

풀잇스쿨 13기

코드로 배우는 추천 시스템 1주차

이수진

Index

- 풀잎 소개
- 일정 및 목표
- Recommender System 기본 - CBF RecSys, CF RecSys model overview

풀잎스쿨 소개

풀잎스쿨 소개

🌱 함께 토론해요

🌱 바로 물어봐요

🌱 일단 던져봐요

🌱 같이 발표해요

🌱 함께 성장해요



풀잎스쿨은 스스로 학습하고, 이를 공유하여 함께 성장하는 방식으로 진행됩니다.

함께 모여서 스터디, 프로젝트 실습, 주제를 분담하여 발표, 자유로운 토론 등

과정에 따라서 다양한 방식으로 진행되지만

풀잎스쿨 과정들의 공통된 목표는 공유를 통한 성장입니다.

모두의연구소는 하나의 풀잎으로 함께하시게 된 여러분들을 진심으로 환영하며,

앞으로 약 3개월 간 풀잎스쿨을 통하여 혼자일 때보다 더 큰 성장을 이루시길 기원합니다.

풀잇스쿨이란

풀잇스쿨이 무엇인가요? ✨

연구에 필요한 기초 지식&기술을 플립러닝 방식으로 2~3개월 간 함께 집중적으로 학습하는 과정입니다.

퍼실이는 누구인가요? ✨

‘facilitator’에서 나온 명칭으로, 풀잇스쿨 각 과정의 조력자입니다. 함께 공부하고 싶은 과정을 기획하고 풀잇 모임을 리딩 및 진행합니다.

다른 곳들과 어떤 차이점이 있나요? ✨

수동적으로 배우게 되는 기존의 학원과 다르게 플립러닝(flipped learning, 거꾸로 학습)으로 진행됩니다. 따라서 질문과 토론을 통해 지식을 인터랙티브하게 공유하고 성장할 수 있습니다.

• 플립러닝(flipped Learning): 거꾸로 학습. 각자 학습을 해온 후 모여서 질문, 토론 등으로 공부하는 방식



이수진

#초보자 #함께공부해요 #배움을주세요

안녕하세요? 이수진이라고 합니다. 저는 이번 풀잇스쿨 12기 추천 시스템 기본반 퍼실을 담당했습니다. 12기 추천 시스템에서는 주로 논문을 읽어왔는데요. 논문을 읽다보니 실제 코드는 어떻게 되어 있고, 어떻게 적용할 수 있을지에 대한 고민이 생겼습니다. 현업에서도 프로젝트를 진행하고 있지만 다른 dataset에서의 다양한 방법도 알고 싶었습니다. 그래서 코드로 배우는 추천 시스템 컨셉의 풀잇을 만들게 되었습니다! 추천 시스템이 궁금하신 분들, 함께 공부하고 싶으신 분들! 모두 환영입니다.

풀잇스쿨 / LAB 참여 이력

- 5기: 딥러닝 기본반
- 6기: 파이토치 기본반
- 7기: 자연어 기본반(퍼실)
- 8기: 실무로 배우는 빅데이터 기술
- 9기: 파이썬으로 배우는 데이터 분석(퍼실)
- 11기: 밀바닥부터 배우는 자연어처리
- 12기: 추천 시스템 기본반(퍼실)

이론으로만 배우는 추천 시스템이 아닌, code로 봅시다!

12기에서 추천 시스템 기본반을 진행하며 이론이 아닌 구현은 어떻게 할까?에 대한 고민을 하게 되었습니다. 따라서 이번 과정에서는 아래와 같이 진행할 예정입니다.

1. 추천 시스템 기본 논문을 Overview합니다.
2. 관련 code를 찾고 필사 및 구현을 해봅니다.
3. 이것을 어떻게 개인/현업에서 적용할 수 있을지 그리고 개선사항 등을 토론합니다.
4. 질문 사항 및 궁금한 점을 토론하고 마칩니다.

이런 분들께 추천합니다

- 파이썬으로 진행됩니다. 따라서 Python에 대한 기본 지식이 있으셔야합니다.
- DL/ML에 대한 기본적인 지식이 있으시면 좋습니다.
- 부족한 것을 채우려는 열정! (기본적인 Recommender System model은 Overview 해드릴 예정입니다. 그러나 Overview이기 때문에 상세한 내용이 빠질 수도 있습니다. 해당 부분을 채울 수 있는 열정이 있으시면 좋습니다!)

일정 및 목표

02 일정 및 목표

이 반을 만든 이유?

- 추천 시스템?
 - 위키백과
 - 추천 시스템은 정보 필터링 기술의 일종으로 특정 사용자가 관심을 가질만한 정보를 추천하는 것
 - 추천 시스템에는 협업 필터링 기법을 주로 사용한다.
- 본인이 생각하는 추천 시스템?
 - 내가 몰랐던 흥미, 관심사를 알려주는 시스템
 - 회사와 사용자간의 Win-Win 할 수 있는 서비스 (반대가 될 수도 있음)
- 회사에서 추천 시스템 프로젝트를 하고 있음
 - (그래도 잘 모르겠ㅇ...)

02 일정 및 목표 이 반을 만든 이유?

• 12기 추천 시스템 기본 반

• 코로나..!

- 코드와 함께 공부하는 것을 실패함
- 논문만 주구장창 읽었음 ㅠ

스케줄 Default View ▼				
📅 날짜	Aa 내용	👤 진행자	📄 진행자료	🔗 진행자료
Jul 18, 2020	📄 OT : 풀잎 소개 및 진행 방향, 추천 시스템 개요	퍼실		풀잎12기_1주차....
Jul 25, 2020	📄 2주차 : Content based recommender system (with Kaggle) & Collaborative Filtering (part 1)	문영주님, 김완기님	https://lsjsj92.tistory.com/563 https://lsjsj92.tistory.com/564 https://lsjsj92.tistory.com/565	https://www.n...
Aug 01, 2020	3주차 : Collaborative Filtering (part 2) (with Kaggle)	양진욱님, 김재윤님	https://lsjsj92.tistory.com/568 https://lsjsj92.tistory.com/569 https://lsjsj92.tistory.com/570	CF_MF_v1.ipynb
Aug 08, 2020	4주차 : Deep neural networks for youtube recommendations	김정원님, 노여명님	https://static.googleusercontent.com/media/research.google.com/ko//pubs/archive/45530.pdf	https://www.n... Candidate Ge...
Aug 15, 2020	5주차 : wide & deep learning for recommender systems	정지혜님, 강현욱님	https://arxiv.org/pdf/1606.07792.pdf	https://www.n... Wide&Deep 추...
Aug 22, 2020	6주차 : ad click prediction a view from the trenches	왕성현님, 김희은님	https://static.googleusercontent.com/media/research.google.com/ko//pubs/archive/41159.pdf	Ad Click Predi... https://www.n...
Aug 29, 2020	7주차 : related pins at pinterest the evolution of a real-world recommender system	박용연님, 서미지님	https://arxiv.org/pdf/1702.07969.pdf	풀잎발표_7주차....
Sep 05, 2020	8주차 : Multi Armed Bandit	이혜진님, 허정준님	각종 오픈된 블로그 자료	
Sep 12, 2020	9주차 : Session based 4RecSys	염은지님, 성은지님	https://arxiv.org/pdf/1511.06939.pdf	
Sep 19, 2020	11주차 : Self-Attentive sequential recommender system(7주차때 논의)	이수진	https://arxiv.org/pdf/1808.09781.pdf	
Sep 26, 2020	12주차 : Graph Convolutional Matrix Completion	이수진	https://arxiv.org/pdf/1706.02263.pdf	

02 일정 및 목표 계획

- 총 11주차
 - 1 ~ 3주차는 퍼실이 진행
 - 4주차부터는 팀(5~6개)으로 진행할 예정
 - 각 팀당 3명씩
 - 2 ~ 3주차에 random.sample

커리큘럼

Chapter 1. RecSys paper overview

1주차 : 스터디 오리엔테이션 및 CBF RecSys, CF RecSys model overview

2주차 : Youtube RecSys overview, Wide & Deep RecSys overview

3주차 : Sequential Based RecSys overview, Graph based RecSys overview

Chapter 2. RecSys with Python (코드 필사 및 구현)

4주차 : Youtube RecSys 코드 필사 및 구현

5주차 : Wide & Deep 코드 필사 및 구현

6주차 : Sequential RecSys 코드 필사 및 구현

Chapter 3. RecSys 대회에서는 어떻게 활용하였을까?

7주차 : Kakao Arena 코드 필사 및 구현 - 1

8주차 : Kakao Arena 코드 필사 및 구현 - 2

Chapter 4. 우리만의 그리고 나만의 RecSys!

9주차 : Open dataset을 활용한 자신만의 RecSys 구현하기

10주차 : Open dataset을 활용 및 팀원들과 함께 RecSys 구현하기

11주차 : 총 정리 및 회고

* 이론+실습으로 진행되는 과정입니다. 참고해주세요.

02 일정 및 목표

RULE

- 우리 반의 규칙
 - 소스 코드 등 자료 조사를 하면서 찾게 된 정보를 slack 저희 채널에 공유해주세요!
 - 옆 사람, 팀 동료가 못하면 같이 이끌어주세요!
 - 질문, 궁금한 것이 있으면 언제든지 적극적으로 질문해주세요!
 - 스터디 날에만 질문하는 것이 아니라 Slack으로도 궁금한 것이 있으면 올려주세요!
 - 본인이 궁금한 것은 다른 사람들도 궁금해합니다
 - 민망한, 부끄러운 질문은 없습니다

Side project로 추천 시스템 만들어봤다!

(우길 수 있도록..)

Recommender System 기본

- CBF RecSys, CF RecSys model overview -

03 추천 시스템 기본

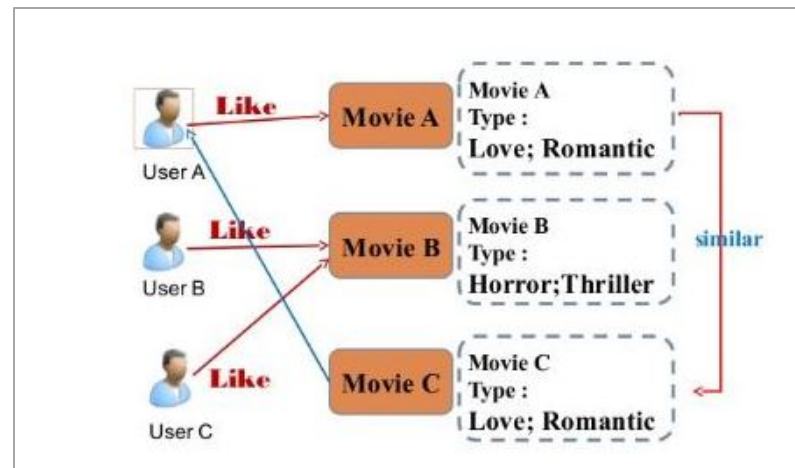
Content Based Filtering

- 추천 시스템 기본
 - Content based filtering
 - Collaborative Filtering
- 더 나아가서
 - Hybrid model
 - Deep Learning / Machine Learning based
 - 2~3주차에 배울 내용이 Deep Learning based
- Content Based Filtering
 - 가장 기본적인 아이디어
 - 추천 시스템 초창기에 많이 활용되던 방법이며 아직까지도 단순한 시스템에서 활용

03 추천 시스템 기본

Content Based Filtering

- Content Based Filtering
 - 예를 들어 ‘이수진’ 감독의 Action 카테고리의 A라는 영화가 존재
 - 어떤 사용자가 A라는 영화를 보았다면
 - ‘이수진’ 감독의 다른 Action 영화를 추천



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

```
In [3]: data.head(2)
```

Out[3]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_c
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}]	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "culture clash"}, {"id": 1465, "name": "culture clash"}]	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Inglorious Film Partners", "id": 289...}]
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 15, "name": "Science Fiction"}]	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "name": "ocean"}, {"id": 727, "name": "ocean"}]	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt Disney Pictures", "id": 289...}]

03 추천 시스템 기본

Content Based Filtering

```
In [16]: data[['genres', 'keywords']].head(2)
```

```
Out[16]:
```

	genres	keywords
0	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 15, "name": "Science Fiction"}]	[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "future space"}, {"id": 1465, "name": "war space colony so..."}]
1	[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 15, "name": "Science Fiction"}]	[{"id": 270, "name": "ocean"}, {"id": 726, "name": "drug abuse"}, {"id": 727, "name": "exotic island"}, {"id": 728, "name": "east india trad..."}]

```
In [19]: data['genres'] = data['genres'].apply(lambda x : [d['name'] for d in x]).apply(lambda x : " ".join(x))
data['keywords'] = data['keywords'].apply(lambda x : [d['name'] for d in x]).apply(lambda x : " ".join(x))
```

```
In [20]: data.head(2)
```

```
Out[20]:
```

	id	genres	vote_average	vote_count	popularity	title	keywords	overview	score
0	19995	Action Adventure Fantasy Science Fiction	7.2	11800	150.437577	Avatar	culture clash future space war space colony so...	In the 22nd century, a paraplegic Marine is di...	7.168053
1	285	Adventure Fantasy Action	6.9	4500	139.082615	Pirates of the Caribbean: At World's End	ocean drug abuse exotic island east india trad...	Captain Barbossa, long believed to be dead, ha...	6.918271

```
In [23]: count_vector = CountVectorizer(ngram_range=(1, 3))
```

```
In [24]: c_vector_genres = count_vector.fit_transform(data['genres'])
```

```
In [25]: c_vector_genres.shape
```

```
Out[25]: (481, 364)
```


03 추천 시스템 기본

Content Based Filtering

```
In [30]: data[data['title'] == 'The Dark Knight Rises']
```

Out[30]:

	id	genres	vote_average	vote_count	popularity	title	keywords	overview	score
3	49026	Action Crime Drama Thriller	7.6	9106	112.31295	The Dark Knight Rises	dc comics crime fighter terrorist secret ident...	Following the death of District Attorney Harve...	7.492998

```
In [29]: get_recommend_movie_list(data, movie_title='The Dark Knight Rises')
```

Out[29]:

	id	genres	vote_average	vote_count	popularity	title	keywords	overview	score
65	155	Drama Action Crime Thriller	8.2	12002	187.322927	The Dark Knight	dc comics crime fighter secret identity scarec...	Batman raises the stakes in his war on crime. ...	8.035690
2091	274	Crime Drama Thriller	8.1	4443	18.174804	The Silence of the Lambs	based on novel psychopath horror suspense seri...	FBI trainee, Clarice Starling ventures into a ...	7.767228
2760	264644	Drama Thriller	8.1	2757	66.113340	Room	based on novel carpet isolation kidnapping imp...	Jack is a young boy of 5 years old who has liv...	7.645138
351	1422	Drama Thriller Crime	7.9	4339	63.429157	The Departed	undercover boston police friends mafia underco...	To take down South Boston's Irish Mafia, the p...	7.621146
1850	111	Action Crime Drama Thriller	8.0	2948	70.105981	Scarface	miami corruption capitalism cuba prohibition b...	After getting a green card in exchange for ass...	7.601698
4337	103	Crime Drama	8.0	2535	58.845025	Taxi Driver	vietnam veteran taxi obsession drug dealer nig...	A mentally unstable Vietnam War veteran works ...	7.564085
1051	146233	Drama Thriller Crime	7.9	3085	88.496873	Prisoners	pennsylvania kidnapping maze vigilante rural s...	When Keller Dover's daughter and her friend go...	7.550121
828	24	Action Crime	7.7	4949	79.754966	Kill Bill: Vol. 1	japan coma martial arts kung fu underworld yak...	An assassin is shot at the altar by her ruthle...	7.500378
3701	641	Crime Drama	7.9	2443	11.573034	Requiem for a Dream	drug addiction junkie heroin speed diet unsoci...	The hopes and dreams of four ambitious people ...	7.497657
1829	6977	Crime Drama Thriller	7.7	3003	53.645267	No Country for Old Men	texas drug traffic hitman united states-mexico...	Llewelyn Moss stumbles upon dead bodies, \$2 mi...	7.420140

고도화 하려면 어떻게 하면 좋을까?

03 추천 시스템 기본

Collaborative Filtering

- Collaborative Filtering

- 현재까지도 많이 사용하는 추천 시스템 방법
- User의 Action data를 활용해서 추천을 진행함
 - 사용자가 아직 평가하지 않은 데이터 중에서 추천

- 크게 2가지로 나뉘어짐

- Nearest Neighbor based

- User based

- 비슷한 고객들이 ~한 Item을 소비했다


- Item based

- 당신이 구매한 Item을 소비한 고객들은 다음과 같은 상품도 구매했다

- Latent factor based

- 잠재 요소 기반

Userid	Item id	rating
User1	itemA	4
User1	ItemC	3
User2	ItemA	3
User2	ItemB	2
User3	itemD	5



	Item1	Item2	Item3	Item4
User1	4		3	
User2	3	3		
user3				5

03 추천 시스템 기본

Collaborative Filtering

- Nearest Neighbor based
 - User based CF (많이 사용 안함)

유사도 높음

	<u>ItemA</u>	<u>ItemB</u>	<u>ItemC</u>	<u>ItemD</u>	<u>ItemE</u>
User1	3	4	4		1
User2	4	4	4	3	
user3	1	1	2	5	

- Item based CF (많이 사용 함)

유사도 높음

	User1	User2	User3	User4	User5
<u>ItemA</u>	5	4	4		5
<u>ItemB</u>	4	4	4	5	
<u>ItemC</u>	1	1	2	3	

03

Collaborative Filtering

- Nearest Neighbor based

```
In [35]: data = ratings.pivot_table('rating', index = 'userId', columns = 'title').fillna(0)
```

```
In [36]: data.head()
```

Out[36]:

title	10 Things I Hate About You	12 Angry Men	1408	15 Minutes	16 Blocks	20,000 Leagues Under the Sea	2001: A Space Odyssey	2046	21 Grams	25th Hour	...	Willy Wonka & the Chocolate Factory	World Trade Center	X-Men Origins: Wolverine	Y Tu Mamá También	You Only Live Twice	You, Me and Dupree	Young Frankenstein
userId																		
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	...	0.0	0.0	5.0	0.0	0.0	0.0	5.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
In [38]: data = data.transpose()
data.head(2)
```

Out[38]:

[illegible]

03 추천 시스템 기본

Collaborative Filtering

- Nearest Neighbor based

```
In [40]: movie_sim = cosine_similarity(data, data)
print(movie_sim.shape)
```

```
(856, 856)
```

```
In [42]: movie_sim_df = pd.DataFrame(data = movie_sim, index = data.index, columns = data.index)
```

```
In [43]: movie_sim_df.head(3)
```

```
Out[43]:
```

title	10 Things I Hate About You	12 Angry Men	1408	15 Minutes	16 Blocks	20,000 Leagues Under the Sea	2001: A Space Odyssey	2046	21 Grams	25th Hour	...	Willy Wonka & the Chocolate Factory	World Trade Center	X-Men Origins: Wolverine	Y Tu Mamá También	You Only Live Twice	You, Me and Dupree	Young Franker
title																		
10 Things I Hate About You	1.0	0.0	0.0	0.182153	0.0	0.022069	0.085323	0.0	0.0	0.10349	...	0.059856	0.0	0.161801	0.088076	0.0	0.0	0.09758
12 Angry Men	0.0	1.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.00000	...	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.00000
1408	0.0	0.0	1.0	0.447214	0.0	0.173381	0.028245	0.0	0.0	0.00000	...	0.146955	0.0	0.148968	0.140265	0.0	0.0	0.19167

03 추천 시스템 기본

Collaborative Filtering

- Nearest Neighbor based

```
In [47]: movie_sim_df["X-Men Origins: Wolverine"].sort_values(ascending=False)[1:10]
```

```
Out[47]: title
Romeo Must Die                0.649625
The Wedding Planner           0.631669
Dogtown and Z-Boys            0.501189
An Unfinished Life            0.485643
Conquest of the Planet of the Apes 0.474626
Reign Over Me                 0.458155
The Terminal                  0.445337
Young Frankenstein            0.423840
Whale Rider                   0.394136
Name: X-Men Origins: Wolverine, dtype: float64
```

```
In [58]: movie_sim_df["Harry Potter and the Half-Blood Prince"].sort
```

```
Out[58]: title
Liar Liar                    1.000000
Family Plot                  1.000000
Once                         1.000000
Synecdoche, New York         1.000000
Rendition                    1.000000
Harry Potter and the Half-Blood Prince 1.000000
The Astronaut Farmer         0.970143
Schindler's List              0.724286
The Last King of Scotland    0.707107
Name: Harry Potter and the Half-Blood Prince, dtype: float64
```

```
In [33]: get_item_based_collabor('Godfather, The (1972)')
```

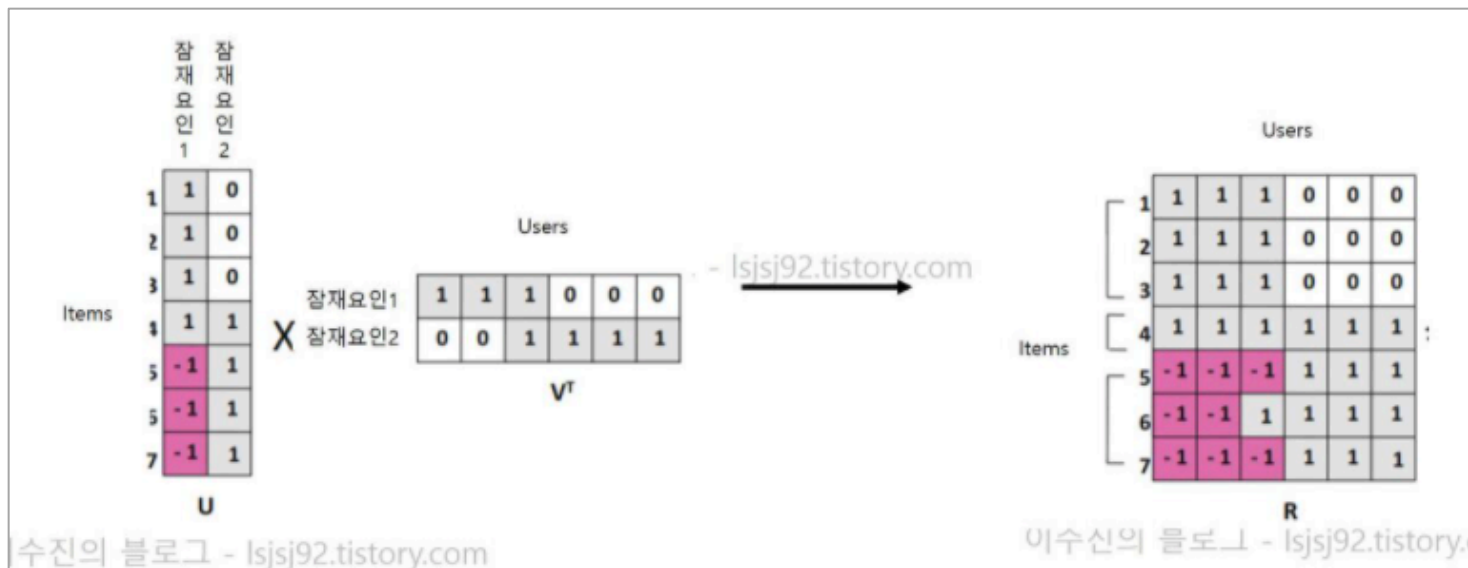
```
Out[33]: title
Godfather, The (1972)                1.000000
Godfather: Part II, The (1974)       0.773685
Goodfellas (1990)                   0.620349
One Flew Over the Cuckoo's Nest (1975) 0.568244
American Beauty (1999)              0.557997
Star Wars: Episode IV - A New Hope (1977) 0.546750
Name: Godfather, The (1972), dtype: float64
```

03 추천 시스템 기본

Collaborative Filtering

- Latent factor based collaborative filtering
 - 정말 많이 사용하는 방법
 - 대규모 다차원 행렬을 분해하는 방법 사용
 - Matrix Factorization
 - 분해하는 과정에서 잠재 요소(latent factor)를 찾아내어 뽑아냄
 - Matrix의 크기도 감소하고 평가하지 않은 부분에 대해서 score를 추측할 수 있음

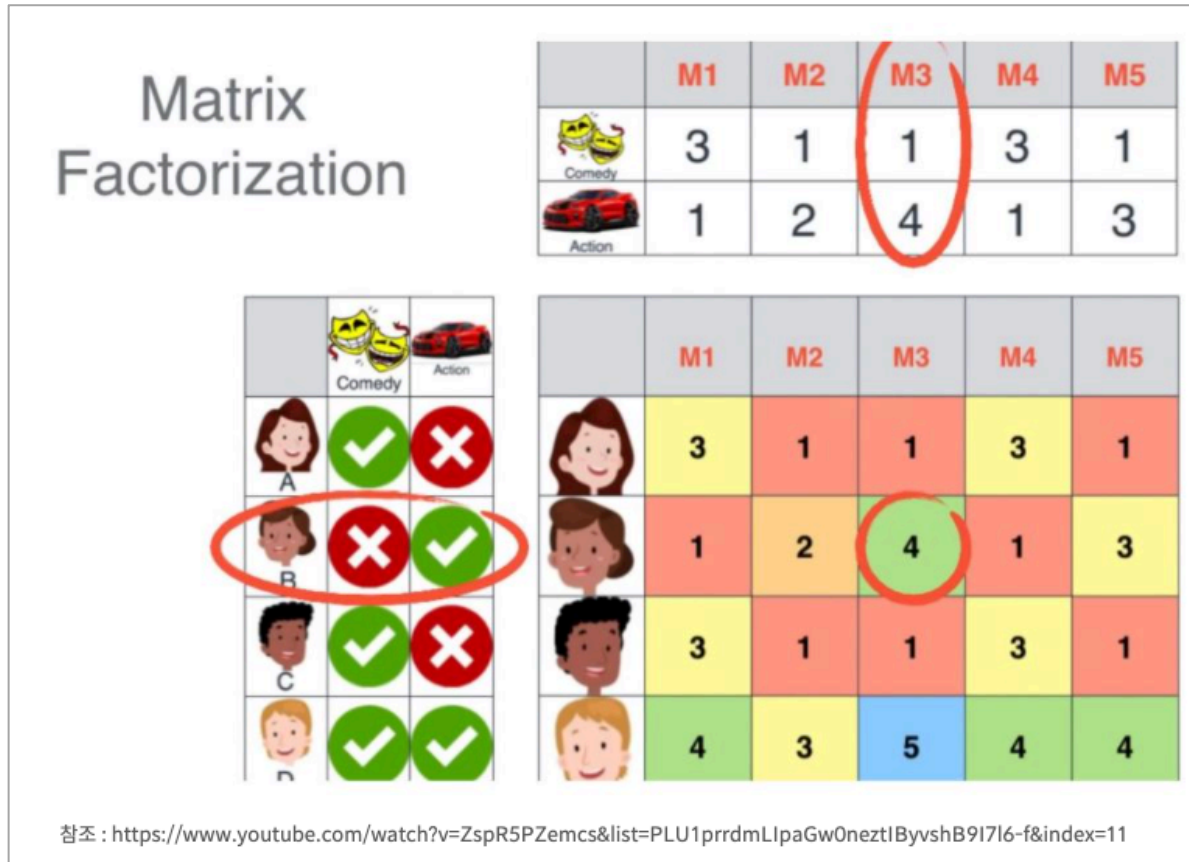
- User - Item Matrix 분해
 - User - K (U)
 - Item - K (V)
 - $U * V^T = \text{User - Item Matrix}$



03 추천 시스템 기본

Collaborative Filtering

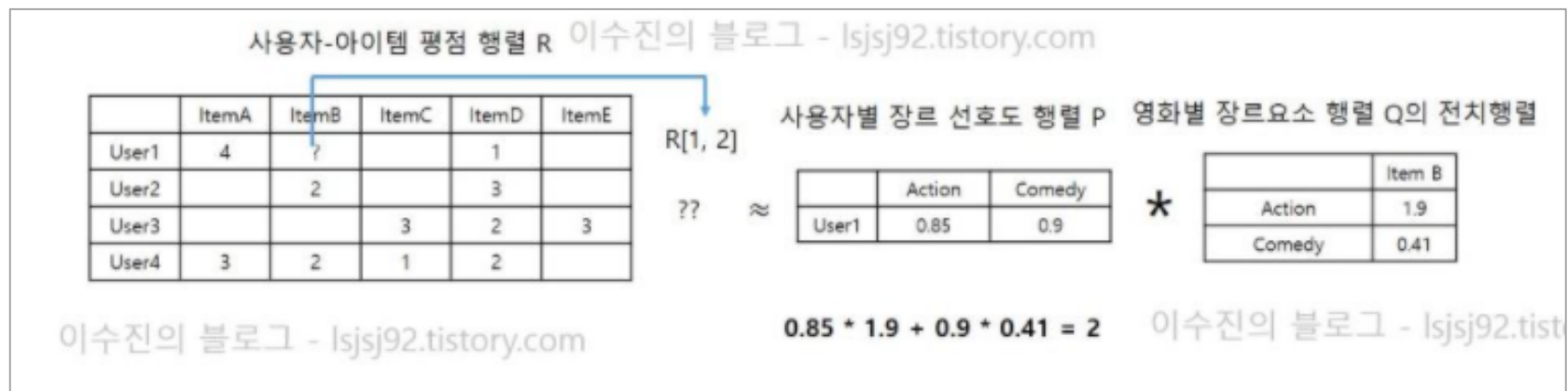
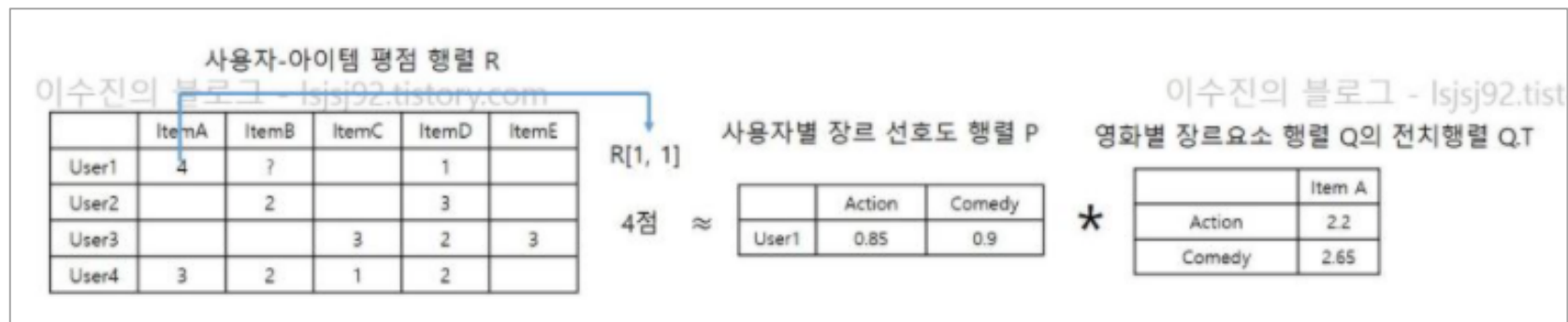
- Latent factor based collaborative filtering
 - example



03 추천 시스템 기본

Collaborative Filtering

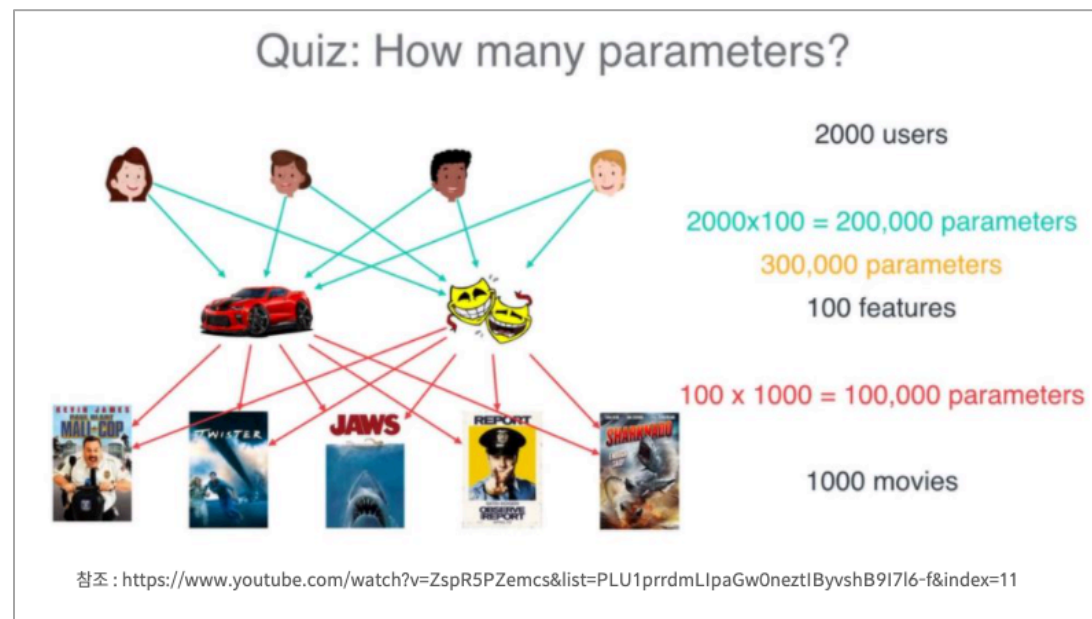
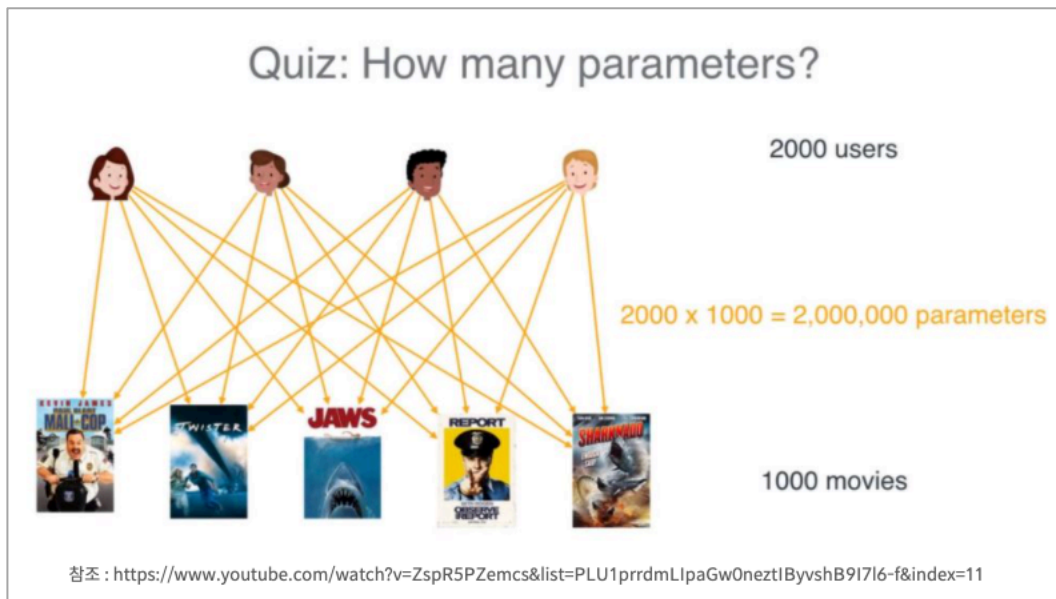
- Latent factor based collaborative filtering



03 추천 시스템 기본

Collaborative Filtering

- Latent factor based collaborative filtering



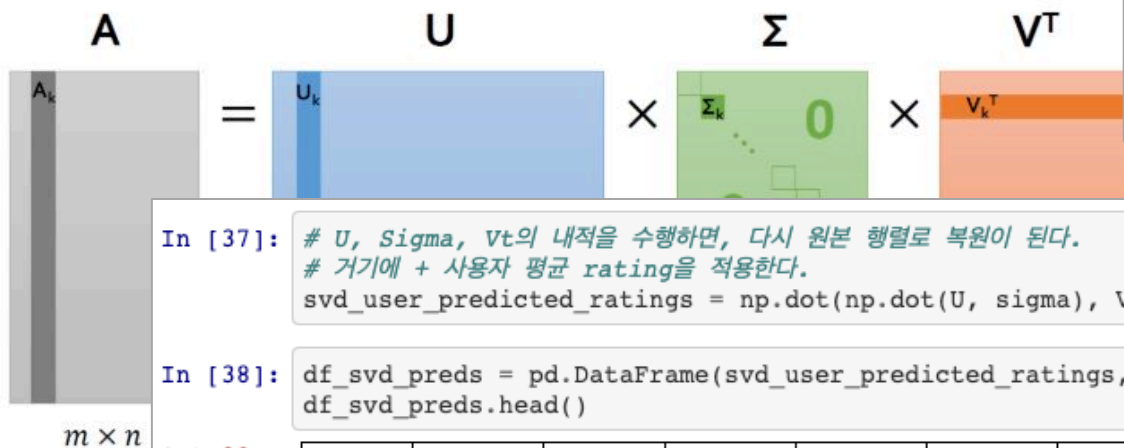
03 추천 시스템 기본

Collaborative Filtering

- Latent factor based collaborative filtering

SVD(Singular Value Decomposition), 특이값 분해란 $m \times n$ 크기의 데이터 행렬 A 를 아래와 같이 분해하는 것을 말합니다.

$$A = U \Sigma V^T$$



```
In [15]: SVD = TruncatedSVD(n_components=12)
matrix = SVD.fit_transform(movie_user_rating)
matrix.shape
```

```
Out[15]: (9064, 12)
```

```
In [16]: matrix[0]
```

```
Out[16]: array([ 0.01227491,  0.00250724,  0.01554347, -0.03396397, -0.01454107,
                  0.00372211, -0.0019783 ,  0.04563075, -0.01597626, -0.02075279,
                  0.00934806, -0.01379995])
```

```
In [37]: # U, Sigma, Vt의 내적을 수행하면, 다시 원본 행렬로 복원이 된다.
# 거기에 + 사용자 평균 rating을 적용한다.
svd_user_predicted_ratings = np.dot(np.dot(U, sigma), Vt) + user_ratings_mean.reshape(-1, 1)
```

```
In [38]: df_svd_preds = pd.DataFrame(svd_user_predicted_ratings, columns = df_user_movie_ratings.columns)
df_svd_preds.head()
```

```
Out[38]:
```

	movieId	1	2	3	4	5	6	7	8	9	10	...	161084	161155	161594	161830
0		-0.079686	0.021779	-0.013837	-0.005870	-0.028877	0.032371	0.000715	-0.004428	-0.005219	0.038195	...	-0.004324	-0.004352	0.010478	-0.0004
1		1.428452	1.608841	0.529476	0.168278	0.520809	1.107473	0.529719	0.089376	0.296270	1.970031	...	0.013227	-0.002275	0.020680	-0.0005
2		0.977246	0.396971	0.000299	0.027444	0.021287	0.141458	-0.057134	0.031633	-0.012538	0.383576	...	0.002761	0.004907	-0.014190	-0.0000
3		1.870844	1.169993	0.252202	0.094831	-0.181713	-0.511953	-0.027820	-0.143080	0.013247	1.461694	...	0.026412	-0.027245	0.054681	0.0184
4		1.182777	0.924903	0.075998	0.061505	0.602680	-0.159825	0.339925	0.081534	-0.079666	0.535018	...	-0.029124	-0.029357	0.009064	-0.0290

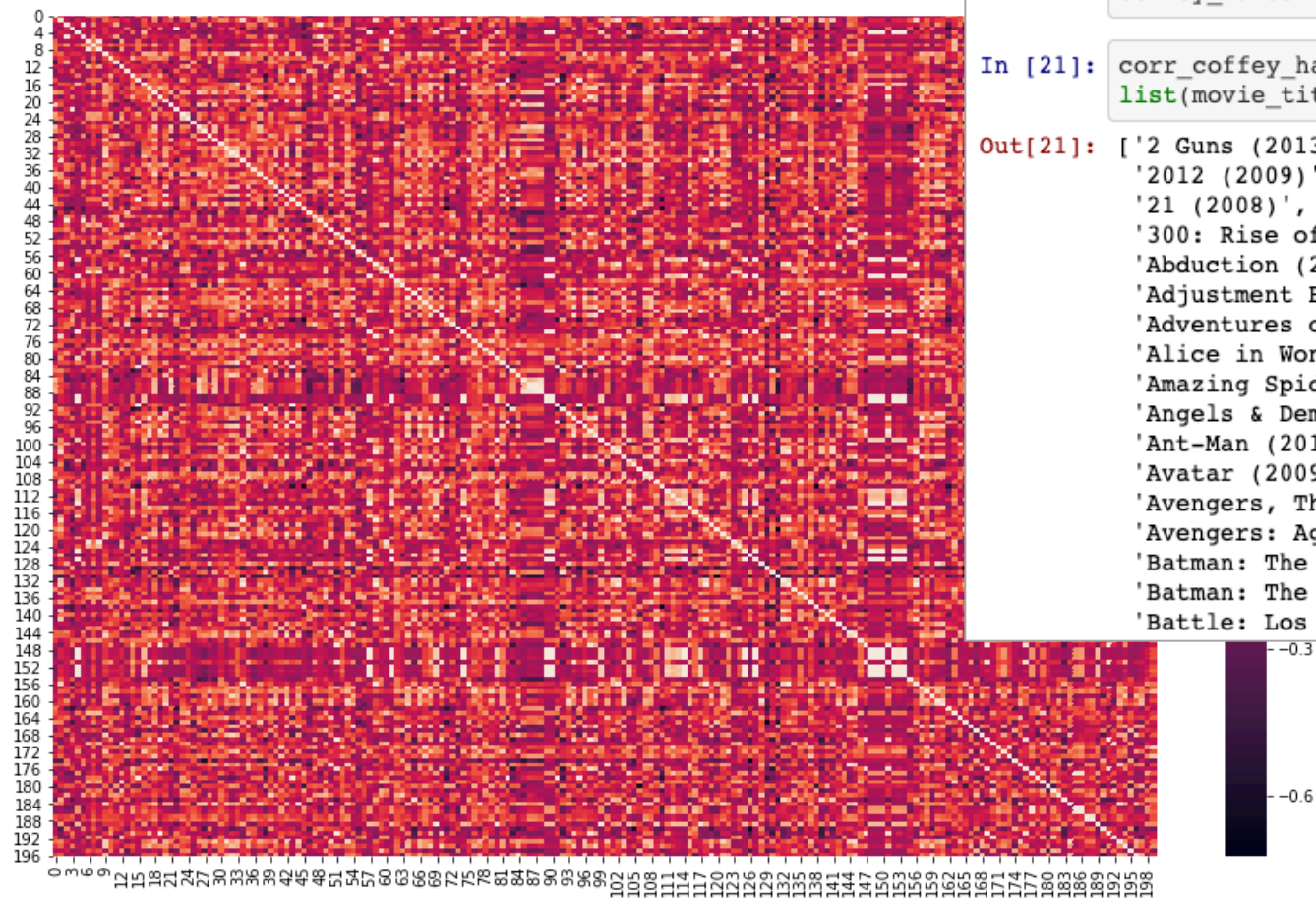
03 추천 시스템 기본

Collaborative Filtering

- Latent factor based collaborative filtering

```
In [19]: plt.figure(figsize=(16, 10))  
sns.heatmap(corr2)
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x2b4f9739be0>
```



```
In [20]: movie_title = user_movie_rating.columns  
movie_title_list = list(movie_title)  
coffey_hands = movie_title_list.index("Guardians of the Galaxy (2014)")
```

```
In [21]: corr_coffey_hands = corr[coffey_hands]  
list(movie_title[(corr_coffey_hands >= 0.9)][:50])
```

```
Out[21]: ['2 Guns (2013)',  
'2012 (2009)',  
'21 (2008)',  
'300: Rise of an Empire (2014)',  
'Abduction (2011)',  
'Adjustment Bureau, The (2011)',  
'Adventures of Tintin, The (2011)',  
'Alice in Wonderland (2010)',  
'Amazing Spider-Man, The (2012)',  
'Angels & Demons (2009)',  
'Ant-Man (2015)',  
'Avatar (2009)',  
'Avengers, The (2012)',  
'Avengers: Age of Ultron (2015)',  
'Batman: The Dark Knight Returns, Part 1 (2012)',  
'Batman: The Dark Knight Returns, Part 2 (2013)',  
'Battle: Los Angeles (2011)',
```

03 추천 시스템 기본

Collaborative Filtering

- Latent factor based collaborative filtering

In [42]: `alreadyRated.head(10)`

Out[42]:

	userId	movieId	rating	timestamp	title	genres
47	330	1094	5.0	948577200	Crying Game, The (1992)	Drama Romance Thriller
14	330	213	5.0	948577039	Burnt by the Sun (Utomlyonnye solntsem) (1994)	Drama
34	330	527	5.0	948577164	Schindler's List (1993)	Drama War
45	330	1035	5.0	948574126	Sound of Music, The (1965)	Musical Romance
48	330	1172	5.0	948575236	Cinema Paradiso (Nuovo cinema Paradiso) (1989)	Drama
53	330	1249	5.0	948576522	Femme Nikita, La (Nikita) (1990)	Action Crime Romance Thriller
27	330	373	5.0	948576522	Red Rock West (1992)	Thriller
58	330	1428	5.0	948731410	Angel Baby (1995)	
61	330	1575	5.0	948577442	Gabbeh (1996)	
67	330	1683	5.0	948578017	Wings of the Dove, The (1997)	

In [43]: `predictions`

Out[43]:

	movieId	title	genres	Predictions
246	296	Pulp Fiction (1994)	Comedy Crime Drama Thriller	2.060064
294	356	Forrest Gump (1994)	Comedy Drama Romance War	1.772814
497	608	Fargo (1996)	Comedy Crime Drama Thriller	1.526111
487	590	Dances with Wolves (1990)	Adventure Drama Western	1.508960
316	380	True Lies (1994)	Action Adventure Comedy Romance Thriller	1.276646
40	47	Seven (a.k.a. Se7en) (1995)	Mystery Thriller	1.265219
313	377	Speed (1994)	Action Romance Thriller	1.142113
214	260	Star Wars: Episode IV - A New Hope (1977)	Action Adventure Sci-Fi	1.076539
295	357	Four Weddings and a Funeral (1994)	Comedy Romance	1.065911
488	592	Batman (1989)	Action Crime Thriller	1.048861



Thanks!