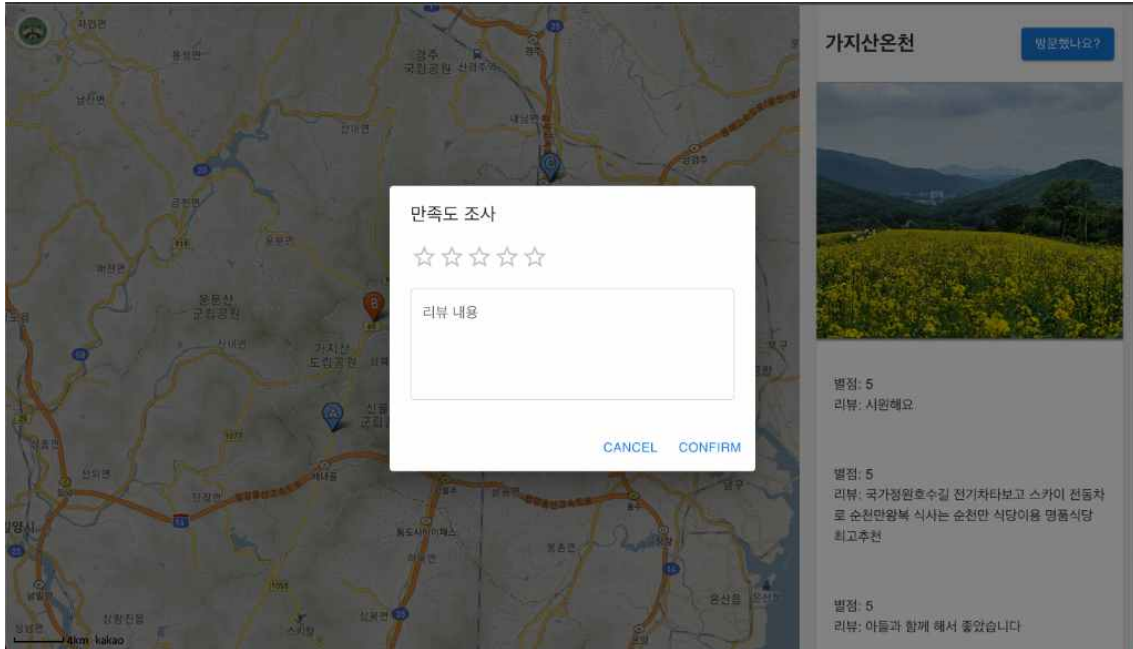
#### 4-3-5. 추천 장소 정보 제공



[Fig 34] 장소 정보 제공 사이드바

추천 장소에 대한 지도 마커를 클릭시 다음과 같이 우측 사이드바가 활성화 되며 해당 장소에 대한 이름, 사진, 기존 리뷰들을 한 눈에 확인할 수 있다.

#### 4-3-6. 추천 장소 리뷰 등록



[Fig 35] 장소 만족도 조사 다이얼로그

추천된 장소에 대하여 '방문했나요?' 버튼을 누를시 만족도 조사 다이얼로그가 활성화되며, 별점과 리뷰 내용을 기입할 수 있다. 리뷰 내용은 해당 유저에 대한 개인화 추천 시스템 모델을 학습할 때 사용될 수 있다.



## 4-4. 사용법

### 4-4-1. 설치 방법

운영체제에 맞는 Git을 설치 후 `git clone https://github.com/UOUCapstoneDesign Team/Ulsan-Recommended-Course-Web.git` 명령어를 실행하여 프로젝트 파일을 가져온다. `node.js(18.15.0)`와 `python(3.10.11)`을 설치하고 프로젝트 내의 폴더 `front-end/react-front` 경로로 이동하여 `npm install` 명령어를 실행해 모듈을 설치 후 `npm start` 명령어를 통해 react를 실행한다. 마찬가지로 프로젝트 내의 폴더 `back-end/test_projec` 경로로 이동하여 `pip install -r requirements.txt`로 모듈을 다운로드 후 `python manage.py runserver`로 django 서버를 실행시킨다. 마지막으로 브라우저 웹 페이지 주소창에 `localhost:8000` 진입한다.

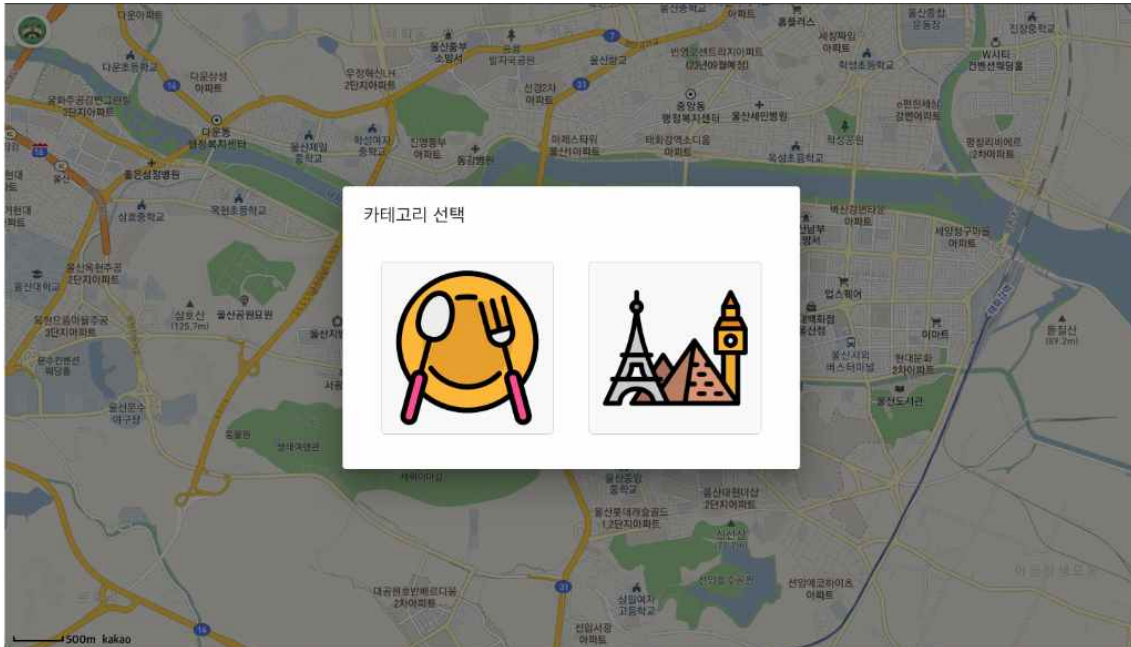
### 4-4-2. 회원가입 및 로그인



[Fig 36] index 페이지 및 회원가입, 로그인

인덱스 페이지에서 우측 상단 Sign up을 눌러 회원가입 페이지로 이동 후 email, 아이디, 비밀번호를 입력하여 회원가입을 완료한다. 그 후 로그인 페이지로 이동해 아이디, 비밀번호를 입력하여 로그인을 진행한다.

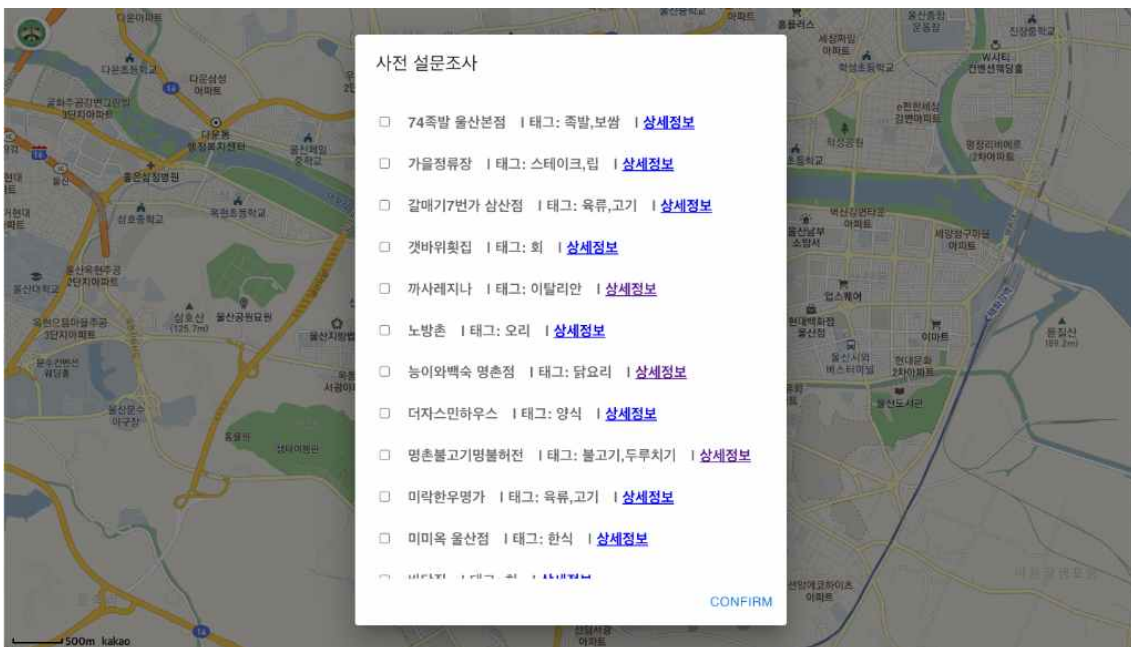
#### 4-4-3. 카테고리 선택



[Fig 37] 카테고리 선택 다이얼로그

음식점 또는 명소를 해당 다이얼로그 그림 버튼을 눌러 추천 받고 싶은 카테고리를 선택한다.

#### 4-4-4. 사전 설문 조사



[Fig 38] 사전 설문조사 다이얼로그

설문 조사에 나오는 이미 방문했던 장소나 상세정보를 눌러 해당 장소에 대한 정보를 얻어 관심있거나 흥미있는 장소를 선택한다.

#### 4-4-5. 추천 장소 표시



[Fig 39] 추천 장소 정보 사이드바

마커로 표시한 추천받은 장소들을 확인하여 마커를 클릭 후 해당 장소에 대한 이름과 사진, 리뷰들을 확인하여 장소에 대한 정보를 얻는다.

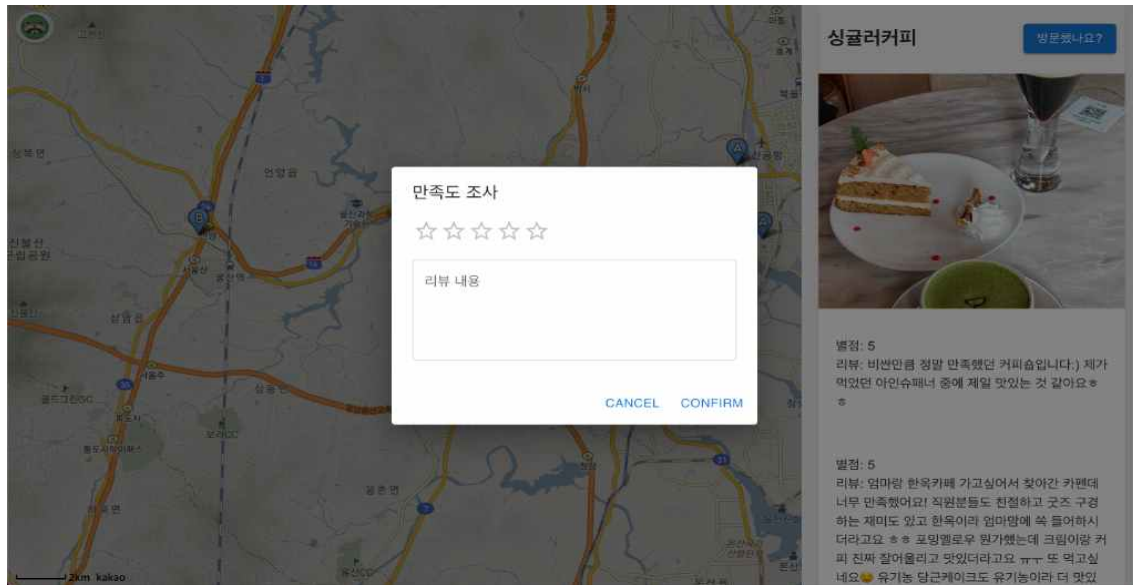
#### 4-4-6. 장소 리뷰 등록

[Fig 40] 장소 만족도 조사 다이얼로그

해당 장소를 실제로 방문하였다면, 우측 위 '방문했나요?' 버튼을 눌러 장소에 대한 별점과 리뷰 내용을 남긴다.

#### 4-5 개발환경

- 언어 : Python, Javascript
- API : Kakaomap API
- 라이브러리 : Numpy, pandas, Selenium, gensim, matplotlib,



seaborn, KoNLPy, MeCab, surprise-python,

scikit-learn, axios, dotenv, react-burger-menu

- 프레임워크 : Tensorflow, Django REST Framework, React, Django
- 하드웨어 : Ubuntu 20.04, CUDA 11.4, NVIDIA RTX A6000
- 운영체제 : windows 11, macOS Ventura
- 개발도구 : Visual Studio, Anaconda
- 데이터베이스 : SQLite

## 4-6 졸업작품 설명

### 4-6-1 파일/모듈 구성

#### 1) react-front/src/choice\_dialog

- ChoiceDialog.js : 음식점/명소 카테고리 선택 다이얼로그 파일

#### 2) react-front/src/kakao\_api

- KakaoMap.js : 카카오맵 로딩 및 지도 마커 표시 API 호출 파일

#### 3) react-front/src/loading\_dialog

- LoadingDialog.js : 추천 결과를 받기 전까지 표시되는 로딩 컴포넌트 파일

#### 4) react-front/src/login\_button

- LoginButton.js : 클릭 시 로그아웃 후 로그인 화면으로 돌아가는 컴포넌트 파일

#### 5) react-front/src/presurvey\_dialog

- PreSurveyDialog.js: 신규 유저에게 제공하는 사전 설문 조사 컴포넌트 파일

#### 6) react-front/src/rating\_dialog

- RatingDialog.js : 추천 받은 장소에 대한 별점과 리뷰를 남기는 컴포넌트 파일

#### 7) react-front/src/rec\_start\_dialog

- RecStartDialog.js : 기존 유저에게 제공하는 장소 추천 시작 컴포넌트 파일

#### 8) react-front/src/sidebar

- Sidebar.js : 마커 클릭 시 해당 장소 정보 제공 사이드바 컴포넌트 파일

#### 9) react-front/src/visit\_button

- VisitButton.js : 추천 장소 방문 여부를 나타내는 버튼 컴포넌트 파일

#### 8) django\_back/accounts

- models.py : 회원 가입된 유저들의 속성 데이터베이스 모델 정의 파일
- urls.py : 회원가입, 로그인, 로그아웃 URL 패턴 정의 파일
- views.py : URL 패턴에 따른 회원가입, 로그인, 로그아웃 로직 정의 파일

#### 10) django\_back/ursapp

- models.py : 음식점/명소 정보와 리뷰의 데이터베이스 모델 정의 파일
- serializers.py : 음식점/명소, 리뷰 데이터를 직렬화, JSON 형식으로 변환하는 파일
- urls.py : 유저 판별 및 추천화 시작, 리뷰 등록 URL 패턴 정의 파일
- views.py : URL 패턴에 따른 유저 판별 및 추천화, 리뷰 등록 로직 정의 파일

#### 11) django\_back/ursapp/recsys

- filtering.py : 콘텐츠기반필터링, 협업 필터링, 하이브리드 추천 실행 파일
- util.py : 추천에 필요한 유틸리티 파일

### 4-6-2 주요 함수/모듈 별 기능

#### 1) ChoiseDialog.js

- function handleClose(url) : 카테고리 선택시 실행되는 메서드로 매개변수인 url 값으로 axios.get 메서드로 GET 요청을 한다. 이때 기존 유저를 나타내는 status code가 200을 응답으로 받으면 RecStartDialog.js를 호출하고, 신규 유저를 나타내는 status code인 201을 응답으로 받으면 json형태로 건네받은 장소 사전 설문 조사 리스트를 PresurveyDialogs.js로 넘겨준다.

- function restClickHandler() : 카테고리 선택시 음식점을 선택하면 실행되는 메서드로 url = 'http://127.0.0.1:8000/recsys\_r2' 값을 할당하고 handleClose(url) 메서드를 호출한다.

- function attrClickHandler() : 카테고리 선택시 음식점을 선택하면 실행되는 메서드로 url = 'http://127.0.0.1:8000/recsys\_a2' 값을 할당하고 handleClose(url) 메서드를 호출한다.

#### 2) KakaoMap.js

- function showMap() : useEffect를 사용하여 지도를 표시할 요소와 위치, 확대 레벨을 정의하고 카카오맵 API를 호출하여 지도를 불러오는 메서드다.
- function saveCoordinatesToAddress(addresses) : 매개변수로 받은 주소들을 (경도, 위도)로 변환을 수행하는 메서드다.

- function initMarker(items) : 마커를 정의하는 메서드로 마커에서 사용할 이미지 좌표 값, 오버 마커 좌표 값, 클릭 마커 좌표 값을 초기화 후 addMarker 메서드를 호출한다.

- function addMarker(index, position, normalOrigin, overOrigin, clickOrigin, items) : 지도에 마커를 표시해주는 메서드로 initMarker에서 정의한 마커 이미지를 사용하여 카카오맵 API를 호출해 마커를 표시하고 클릭 이벤트를 등록해준다.

### 3) PreSurveyDialog.js

- function handleConfirm() : 사전 설문 조사 다이얼로그에서 장소 선택을 마친 후 Confirm 버튼을 누를시 발생하는 메서드로 선택한 장소들을 json 메시지로 담아 'http://localhost:8000/cbf2/' url로 axios.post 메서드를 이용하여 POST 요청을 수행한다. 응답으로 받은 json 데이터를 이용하여 saveCoordinatesToAddress 메서드와 initMarker 메서드를 수행한다.

### 4) RatingDialog.js

- function handleConfirm() : 추천받은 장소 리뷰 다이얼로그에서 리뷰 작성을 마치고 Confirm을 하면 발생하는 메서드로 리뷰 작성 내용을 json 메시지에 담아 axios.post 메서드를 이용하여 'http://localhost:8000/review/' POST 요청을 보낸다.

### 5) RecStartDialog.js

- function onConfirm() : 추천 시작 버튼을 누를시 발생하는 메서드로 'http://localhost:8000/recsys' url로 axios.get 메서드를 이용하여 GET 요청을 보낸다.  
응답으로 받은 json 데이터를 이용하여 saveCoordinatesToAddress 메서드와 initMarker 메서드를 수행한다.

### 6) Sidebar.js

- function pullReviews(title) : 추천 받은 장소의 마커를 클릭시 발생하는 메서드로 axios.post 메서드를 이용하여 'http://localhost:8000/pull\_reviews' url로 매개변수인 title 값을 메시지에 담아 POST 요청을 보낸다.

## 7) ursapp/views.py

- def recsys\_r2(request) : GET 요청에서 'username' 속성의 값을 꺼낸 후 해당 속성의 값이 UserInfo 테이블 존재 여부 확인 후에 기존 유저일 경우 status code 200 응답을 해주고 신규 유저일 경우에는 RestInfo 테이블에서 평점 속성을 내림차순으로 정렬 후 상위부터 20개를 가져와 json 데이터에 담고 status code 201과 함께 응답을 해주는 메서드

-def recsys\_a2(request) : GET 요청에서 'username' 속성의 값을 꺼낸 후 해당 속성의 값이 UserInfo 테이블 존재 여부 확인 후에 기존 유저일 경우 status code 200 응답을 해주고 신규 유저일 경우에는 AttrInfo 테이블에서 평점 속성을 내림차순으로 정렬 후 상위부터 20개를 가져와 json 데이터에 담고 status code 201과 함께 응답을 해주는 메서드

-def cbf2(request) : POST 요청으로 온 데이터들을 'type' 속성 별로 culc\_sim 메서드와 recom\_cbf 메서드를 통해 컨텐츠 기반 추천을 수행하고 결과로 나온 오브젝트들을 json 메시지에 담아 status code 200과 같이 응답을 해준다.

-def recsys(request) : GET 요청에서 'username' 속성을 통해 UserInfo 테이블에서 오브젝트를 가져온 후 'kind' 속성 별로 'ursapp/recsys/model' 내에 위치한 MLP.h5 모델과 attraction\_MLP.h5 모델 중 결정하고 load\_model 메서드를 통해 불러온 후 culc\_sim, recom\_hybrid 메서드를 통해 하이브리드 기반 추천을 수행하고 결과로 나온 오브젝트들을 json 메시지에 담아 status code 200과 같이 응답을 해준다.

-def review(request) : POST 요청으로 온 데이터인 'u\_id', 'title', 'rating', 'review' 속성 값들을 종류에 따라 ReviewRestInfo 테이블 또는 ReviewAttrInfo 테이블에 저장하는 메서드이다.

-def pull\_reviews(request) : POST 요청으로 온 데이터인 'title' 속성 값에 따라 'p\_id'를 결정하고 해당 값에 따른 리뷰들을 최대 20개까지 가져온 후 json 메시지에 담아 status code 200과 같이 응답하는 메서드이다.

## 8) ursapp/model/filltering.py

- def recom\_cf(model, u\_id, data\_df, top\_n) : 협업 필터링을 하는 함수이며 사



용자 ID와 각 카테고리에 맞는 모델, 추천해 줄 개수를 매개변수로 받아 추천 결과를 반환한다.

- def recom\_cbf(p\_id, sim) : 콘텐츠 기반 필터링을 하는 함수로 코사인 유사도를 계산하여 해당 아이템과 유사한 상위 3개의 아이템을 반환한다.

- def recom\_hybrid(model, u\_id, data\_df, top\_n, sim) : 콘텐츠 기반 필터링과 협업 필터링을 조합하는 과정으로 하이브리드 추천 결과를 반환한다.

#### 9) ursapp/model/util.py

- get\_unvisted\_item(data\_df, u\_id) : 해당 사용자가 방문하지 않은 장소들을 리턴하는 함수

- get\_items(data\_df, u\_id) : 방문한 장소와 방문하지 않은 장소를 반환하는 함수

- def culc\_sim(kind) : 코사인 유사도를 계산하여 결과 값을 리턴하는 함수

- def load\_data(kind) : 카테고리를 매개변수로 받아 해당 카테고리에 맞는 데이터 반환

#### 10) accounts/views.py

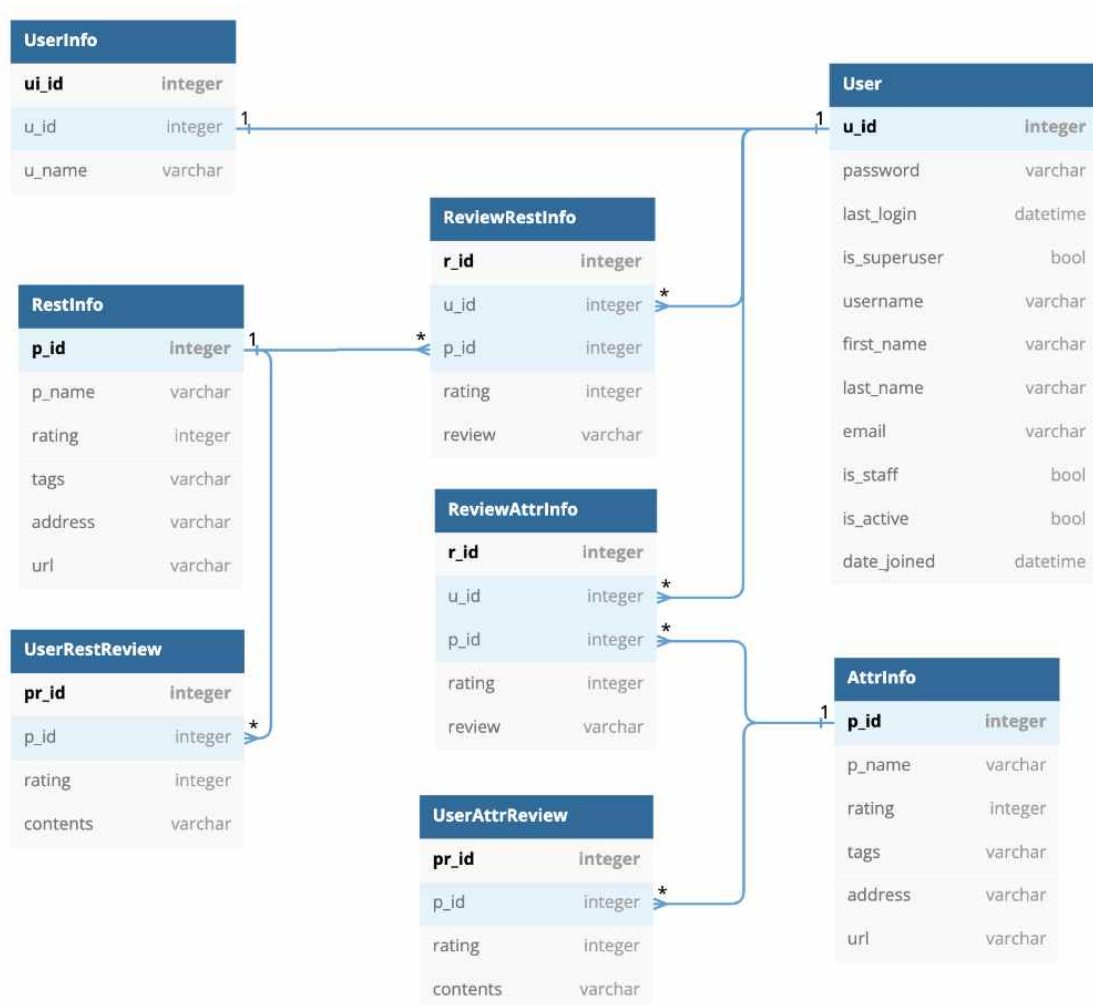
- def signup(request) : POST 요청에서 'username', 'email', 'password1', 'password2' 속성의 값을 꺼낸 후 email과 password의 조건 정규식을 담은 변수를 설정 후 email을 정규식으로 검사해서 조건을 통과하지 못하면 '이메일을 다시 입력해주세요.'라는 문구를 예러 문구로 회원가입 페이지로 보내고 통과하면 password1을 정규식으로 검사해서 조건을 통과하지 못하면 '비밀번호를 다시 입력해주세요.'라는 문구를 예러 문구로 회원가입 페이지로 보내고 통과하면 password1과 password2가 일치하는지 검사하고 일치하면 User 테이블에 해당 정보로 객체를 만들어 저장하고 통과하지 못하면 다시 회원가입 페이지로 돌아가는 메서드

- def login(request) : POST 요청에서 'username', 'password' 속성의 값을 꺼낸 후 username, password에 해당하는 유저가 존재하는지 검사하고 존재하면 로그인을 하고 'http://localhost:3000'으로 보내고 존재하지 않으면 로그인페이지로 예러

문구 ‘아이디, 비밀번호를 확인해주세요.’를 보내는 메서드

-logout(request) : GET 요청을 확인하고 현재 로그인 유저를 로그아웃 시키고 홈 페이지로 이동시키는 메서드

### 4-6-3. 데이터베이스 구조

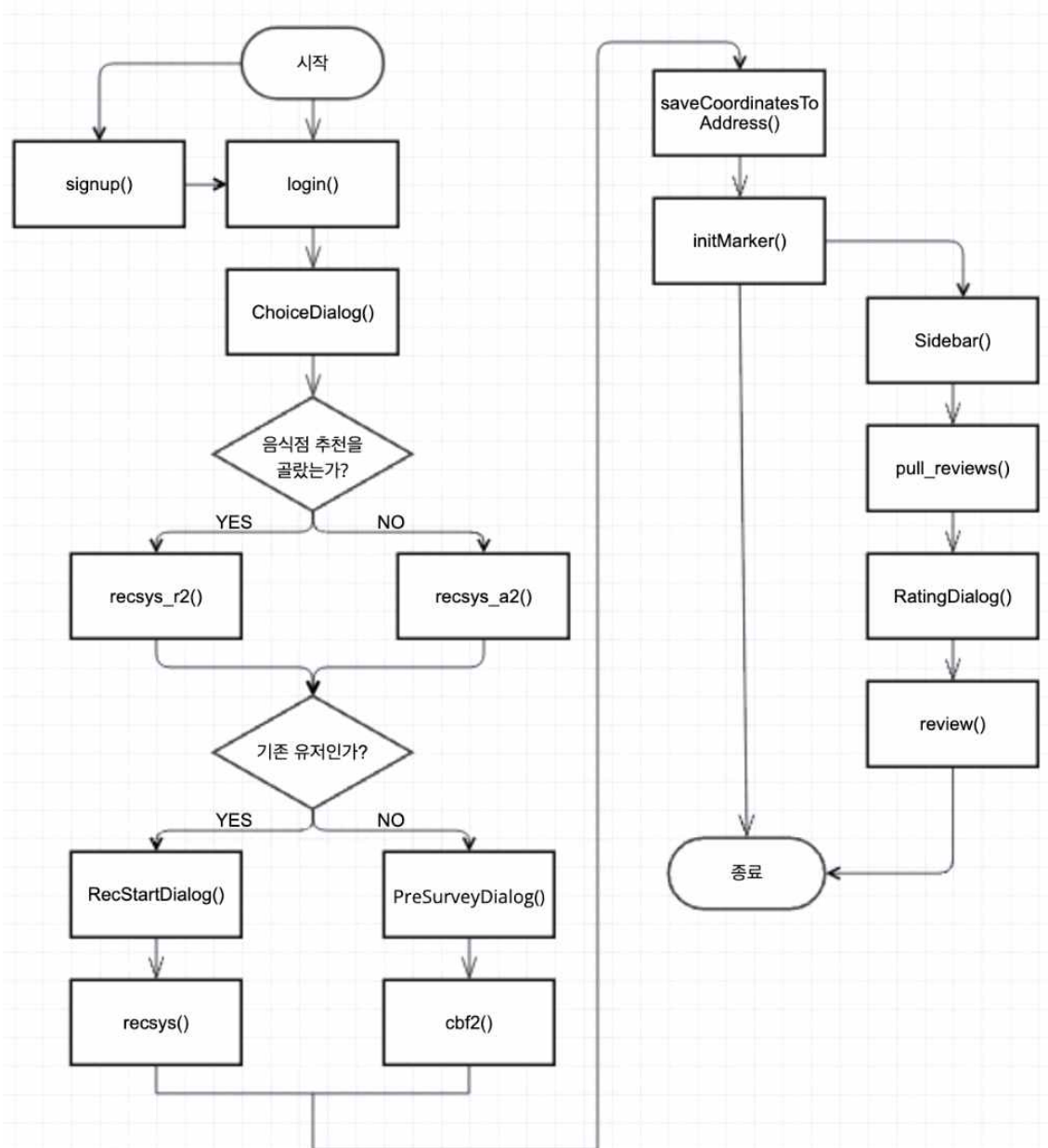


[Fig 41] 개체 관계 모델 (ERD)

- UserInfo == 1 : 1 == User
- User == 1 : N == ReviewRestInfo
- User == 1 : N == ReviewAttrInfo
- RestInfo == 1 : N == ReviewRestInfo
- RestInfo == 1 : N == UserRestReview
- AttrInfo == 1 : N == ReviewAttrInfo

- AttrInfo == 1 : N == UserAttrReview

#### 4-6-4. 주요 기능 흐름도



[ Fig 42 ] 소프트웨어 주요 기능 흐름도

## 5. 제작 일정

(제작기간 : 2022.12.20 ~ 2023.05.25 )							담당학생
제작내용	월 별						
	12월	1월	2월	3월	4월	5월	
주제 선정 및 기존 애플리케이션 조사	O	O					박영준, 김창영, 박상원
자료 조사, 요구 분석 및 필수 기능 선정		O					박영준, 김창영, 박상원
데이터 수집		O	O				박영준, 김창영, 박상원
데이터 전처리 및 EDA			O	O	O		김창영, 박상원
모델링				O	O		김창영, 박상원
도커 환경 구성				O	O		박영준
프론트 화면 구성				O	O		박영준
회원가입, 로그인, 로그아웃 구현					O		박상원
모델 서빙 및 추천 기능 구현					O		김창영
데이터베이스 구축					O	O	박영준, 김창영, 박상원
리뷰 등록 기능 구현						O	박영준
프론트, 백 연결작업						O	박영준, 김창영, 박상원
오류 검출 및 성능 개선						O	박영준, 김창영, 박상원

## 6. 향후 연구와 기대효과

이 졸업작품은 울산 지역의 맛집/명소 추천 시스템 플랫폼으로 개발되었다. 하지만 이 작품은 여기서 끝나지 않고, 향후 연구와 발전을 통해 더욱 많은 기대효과를 가져올 수 있다.

첫째로, 개발된 시스템은 사용자들의 취향과 기호를 분석하여 맞춤형 추천을 제공하는데, 이를 향상시키기 위한 연구가 필요하다. 예를 들어, 더 정교한 사용자 프로파일링 기법을 도입하여 사용자의 선호도와 관심사를 더욱 정확하게 파악할 수 있도록 개선할 수 있다. 또한, 추천 알고리즘의 성능을 향상시켜서 사용자들에게 더욱 타겟팅된 추천을 제공할 수 있는 방안을 연구할 필요가 있다. 이를 통해 사용자들이 더욱 효과적으로 울산의 맛집과 명소를 탐색하고 경험할 수 있을 것이다.

둘째로, 지속적인 데이터 수집과 관리가 필요하다. 졸업작품에서는 초기 데이터를 수집하고 분석하여 시스템을 구축하였지만, 향후에는 새로운 데이터의 수집과 기존 데이터의 업데이트가 필요하다. 이를 위해 MLOps (Machine Learning Operations) 개념을 도입하여 데이터 관리 및 시스템 업데이트 프로세스를 자동화하고 효율화할 수 있다.

MLOps를 적용함으로써, 우리 시스템은 지속적으로 발전하고 개선될 수 있으며, 데이터 관리와 시스템 업데이트에 효율성과 일관성을 제공할 수 있다. 이를 통해 사용자들에게 신뢰할 수 있는 추천 서비스를 제공하고, 울산의 맛집과 명소를 효과적으로 소개하여 지역 경제 활성화에 기여할 수 있을 것이다.

셋째로, 향후에는 시스템의 확장성과 유연성을 고려한 개선이 필요하다. 우리는 사용자들의 다양한 요구사항에 대응하기 위해 울산 맛집 명소 추천 시스템을 스마트폰 어플리케이션 형태로 발전시킬 계획이다. 이를 통해 사용자들은 언제 어디서나 손쉽게 울산의 맛집과 명소 정보를 확인하고 추천받을 수 있다.

스마트폰 어플리케이션 형태의 발전을 통해 우리 시스템은 보다 사용자 친화적이고 접근성이 용이한 서비스를 제공할 수 있다. 사용자들은 울산의 맛집과 명소를 손쉽게 탐색하고, 예약과 리뷰 작성 등의 기능을 통해 더욱 풍부한 경험을 할 수 있다. 또한, 다양한 플랫폼과의 연동을 통해 사용자들은 편리하게 시스템을 이용할 수 있으며, 울산의 매력을 더욱 널리 알릴 수 있을 것이다.

이러한 연구와 개발을 통해 울산 지역의 맛집/명소 추천 시스템은 다음과 같은 기대효과를 가져올 수 있다. 첫째로, 사용자들은 개인화된 추천을 받음으로써 울산의 다양한 맛집과 명소를 보다 효과적으로 발견하고 경험할 수 있을 것이다. 두번째로, 울산의 문화와 관광 산업이 활성화될 것으로 기대된다. 시스템을 통해 울산의 특색 있는 맛집과 명소가 더욱 알려지고, 이를 방문하는 관광객들의 수가 증가함으로써 지역 경제에 긍정적인 영향을 미칠 것이다. 마지막으로, 지속적인 연구와 발전을 통해 맛집/명소 추천 시스템의 기술력과 활용도가 높아질 것이다. 이는 다른 지역이나 업종에도 확장될 수 있으며, 관광 산업에 대한 연구와 개발의 지속적인 발전을 이끌어낼 수 있을 것이다.

## 참고 문헌

- [1] 현지연, 유상이, 이상용(2019), 평점과 리뷰 텍스트 감성분석을 결합한 추천시스템 향상 방안 연구, 한양대학교
- [2] 최환석, 팽전, 이우섭(2020), 머신러닝 기반 음식점 추천시스템 설계 및 구현, 한밭대학교
- [3] 윤소영, 윤성대(2020), 상품리뷰 감성분석을 이용한 아이템 기반 협업 필터링 추천 기법, 부경대학교