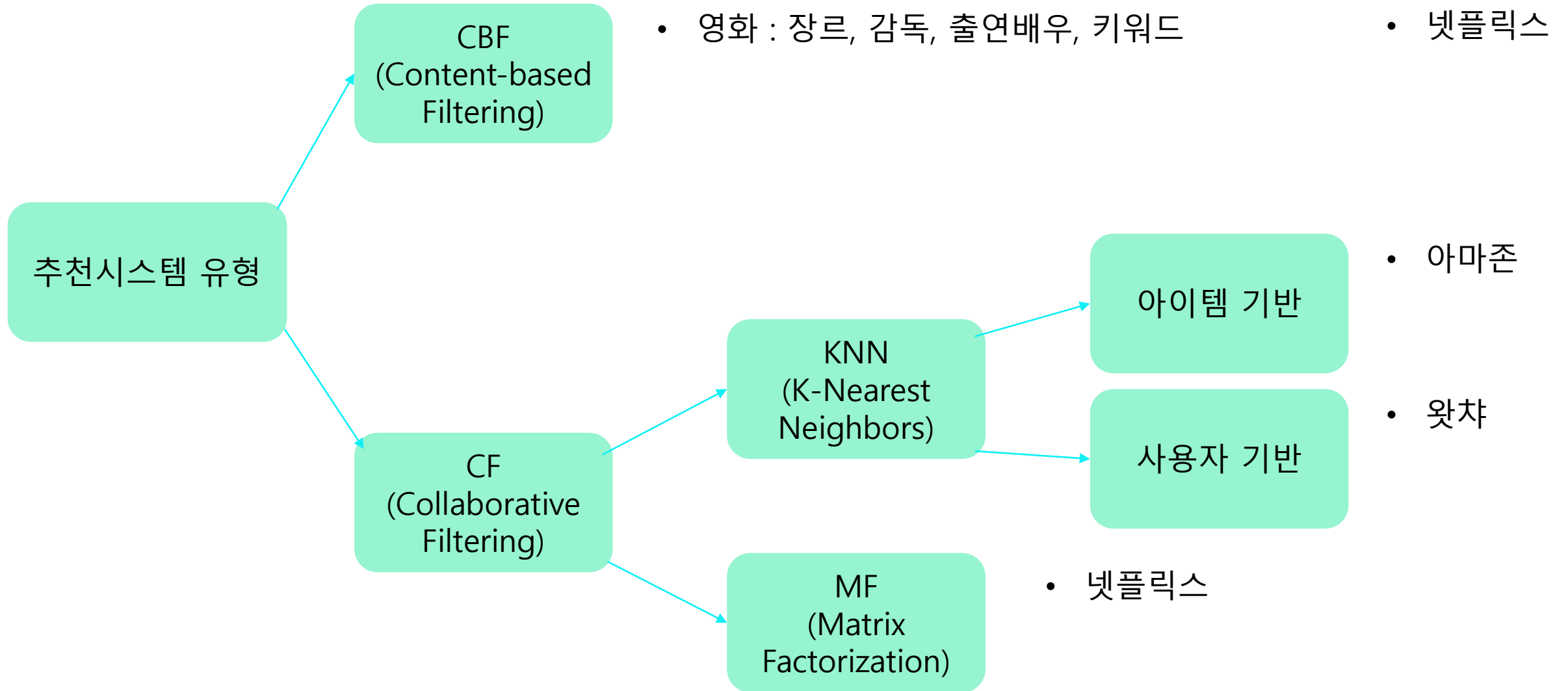


추천시스템의 이해 (Recommendation)

추천시스템 종류




추천 시스템의 중요성

아마존 등과 같은 전자상거래 업체부터 넷플릭스, 유튜브, 애플 뮤직 등 콘텐츠 포털까지 추천 시스템을 통해 사용자의 취향을 이해하고 맞춤 상품과 콘텐츠를 제공해 조금이라도 오래동안 자기 사이트에 고객을 머무르게 하기 위해 전력을 기울이고 있습니다







사용자에게 맞춤 콘텐츠를 제공




검색


Q




맞춤 동영상




Hans Zimmer - Time (Extended) | Inception
Music Hole
조회수 14.4만회 · 1년 전




Jóhann Jóhannsson - Orphée HD
Soundtracked Universe
조회수 4.7만회 · 1개월 전




25m Extended - 인터스텔라 Interstellar OST : "First Step"
Reynah
조회수 99.5만회 · 2년 전




The Last of the Mohicans - Promontory (Main Theme)
EpicMusicChannel8
조회수 420만회 · 5년 전




자우림 (Jaurim) - 스물다섯, 스물하나 (Twenty-five, Twenty-one)
KpopMar3
조회수 36.3만회 · 3년 전




The Grey Soundtrack - Into The Fray (Long Version)
Supercali006
조회수 1.1만회 · 1년 전




The Last Samurai • A Way Of Life • Hans Zimmer
MusicAreaHQ
조회수 1.1만회 · 1년 전



Harry Escott: Unravelling ("Shame" Motion Picture)
Aritivista-Abogado
조회수 1.1만회 · 1년 전




Hans Zimmer - S.T.A.Y. (Madis Remix) Interstellar
madis-music
조회수 1.1만회 · 1년 전



Wonder Woman Theme Song - From OST
Michael Mega
조회수 1.1만회 · 1년 전

사용자에게 맞춤 콘텐츠를 제공



All ▾

Q

The Halloween Shop

Departments ▾Browsing History ▾chulmin's Amazon.comToday's DealsGift Cards & RegistrySellHelp

EN ▾Hello, chulminAccount & Lists ▾OrdersTry Prime ▾Cart

Your recently viewed items and featured recommendations

Best Sellers

Page 1 of 7



Wheat Belly: Lose the Wheat, Lose the Weight, and How the World's Most Dangerous Food is Destroying Your Health

William Davis MD

★★★★☆ 6,000

Kindle Edition

\$8.42



From Sand and Ash

Amy Harmon

★★★★☆ 2,948

Kindle Edition

\$4.99



A Dark Lure

Loreth Anne White

★★★★☆ 4,817

Kindle Edition

\$4.99



The Naturalist (The Naturalist Series Book 1)

Andrew Mayne

★★★★☆ 1,691

Kindle Edition

\$4.99



I Am Watching You

Teresa Driscoll

★★★★☆ 1,392

Kindle Edition

\$4.99



Gulp: Adventures on the Alimentary Canal

Mary Roach

★★★★☆ 1,072

Kindle Edition

\$9.99

추천 엔진을 통한 사용자의 취향 저격

- 사용자가 어떤 상품을 구매했는가?
- 사용자가 어떤 제품을 Browse 했는가?
- 사용자가 무엇을 클릭했는가?
- 사용자가 평가한 영화 평점은?



추천 엔진

“ 당신만을 위한 최신 상품 ”

“ 이 상품을 선택한 다른
사람들이 좋아하는 상품들 ”

“ 이 상품을 좋아하시나요? 아래
있는 다른 상품은 어떠신가요? ”

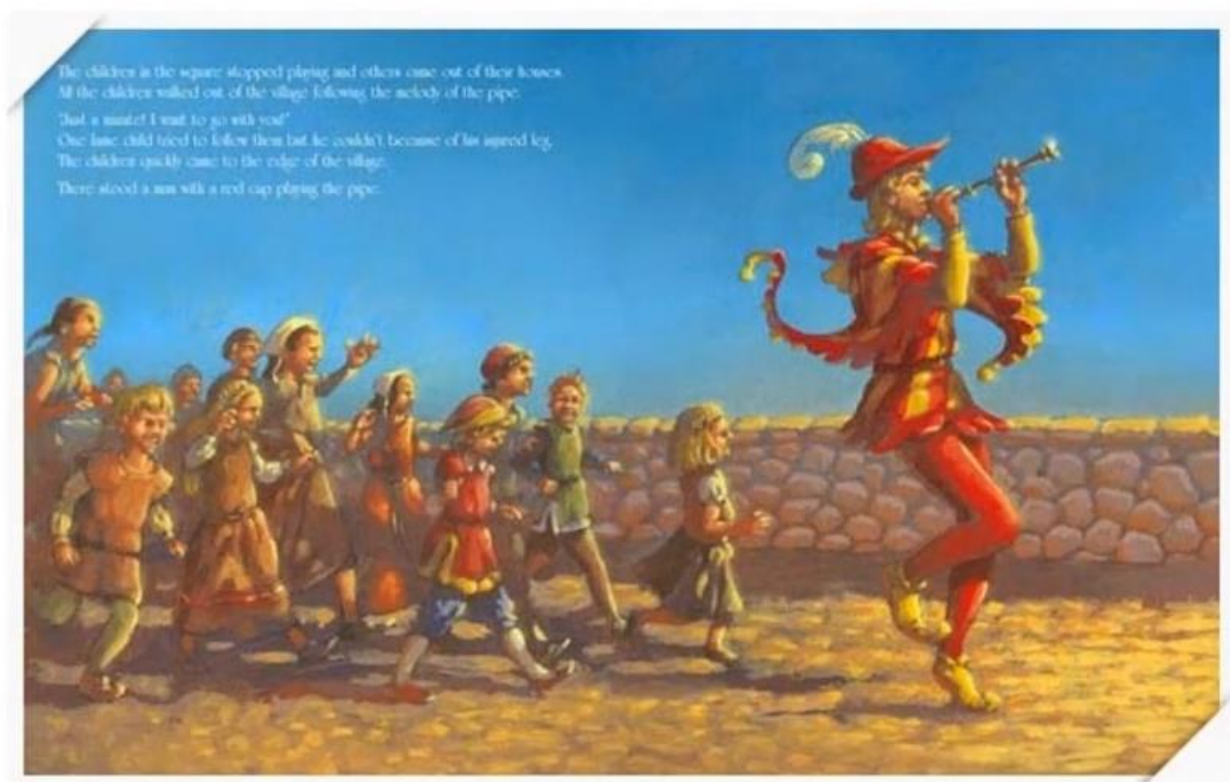
추천 시스템의 묘미



추천 시스템의 묘미는 사용자 자신도 좋아하는지 모르고 있었던 취향을 발견하는 것이다.

추천 시스템에 신뢰가 높아지면서 사용자는 추천 아이템을 더 많이 선택하게 되고, 이로 인해 더 많은 데이터가 추천 시스템에 축적되면서 추천이 정확해지고 다양해 진다.

추천이 지배하는 사회



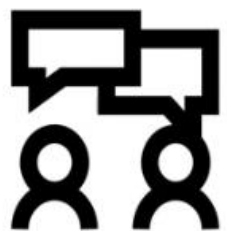
The children in the square stopped playing and others came out of their houses.
All the children walked out of the village following the melody of the pipe.
"Just a minute! I want to go with you!"
One lame child tried to follow them but he couldn't because of his injured leg.
The children quickly came to the edge of the village.
There stood a man with a red cap playing the pipe.

정교한 추천 시스템은 사용자에게 높은 신뢰도를 얻게 되며, 맹목적으로 사용자가 이에 의존하게 만든다. 이를 기반으로 서비스 프로바이더는 고객 충성도를 크게 향상시킬 수 있다.

추천 시스템 방식



콘텐츠 기반 필터링
Content Based Filtering



협업 필터링
Collaborative Filtering

추천 시스템은 이들 방식중 1가지를
선택하거나 이들을 결합하여 hybrid
방식으로 사용

(예 : Content Based + Collaborative
Filtering)

하이브리드 기반 추천

넷플릭스의 경우 자사가 생성한 콘텐츠
위주로 추천 영화가 치우치는 경향이 시작됨.



컨텐츠 기반 필터링 실습

(Contents Based Filtering)

콘텐츠 기반 필터링 - Contents Based Filtering

장르 : SF, 드라마, 미스터리

감독 : 드니 빌뇌브

출연 : 에이미 아담스,
제레미 러너

키워드 : 외계인 침공,
예술성, 스릴러 요소



평점 : 8.0

장르 : SF, 액션, 스릴러

감독 : 리들리 스콧

출연 : 노미 라마스,
마이클 패스벤더

키워드 : 에일리언 프리퀄,
액션과 스릴러의 조화



평점 : 9.0



사용자 선호 프로필

선호 장르 : SF, 액션, 스릴러

선호 배우 : 에이미 아담스, 마이클
패스벤더 등

선호 감독 : 리들리 스콧, 드니 빌뇌브



장르 : SF, 액션, 스릴러

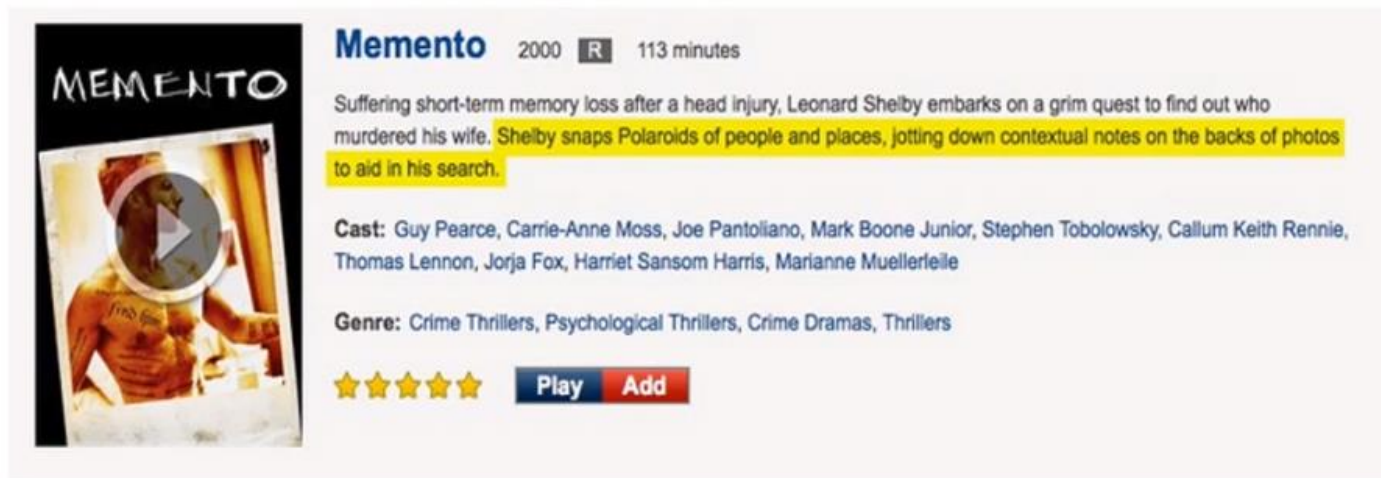
감독 : 드니 빌뇌브

출연 : 라이언 고슬링,
해리슨 포드

키워드 : 리들리 스콧
감독의 전작을 리메이크

콘텐츠 기반 필터링 기법

감독, 배우, 영화 설명, 장르 등 영화를 구성하는 다양한 콘텐츠들을 텍스트 기반 문서 유사도로 비교하여 추천



영화 구성
콘텐츠 텍스트



피처 벡터화
(Count, TF-IDF)

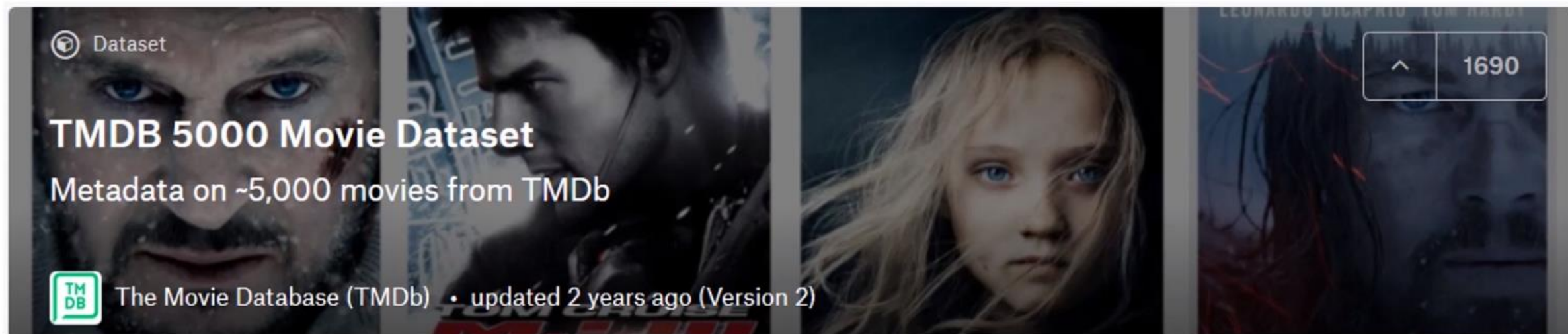


코사인 유사도



유사도 및
평점에 따른
영화 추천

콘텐츠 기반 필터링 실습 - TMDB 5000



콘텐츠 기반 필터링 실습

콘텐츠 기반 필터링 구현 프로세스

1. 콘텐츠에 대한 여러 텍스트 정보들을 피쳐 벡터화
2. 코사인 유사도로 콘텐츠별 유사도 계산
3. 콘텐츠 별로 가중 평점을 계산
4. 유사도가 높은 콘텐츠 중에 평점이 좋은 콘텐츠 순으로 추천

영화 데이터 셋

캐글 TMDb 데이터 : <https://www.kaggle.com/tmdb/tmdb-movie-metadata>

예산	장르	홈페이지	id	키워드	언어	원제목	오버뷰	관람객	제작사	제작국가	개봉일	수익	상영시간	언어	상태	한줄설명	제목	평점	평가참여자수
budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	production_countries	release_date	revenue	runtime	spoken_languages	status	tagline	title	vote_average	vote_count
2.37E+08	[{"id": 28, "name": "Action"}]	http://www.avatar.com	1999	[{"id": 146, "text": "Avatar"}]	en	Avatar	In the 22nd century, a group of humans are transported to the moon's surface to mine for rare minerals. A group of Na'vi, a native tribe, are also present on the moon. A conflict arises between the humans and the Na'vi.	150.4376	[{"name": "Twentieth Century Fox Film Corporation"}]	[{"iso_3166_2": "US"}]	2009-12-18	2.0735E+09	162	[{"iso_639_1": "en"}]	Released	Enter the Avatar	Avatar	7.2	11800
3E+08	[{"id": 12, "name": "Adventure"}]	http://disney.com	285	[{"id": 270, "text": "Pirates of the Caribbean"}]	en	Pirates of the Caribbean: The Curse of the Black Pearl	Captain Jack Sparrow leads a band of misfits to defeat a vengeful goddess and a ruthless pirate.	139.0826	[{"name": "Walt Disney Pictures"}]	[{"iso_3166_2": "US"}]	2003-07-09	6.5436E+08	169	[{"iso_639_1": "en"}]	Released	At the end of the world	Pirates of the Caribbean: The Curse of the Black Pearl	6.9	4500
2.45E+08	[{"id": 28, "name": "Action"}]	http://www.spectre.com	470	[{"id": 470, "text": "Spectre"}]	en	Spectre	A cryptic riddle leads James Bond to a shadowy enemy whose only goal is to destroy the world.	107.3768	[{"name": "Eon Productions"}]	[{"iso_3166_2": "GB"}]	2015-10-26	663.6521E+06	148	[{"iso_639_1": "en"}]	Released	A Plan No Spectre	Spectre	6.3	4466
2.5E+08	[{"id": 28, "name": "Action"}]	http://www.dcninja.com	849	[{"id": 849, "text": "The Dark Knight Rises"}]	en	The Dark Knight Rises	Following the events of The Dark Knight, Batman must prepare to fight his next enemy: a new generation of villains.	112.313	[{"name": "Warner Bros. Entertainment Inc."}]	[{"iso_3166_2": "US"}]	2012-07-20	1.0841E+09	165	[{"iso_639_1": "en"}]	Released	The Legend of Batman	The Dark Knight Rises	7.6	9106
2.6E+08	[{"id": 28, "name": "Action"}]	http://movie.com	818	[{"id": 818, "text": "John Carter"}]	en	John Carter	John Carter is transported to Mars, where he becomes a hero and a warrior.	43.927	[{"name": "MGM Studios"}]	[{"iso_3166_2": "US"}]	2007-03-03	2.707E+07	132	[{"iso_639_1": "en"}]	Released	Lost in our world	John Carter	6.1	2124
2.58E+08	[{"id": 14, "name": "Fantasy"}]	http://www.sony.com	559	[{"id": 851, "text": "Spider-Man 3"}]	en	Spider-Man 3	The seemingly perfect Spider-Man must face off against his greatest foe, Doctor Octopus, who has become a powerful and ruthless villain.	115.6998	[{"name": "Sony Pictures Entertainment Inc."}]	[{"iso_3166_2": "US"}]	2007-05-04	3.3643E+08	139	[{"iso_639_1": "en"}]	Released	The battle of the century	Spider-Man 3	5.9	3576
2.6E+08	[{"id": 16, "name": "Family"}]	http://disney.com	156	[{"id": 156, "text": "Tangled"}]	en	Tangled	When the princess escapes her evil father's control, a search for true love begins.	48.68197	[{"name": "Walt Disney Pictures"}]	[{"iso_3166_2": "US"}]	2010-11-24	5.4361E+08	100	[{"iso_639_1": "en"}]	Released	They're taking over	Tangled	7.4	3330
2.8E+08	[{"id": 28, "name": "Action"}]	http://marvel.com	882	[{"id": 882, "text": "Avengers: Age of Ultron"}]	en	Avengers: Age of Ultron	When Tony Stark and his fellow Avengers discover that a powerful new villain is on the rise, they must band together to stop him.	134.2792	[{"name": "Marvel Studios"}]	[{"iso_3166_2": "US"}]	2015-05-01	6.7881E+08	141	[{"iso_639_1": "en"}]	Released	A New Age of Heroes	Avengers: Age of Ultron	7.3	6767
2.5E+08	[{"id": 12, "name": "Adventure"}]	http://harrypotter.com	767	[{"id": 616, "text": "Harry Potter and the Half-Blood Prince"}]	en	Harry Potter and the Half-Blood Prince	As Harry battles the forces of evil, he discovers the truth about his parents and the wizarding world.	98.88564	[{"name": "Warner Bros. Entertainment Inc."}]	[{"iso_3166_2": "GB"}]	2009-07-07	3.5012E+08	153	[{"iso_639_1": "en"}]	Released	Dark Secrets	Harry Potter and the Half-Blood Prince	7.4	5293

특정 영화에 대해
장르가 유사한 영화를
추천해주는 서비스를
기획해보자

데이터 전처리

```
In [1]: ▶ import pandas as pd
import numpy as np
import warnings; warnings.filterwarnings('ignore')
```

```
In [2]: ▶ movies = pd.read_csv('./data/tmdb_5000_movies.csv')
```

```
In [3]: ▶ print(movies.shape)
movies.head(1)
```

(4803, 20)

Out[3]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularit
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id": ...}]	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.43757

```
In [5]: ▶ movies_df['genres'][0]
```

Out[5]: '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'

```
In [6]: ▶ type(movies_df['genres'][0])
# str 타입인 것을 확인할 수 있다.
```

Out[6]: str

genres 칼럼

- str 형태

- 리스트 안에 딕셔너리로 여러 개의 장르 키워드가 저장된 형태

전처리 필요

데이터 전처리

literal_eval : str 형태를 list형태로 바꿔준다.

genres, keywords 칼럼들의 str형태를 list형태로 바꾸주기

```
In [38]: ▶ from ast import literal_eval
          movies_df['genres'] = movies_df['genres'].apply(literal_eval)
          movies_df['keywords'] = movies_df['keywords'].apply(literal_eval)
```

```
In [39]: ▶ movies_df['genres'][0]
```

```
Out[39]: [{'id': 28, 'name': 'Action'},
          {'id': 12, 'name': 'Adventure'},
          {'id': 14, 'name': 'Fantasy'},
          {'id': 878, 'name': 'Science Fiction'}]
```

```
In [40]: ▶ type(movies_df['genres'][0])
          # str타입에서 list타입으로 바뀐 것을 확인할 수 있다.
```

```
Out[40]: list
```


데이터 전처리

list 내 여러개 딕셔너리의 name키에 해당하는 값들을 리스트로 변환

```
In [41]: ▶ movies_df['genres'] = movies_df['genres'].apply(lambda x : [ y['name'] for y in x])
movies_df['keywords'] = movies_df['keywords'].apply(lambda x : [ y['name'] for y in x])
```

```
In [45]: ▶ movies_df[['genres', 'keywords']][:1]
```

Out [45]:

	genres	keywords
0	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...]

genres 안의 딕셔너리들 중 키워드들만 뽑아
냄

장르 유사도 기반 영화 추천 시스템

@장르 CBF 추천 : 장르를 피처 벡터화한 후 행렬 데이터 값을 코사인 유사도로 계산하기
<프로세스>

1. 장르 피처 벡터화: 문자열로 변환된 genres 칼럼을 Count 기반으로 피처 벡터화 변환
2. 코사인 유사도 계산 : genres 문자열을 피처 벡터화한 행렬로 변환한 데이터 세트를 코사인 유사도로 비교
3. 평점으로 계산 : 장르 유사도가 높은 영화 중 평점이 높은 순으로 영화 추천

CountVectorizer

```
In [47]: from sklearn.feature_extraction.text import CountVectorizer
```

```
In [50]: # 참고 : CountVectorizer에 대하여  
# CountVectorizer는 다음 세가지를 수행한다.  
# 1. 문서를 토큰 리스트로 변환한다.  
# 2. 각 문서에서 토큰의 출현 빈도를 센다.  
# 3. 각 문서를 BOW(Bag of Words) 인코딩 벡터로 변환한다.  
  
from sklearn.feature_extraction.text import CountVectorizer  
corpus = [  
    'This is the first document.',  
    'This is the second second document.',  
    'And the third one.',  
    'Is this the first document?',  
    'The last document?',  
]  
vect = CountVectorizer()  
vect.fit(corpus)  
vect.vocabulary_
```

문서를 토큰 리스트로 변환 후,
각 문서에서 토큰의 출현 빈도를 계산
-> BOW 인코딩 벡터로 변환 가능

```
Out[50]: {'this': 9,  
          'is': 3,  
          'the': 7,  
          'first': 2,  
          'document': 1,  
          'second': 6,  
          'and': 0,  
          'third': 8,  
          'one': 5,  
          'last': 4}
```

장르 유사도 기반 영화 추천 시스템

@장르 CBF 추천 : 장르를 피처 벡터화한 후 행렬 데이터 값을 코사인 유사도로 계산하기
<프로세스>

1. 장르 피처 벡터화: 문자열로 변환된 genres 칼럼을 Count 기반으로 피처 벡터화 변환
2. 코사인 유사도 계산 : genres 문자열을 피처 벡터화한 행렬로 변환한 데이터 세트를 코사인 유사도로 비교
3. 평점으로 계산 : 장르 유사도가 높은 영화 중 평점이 높은 순으로 영화 추천

장르 유사도 기반 영화 추천 시스템

장르 칼럼의 키워드들을 공백문자 기준으로 join하여 최종적으로 장르 매트릭스(4803행, 276열) 생성

```
In [48]: # CountVectorizer를 적용하기 위해 공백문자로 word 단위가 구분되는 문자열로 변환.  
movies_df['genres_literal'] = movies_df['genres'].apply(lambda x : (' ').join(x))  
movies_df['genres_literal']
```

```
Out[48]: 0      Action Adventure Fantasy Science Fiction  
1              Adventure Fantasy Action  
2              Action Adventure Crime  
3              Action Crime Drama Thriller  
4      Action Adventure Science Fiction  
5              Fantasy Action Adventure  
6              Animation Family  
7      Action Adventure Science Fiction  
8              Adventure Fantasy Family  
9      Action Adventure Fantasy  
10     Adventure Fantasy Action Science Fiction  
11     Adventure Action Thriller Crime  
12     Adventure Fantasy Action  
13     Action Adventure Western  
14     Action Adventure Fantasy Science Fiction  
15     Adventure Family Fantasy  
16     Science Fiction Action Adventure  
17     Adventure Action Fantasy  
18     Action Comedy Science Fiction  
19     Action Adventure Fantasy
```

```
In [54]: # CountVectorizer로 학습시켰더니 4803개 영화에 대한 276개 장르의 '장르 매트릭스'가 생성되었다.  
count_vect = CountVectorizer(min_df=0, ngram_range=(1,2)) #min_df: 단어장에 들어갈 최소빈도, ngram_rang  
genre_mat = count_vect.fit_transform(movies_df['genres_literal'])  
print(genre_mat.shape)
```

(4803, 276)

장르 유사도 기반 영화 추천 시스템

코사인 유사도(cosine_similarity)이용해서 영화별 유사도 계산

```
In [66]: ▶ # 코사인 유사도에 의해 4803개 영화 각각 유사한 영화들이 계산됨
from sklearn.metrics.pairwise import cosine_similarity
genre_sim = cosine_similarity(genre_mat, genre_mat)
print(genre_sim.shape)
print(genre_sim[:5])
```

```
(4803, 4803)
[[1.          0.59628479 0.4472136  ... 0.          0.          0.          ]
 [0.59628479 1.          0.4       ... 0.          0.          0.          ]
 [0.4472136  0.4        1.          ... 0.          0.          0.          ]
 [0.12598816 0.16903085 0.3380617  ... 0.12598816 0.          0.          ]
 [0.75592895 0.3380617  0.50709255 ... 0.          0.          0.          ]]
```

영화별 장르 유사도가 계산된
매트릭스(4803, 4803) 생성

```
In [68]: ▶ # 자료를 정렬하는 것이 아니라 순서만 알고 싶다면 argsort
# 유사도가 높은 영화를 앞에서부터 순서대로 보여줌
# 0번째 영화의 경우 유사도 순서 : 0번, 3494번, 813번, ..., 2401 순서
genre_sim_sorted_ind = genre_sim.argsort()[::-1] # 전체를 -1칸 간격으로
print(genre_sim_sorted_ind[:1])
```

```
[[ 0 3494 813 ... 3038 3037 2401]]
```

특정 영화와 유사도가 높은 순서대로 인덱스 번호를 보여줌

장르 유사도 기반 영화 추천 시스템

추천 ver1. 장르 코사인 유사도에 의해 영화를 추천하는 함수

```
In [93]: ▶ def find_sim_movie_ver1(df, sorted_ind, title_name, top_n=10):  
  
    # 인자로 입력된 movies_df DataFrame에서 'title' 컬럼이 입력된 title_name 값인 DataFrame추출  
    title_movie = df[df['title'] == title_name]  
  
    # title_named을 가진 DataFrame의 index 객체를 ndarray로 반환하고  
    # sorted_ind 인자로 입력된 genre_sim_sorted_ind 객체에서 유사도 순으로 top_n 개의 index 추출  
    title_index = title_movie.index.values  
    similar_indexes = sorted_ind[title_index, :(top_n)]  
  
    # 추출된 top_n index들 출력. top_n index는 2차원 데이터 임.  
    # dataframe에서 index로 사용하기 위해서 1차원 array로 변경  
    print(similar_indexes)  
    # 2차원 데이터를 1차원으로 변환  
    similar_indexes = similar_indexes.reshape(-1)  
  
    return df.iloc[similar_indexes]
```

장르 유사도 기반 영화 추천 시스템

영화 Godfather와 장르가 유사한 영화 10개 추천

```
In [94]: ▶ similar_movies = find_sim_movie_ver1(movies_df, genre_sim_sorted_ind, 'The Godfather', 10)
similar_movies[['title', 'vote_average', 'genres', 'vote_count']]
# 문제 ; 평점 기반으로 추천하고자 하는데, vote_count가 낮은 영화는 제외하고 싶음

[[2731 1243 3636 1946 2640 4065 1847 4217 883 3866]]
```

Out[94]:

	title	vote_average	genres	vote_count
2731	The Godfather: Part II	8.3	[Drama, Crime]	3338
1243	Mean Streets	7.2	[Drama, Crime]	345
3636	Light Sleeper	5.7	[Drama, Crime]	15
1946	The Bad Lieutenant: Port of Call - New Orleans	6.0	[Drama, Crime]	326
2640	Things to Do in Denver When You're Dead	6.7	[Drama, Crime]	85
4065	Mi America	0.0	[Drama, Crime]	0
1847	GoodFellas	8.2	[Drama, Crime]	3128
4217	Kids	6.8	[Drama, Crime]	279
883	Catch Me If You Can	7.7	[Drama, Crime]	3795
3866	City of God	8.1	[Drama, Crime]	1814

문제

평가횟수가 현저히 적은 영화들이 추천되는 것도 있음
low quality 추천 문제

우리가 전혀 모르는 영화를 추천받는 것은 엉뚱한 추천 결과를 낼 수 있음

-> 평가횟수를 반영한 추천 시스템이 필요

장르 유사도 기반 영화 추천 시스템

가중평점(평점&평가횟수) 반영한 영화 추천

@ 가중평점 (Weighted Rating):

$$(v/(v+m))*R + (m/(v+m))*C$$

- v : 영화별 평점을 투표한 횟수 (vote_count) ★ 투표횟수가 많은 영화에 가중치 부여
 - m : 평점을 부여하기 위한 최소 투표 횟수 → 여기서는 투표수 상위 60%
 - R : 개별 영화에 대한 평균 평점 (vote_average)
 - C : 전체 영화에 대한 평균 평점 (movies_df['vote_average'].mean())
- # C, m은 고정값
v, R은 영화마다 변동값

투표 횟수가 많으면
가중치가 붙는다.

최종적으로

1. 장르가 유사한 영화 중
2. 가중평점이 높은 영화가 추천되게 된다.

```
In [86]: ▶ # 상위 60%에 해당하는 vote_count를 최소 투표 횟수인 m으로 지정
C = movies_df['vote_average'].mean()
m = movies_df['vote_count'].quantile(0.6)
```

```
In [84]: ▶ # C: 전체 영화에 대한 평균평점 = 약 6점
# m: 평점을 부여하기 위한 최소 투표 횟수 = 370회(상위 60% 수준)
print('C:', round(C, 3), 'm:', round(m, 3))
```

C: 6.092 m: 370.2

장르 유사도 기반 영화 추천 시스템

가중평점을 계산하는 함수

```
In [85]: ▶ def weighted_vote_average(record):  
    v = record['vote_count']  
    R = record['vote_average']  
  
    return ( (v/(v+m)) * R ) + ( (m/(m+v)) * C )
```

가중평점을 return값으로 돌려준다.

```
In [87]: ▶ # 기존 데이터에 가중평점 칼럼 추가  
movies_df['weighted_vote'] = movies_df.apply(weighted_vote_average, axis=1)
```


장르 유사도 기반 영화 추천 시스템

추천 ver2. 먼저 장르 유사성 높은 영화 20개 선정 후, 가중평점순 10개 선정

```
In [95]: ▶ def find_sim_movie_ver2(df, sorted_ind, title_name, top_n=10):  
    title_movie = df[df['title'] == title_name]  
    title_index = title_movie.index.values  
  
    # top_n의 2배에 해당하는 장르 유사성이 높은 index 추출  
    similar_indexes = sorted_ind[title_index, :(top_n*2)]  
    similar_indexes = similar_indexes.reshape(-1)  
  
    # 기준 영화 index는 제외  
    similar_indexes = similar_indexes[similar_indexes != title_index]  
  
    # top_n의 2배에 해당하는 후보군에서 weighted_vote 높은 순으로 top_n 만큼 추출  
    return df.iloc[similar_indexes].sort_values('weighted_vote', ascending=False)[:top_n]
```

먼저, 장르 유사성 높은 것 20개를 먼저 뽑은 뒤

최종적으로 상위 10개를
sort해서 보여준다

장르 유사도 기반 영화 추천 시스템

영화 Godfather에 대해 장르 유사성, 가중평점 반영한 추천 영화 10개를 뽑아보자

```
In [96]: > similar_movies = find_sim_movie_ver2(movies_df, genre_sim_sorted_ind, 'The Godfather', 10)
similar_movies[['title', 'vote_average', 'weighted_vote', 'genres', 'vote_count']]
```

Out[96]:

	title	vote_average	weighted_vote	genres	vote_count
2731	The Godfather: Part II	8.3	8.079586	[Drama, Crime]	3338
1847	GoodFellas	8.2	7.976937	[Drama, Crime]	3128
3866	City of God	8.1	7.759693	[Drama, Crime]	1814
1663	Once Upon a Time in America	8.2	7.657811	[Drama, Crime]	1069
883	Catch Me If You Can	7.7	7.557097	[Drama, Crime]	3795
281	American Gangster	7.4	7.141396	[Drama, Crime]	1502
4041	This Is England	7.4	6.739664	[Drama, Crime]	363
1149	American Hustle	6.8	6.717525	[Drama, Crime]	2807
1243	Mean Streets	7.2	6.626569	[Drama, Crime]	345
2839	Rounders	6.9	6.530427	[Drama, Crime]	439

평가 횟수가 반영된
고품질의 추천이 적용된 모습

요약 : **Godfather**를 좋아하는 사람에게 영화 추천해주기

Godfather 장르가 Drama, Crime이다.

우선 Drama, Crime 장르 기준으로 상위 20개 영화를 뽑아보고,
그 중 평가횟수를 반영한 가중평점 기준 상위 10개 영화를 뽑아서 추천해준다.

장르 유사도 기반 영화 추천 시스템

응용 : Spider-Man 3 좋아하는 사람 기준으로 장르가 유사한 영화를 추천해주자

```
In [30]: > similar_movies = find_sim_movie_ver2(movies_df, genre_sim_sorted_ind, 'Spider-Man 3', 10)
similar_movies[['title', 'vote_average', 'weighted_vote', 'genres', 'vote_count']]
```

Out[30]:

	title	vote_average	weighted_vote	genres	vote_count
329	The Lord of the Rings: The Return of the King	8.1	8.011871	[Adventure, Fantasy, Action]	8064
262	The Lord of the Rings: The Fellowship of the Ring	8.0	7.922175	[Adventure, Fantasy, Action]	8705
330	The Lord of the Rings: The Two Towers	8.0	7.910111	[Adventure, Fantasy, Action]	7487
19	The Hobbit: The Battle of the Five Armies	7.1	7.027274	[Action, Adventure, Fantasy]	4760
98	The Hobbit: An Unexpected Journey	7.0	6.961224	[Adventure, Fantasy, Action]	8297
126	Thor: The Dark World	6.8	6.748873	[Action, Adventure, Fantasy]	4755
30	Spider-Man 2	6.7	6.652034	[Action, Adventure, Fantasy]	4321
129	Thor	6.6	6.572735	[Adventure, Fantasy, Action]	6525
20	The Amazing Spider-Man	6.5	6.478296	[Action, Adventure, Fantasy]	6586
38	The Amazing Spider-Man 2	6.5	6.466812	[Action, Adventure, Fantasy]	4179

장르 유사도 기반 영화 추천 시스템

응용 : Enemy at the Gates 좋아하는 사람 기준으로 장르가 유사한 영화를 추천해주자

```
In [29]: ► similar_movies = find_sim_movie_ver2(movies_df, genre_sim_sorted_ind, 'Enemy at the Gates', 10)
similar_movies[['title', 'vote_average', 'weighted_vote', 'genres', 'vote_count']]
```

Out[29]:

	title	vote_average	weighted_vote	genres	vote_count
1525	Apocalypse Now	8.0	7.708775	[Drama, War]	2055
2798	The Boy in the Striped Pyjamas	7.7	7.373173	[War, Drama]	1451
2536	The Deer Hunter	7.8	7.310348	[Drama, War]	921
1662	Glory	7.4	6.757199	[War]	383
585	War Horse	7.0	6.753283	[Drama, War]	992
2662	Sarah's Key	7.2	6.475666	[Drama, War]	196
2016	The Water Diviner	6.8	6.472943	[War, Drama]	431
557	Jarhead	6.6	6.434392	[Drama, War]	765
2671	Born on the Fourth of July	6.7	6.405936	[Drama, War]	395
3310	Far from Men	6.6	6.145019	[Drama, War]	43

화장품 cbf 추천시스템 구현

수분로션 2위



닥터뉴엘 (Dr.NUELL)

워터풀 스킨베리어 로센스

150ml / 24,000원

4.17 ♥♥♥♥ (70)

도움말

본문

도움말

본문

카테고리

로션/에멀전 수분로션

정밀확인

로션/에멀전

정밀확인

설명

에센스와 로션을 한 병에 담아 촉촉하고 산뜻하게 미네랄 밸런스

- 인체와 가장 유사한 미네랄 밸런스인 음이온양성수 79.9% 함유로 예민해진 피부를 진정하는데 도움
- 판테놀과 히알루론산으로 피부 수분 공급 및 장벽 형성에 도움
- 병물추출물과 오일물추출물로 자극 받은 피부를 편안하게 보호

태그

보습

수분공급

저자극

피부진정

도움말

본문



필터

초기화

성별

전체

여자

남자

연령대

전체

10대

20대 초반

20대 후반

30대 초반

30대 후반

피부타입

전체

건성

지성

중성

복합성

민감성

평점

전체

평평!

굿!

사사!

병!

최악!

평점

4.17

♥♥♥♥ (총 70명)

평평!

74

굿!

36

사사!

8

병!

7

최악!

0

리뷰

최신순

물루알라라라라

26세 · 건성 · ♀

👍👍👍

평평 제품이라 깨끗하게 사용 할수 있어서 좋고 제 피부에 촉촉하게 잘 맞아서 좋았어요!!

뽕

20세 · 복합성 · ♀

👍👍

처음 보는 브랜드라 걱정했지만 좋았어요 향도 좋고 트러블도 안올라오고 양도 좋아요!