Recommender를 이용한 영화 추천

경제학부 응용통계전공 / 강도형

경제학부 응용통계전공 / 김기선

경제학부 응용통계전공 / 김수현

경제학부 응용통계전공 / 이영훈

경제학부 응용통계전공 / 이호창

- 사례 소개
  - 왓챠플레이
  - 넷플릭스
- Recommender System
- Dataset
- 결론 및 코드 설명

영화 추천시스템의 대표적 사례

WATCHA PLAY

상세한 컨텐츠 분석 데이터

취향기록 및 개인화 추천

예상 별점으로
사용자에게 추천

'보고싶어요'등으로
추천 받은 컨텐츠 차후추가

영화 관련 각종
태그를 이용한 추천

사용자의 컨텐츠 평가 데이터
5점 만점

유사한 취향을 보인
다른 유저의 시청내역 추천

영화 추천시스템의 대표적 사례

NETFLIX

영화와 관련된 여러 정보
장르, 배우, 출시연도,
러닝타임 등

하루중 시청 시간대
출근, 퇴근, 여가 시간 등을
고려

사용자의 컨텐츠 평가
Ex)좋아요, 별로 에요

시청하는 디바이스
패드, 모바일, 컴퓨터 등

영화 관련 각종
태그를 이용한 추천

사용자의 컨텐츠 평점 데이터
(시청기록, 다른 컨텐츠 평가)

유사한 취향을 보인
다른 유저의 시청내역 추천

협업 필터링 이란?

- 정의

- 많은 사용자들로부터 얻은 기호정보(taste information)에 따라 사용자들의 관심사들을 자동적으로 예측하게 해주는 방법

- 특징

- 잠재적 특징을 고려하여 다양한 추천 범위를 가짐

- 복잡한 아이템을 어렵게 분석할 필요 없음

- **평가내역이 없는 영화의 경우 추천이 어려움**

- UBCF방식과 IBCF방식으로 나뉨

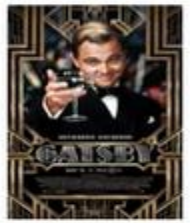
Which movie will you recommend for user 3?

user1

user2

user3

user4

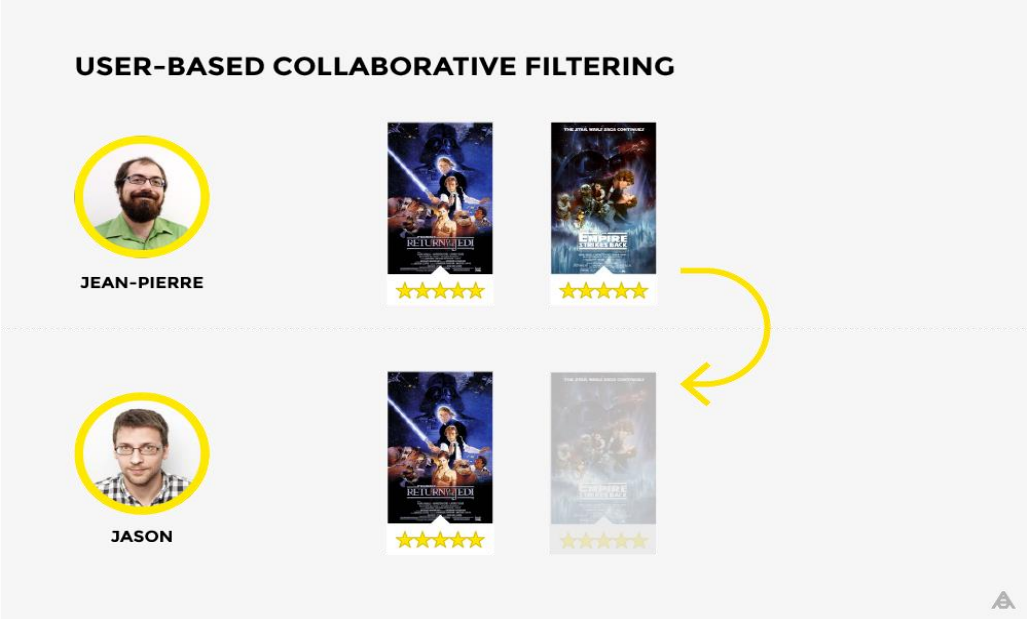


협업 필터링의 종류와 개념

User based

취향이 비슷한 유저B가 어떤 아이템을
구매했는지 확인 후 B가 구매한 상품을 추천

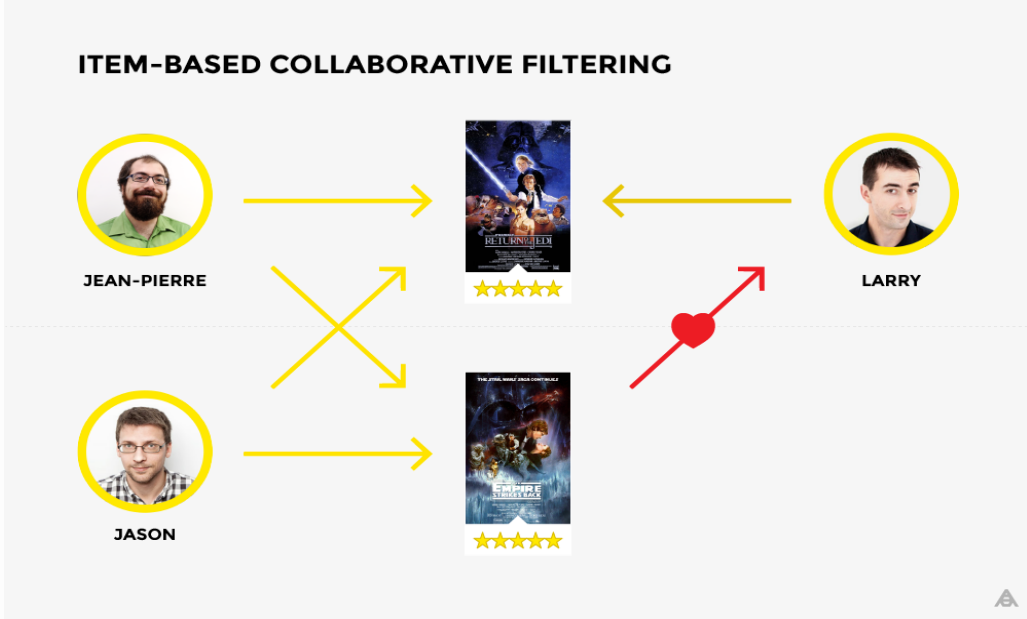
특징
- 사용자 수가 많을 수록 메모리가 많이 소요
- 시간이 지남에 따라 사용자의 특징이 변화



Item based

내가 구매했던 상품들을 기반으로,
연관성이 있는 상품을 추천

특징
- 사용자 대비 아이템 수가 적어 메모리 부담 완화
- 시간이 지남에 따른 아이템의 특징 변화가 드뭄



유사도 계산

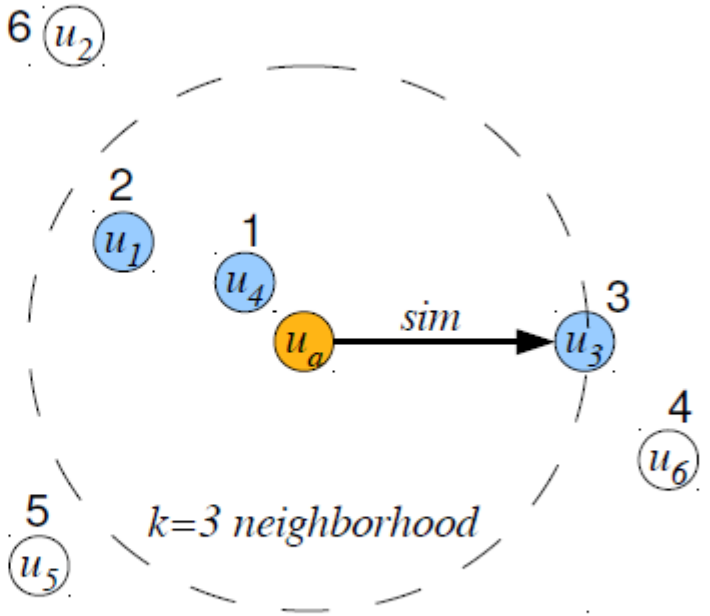
- 유사도는 피어슨 유사도, 유클리디안 거리, 코사인 유사도 등 사용 가능
- 코사인 유사도를 기본으로 설명

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

UBCF(User Based Collaborate filtering)

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
u_a	?	?	4.0	3.0	?	1.0	?	5.0
$\hat{r}_a$	3.5	4.0			1.3		2.0	

(a)



(b)

UBCF(User Based Collaborate filtering)

$$u_a, u_1 \text{코사인 유사도} : \frac{4*4+2*3+1*2}{\sqrt{4^2+3^2+1^2} * \sqrt{4^2+2^2+2^2}} = \frac{16+6+2}{\sqrt{24} * \sqrt{26}} = 0.96$$

$$u_a, u_2 \text{코사인 유사도} : \frac{3*3+2*1+3*5}{\sqrt{3^2+1^2+5^2} * \sqrt{3^2+2^2+3^2}} = \frac{9+2+15}{\sqrt{35} * \sqrt{20}} = 0.98$$

$$u_a, u_4 \text{코사인 유사도} : \frac{3*2+1*1+5*4}{\sqrt{3^2+1^2+5^2} * \sqrt{2^2+1^2+4^2}} = \frac{6+1+20}{\sqrt{35} * \sqrt{21}} = 0.99$$

$$r_{a(i1)} = \frac{3+4}{2}, r_{a(i2)} = \frac{4}{1}, r_{a(i5)} = \frac{1+2+1}{3}, r_{a(i7)} = \frac{2}{1}$$

1. 분석하고 싶은 대상과 모든 대상의 코사인 유사도 도출
2. 코사인 유사도가 큰 k개의 대상을 선택
3. K의 개의 대상이 대상자가 시청하지 않은 것을 예상 평점 도출
4. 평점이 높은 순으로 대상에 대해 추천 시행

IBCF(Item Base Collaborate filtering)

Item-Item 간의

코사인 유사도 행렬

S	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	$\hat{r}_a$
i_1	-	0.1	0	**0.3**	**0.2**	**0.4**	0	0.1	-
i_2	0.1	-	**0.8**	**0.9**	0	**0.2**	0.1	0	0.0
i_3	0	**0.8**	-	0	**0.4**	0.1	0.3	**0.5**	4.6
i_4	**0.3**	**0.9**	0	-	0	0.1	0	**0.2**	3.2
i_5	**0.2**	0	**0.4**	0	-	0.1	**0.2**	0.1	-
i_6	**0.4**	**0.2**	0.1	**0.3**	0.1	-	0	0.1	2.0
i_7	0	**0.1**	**0.3**	0	**0.2**	0	-	0	4.0
i_8	**0.1**	0	**0.5**	**0.2**	0.1	0.1	0	-	-
u_a	2	?	?	?	4	?	?	5	

$k=3$

IBCF(Item Base Collaborate filtering)

$$r_{a(i3)} = \frac{0.9}{0.4} * 4 + \frac{0.9}{0.5} * 5 = 4.6$$

$$r_{a(i4)} = \frac{0.5}{0.3} * 2 + \frac{0.5}{0.2} * 5 = 3.2$$

$$r_{a(i6)} = \frac{0.4}{0.4} * 2 = 2$$

$$r_{a(i2)} = \frac{0.2}{0.2} * 4 = 4$$

1. 분석하고 싶은 대상의 평점을 k개만 선택
2. 한 개의 아이템 행에 대해서 코사인 유사도를 모두 더한다
3. $\frac{\text{특정 아이템의 코사인 유사도}}{\text{한 행의 모두 합한 코사인 유사도}}$ 를 가중치를 두고 곱하여 합해준다.
4. 높은 순으로 분석하고 싶은 대상에게 추천을 한다.

Data 출처

kaggle

'The Movies Dataset'

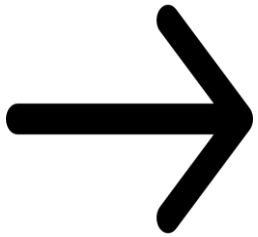
45,000개의 영화 데이터, 270,000 유저의 평점 데이터

https://www.kaggle.com/rounakbanik/the-movies-dataset

Data 형식

movies_metadata.csv

adult	성인물 여부
belongs_to_collection	시리즈 여부
budget	예산
genres	장르
homepage	영화 홈페이지
id	영화 Code
original_language	언어
original_title	원작 이름
overview	스토리
poster_path	포스터 링크
production_companies	제작사
production_countries	제작 국가
release_date	개봉일
revenue	수익
runtime	상영시간
spoken_languages	언어
status	개봉 상태
tagline	슬로건
title	제목
vote_average	평점 평균
vote_count	평점 수



45,000개의 영화 데이터 중
결측 값 및 평점 낮은 영화 등의
데이터 제거

3000여개의 영화로 분석 진행

Data 전처리

new_movies_metadata.csv

genres	장르
id	영화 Code
revenue	수익
title	제목
vote_average	평점 평균
vote_count	평점 수

결측 값 및 특이 값 제거

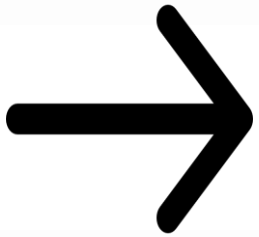
평점 5.5점 미만 영화 제거
평점 수 100개 미만 영화제거
Json 형식이던 장르 처리

3000여개의 영화로 분석 진행

Data 형식

ratings.csv

id	유저 id
movieId	영화 id
rating	평점



약 600MB의 파일 크기
메모리 적재가 어려우므로
100,000 row (약 1.1MB)의
Small Data로 분석 진행

Data 전처리

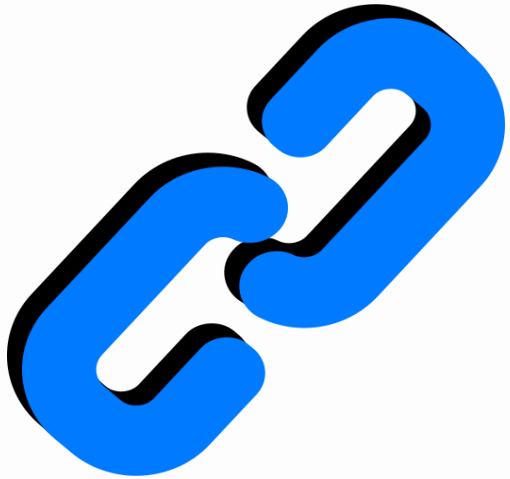
new_movies_metadata.csv

	title	id	revenue	vote_avera	vote_coun	genre1	genre2	genre3	genre4	genre5
0	Toy Story	862	3.74E+08	7.7	5415	Animation	Comedy	Family		
1	Jumanji	8844	2.63E+08	6.9	2413	Adventure	Fantasy	Family		
2	Heat	949	1.87E+08	7.7	1886	Action	Crime	Drama	Thriller	
3	GoldenEye	710	3.52E+08	6.6	1194	Adventure	Action	Thriller		
4	Casino	524	1.16E+08	7.8	1343	Drama	Crime			
5	Sense and Sensibility	4584	1.35E+08	7.2	364	Drama	Romance			
6	Four Rooms	5	4300000	6.5	539	Crime	Comedy			
7	Ace Ventura: When Nature Calls	9273	2.12E+08	6.1	1128	Crime	Comedy	Adventure		
8	Get Shorty	8012	1.15E+08	6.4	305	Comedy	Thriller	Crime		
9	Assassins	9691	30303072	6	394	Action	Adventure	Crime	Thriller	
10	Leaving Las Vegas	451	49800000	7.1	365	Drama	Romance			
11	The City of Lost Children	902	1738611	7.6	308	Fantasy	Science Fi	Adventure		
12	Twelve Monkeys	63	1.69E+08	7.4	2470	Science Fi	Thriller	Mystery		
13	Babe	9598	2.54E+08	6	756	Fantasy	Drama	Comedy	Family	
14	Dead Man Walking	687	39363635	7.3	350	Drama				
15	Se7en	807	3.27E+08	8.1	5915	Crime	Mystery	Thriller		
16	Pocahontas	10530	3.46E+08	6.7	1509	Adventure	Animation	Drama	Family	
17	The Usual Suspects	629	23341568	8.1	3334	Drama	Crime	Thriller		

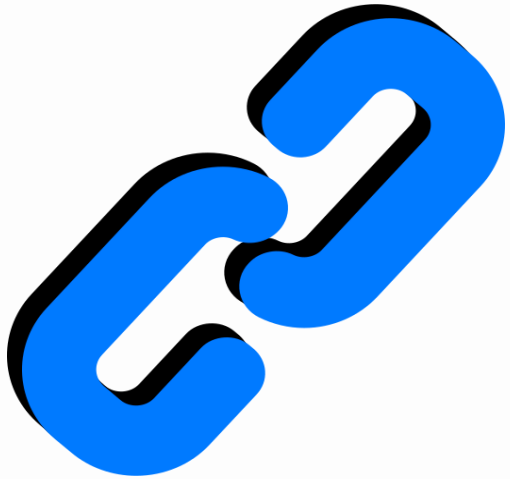
ratings.csv

userId	movieId	rating
1	31	2.5
1	1029	3
1	1061	3
1	1129	2
1	1172	4
1	1263	2
1	1287	2
1	1293	2
1	1339	3.5
1	1343	2
1	1371	2.5
1	1405	1
1	1953	4
1	2105	4
1	2150	3
1	2193	2
1	2294	2
1	2455	2.5
1	2968	1
1	3671	3
2	10	4
2	17	5

데이터 전처리 Code



Recommender Code



끝!

감사합니다!

References

1. Introduction to Recommender Systems
   https://dzone.com/articles/introduction-to-recommender-systems
2. 추천 시스템(Recommendation System)
   https://khanrc.tistory.com/entry
3. Recommendation System : 협업 필터링을 중심으로
   http://rosaec.snu.ac.kr/meet/file/20120728b.pdf
4. 협업 필터링 추천 시스템(Collaborative Filtering Recommendation System)
   https://scvgoe.github.io/2017-02-01-%ED%98%91%EC%97%85-%ED%95%84%ED%84%B0%EB%A7%81-%EC%B6%94%EC%B2%9C-%EC%8B%9C%EC%8A%A4%ED%85%9C-(Collaborative-Filtering-Recommendation-System)/
5. 추천시스템과 협업필터링
   https://www.slideshare.net/bage79/ss-45783615?qid=a2355e69-d3e6-48e0-93ae-33f870c8207f&v=&b=&from_search=3