TF-IDF를 활용한
사용자 취향 및 재료 기반 메뉴 추천 시스템

전계범, 이지석, 오유솔
경희대학교 산업경영공학과, 경희대학교 소프트웨어융합학과
bum0209@khu.ac.kr , jiseok0106@khu.ac.kr . wina0817@khu.ac.kr

Users' Preference and Ingredient based Menu Recommendation System using TF-IDF

영문이름
소속 영문명

요 약

- - -

1 서론

1.1 프로젝트 주제 선정 이유 및 목적

환경부 산하 한국환경공단이 서울시의 생활폐기물 발생량을 검토하여 2022년 2월 말에 발표한 바에 따르면, 서울 시민 1인당 하루 플라스틱 배출량이 2016년 110g에서 2020년 236g으로 2배가 넘게 증가했다고 한다. 이는 코로나로 인해 재택근무와 외식을 대신한 배달서비스 이용이 늘면서 일회용품 소비가 급증한 것을 주요 원인으로 예상해볼 수 있다. 코로나로 인해 증가한 쓰레기를 칭하는 '코로나 트래쉬(Trash)'라는 신조어까지 생긴 것을 보면, 배달 음식의 소비 증가가 환경에 미치는 영향을 엿볼 수 있다.

또한, 코로나19로 인해 배달 서비스들이 더욱 편리해지고 발전함에 따라, 소비자들이 인스턴트 음식들을 많이, 그리고 자주 먹을 수 있는 환경이 만들어졌다. 이로 인해 소비자들의 건강에도 우려가 가는 상황이다.

이러한 상황의 개선을 위해, '집에 가지고 있는 재료들로 쉽게 할 수 있으면서, 자신이 좋아하는 음식 취향이 반영된 메뉴와 요리 방법'을 알려주고자 한다. 따라서 '개인의 음식 취향 및 사용하고자 하는 재료 기반 메뉴 추천'이라는 주제를 프로젝트의 주제로 선정하게 되었다. 이를 통해, 배달 음식에 대한 소비를 줄여 일회용품의 사용량을 줄이고, 소비자가 조금 더 건강한 식생활을 할 수 있도록 유도하고자 한다.

1.2 프로젝트 개요

프로젝트는 다음의 큰 네 단계로 진행된다.

① 데이터 확보 ② 데이터 전처리 ③ 추천시스템 모델 구현 ④ 추천시스템 평가 및 보완

① 사용자에게 추천 될 메뉴, 재료와 레시피에 대한 데이터를 직접 수집하고, 사용자에게 다양한 레시피를 제공하기 위해 최대한 많은 데이터를 확보한다. ② 정확한 추천 시스템 구현을 위하여, 모델 구현에 알맞은 형태로 전처리를 진행한다. ③ 그 후, 사용자의 취향에 맞고, 사용자가 입력한 재료 조건을 잘 만족하는 메뉴 추천을 위해, 전체 메뉴 풀을 세 단계에 걸쳐 마지막으로 필터링 한 뒤, 추천하는 모델을 구현한다. 마지막으로, ④ 모델이 개인의 취향을 잘 반영하여 추천을 하는지, 추천된 메뉴들이 상이하지 않고 일관성 있는지를 personalization과 intra-list similarity 지표를 사용하여 성능평가를 진행한다. 성능 평가 결과를 바탕으로, 최고의 성능을 갖기 위해 필요한 사용자 입력 조건을 결정하는 것으로 프로젝트의 결론을 맺는다.

2. 방법

2.1 데이터 수집

사용자의 취향에 맞는 동시에, 사용자가 가지고 있는 재료로 요리 할 수 있는 메뉴와 레시피를 추천하기 위해서는, '한 가지 메뉴에 대해, 다양한 재료를 사용한 여러 레시피 데이터'가 필요하다. 또한, 사용자들의 다양한 음식 취향을 반영하기 위해서는 많은 양의 데이터가 필요하다. <만개의 레시피>라는 웹 사이트에는 약 200만 개의 '메뉴 & 레시피' 데이터가 있었기 때문에, <만개의 레시피> 데이터가 위의 조건에 가장 적합하다고 판단하였다.

데이터 확보를 위해 <만개의 레시피> 웹 사이트를 직접 크롤링하고, 추후에 데이터가 '음식의 대분류' 기준으로 사용될 것이기 때문에, 크롤링은 페이지 속 '음식의 대분류' 기준으로 나누어 진행하였다. 필요한 데이터들을 모두 웹 크롤링 한 결과, 총 2293717개의 '메뉴 & 레시피' 데이터를 확보하였다.

2.2 데이터 전처리

추천 모델을 구현할 때에, TF-IDF (텍스트의 벡터화)를 통해 메뉴 간의 유사도를 측정해야하기 때문에, 전처리를 통해 데이터의 텍스트들을 최대한 정형화하고자 했다. 데이터 전처리를 다음의 두 단계로 진행한다.

① 1차 전처리 : 모든 데이터에 대해 특수 문자 제거 및 맞춤법 검사
② 2차 전처리 : '메뉴명'에서 불필요한 수식어들을 제거

Hanspell(spell_checker) 등의 python 라이브러리를 이용하여 데이터들에 존재하는 특수 문자를 제거하고, 맞춤법 검사를 진행하였다. '메뉴 & 레시피' 데이터의 '메뉴 이름'에서 불필요한 수식어들을 제거하였고, 토큰화를 통한 자연어처리로 명사들과 주요 키워드들을 추출하여 '메뉴 이름'을 정제하려고 했지만, 원하는 형태로 결과가 도출되지 않았다. 그러나 '만개의 레시피' 회사에서 제공하는 '정제된 메뉴명' 데이터를 추가 확보하였고, 기존에 직접 크롤링한 데이터와 이 데이터를 통합하여, 원하는 형태로 '메뉴명'을 수정하였다.

3. 추천 시스템 개발

3.1 모델 개발 방법 TF-IDF 소개

3.1.1 문서 단어 행렬과 TF-IDF

문서 단어 행렬이란 다수의 문서에서 등장하는 각 단어들의 빈도를 행렬로 표현한 것을 의미한다. 예를 들어, 미역국의 레시피 재료와 떡국의 레시피 재료, 2개의 문서가 있다고 가정한다.

- 레시피 재료 1 : 미역 소고기 참기름 다진마늘 국간장
- 레시피 재료 2 : 떡 참기름 계란 대파 김가루 다진마늘 국간장

2개의 문서를 문서 단어 행렬로 나타내면 [표1]과 같다.

[표1]

	계란	국간장	김가루	다진 마늘	대파	미역	소고기	참기름
재료1	0	1	0	1	0	1	1	1
재료2	1	1	1	1	1	0	0	1

각 문서에 등장한 단어의 빈도를 행렬로 나타낸 것이다. 이 결과를 각 재료 문서에 대해 0과 1로 구성된 벡터로 표현한 것으로 해석할 수 있다.

3.1.2 희소 표현 (Sparse Representation)

각 행을 문서 벡터라고 가정한 문서 단어 행렬에서, 각 문서 벡터의 차원은 전체 데이터(레시피)에 포함된 재료 단어 집합의 크기를 가진다. 만약 가진 데이터의 수가 많을 경우, 문서 벡터의 차원은 수천수만 차원이 될 수도 있다. 또한, 많은 문서 벡터가 0의 값을 가지고 있는데, 이와 같이 대부분의 값이 0인 표현을 희소 벡터 또는 희소 행렬이라고 부른다.

3.1.3 TF-IDF (Term Frequency-Inverse Document Frequency)

문서 단어 행렬 내의 각 단어에 대해 중요도를 계산할 수 있는 방법이 TF-IDF이다. TF-IDF를 사용하면, 단어가 더 많은 경우에 더 좋은 결과를 얻을 수 있는데, 자세한 설명은 아래에서 예시와 함께 설명한다.

・TF : 특정 문서에서 특정 단어의 등장 횟수
・DF : 전체 문서 대비 특정 단어가 등장한 문서의 수
・IDF : DF에 반비례하는 값

$$Idf(d(t),t) = \log(\frac{n}{1+d(t)})$$

d(t)는 DF값을 의미하고 t는 특정 단어를 의미한다. n은 전체 문서의 수를 의미한다. 역수를 취하는 이유는 문서에 자주 등장하는 단어일수록 낮은 가중치를 부여하고, 자주 등장하지 않는 단어일수록 높은 가중치를 부여하기 위함이다. log를 사용하는 이유는, 총 문서의 수(데이터의 수)가 커질수록 값이 기하급수적으로 커지기 때문이다.

3.1.4 Sklearn을 이용한 특징 추출

자연어 처리에서 특징 추출이란 텍스트 데이터에서 단어나 문장들을 어떤 특징 값으로 바꿔주는 것을 의미한다. 기존에 문자로 구성되어 있던 데이터를 모델에 적용할 수 있도록 특징을 뽑아 어떤 값으로 바꿔 수치화한다. Sklearn을 사용해 텍스트 데이터를 수치화하는 방법은 크게 3가지가 있다. 3가지 방법 모두 텍스트 데이터를 다루면서 자주 사용되는 기법이다. 관련 모듈의 목록은 다음과 같다. 각 모듈에 대해 간단히 소개를 하고, 이번 주제인 TF-IDF에 대해서 다뤄보고자 한다.

・CountVectorizer ・TfidfVectorizer ・HashingVectorizer

CounterVectorizer는 단순히 각 텍스트에서 횟수를 기준으로 특징을 추출하는 방법이다. TfidfVectorizer는 TF-IDF라는 값을 사용해 텍스트에서 특징을 추출한다. 마지막으로 HashingVectorizer는 앞서 설명한 CountVectorizer와 동일한 방법이지만, 텍스트를 처리할 때 해시함수를 사용하기 때문에 실행 시간을 크게 줄일 수 있다. 따라서 텍스트의 크기가 클수록 HashingVectorizer를 사용하는 것이 효율적이다. 세 가지 방법 모두 텍스트를 벡터로 만드는 방법이다. 하지만, 이번 주제인 사용자의 취향을 반영한 재료 기반 레시피 추천 시스템에서 중심적으로 사용된 TfidfVectorizer에 대해서 중심적으로 다룬다.

CounterVectorizer는 이름에서 확인할 수 있듯이 텍스트 데이터에서 횟수를 기준으로 특징을 추출하는 방법이다. 일반적으로 텍스트에서 단어를 기준으로 횟수를 측정한다. 예를 들어, '미역국'이란 데이터(요리명)에 포함된 '미역 소고기 참기름 다진마늘 국간장'이라는 텍스트 데이터(재료)를 횟수 값으로 이루어진 벡터로 만든다면, 우선 단어 사전을 정의해야 한다. 이때, 단어 사전이 {계란, 국간장, 김가루, 다진마늘, 대파, 미역, 소고기, 참기름}이라는 8개의 단어로 구성되어 있다고 한다면, '미역국'이란 데이터는 [0,1,0,1,0,1,1,1]이라는 벡터로 바뀔 것이다. 이와 같이 매우 간단하게 텍스트 데이터에서 특징을 추출하는 것이 가능하지만, 큰 의미가 없지만 자주 사용되는 단어들로 인해 단점도 존재한다.

TfidfVectorizer는 앞서 설명한 CountVectorizer의 단점을 보완하는 것이 가능하다. TfidfVectorizer는 TF-IDF라는 특정한 값을 사용해서 텍스트 데이터의 특징을 추출하는 방법이다. 각 값이 의미하는 바를 간단히 설명하면, TF(Term Frequency)란 특정 단어가 하나의 데이터 안에서 등장하는 횟수를 의미한다. DF(Document Frequency)란 특정 단어가 여러 데이터에 자주 등장하는지를 알려주는 지표이고, IDF(Inverse Document Frequency)는 이 값에 역수를 취해서 구할 수 있으며, 특정 단어가 다른 데이터(문서)에 등장하지 않을수록 값이 커지는 것을 의미한다. 한글 중 조사와 같은 불용어의 경우, 다른 데이터(문서)에 빈번하게 등장한다. 하지만, 불용어의 경우는 문서 중에서 의미적 중요도가 낮은 단어이기 때문에 역수를 취함으로써 IDF 값을 낮춘다. 반면, 중요한 단어의 경우는 해당 단어가 출현한 문서에만 등장하고 다른 문서에는 등장할 가능성이 상대적으로 낮기 때문에 DF 값이 작다. 따라서 이를 반영하기 위해 IDF 값을 계산함으로써 높은 가중치를 준다. TF-IDF는 이 두 값을 곱해서 사용하므로, 어떤 단어가 해당 문서에 자주 등장하지만, 다른 문서에는 많이 없는 단어일수록 높은 값을 가지게 된다. 따라서 조사나 지시대명사와 같이 자주 등장하는 단어는 TF 값은 크지만, IDF 값은 작아지게 되므로 상대적으로 일반 명사나 고유 명사에 비해 낮은 가중치를 부여 받는다. [그림1]은 위에서 사용한 예시와 동일한 '미역국' 데이터를 텍스트 데이터(재료)를 통해 TfidfVectorizer를 사용해 특징을 추출한 것이다.

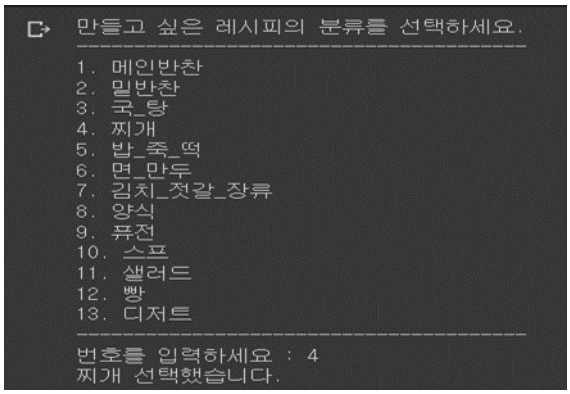
[그림 1]

위 [그림1]의 마지막 결과를 보면, 문서를 벡터로 만든 값이 위의 CountVectorizer를 사용한 결과와 다르다. 떡, 계란, 대파, 김가루라는 단어는 해당 문서(미역국)에 사용되지 않아서 모두 0의 값이 나왔다. 또한, 국간장, 다진마늘, 참기름이라는 단어는 0.38 정도의 TF-IDF 값을 가지고, 미역, 소고기라는 단어는 0.53 정도의 값을 가진다. 이처럼 TF-IDF 값을 사용할 경우, 단순 횟수를 이용해 특징을 추출하는 것보다 각 단어의 특성을 좀 더 잘 반영할 수 있다. 따라서 레시피에 사용되는 재료의 경우, 각 단어를 동일하게 가중치를 부여하는 것보다, 중요한 단어(주재료)에 대해 높은 가중치를 부여하고, 덜 중요한 단어(보조재료)에 대해 상대적으로 낮은 가중치를 부여하는 TF-IDF 방식이 조금 더 좋은 결과를 만들어낸. 그렇기 때문에, 이 프로젝트에서는 TF-IDF방식을 사용하여 추천 모델을 구현한다.

3.2 추천 시스템 모델 내용

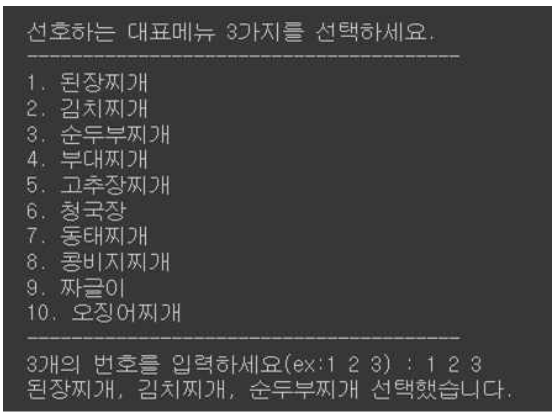
본 프로젝트에서 개발한 추천 시스템 모델은, 확보한 데이터의 절대적인 양이 많기 때문에, 사용자로 하여금 조건들을 입력하게 하여 데이터를 필터링 하는 방식으로 진행되며, 최종적으로 10개의 메뉴와 레시피를 추천해준다. 필터링하는 과정은 총 3번 진행된다.

1차 필터링은 사용자가 본인이 만들고 싶은 메뉴의 분류를 선택하여 진행된다. 본 프로젝트에서는 이 분류를 대분류라고 정의 내렸다. 사용한 데이터는 총 13가지의 대분류로 나뉜다. [그림 2]처럼 '1.메인반찬'부터 '13.디저트'까지 13가지의 대분류 중 사용자가 만들고 싶은 대분류를 선택하면, 해당 대분류의 레시피 데이터만 불러와 추천 시스템 모델이 동작한다.



[그림 2]

다음으로 2차 필터링을 진행하기 전에, 1차 필터링 된 데이터(요리 재료)를 TF-IDF를 통해 텍스트 임베딩을 하여 벡터화를 한다. 그 후, 사용자로 하여금 선호하는 대표메뉴 3가지를 선택하게 한다. 여기서 대표메뉴라는 것은, 1차 필터링에서 선택한 대분류의 음식 중 가장 대중적인 메뉴 10가지를 의미한다. 대중적인 메뉴의 기준은 <만개의 레시피> 사이트에서 '레시피 조회 순'으로 정렬한 뒤 선정하였다. 2차 필터링에서 선택한 대표메뉴 3가지를 사용자의 취향이라고 판단한다. 그 후 3가지 대표메뉴들과 기존의 데이터 간의 코사인 유사도를 계산하여 필터링한다. 이 과정이 2차 필터링이다. 이 때 필터링하는 기준은, 코사인 유사도의 값이 전체 데이터의 평균(mean) + 표준편차(std) 값을 넘기는지 넘기지 못하는지로 한다. 이러한 방식으로 2차 필터링을 통해, 사용자의 취향을 반영하여 메뉴와 레시피 리스트를 다음 과정(3차 필터링)으로 넘겨준다.



[그림 3]

3차 필터링을 진행하기 전에, 2차 필터링 된 데이터(요리 재료)를 TF-IDF를 통해 텍스트 임베딩을 하여 벡터화를 해준다. 그 후 3차 필터링에서는 사용자가 현재 가지 있는 재료나 사용하고 싶은 재료를 입력한다. 입력한 재료들에 대해서 2차 필터링 된 데이터들과 코사인 유사도를 계산하여 내림차순으로 정렬한다. 정렬된 데이터 중 상위 10가지, 즉 코사인 유사도가 가장 높은 10가지 메뉴와 레시피를 최종적으로 추천해주는 과정이 3차 필터링이다. [그림 4]는 3차 필터링을 위해 사용자가, 본인이 사용하고자 하는 재료를 입력하는 과정을 나타낸다.

[그림 4]

이렇게 1~3차 필터링 과정을 통해 사용자의 취향 및 재료 기반 레시피 추천 시스템 모델이 최종적으로 10가지의 레시피를 추천해주게 된다. 추천된 메뉴와 레시피는 [그림 5]와 같이 출력된다.

[그림 5]

4. 추천 시스템 성능 평가

4.1 성능 평가 지표 선정

최종 결과물의 추천 성능의 정확도를 보기 위해, ‘Intra-list Similarity’와 ‘Personalization’의 성능 지표를 이용하여 평가를 해보았다.

Intra-list Similiarity의 경우, 여러 사용자의 모든 추천 메뉴들 간의 평균 코사인 유사도를 구하여, 추천 시스템이 얼마나 비슷한 성격의 메뉴들을 잘 추천하는지를 평가하기 위하여 선택하였다. 한 사용자에게 성격이 많이 다른 메뉴들이 추천되는 경우, 추천 시스템의 성능이 좋지 않은 것이라고 판단하였다. 즉, Intra-list Similarity 값이 높다면, 추천 시스템이 각 사용자에게 일관성 있게 매우 비슷한 성격의 메뉴들을 잘 추천해주고 있음을 의미하는 것이다.

Personalization의 경우, 추천 시스템이 사용자의 취향을 얼마나 잘 반영하는지를 평가하기 위해, 사용자들의 추천 메뉴 리스트들 간의 dissimilarity를 비교하기 위하여 선택하였다. Personalization 값이 높으면, 추천 시스템이 각 사용자의 취향을 잘 반영하고 있음을 의미하는 것이다.

4.2 Intra-list Similarity를 이용한 평가 결과

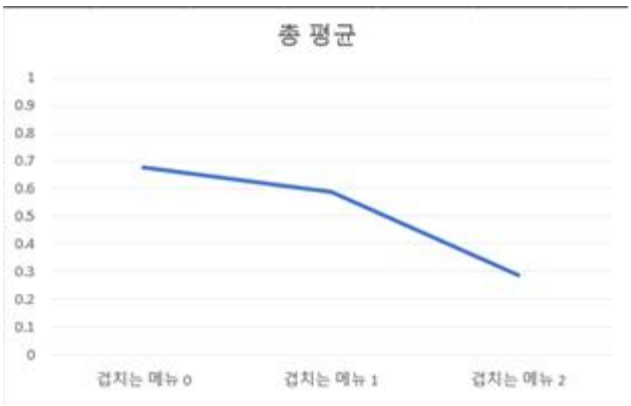
추천 시스템이 일관성 있게 메뉴를 추천하는지를 평가하기 위해, 동일한 음식의 대분류 내에서 동일한 대표메뉴를 선택한 사용자들이 재료를 입력할 때에, 입력한 주재료와 보조재료(ex. 양념)의 비율에 차이를 두어 값을 비교해보았다. 주재료의 비율을 높게 입력하였을 때가 보조 재료의 비율을 높게 입력 했을 때에 비해, 비슷한 성격의 메뉴들을 더 정확하게 추천해 줄 것이라고 가정하고 진행하였다. 실제로 8개의 대분류에 대해서 평가를 진행했을 때에, [그래프 1]과 같이 ‘주재료의 비율이 높은 경우’에 Intra-list Similarity의 값도 높게 나왔다.

[그래프 1]

4.3 Personalization을 이용한 평가 결과

모델이 사용자의 취향을 잘 반영하여 추천해주는지를 평가하기 위하여, 동일한 음식의 대분류 내에서 동일한 재료를 입력했을 때, 사용자가 선택한 대표 메뉴에 따라 추천되는 결과가 어떻게 다른지를 비교하여 평가를 진행했다. 추천 시스템이 개인의 취향을 잘 반영하는지를 보기 위해, 사용자들이 선택한 대표 메뉴가 2개 겹칠 때, 1개 겹칠 때, 아예 겹치지 않을 때로 나눠서 계산 결과를 관찰하였다. 실제로 여러 번 시행을 해본 결과, 사용자들 간에 겹치는 대표메뉴가 없었을 때, 즉, 사용자들의 취향을 모두 다르게 입력하였을 때에 personalization 값이 높게 나왔다. [그래프 2]는 각 경우에 대해서 평균 값을 계산한 뒤, 증감을 보기 위해 나타낸 그래프이다.

[그래프 2]



5. 결론

5.1 기대 효과

개인 선호도 기반 맞춤형 레시피 추천 시스템을 활용하면, 서비스 이용자들은 보유하고 있는 식재료를 최대한 활용함으로써 배출되는 음식물 쓰레기량을 효과적으로 감소시킬 수 있을 뿐만 아니라 배달음식의 소비량을 감소시킴으로써 결과적으로는 일회용품 소비량을 줄여, 환경 개선에 영향을 미칠 수 있다. 또한, 서비스 이용자들은 가지고 있는 식재료를 활용해 어떤 요리를 할지 고민하고 레시피를 검색하는 등의 노력을 줄일 수 있다는 장점이 있다.

5.2 활용방안

이를 활용하면 가지고 있는 식재료를 최대한 활용하는 것도 가능하지만, 선택한 레시피에 대해 필요한 식재료를 효과적으로 파악할 수 있어서 식재료 구매 시에 효율적이게 된다. 또한, 제공되는 요리에 대한 칼로리, 영양성분 등의 정보를 추가하여 시스템을 보완한다면, 이를 활용해 균형 잡힌 식사를 유도하거나 다이어트에 식단 보조 프로그램으로도 이용 가능할 것이다.

이와 같이, 현재의 '사용자 취향 기반 및 사용하고자 하는 재료 기반' 메뉴 추천 시스템에, 메뉴와 재료, 레시피에 대한 다양한 메타데이터를 추가하여 시스템을 목적에 맞게 구체화하고 발전시킨다면, 다양한 목적을 충족시키는 시스템으로 변형하여 사용이 가능하다.

5.3 결론 및 제언

코로나19로 인해 배달 음식과 인스턴트식품의 수요가 증가하였고, 여기서 배출되는 쓰레기들과 인스턴트식품의 좋지 않은 재료들이 환경과 소비자들의 건강에 나쁜 영향을 미치고 있다. 이러한 상황을 조금이나마 개선시킬 수 있는 프로젝트를 하고 싶어, '사용자의 취향과 사용하고자 하는 재료 기반 메뉴 추천 시스템'을 만들게 되었다. TF-IDF를 활용하여 메뉴 데이터들을 여러 단계를 통해 선별하였고, 최종적으로 10개의 메뉴를 추천해주는 형식으로 추천 시스템을 구현하였다.

프로젝트를 진행하며, 프로젝트에 적합한 데이터를 구하지 못하거나 데이터 전처리에서 기술적으로 해결하지 못한 부분에 있어서 아쉬움이 남는다. 프로젝트 계획 초반에 확보하고자 했던 사용자 데이터를 추후에 이용할 수 있게 된다면, 사용자 데이터와 지금의 추천 모델을 결합하여, 단순 'TF-IDF'를 통한 추천이 아닌 머신러닝을 통한 추천 시스템을 구현하는 프로젝트를 진행해보고자 한다.