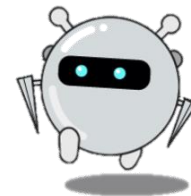


추천시스템

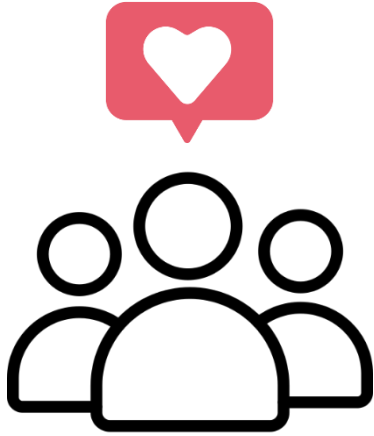


OUTTA 이윤지

1. 개요

추천시스템이란?

다양한 아이템 중 사용자가 선호할 아이템을 제공하는 시스템



사용자 만족도 상승

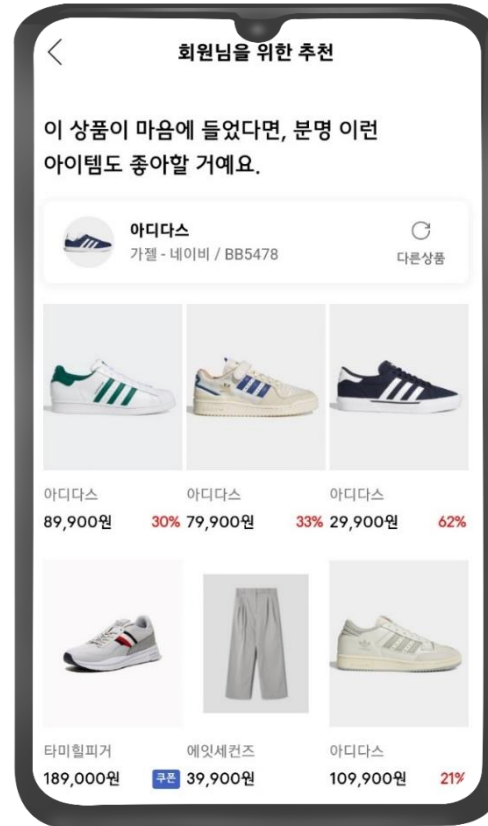


서비스 제공자 수익 상승

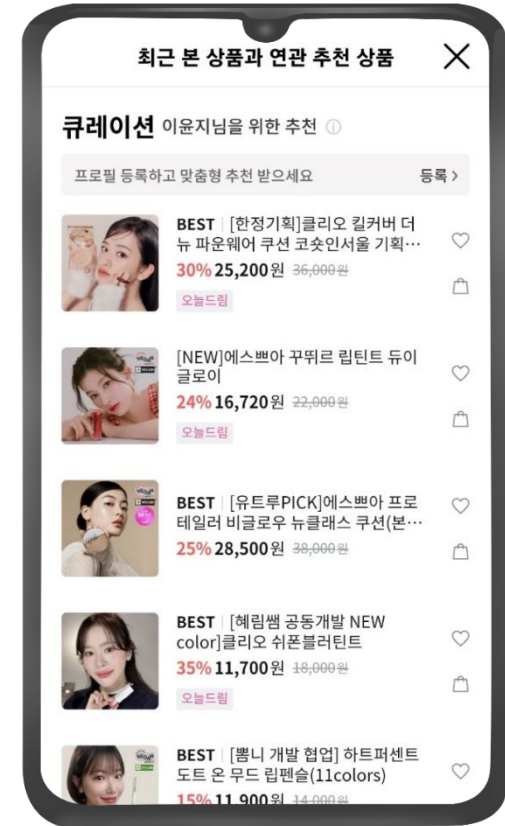
일상 속 추천시스템



▲ 넷플릭스



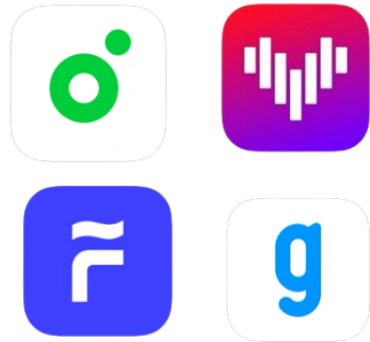
▲ 무신사



▲ 올리브영

다양한 추천시스템

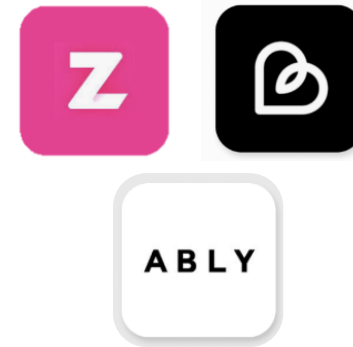
음악



친구



패션



영화



Netflix Prize

- 넷플릭스 상

2006년부터 2009년까지 진행된 넷플릭스의 온라인 영화 추천 시스템 개선 대회

- 대회 목표

넷플릭스의 영화 추천 정확도를 10% 이상 개선

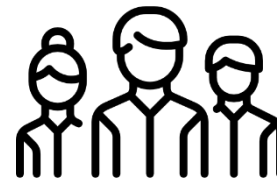
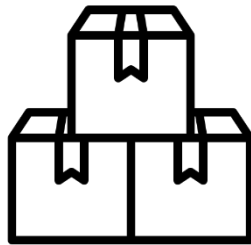
- 상금 기준

넷플릭스의 자체 알고리즘인 Cinematch의 성능보다 10%이상 향상한 경우 100만 달러

고민해봐야 할 것

- 어떤 사용자에게 무슨 아이템을 추천할 것인가
사용자의 취향, 흥미, 의도, 상황에 맞는 아이템
- 어떤 데이터를 활용할 것인가

유저 데이터 - 나이 성별 직업 | 행동 데이터 - 클릭 구매 평가 | 아이템 데이터 - 가격 색상 이름



2. 분류

추천시스템 분류

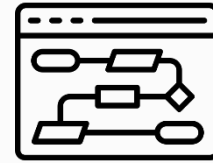
추천목적



데이터



알고리즘



모델



● 추천목적

데이터

알고리즘

모델

1. Best Recommendation

단 7일, 지금 가입하고 1만원 할인 쿠폰 받아주세요!

회원가입 | 로그인 | 고객센터

Kurly마켓컬리 | 뷰티컬리

검색어를 입력해주세요

📍❤️🛒

셋별 · 택배 배송안내

카테고리

신상품

베스트

알뜰쇼핑

특가/혜택

베스트

필터

초기화

총 291건

추천순 | 신상품순 | 판매량순 | 혜택순 | 낮은 가격순 | 높은 가격순

카테고리

샐러드·간편식 72

국·반찬·메인요리 45

정육·계란 38

과일·견과·쌀 27

간식·과자·떡 19

생수·음료·우유·커피 16


수산물·해산물·건어물 14

베이커리·치즈·델리 13


해물·바다·구강 11

생활용품·리빙·캠핑 11


카테고리 더보기



셋별배송
[사미현] 갈비탕
진짜 갈비로 우려낸 전통 갈비탕
12,000원
후기 9,999+



셋별배송
[KF365] 양념 소불고기 1kg (냉장)
100g당 가격: 1,899원
5% 18,990원
19,990원



셋별배송
[태우한우] 1+ 한우 안심 구이용 200g (냉장)
100g당 가격: 19,950원
39,000원

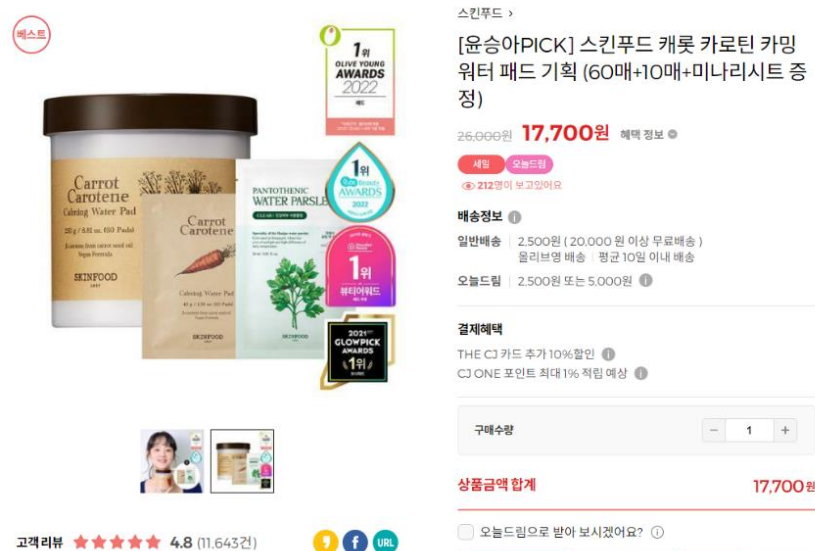
▲ 마켓컬리

2. Related Recommendation

데이터

알고리즘

모델



대체재



보완재



▲ 올리브영

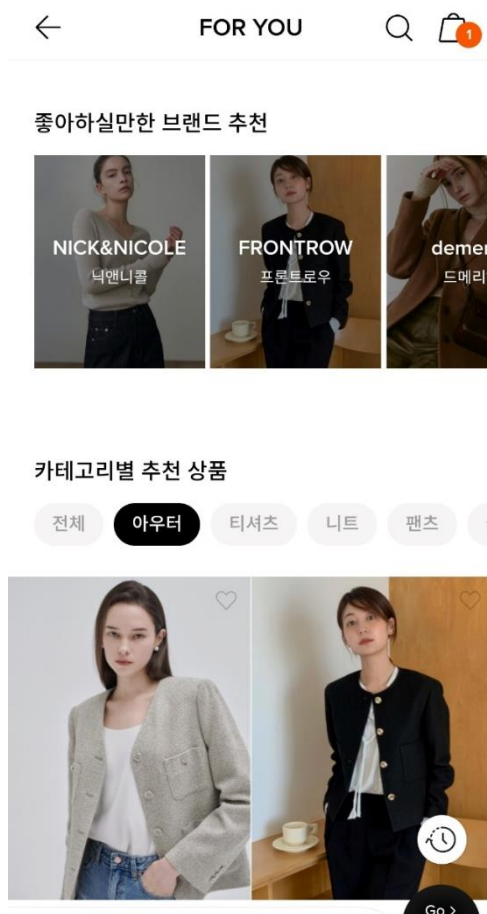
추천목적

데이터

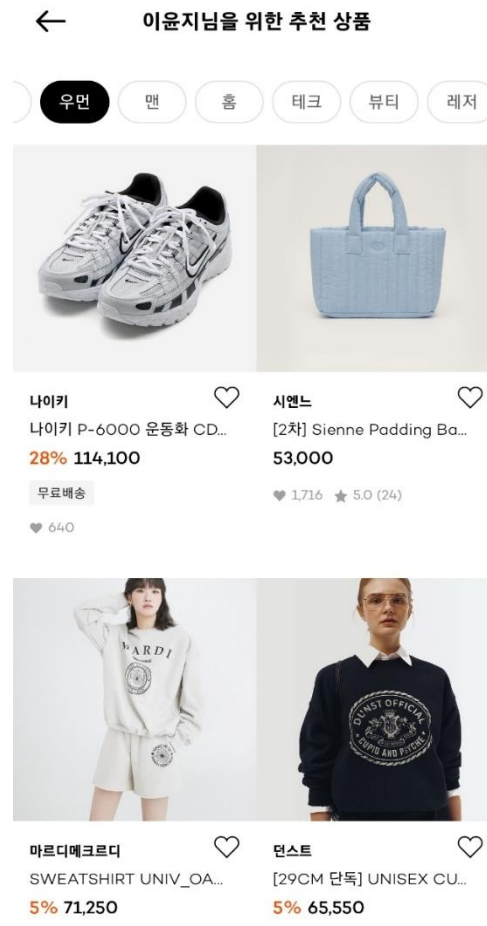
알고리즘

모델

3. Personalized Recommendation



▲더블유컨셉



▲29CM

- 추천목적

데이터

알고리즘

모델

4. Context-aware Recommendation

회색 구름으로 채워진 날

서울 🌱



부드러운 록에 취해
#얼터너티브록 #리드미컬



흥+흥 댄스 맛집
#댄스 #흥

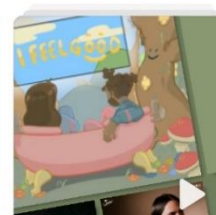


돌고
#가요

봄의 설렘을 담은 뮤직



봄바람 맞으며 듣는 감성 발라드



기분 전환 하기 좋은 리드미컬 팝

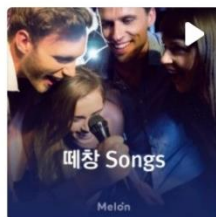


감성 R&B

나만의 음악으로 채우는 밤



미니멀한 비트 + 뽀센 랩 = 근본
#외힙 #dope



떼창 Songs
#노래방 #기분전환



Chill
#저브

밤에 듣기 좋은 음악



어둑해진 하늘에 띄운 우타이테



소중한 아이를 위한 명품 클래식 자장가



파티 EDM

→ 행복했던 날들이었다
DAY6 (데이식스)



재생목록이 비었습니다.



▲멜론

▲FLO

추천목적

• 데이터

알고리즘

모델

데이터에 따른 추천시스템 분류

Explicit Data

사용자가 아이템에 대한 자신의 선호도를 분명하게 표현한 데이터

- 평점
- 좋아요/싫어요
- 관심 상품/찜



Rate

-> 수집이 어렵다

Implicit Data

사용자가 아이템에 대해 간접적으로 선호도, 취향을 나타내는 데이터

- 클릭
- 시청
- 구매



Consume

-> 선호도 파악이 어렵다

추천목적

데이터

• 알고리즘

모델

알고리즘에 따른 추천시스템 분류

Content-based Filtering

사용자가 어떤 아이템을 선호
하면 해당 아이템의 속성을
파악해 유사한 아이템을 추천

가시적 특성을 기반으로 아이템
유사도 측정

Item cold-start 문제 X

User cold-start 문제
다양성이 떨어지는 추천

Collaborative Filtering

아이템과 사용자의 관계를 이
용하여 사용자의 흥미에 맞는
아이템을 추천

잠재적 특성을 기반으로 아이템,
사용자의 유사도 측정

전반적으로 추천 정확도가 높음

Item/User cold start
문제 발생

Hybrid

Content-based
&
Collaborative
Filtering

아이템 속성, 사용자 행동 이력
데이터 모두 사용한 추천 시스템

추천목적

데이터

알고리즘

- 모델

모델에 따른 추천시스템 분류

Rating Prediction

점수를 예측하고 이를 토대로 추천하는 방식

e.g. 왓챠 - 사용자가 콘텐츠에 남길 평점을 예측하여 추천

Top - k Recommendation

정확한 점수 예측보다는 아이템의 순위에 초점을 둔 방식

3. 영화 추천 시스템

학습 목표

“

사용자가 영화에 남길 평점을 예측하여 영화를 추천

”

효과 예상 평점을 제시하여 작품 선택에 도움을 줌 | 유사한 작품 추천 가능

실습 데이터

● Movielens Dataset

671명의 사용자 | 약 9000개의 영화 | 약 10만 개의 평점

출처 : GroupLens Research - ([무비렌즈](http://movielens.org) | [그룹렌즈 \(grouplens.org\)](http://grouplens.org))

● 평점 예측

UserId	Movielid	Title	genres	Rating
12	0	Toy Story	Adventure Animation Children Comedy Fantasy	4.1
12	1	Jumanji	Adventure Children Fantasy	3.2
12	2	Grumpier Old Men	Comedy Romance	예측
12	3	Waiting to Exhale	Comedy Drama Romance	3.7

실습할 추천시스템

추천목적



Personalized

데이터



Explicit

알고리즘



CBF, CF

모델



Rating Prediction

4. CBF

CBF (content - based filtering)

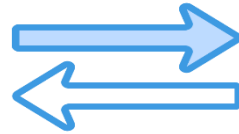
유사한 두 영화에 대해 비슷한 평점을 남길 것이라는 생각에서 출발

<원스>

- 드라마
- 멜로/로맨스



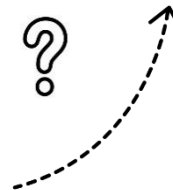
“유사한 장르”



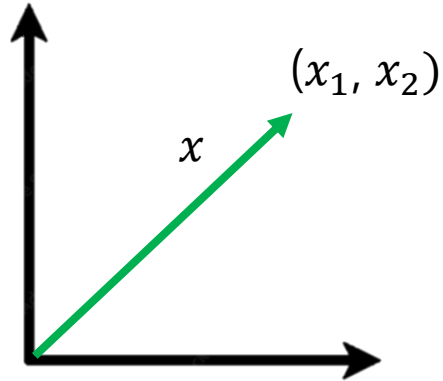
<비긴 어게인>

- 드라마
- 멜로/로맨스
- 코미디

3.9점



사전지식 for 유사도



장르의 벡터화

genres	Adventure	Comedy	Fantasy	Romance
비긴 어게인	0	1	0	1

$$b = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

내적

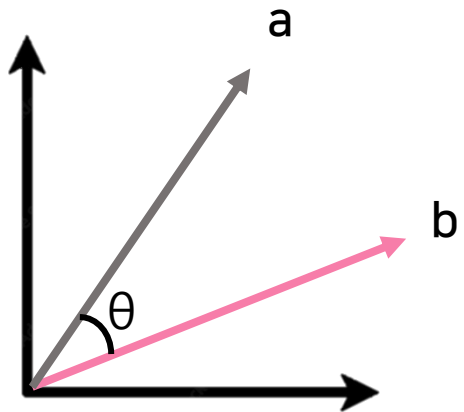
$$a_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad a_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} a_1 \cdot a_2 &= a_1^T a_2 \\ &= 1 \times 1 + 2 \times 1 + 3 \times 0 + 4 \times 1 = 7 \end{aligned}$$

Euclidean Norm (l_2)

$$\begin{aligned} v &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} & \|v\| \\ &= \sqrt{v \cdot v} \\ &= \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2} \end{aligned}$$

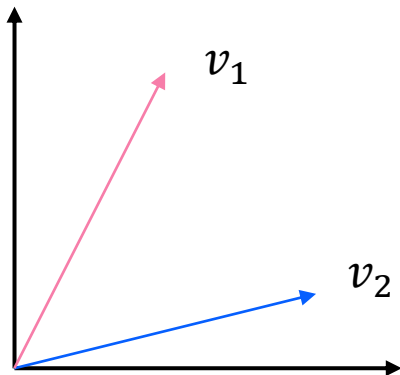
코사인 유사도



코사인 유사도

$$\text{sim}(a, b) = \cos \theta = \frac{a \cdot b}{\|a\| \|b\|}$$

예제 $v_1 = [2, 4]$, $v_2 = [4, 1]$ 일 때, 두 벡터의 코사인 유사도는?



$$\begin{aligned} \text{sim}(v_1, v_2) &= \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|} \\ &= \frac{2 \times 4 + 4 \times 1}{\sqrt{2^2 + 4^2} \sqrt{4^2 + 1^2}} \\ &= 0.65 \end{aligned}$$

Bag of word

Beautiful 크러쉬

It's a beautiful life

난 너의 곁에 있을게

It's a beautiful life

너의 뒤에 서 있을게

beautiful love



Bag of word



<Bag of word embedding>

단어	beautiful	It's	life	a	너의	있을게	난	곁에	뒤에	서	love
개수	3	2	2	2	2	2	1	1	1	1	1

TF-IDF

Term Frequency-Inverse Document Frequency

$$\text{TF-IDF}(w,d) = \text{TF}(w,d) \times \log\left(\frac{N}{1+\text{DF}(w)}\right)$$
$$\text{IDF}(w) = \log\left(\frac{N}{1+\text{DF}(w)}\right)$$

- $\text{TF}(w)$: 특정 단어 w 가 특정 문서 d 에 나온 빈도
- $\text{DF}(w)$: 특정 단어 w 가 나타난 문서의 수
- $\text{IDF}(w)$: 전체 문서 수 N 을 해당 단어의 DF 로 나눈 뒤 로그 취한 값
- N 은 전체 문서 수

예제

d1: 귀여운 강아지

d2: 물을 마시는 강아지 고양이

d3: 잠자는 고양이

단어	귀여운	강아지	고양이	물을	마시는	잠자는
DF						
IDF						
TF	d1 d2 d3					
TF-IDF	d1 d2 d3					

TF-IDF

Term Frequency-Inverse Document Frequency

예제 답

d1: 귀여운 강아지

d2: 물을 마시는 강아지 고양이

d3: 잠자는 고양이



단어	귀여운	강아지	고양이	물을	마시는	잠자는
DF	1	2	2	1	1	1
IDF	$\log(1.5)$	$\log(1)$	$\log(1)$	$\log(1.5)$	$\log(1.5)$	$\log(1.5)$
TF	d1	1	0	0	0	0
	d2	0	1	1	1	0
	d3	0	0	0	0	1
TF-IDF	d1	$\log(1.5)$	$\log(1)$	0	0	0
	d2	0	0	$\log(1)$	$\log(1.5)$	$\log(1.5)$
	d3	0	0	$\log(1)$	0	$\log(1.5)$

TF-IDF 이용한 영화 유사도

TF-IDF 결과 예시

MovieID	Adventure	Fantasy	Comedy	Romance	Drama	Thriller
1	0.00	0.00	0.38	0.00	0.59	0.00
2	0.12	0.66	0.45	0.15	0.00	0.00
3	0.27	0.00	0.00	0.62	0.34	1.03
4	0.00	0.13	0.22	0.18	0.55	0.00

영화 ID 1과 2 유사도

$$\text{sim}(a, b) = \frac{a^T b}{\|a\| \cdot \|b\|} \quad \text{sim}(1,2) = \frac{0.38 \times 0.45}{\sqrt{0.59^2 + 0.38^2} \cdot \sqrt{(0.12^2 + 0.66^2 + 0.45^2 + 0.15^2)}} = \frac{0.171}{0.7018 \times 0.8216} = 0.2966$$

CBF 평점 예측

$$\hat{r}_{u,i} = \frac{\sum_{j \in I_u} sim(i,j) \cdot r_{u,j}}{\sum_{j \in I_u} sim(i,j)}$$

사용자 u가 영화 i에 남길 예상 평점

- I_u 사용자 u가 평점을 남긴 영화 전체 집합
- $r_{u,j}$ 사용자 u가 영화 j에 남긴 평점
- $sim(i,j)$ 영화 i와 j의 유사도

유사도와 평점

<예시 : 영화 컨택트 예상 평점 구해보기>



컨택트(Arrival)

영화 (I_u)	평점 $r_{u,j}$	$sim(\text{컨택트}, j)$	$sim(\text{컨택트}, j) \cdot r_{u,j}$
더 기버:기억 전달자	3.1	0.4	1.24
인터스텔라	4.2	0.35	1.47
더 문	3.8	0.6	2.28
합계		1.35	4.99

$$\hat{r}_{u,i} = \frac{\sum_{j \in I_u} sim(i, j) \cdot r_{u,j}}{\sum_{j \in I_u} sim(i, j)}$$

사용자 u가 영화 i에 남길 예상 평점

// 사용자 u가 컨택트에 남길 예상 평점 : $\frac{4.99}{1.35} = 3.696$ //

5. CF

CF (collaborative filtering)

사용자들이 영화에 남긴 평점 즉, 행동 이력을 통해 영화의 평점을 예측하여 추천

학습 내용

01 ITEM – BASED CF 아이템 간의 유사도를 구하여 영화의 평점을 예측해 추천하는 방식

02 USER – BASED CF 사용자 간의 유사도를 구하여 영화의 평점을 예측해 추천하는 방식

Item-based CF (collaborative filtering)

유사한 두 영화에 대해 비슷한 평점을 남길 것이라는 생각에서 출발

➡ CBF와 동일한 가정 단, 유사도 구하는 과정에 차이 있음

CBF

아이템 속성인 영화 장르를 이용한 유사도 측정

장르	비긴 어게인	어바웃 타임
Adventure	0	0
Comedy	1	1
Romance	1	1
Drama	1	0

Item-based CF

사용자들이 영화에 남긴 평점을 이용하여 유사도 측정

사용자 ID	비긴 어게인	어바웃 타임
1	4.5	3.9
2	4.3	4.4
3	3.6	4.0
4	3.7	3.5

Item-based CF (collaborative filtering)

사용자 u 가 영화 i 에 남길 예상 평점

$$\hat{r}_{u,i} = \frac{\sum_{j \in I_u} \text{sim}(i, j) \cdot r_{u,j}}{\sum_{j \in I_u} \text{sim}(i, j)}$$

- I_u 사용자 u 가 평점을 남긴 영화 전체 집합
- $r_{u,j}$ 사용자 u 가 영화 j 에 남긴 평점

영화 a 와 b 의 평점을 이용한 유사도

$$\text{sim}(a, b) = \frac{a^T b}{\|a\| \|b\|}$$

- 영화 장르의 TF-IDF를 계산하지 않고 영화에 남겨진 평점으로 유사도 계산

Item-based CF (collaborative filtering)

평점을 이용한 영화 유사도 구하기 주의 : 두 영화에 대해 평점 기록이 모두 있는 경우에만 계산

사용자 ID	A	B	C	D
1	4.8	3.6	3.7	1.5
2	2.7	4.3		4.6
3	5.0	4.1	4.4	1.8
4		3.7	4.1	3.6

▲영화 평점 예시

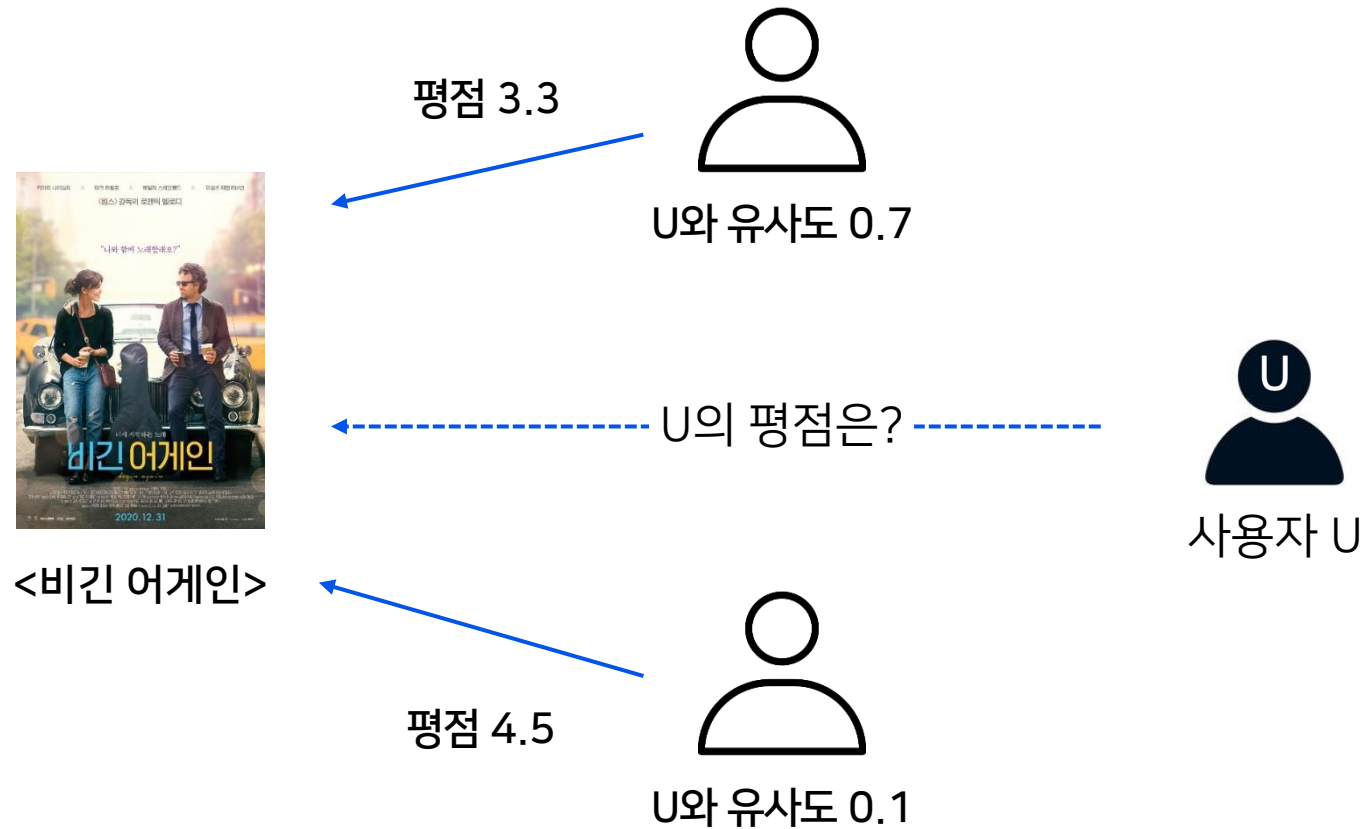
예시 영화 A와 D의 유사도 구하기

$$\text{sim}(a, b) = \frac{a^T b}{\|a\| \|b\|}$$

$$\begin{aligned}\text{sim}(A, D) &= \frac{4.8 \times 1.5 + 2.7 \times 4.6 + 5.0 \times 1.8}{\sqrt{4.8^2 + 2.7^2 + 5.0^2} \sqrt{1.5^2 + 4.6^2 + 1.8^2}} \\ &= \frac{28.62}{29} = 0.74\end{aligned}$$

User-based CF (collaborative filtering)

유사한 유저는 같은 영화에 대해 비슷한 평점을 남길 것이라는 생각에서 출발



“사용자 간의 유사도를 가중치처럼 적용하여 예상 평점을 구함”

User-based CF (collaborative filtering)

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in U_i} \text{sim}(u, v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in U_i} \text{sim}(u, v)}$$

사용자 u가 영화 i에 남길 예상 평점

- U_i 영화 i에 대해 평점을 남긴 사용자 전체 집합
- $r_{v,i}$ 사용자 v가 영화 i에 남긴 평점
- \bar{r}_v 사용자 v가 영화에 남긴 평점들의 평균
- \bar{r}_u 사용자 u가 영화에 남긴 평점들의 평균
- $\text{sim}(u, v)$ 사용자 u와 v의 유사도

User-based CF (collaborative filtering)

평점을 이용한 유저 유사도 구하기

주의 : 두 유저에 대해 평점 기록이 모두 있는 경우에만 계산

사용자 ID	A	B	C	D
1	4.8	3.6	3.7	1.5
2	2.7	4.3		4.6
3	5.0	4.1	4.4	1.8
4		3.7	4.1	3.6

▲영화 평점 예시

예시 유저 1과 4의 유사도 구하기

$$\text{sim}(a, b) = \frac{a^T b}{\|a\| \|b\|}$$

$$\begin{aligned}\text{sim}(1, 4) &= \frac{3.6 \times 3.7 + 3.7 \times 4.1 + 1.5 \times 3.6}{\sqrt{3.6^2 + 3.7^2 + 1.5^2} \sqrt{3.7^2 + 4.1^2 + 3.6^2}} \\ &= \frac{33.89}{35.44} = 0.95\end{aligned}$$

User-based 유사도와 평점

<예시 : 사용자 u의 컨택트(i) 예상 평점 구하기>

	사용자 1 (v=1)	사용자 2 (v=2)	사용자 3 (v=3)
컨택트 ($r_{v,i}$)	4.4	3.5	4.7
평점 평균 ($\overline{r_v}$)	4.2	3.9	4.15
사용자 유사도 ($sim(u, v)$)	0.6	0.3	0.8

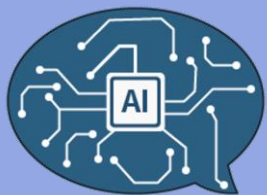
$$\hat{r}_{u,i} = \overline{r_u} + \frac{\sum_{v \in U_i} sim(u, v) \cdot (r_{v,i} - \overline{r_v})}{\sum_{v \in U_i} sim(u, v)}$$

$$\begin{aligned}\hat{r}_{u,i} &= 4.0 + \frac{0.6 \times 0.2 + 0.3 \times (-0.4) + 0.8 \times 0.55}{0.6 + 0.3 + 0.8} \\ &= 4.259\end{aligned}$$



사용자 u의 평점 평균 ($\overline{r_u}$) : 4.0

사용자 u가 컨택트에 남길 예상 평점 : 4.259



감사합니다

