NETFLIX.

추천 시스템

2022.11.10

발표자 안민재

Part 1

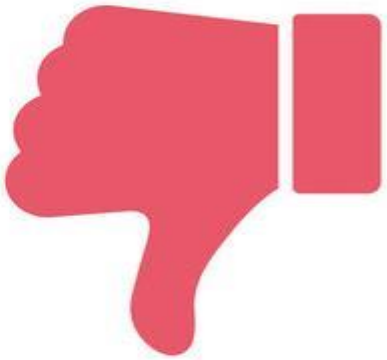
개요

1. 다양한 사람이 다양한 영화를 봄

2. 다양한 영화들 중 본인의 취향에 맞는 영화를 고르기에 어려움이 있음

3. 여러가지의 추천 시스템 모델이 존재함

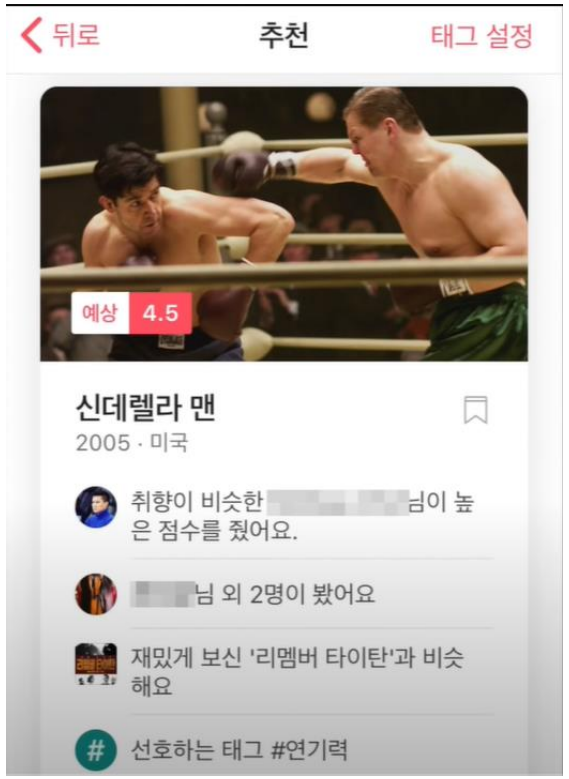
Part 1 개요

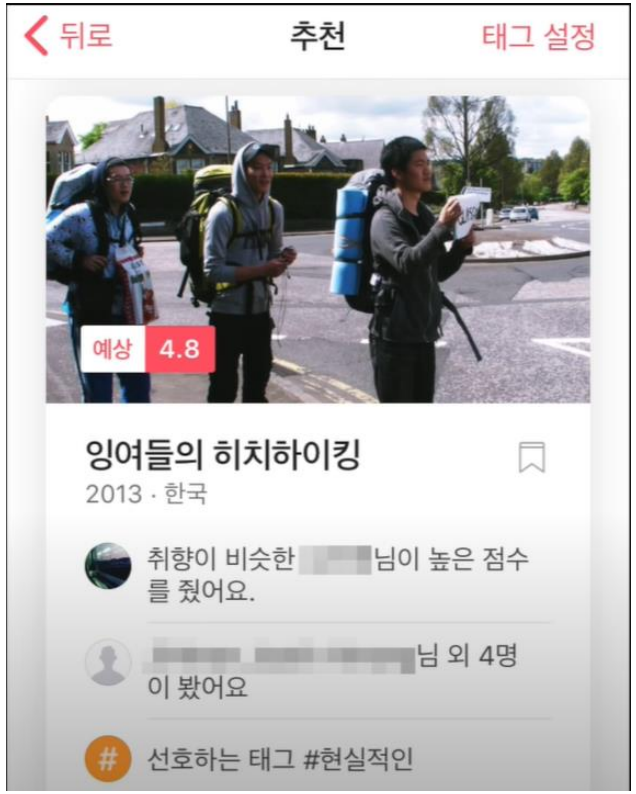


4.6

Part 1

개요





Part 2

데이터 로드 및 셋

Load Dataset

```
import numpy as np
import pandas as pd
import json
```

```
meta = pd.read_csv('the-movies-dataset/movies_metadata.csv')

meta.head()
```

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_title	overview	release_date	revenue	runtime	spoken_languages	status	tagline	title	video	vote_average	vote_count
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt0114709	en	Toy Story	Led by Woody, Andy's toys live happily in his ...	1995-10-30	373554033.0	81.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	NaN	Toy Story	False	7.7	5415.0
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt0113497	en	Jumanji	When siblings Judy and Peter discover an encha...	1995-12-15	262797249.0	104.0	[{'iso_639_1': 'en', 'name': 'English'}, {'iso...	Released	Roll the dice and unleash the excitement!	Jumanji	False	6.9	2413.0
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collect...	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	NaN	15602	tt0113228	en	Grumpier Old Men	A family wedding reignites the ancient feud be...	1995-12-22	0.0	101.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	Still Yelling. Still Fighting. Still Ready for...	Grumpier Old Men	False	6.5	92.0
3	False	NaN	16000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	31357	tt0114885	en	Waiting to Exhale	Cheated on, mistreated and stepped on, the wom...	1995-12-22	81452156.0	127.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	Friends are the people who let you be yourself...	Waiting to Exhale	False	6.1	34.0
4	False	{'id': 96871, 'name': 'Father of the Bride Col...	0	[{'id': 35, 'name': 'Comedy'}]	NaN	11862	tt0113041	en	Father of the Bride Part II	Just when George Banks has recovered from his ...	1995-02-10	76578911.0	106.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	Just When His World Is Back To Normal... He's ...	Father of the Bride Part II	False	5.7	173.0

Part 2

데이터 로드 및 셋

Load Dataset

```
meta = meta[['id', 'original_title', 'original_language', 'genres']]
meta = meta.rename(columns={'id':'movieId'})
meta = meta[meta['original_language'] == 'en']
meta.head()
```

	movieId	original_title	original_language	genres
0	862	Toy Story	en	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...
1	8844	Jumanji	en	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...
2	15602	Grumpier Old Men	en	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...
3	31357	Waiting to Exhale	en	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...
4	11862	Father of the Bride Part II	en	[{'id': 35, 'name': 'Comedy'}]

Part 2

데이터 로드 및 셋

Load Dataset

```
ratings = pd.read_csv('the-movies-dataset/ratings_small.csv')
ratings = ratings[['userId', 'movieId', 'rating']]
ratings.head()
```

	userId	movieId	rating
0	1	31	2.5
1	1	1029	3.0
2	1	1061	3.0
3	1	1129	2.0
4	1	1172	4.0

Part 2

데이터 로드 및 셋

Load Dataset

```
ratings.describe()
```

	userId	movieId	rating
count	100004.000000	100004.000000	100004.000000
mean	347.011310	12548.664363	3.543608
std	195.163838	26369.198969	1.058064
min	1.000000	1.000000	0.500000
25%	182.000000	1028.000000	3.000000
50%	367.000000	2406.500000	4.000000
75%	520.000000	5418.000000	4.000000
max	671.000000	163949.000000	5.000000

Part 2

데이터 로드 및 셋

Refine Dataset

```
def parse_genres(genres_str):
    genres = json.loads(genres_str.replace('₩'', '"'))

    genres_list = []
    for g in genres:
        genres_list.append(g['name'])

    return genres_list

meta['genres'] = meta['genres'].apply(parse_genres)

meta.head()
```

	movieId	original_title	original_language	genres
0	862	Toy Story	en	[Animation, Comedy, Family]
1	8844	Jumanji	en	[Adventure, Fantasy, Family]
2	15602	Grumpier Old Men	en	[Romance, Comedy]
3	31357	Waiting to Exhale	en	[Comedy, Drama, Romance]
4	11862	Father of the Bride Part II	en	[Comedy]

Merge Meta and Ratings

```
data = pd.merge(ratings, meta, on='movieId', how='inner')

data.head()
```

	userId	movieId	rating	original_title	original_language	genres
0	1	1371	2.5	Rocky III	en	[Drama]
1	4	1371	4.0	Rocky III	en	[Drama]
2	7	1371	3.0	Rocky III	en	[Drama]
3	19	1371	4.0	Rocky III	en	[Drama]
4	21	1371	3.0	Rocky III	en	[Drama]

Part 2

데이터 로드 및 셋

Pivot Table

```
matrix = data.pivot_table(index='userId', columns='original_title', values='rating')

matrix.head(20)
```

original_title	!Women Art Revolution	'Gator Bait	'Twas the Night Before Christmas	10 Items or Less	10 Things I Hate About You	10,000 BC	11'09"01 - September 11	12 + 1	12 Angry Men	1408	...	Young and Innocent	Zaat	Zabriskie Point	Zapped Again!	Zardoz	Zodiac
userId																	
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.5	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.5	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.5	NaN	NaN	NaN	NaN	NaN
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.0	NaN	NaN	NaN	NaN	NaN
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	NaN	NaN	NaN	NaN	NaN
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	NaN	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
15	NaN	0.5	NaN	3.0	NaN	NaN	NaN	1.0	NaN	NaN	...	1.0	NaN	NaN	NaN	NaN	NaN
16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.5	NaN	NaN	NaN	NaN	NaN
18	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
19	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.5	5.0	NaN	NaN	NaN	NaN

Part 3

Pearson Correlation

Part 3

Pearson Correlation

n : 샘플 크기

x_i , y_i : i로 인덱싱된 개별 샘플 포인트

$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$: 샘플의 평균

$$\text{피어슨상관계수} = \frac{\text{공분산}}{\text{표준편차} \cdot \text{표준편차}}$$

$$r_{XY} = \frac{\frac{\sum_i^n \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{n-1}}{\sqrt{\frac{\sum_i^n \left(X_i - \overline{X}\right)^2}{n-1}}\sqrt{\frac{\sum_i^n \left(Y_i - \overline{Y}\right)^2}{n-1}}}$$

따라서

$$r_{XY} = \frac{\sum_i^n \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_i^n \left(X_i - \overline{X}\right)^2}\sqrt{\sum_i^n \left(Y_i - \overline{Y}\right)^2}}$$

Part 3

Pearson Correlation

userId/title	영화1	영화2	영화3
1	2.5		1.0
2	5.0	4.5	2.5
3	0.5	1.5	
4			0.5

Part 3

Pearson Correlation

userId/title	영화1 S1	영화2 S2
1	2.5	
2	5.0	4.5
3	4.5	1.5
평균	4	3
	S1.mean()	S2.mean()

Part 3

Pearson Correlation

S1_c = S1 – S1.mean() S2_c = S2 – S2.mean()

userId/title	S1_c	S2_c
1	-1.5	
2	1.0	1.5
3	0.5	-1.5
평균	4	3

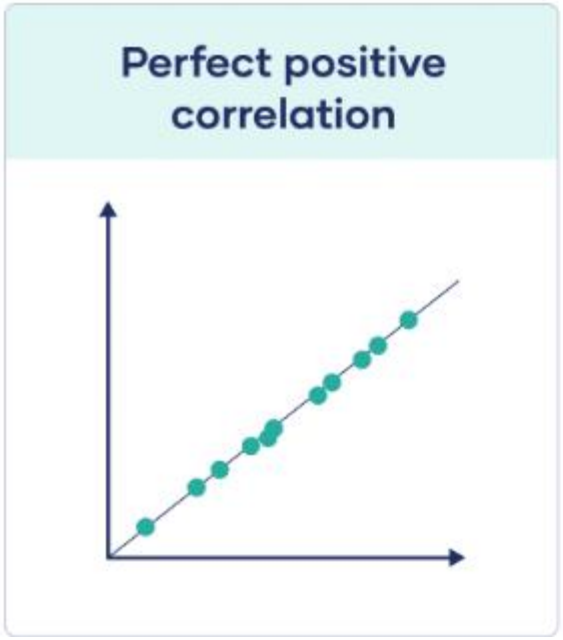
Part 3

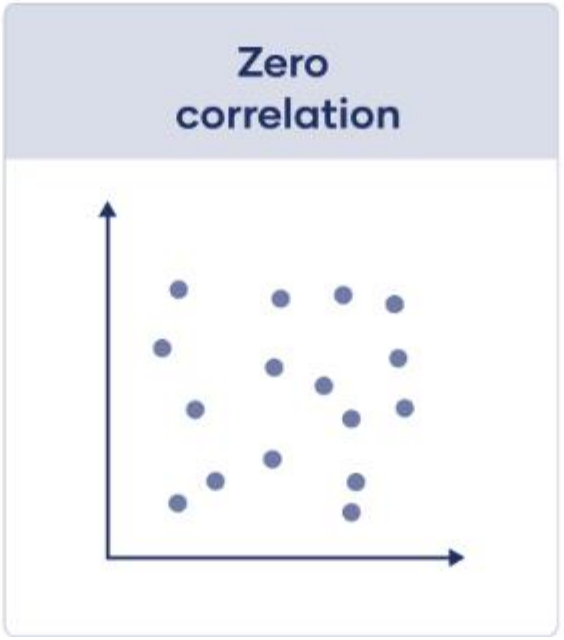
Pearson Correlation

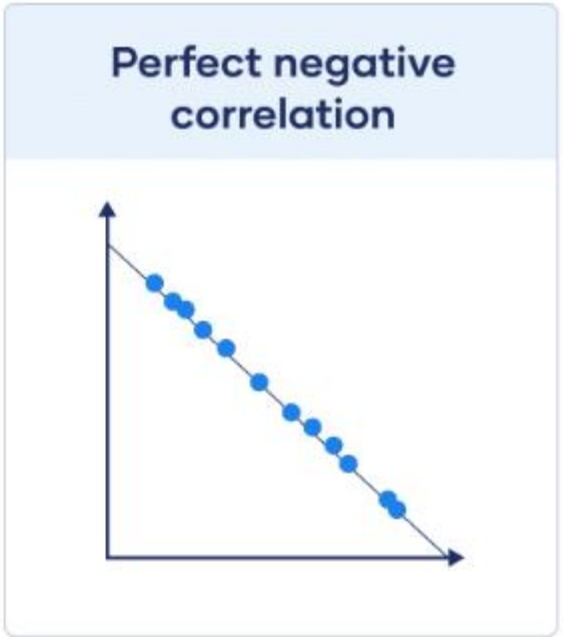
userId/title	S1_c	S2_c	S1_c * S2_c
1	-1.5		
2	1.0	1.5	1.5
3	0.5	-1.5	-0.75
Sum			0.75

Part 3

Pearson Correlation







Part 4

추천 시스템

```
GENRE_WEIGHT = 0.1

def pearsonR(s1, s2):
    s1_c = s1 - s1.mean()
    s2_c = s2 - s2.mean()
    return np.sum(s1_c * s2_c) / np.sqrt(np.sum(s1_c ** 2) * np.sum(s2_c ** 2))

def recommend(input_movie, matrix, n, similar_genre=True):
    input_genres = meta[meta['original_title'] == input_movie]['genres'].iloc(0)[0]

    result = []
    for title in matrix.columns:
        if title == input_movie:
            continue

        # rating comparison
        cor = pearsonR(matrix[input_movie], matrix[title])

        # genre comparison
        if similar_genre and len(input_genres) > 0:
            temp_genres = meta[meta['original_title'] == title]['genres'].iloc(0)[0]

            same_count = np.sum(np.isin(input_genres, temp_genres))
            cor += (GENRE_WEIGHT * same_count)

        if np.isnan(cor):
            continue
        else:
            result.append((title, '{:.2f}'.format(cor), temp_genres))

    result.sort(key=lambda r: r[1], reverse=True)

    return result[:n]
```

Part 4

테스트 결과

Prediction

```
recommend_result = recommend('The Dark Knight', matrix, 10, similar_genre=True)

pd.DataFrame(recommend_result, columns = ['Title', 'Correlation', 'Genre'])
```

	Title	Correlation	Genre
0	Prom Night	0.87	[Horror, Mystery, Thriller]
1	Wild Wild West	0.87	[Action, Adventure, Comedy, Science Fiction, W...
2	Blue Thunder	0.73	[Science Fiction, Action, Thriller, Crime, Drama]
3	Topaz	0.68	[Action, Drama, Mystery, Thriller]
4	Yamakasi - Les samouraïs des temps modernes	0.68	[Action, Crime, Drama]
5	Best Seller	0.67	[Action, Crime, Drama, Thriller]
6	Midnight in the Garden of Good and Evil	0.67	[Crime, Drama, Mystery, Thriller]
7	Big Bad Mama	0.64	[Action, Comedy, Crime, Drama]
8	The Enforcer	0.63	[Action, Crime, Thriller]
9	The River Wild	0.63	[Action, Adventure, Crime, Thriller]

감사합니다