<맛집 추천시스템 알고리즘 연구 및 개발>

한상현 안우진

백우현 강지원

전희연

목차

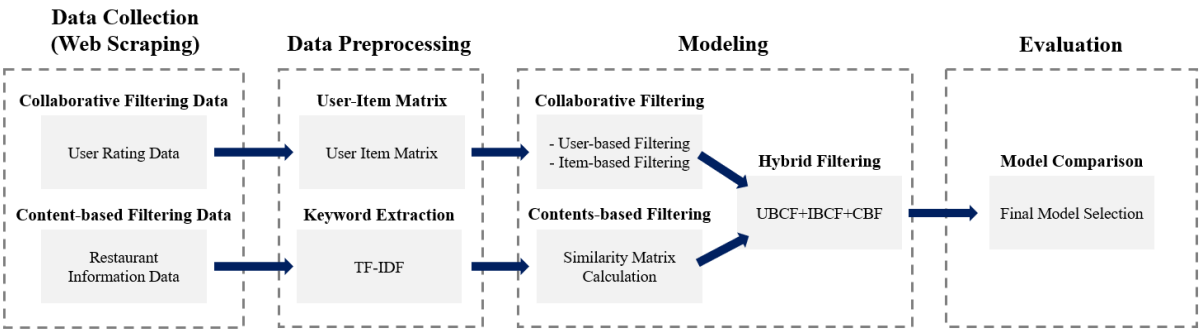
I. 서론

Netflix의 성공신화는 추천시스템으로부터 시작되었다. 1997년 DVD 대여사업으로 시작한 Netflix는 콘텐츠 온라인 스트리밍 서비스로 점차 영역을 넓혀갔고, 추천시스템의 상업화로 전 세계 7천5백만 명의 사용자를 확보하여 대표적인 콘텐츠 제공기업으로 자리 잡았다. 현재 콘텐츠 수요의 약 75%가 고객맞춤 추천으로 이루어진다고 하니, 그 영향력을 실감할 수 있다. 이후 Netflix를 시작으로 아마존, 구글 뉴스 등에서 다양한 콘텐츠들이 추천시스템을 통해 소비되었다.

추천 알고리즘의 개발을 위해 Netflix는 Netflix Prize라는 100만 달러의 상금이 걸린 제한 없는 공개 공모전을 개최했다. 기존 알고리즘보다 오차를 10% 넘게 줄이는 첫 번째 팀이 상금을 가져가는 이 공모전은 2007년부터 약 2년간에 걸쳐 이루어졌고, 데이터 애널리스트들의 수많은 추천 알고리즘의 연구 및 발전을 촉진했다.

이로 인해 추천시스템의 이윤 창출과 고객 만족에 대한 영향력과 알고리즘의 개발 및 테스트에 대한 흥미가 생겼는데, 그중에서도 콘텐츠 마케팅에 가장 많은 영향을 받는 음식점에 대한 추천시스템의 아이디어가 떠올랐다. 현재 망고플레이트, 식신 같은 음식점에 대한 단순 평가와 별점이 많은 순서대로의 음식점 나열을 한 홈페이지 및 애플리케이션은 존재하나, 개개인의 취향을 고려해 사용자 맞춤 음식점 추천을 해주는 시스템은 존재하지 않았다. 'YOLO', '소확행'이라는 단어에서 볼 수 있듯 현재 소비자들은 맛있는 것을 먹고 좋은 것을 보러 다니는 자신을 위한 소비가 많은 부분을 차지하며, 소비자 맞춤 음식점 추천 시스템의 파급력은 클 것으로 예상하였다.

이에 따라 새로운 것을 배우고 실용화시켜보자는 목표하에 '맛집 추천시스템 알고리즘 연구 및 개발'을 주제로 프로젝트를 진행하였다.



<알고리즘 순서도>

II. 데이터

2.1 개요

식당 관련 사이트 중 대한민국 최대 규모 "식신"에서 웹크롤링을 진행하였다. "식신"을 선정한 이유는 타 사이트 대비 압도적으로 많은 유저 리뷰와 구체적인 식당 소개가 있었기 때문이다. 수도권 지역인 서울, 경기, 인천에서 방문 된 모든 식당의 이름, 리뷰를 남긴 유저 아이디, 유저가 매긴 평점을 크롤링하였다. 이는 다음 [Ⅲ. 연구모델]에서 자세히 설명될 협업 필터링 (Collaborative Filtering)에 사용되는 평점 데이터이다. 또, 수도권 식당들에 대한 지역 정보, 업종 정보, 매장 소개 등을 웹크롤링 하였다. 그중 매장 소개는 비정형데이터로 텍스트 마이닝 기법을 통해 콘텐츠 기반 필터링 (Content-based Filtering)에 사용되는 정보이다.

2.2 Collaborative Filtering Data

식당 이름, 유저 아이디, 그리고 평점, 총 3가지 변수로 구성된 Collaborative Filtering용 데이터는 5단계를 거쳐 전처리를 진행하였다.

1) 먼저, 리뷰는 하였지만, 평점은 매기지 않아 NA로 남겨진 행을 삭제하였다.
2) 한 유저가 같은 식당을 여러 번 평가했다면 더 최근에 부여한 평점을 유지하였다.
3) 추천의 정확도와 평가를 위해 평점을 여섯 번 미만으로 받은 식당을 삭제하였다.
4) 추천의 정확도를 위해 평점을 여섯 개 이상 남긴 유저가 방문한 식당만을 유지하였다.
5) Long format의 데이터를 Wide format인 User-Item Matrix로 변환하였다.

2.3 Content-based Filtering Data

식당이름, 지역, 업종, 평균 평점, 매장소개로 이루어진 Content-based Filtering용 데이터는 총 4단계를 거쳐 전처리를 진행하였다.

1) 위의 Collaborative Filtering 데이터 전처리 과정에서 살아남은 식당 외 삭제하였다.
2) 추천의 정확도를 위해 매장소개의 글자 수가 100자 이하인 식당을 삭제하였다.
3) 프랜차이즈와 같은 체인점은 더 높은 평균 평점을 받은 지점으로 대체하였다.
4) 텍스트변수인 [매장소개] 변수에서 TF-IDF 기법을 통해 키워드를 추출하였다.

네 번째 단계인 TF-IDF 기법을 통한 키워드 추출은 다음 2.3.1에서 자세히 설명하겠다.

2.3.1 TF-IDF를 통한 키워드 추출

TF-IDF는 텍스트 마이닝에서 사용되는 가중치로 여러 문서로 이루어진 문서군이 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내는 통계적 수치이다. 단어에 중요도를 의미하는 수치를 부여하기 때문에 내림차순으로 정렬하여 상위 키워드를 추출할 수 있다.

하지만 단어 자체가 문서군 내에서 자주 사용되는 경우, 이것은 그 단어가 흔하게 등장한다는 것을 의미한다. 이것을 DF(문서 빈도, document frequency)라고 하며, 이 값의 역수를 IDF(역문서 빈도, inverse document frequency)라고 한다. TF-IDF는 TF와 IDF를 곱한 값이다.

TF(Term Frequency)는 특정 단어가 문서 내에서 얼마나 자주 등장하는가 나타내는 값이다. 단어의 등장이 잦을수록 문서에서 중요하다고 할 수 있다. 하지만 해당 단어가 문서군 내의 모든 문서에서 자주 등장한다면 관사나 조사와 같이 쓸모없는 단어일 가능성이 크다. 이를 해결하기 위해, 단어가 몇 개의 문서에서 나타나는지에 대한 값인 DF(Document Frequency)를 구하여 이의 역수인 IDF(Inverse Document Frequency)를 산출한다. TF-IDF는 TF와 IDF를 곱한 값이다.

매장소개 변수에서 해당 식당을 대표하는 키워드를 추출하기 전, 불필요한 문자를 제거하는 작업을 거쳤다. 먼저, 컴퓨터가 가공할 수 있는 언어로 만들어 주기 위해 Corpus 처리를 해주었다. 분석의 용이성을 위해 숫자, 구두점, 불필요한 공백 등을 제거하는 Parsing 과정을 거친 뒤, 남은 단어들의 어근만을 추출하는 Stemming을 진행하였다. 이후 TF-IDF를 통해 매장소개에 등장하는 단어들에 중요도를 나타내는 수치 값을 부여한 뒤, 이를 정렬하여 상위 5개의 키워드를 추출하였다. 최종적으로, Contents-based Filtering 데이터는 식당 이름, 지역, 업종, 키워드1, 키워드2, ..., 키워드5로 구성하였다.

III. 연구모델

3.1 Collaborative Filtering

3.1.1 개요

대표적인 추천시스템 알고리즘으로는 Collaborative filtering(협업 필터링)과 Content-based filtering (콘텐츠 기반 필터링)이 있다. 첫 번째 모델로 '사용자 간의 관계'에 초점을 맞추어 같은 취향의 사용자가 선호하는 아이템을 추천하는 Collaborative Filtering을 사용하였다. Collaborative Filtering의 기본적인 아이디어는 이렇다. 만약 두명의 사용자가 과거에도 비슷한 관심사를 가지고 있다면 그들은 미래에도 비슷한 취향을 가질 것이라는 얘기다. 아마존의 도서 추천 시스템이 이러한 추천 시스템의 좋은 예다. 협업 필터링은 사용자 간의 선호도를 서로 고려해 많은 선택 사항들 로부터 아이템을 걸러내는데 이에는 사용자 간 유사도 차이를 계산하는 방법, 아이템 간 유사도 차이를 계산하는 방법 두가지가 고려된다.

3.1.2 User-Based Collaborative Filtering

User-Based Collaborative Filtering(사용자 기반 협업 필터링)이란 대규모의 기존 사용자 행동 정보를 분석하여 '사용자 간의 유사도'를 계산해 해당 사용자와 비슷한 성향의 사용자들이 기존에 선호하던 항목을 추천하는 기술을 말한다. 예를 들어, 피자와 샐러드를 구매한 사용자들이 콜라를 구매한 경우가 많으면, 피자와 샐러드를 구매한 사용자에게 콜라를 추천하는 시스템을 말한다.

	나와의 유사도	item1 평점	item1 평점 × 나와의 유사도	item2 평점	item2 평점 × 나와의 유사도
user1	0.9	5	4.5	1	0.9
user2	0.4			3	1.2
user3	0.1	1	0.1	5	0.6
유사도×평점 합			4.6		2.9
유사도 합			1.0		1.4
평점 예측값			**4.6** (4.6 ÷ 1.0)		**2.0** (2.9 ÷ 1.4)

<User-Based Collaborative Filtering 계산 과정>

즉, 위와 같이 사용자 간의 유사도와 아이템1, 아이템2에 대한 다른 사용자들의 평점이 나열되어 있다면, 사용자 기반 협업 필터링의 원리에 따라 사용자 간의 유사도와 다른 사용자들의 아이템

에 대한 평점을 가중평균해 나의 평점 예측 값을 계산할 수 있다. 이 경우는 평점 예측 값이 더 높은 아이템 1을 추천하게 된다.

사용자 기반 협업 필터링의 장점으로는 첫째, 사용자들이 사이트를 사용하며 얻어지는 자료를 활용하므로 데이터 셋을 구축하기 수월하다는 점이 있다. 둘째, '유유상종'이라는 현실세계에서 잘 적용되는 아이디어에 착안해 좋은 성능을 보여준다는 장점을 지닌다.

반면, 단점으로 첫째는 초기에 정보가 부족해 신규 사용자의 경우 아이템의 선호도가 제대로 반영되지 않는다는 Cold Start 문제가 있다. 둘째는 사용자에게 소수의 항목만이 인기가 있으므로 사용자의 관심이 적은 다수의 항목은 추천을 위한 충분한 정보를 제공하지 못한다는 Long tail 문제가 있다.

3.1.3 Item-Based Collaborative Filtering

Item-Based Collaborative Filtering(아이템 기반 협업 필터링)이란 고객이 선호도를 입력한 기존의 상품들과 예측하고자 하는 '상품 간의 유사도'를 계산하여 고객의 선호도를 예측해 항목을 추천하는 기술을 말한다. 예를 들어, 특정 사용자가 A맥주를 구매했고, 여러 사용자들의 구매데이터들을 분석하여 A맥주와 B티셔츠의 유사도, 즉 '상품 간의 유사도'가 높다면, 사용자에게 B티셔츠를 추천하는 시스템을 말한다.

	나의 평점	item4와 유사도	나의 평점 × item4와 유사도	item5 유사도	나의 평점 × item5와 유사도
item1	5	0.9	4.5	0.1	0.5
item2	3	0.5	1.5	0.5	1.5
item3	1	0.1	0.1	0.9	0.9
유사도×평점 합			6.1		2.9
유사도 합			1.5		1.5
평점 예측값			**4.0** (6.1 ÷ 1.5)		**1.9** (2.9 ÷ 1.5)

<Item-Based Collaborative Filtering 계산 과정>

위와 같이 아이템들에 대한 나의 평점, 아이템들과 아이템 4,5와의 유사도가 나열되어 있다면, Item-Based Collaborative Filtering의 원리에 따라 아이템들에 대한 나의 평점과 각 상품 간의 유사도를 가중평균해 나의 평점 예측 값을 계산할 수 있다. 이 경우, 평점 예측 값이 더 높은 아이템 4를 추천하게 된다.

Item-Based Collaborative Filtering의 장점으로는 사용자 기반 필터링에서 발생하는 콜드 스타트 문제가 없이 새로운 유저에 대해 추천이 수월하고, 아이템 간의 유사도를 저장하며 사용할 수 있다는 점이 있다. 반면, 고객들 간의 유사도가 고려되지 않아 추천 시스템의 추천 능력이 저하될 수 있다는 단점을 지니고 있다.

3.1.4 매개변수 최적화

3.1.2와 3.1.3에 설명된 User-Based Collaborative Filtering과 Item-Based Collaborative Filtering은 사용자와 아이템 간의 유사도를 고려해 작동하고 이러한 유사도 측정 방법은 코사인 유사도, 피어슨 상관계수, 자카드 유사도를 포함한다. 그 중, 최적의 유사도 측정 방법을 결정하기 위해 ROC곡선을 사용하였다.

5-fold Cross-Validation을 통해 서로 다른 유사도 측정 방법을 사용했을 때 User-Based Collaborative Filtering과 Item-Based Collaborative Filtering이 그리는 ROC 곡선을 살펴보았다. 위의 그림과 같이 두 방법 모두 피어슨 상관계수를 유사도로 사용할 때 최적의 성능을 나타냈고 따라서 피어슨 상관계수를 유사도 측정 방법으로 선택하였다. 또, 모든 식당에 높은(혹은 낮은) 평점을 준 유저는 편향된 결과를 낳을 수 있어, 개개인의 평점 편향을 줄이기 위해 정규화(Normalize)를 하였다.

매개변수 최적화 이후, 서로 다른 두개의 Collaborative Filtering 기법 추천시스템을 적합하였다.

3.2 Content-based Filtering

3.2.1 개요

앞서 우리는 웹크롤링을 통해 식당들에 대한 지역, 업종(메뉴) 정보를 모으고 TF-IDF를 통해 매장 소개 글로부터 각 식당의 키워드를 추출한 바 있다. 이제 하나의 식당을 표현하는 변수가 준비되었으니 이를 통해 식당간 거리를 계산하여 최종 추천이 이루어질 수 있다. 사실 이 과정은 앞선 Item-based Collaborative Filtering의 방식과 크게 다르지 않다. 다만 IBCF에서는 User-Item Rating Matrix를 통해 Item 간의 거리를 계산했다면, Content-Based Filtering에서는 구해진 텍스트 변수를 활용하여 거리를 계산하는 것이 차이점이라 할 수 있겠다.

3.2.2 유사도 행렬 계산 방법

Content-based Filtering용 데이터 전처리 후 식당에 대한 데이터 예시는 다음과 같다.

	Location1	Location2	Menu1	Menu2	Keyword1	Keyword2	Keyword3	Keyword4	keyword5
율전돈가스	경기	수원	세계음식	돈 가스	cutlet	pork	soup	japan	smart
혜화스시	서울-강북	대학로	세계음식	초밥	sushi	largest	japan	variety	enjoy
강남쌀국수	서울-강남	강남역	세계음식	쌀국수	soup	noodle	flavor	vietnam	hot

Item간의 유사도가 높다는 것은 두 Item이 공유하는 정보가 많다는 것을 의미한다. 위 예시를 보면 '율전돈가스'와 '혜화스시'는 Menu1과 Keyword4의 japan이라는 변수를 공유한다. 그렇다면 우리는 두 Item간의 거리를 계산하기 위해 다음과 같은 방법을 고안할 수 있을 것이다.

	Location1	Location2	Menu1	Menu2	Keyword1	Keyword2	Keyword3	Keyword4	keyword5
율전돈가스	1	1	1	1	1	1	1	1	1
혜화스시	0	0	1	0	0	0	1	0	0
강남쌀국수									

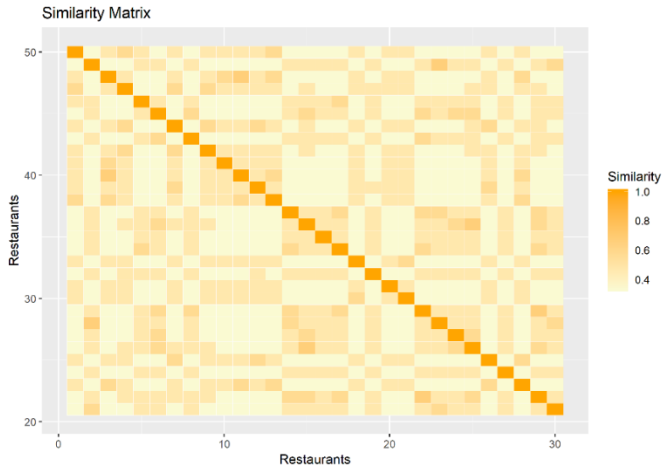
기준 Item값을 모두 1로 채우고 비교하려는 Item의 변수 중 기준 Item과 겹치는 것들만 1로 채워주면 0과 1로 이루어진 두 Vector가 구해진다. 위에서는 Menu1이 겹치고 Japan이라는 keyword를 두 Item 모두 공유하고 있기 때문에 해당하는 두 Feature만 1이 되고 나머지는 0으로 채워진다. 이제 이 두 Vector간의 유사도를 구할 수 있다. 유사도는 Cosine 유사도를 사용했으며 위의 예시에서 두 식당의 유사도는 다음과 같이 구해진다.

$$\text{율전돈가스 } \mathrm{a} = (1,1,1,1,1,1,1,1)$$

$$\text{혜화스시 } \mathrm{b} = (0,0,1,0,0,0,1,0,0)$$

$$\therefore \cos\theta = \frac{a \cdot b}{\|a\|\|b\|} = \frac{\sqrt{2}}{3} \approx 0.4714$$

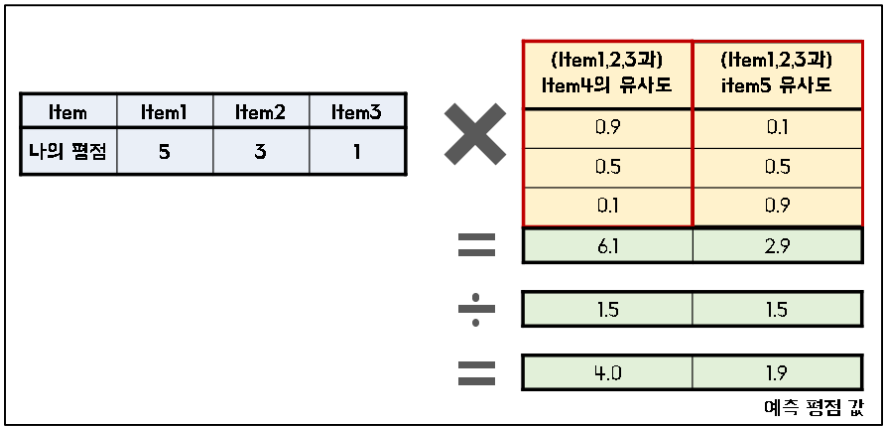
이와 같은 방법으로 모든 Item간의 유사도를 구하여 Item-Item 유사도 행렬을 구할 수 있다. 대각성분은 모두 1인 대칭행렬의 형태이며, 차원은 고유한 식당 개수인 2658개가 된다.



<Contents-Based Filtering에서 사용되는 유사도 행렬>

3.2.3 Contents-Based Filtering 추천 방식

이후의 과정은 앞서 언급했듯 Item-Based Collaborative Filtering의 과정과 유사하다. 유저가 식당에 기록한 평점 벡터를 식당 수만큼의 차원으로 구해진 유사도 행렬에 곱하여 해당 유저의 모든 식당에 대한 예측평점을 구하는 것이다. 예를 들어 어떤 유저가 식당 1, 2, 3에 각각 5점, 3점, 1점의 평점을 남겼고 식당 4, 5는 방문한 적이 없다고 하자. 우리는 위의 과정에서 식당 간의 유사도 행렬을 텍스트변수를 활용하여 구한 바 있다. 아래의 그림처럼 식당 1, 2 ,3에 대한 평점을 유사도 행렬의 식당 4, 5에 해당하는 부분과 곱해준 후, 유사도 합으로 나눠주면 식당 4와 식당 5에 대한 유저의 예측평점을 구할 수 있게 된다. 이렇게 유저가 방문하지 않은 모든 식당에 대한 예측 평점을 구하여 그것이 가장 큰 Top N개의 식당을 추천해주는 방식이다.



<Contents-Based Filtering 계산>

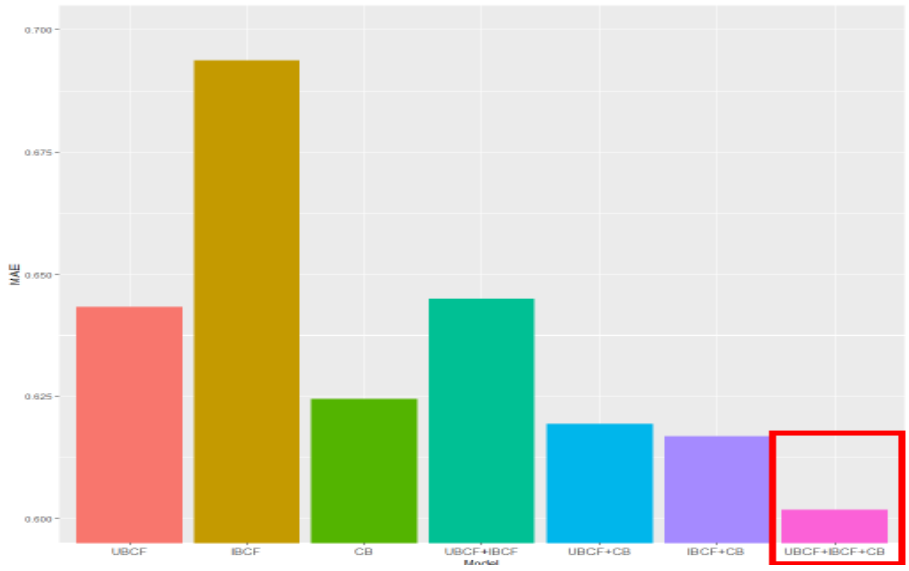
3.3 Hybrid Filtering

Hybrid Filtering은 지금까지 살펴본 UBCF, IBCF, CBF의 세 모델 결과를 앙상블하는 과정이다. 모델 앙상블은 기본적으로 서로 다른 알고리즘을 가진 여러 모델들이 각자가 가진 단점을 상호보완하여 오차를 줄일 수 있다는 가정에서 시작한다. 우리의 추천시스템도 평점을 통한 유사도 행렬을 활용하는 Collaborative Filtering 모델들과, 콘텐츠를 통한 유사도 행렬을 활용하는 Contents-Based Filtering 모델이 긍정적으로 종합될 수 있다고 판단해 앙상블을 진행하였다. 모델 앙상블은 결과값, 즉 예측평점의 단순 평균값을 사용했으며 UBCF+IBCF, UBCF+CBF, IBCF+CBF, UBCF+IBCF+CBF 네 가지 경우를 모두 고려하여 최종 평가하였다.

IV. 모델평가

우리가 만든 모델의 출력 값은 각 식당들의 예상 평점이고 더 나아가, 추천시스템의 출력 값은 평점기준 상위 N개 식당명이다. 즉, 모델 Output의 Class로 연속형과 범주형 둘 다 가능하므로 연속형과 범주형 평가지표를 사용할 수 있다. 추천시스템에 적합한 연속형 평가지표로는 Mean Absolute Error(MAE)가 있고 범주형 평가지표로는 Precision과 Recall이 있다.

Precision은 추천시스템이 추천한 식당 중 사용자가 실제로 선호하는 식당인 비율, Recall은 실제로 사용자가 선호하는 식당 중 추천받은 식당의 비율이다. 하지만, 가지고 있는 데이터 셋의 Sparsity가 높아 (즉, 사용자가 평가하는 식당들의 극히 일부만 평가했기 때문에) Precision, Recall이 낮게 나올 수밖에 없다는 한계점이 있다. 이러한 데이터 특유의 한계점을 고려해 사용자가 평가한 식당의 평점과 모델이 구한 예상 평점 간의 차이의 절댓값의 합인 MAE를 평가요소로 선정했다.



<개별 모델 및 앙상블 모델 Mean Absolute Error>

예상대로 모든 모델을 앙상블한 모델(IBCF+UBCF+CBF)의 MAE가 가장 낮았다.

V. 결론

오랜만에 친구를 만나서 식사하기로 했는데 메뉴 절충이 도무지 안돼서 길에서 하염없이 30분 이상 걸은 적이 누구나 있을 것이다. 그리고 한껏 기대에 부푼 마음으로 블로그나 맛집 추천 어플리케이션에서 입을 모아 추천하는 식당에 갔는데, 그 기대가 박살 난 적이 한 두 번이 아닐 것이다. 그런 분들을 위해 솔루션을 제공한다. 입맛이 서로 다른 친구와 식사를 하는 경우, 지금 당장 먹고 싶은 음식을 정하지 못하는 결정 장애를 위해서, 빅데이터에 의존하는 개인화 맛집 추천시스템을 활용하고자 한다.