



# 캡스톤 디자인(2)

## 중간발표

---

머신러닝을 이용한 영화 탐색 및 추천 웹

# 목차

---

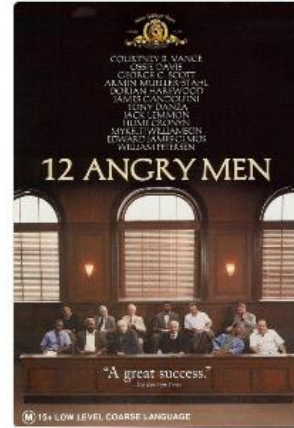
1. 개요 및 목적
2. 요구명세
3. 진행상황
4. 예상 결과
5. 향후 계획

# 개요 및 목적



영화 검색

예상별점이 높은 작품



12인의 노한 사람들  
예상 ★4.5

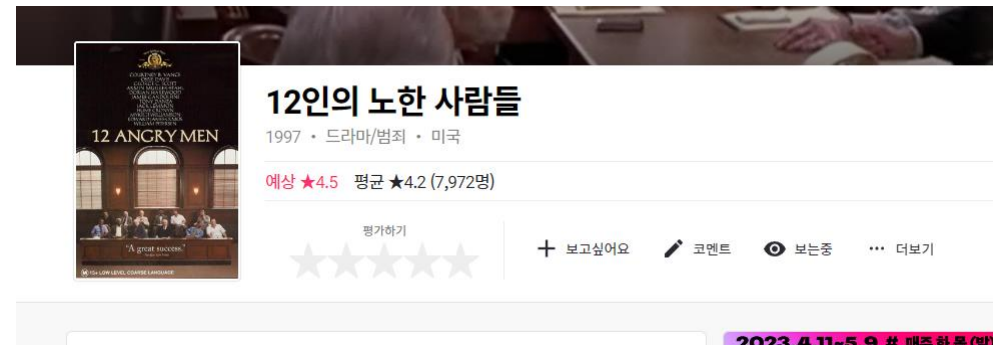


피어나  
예상 ★4.1



천국보다 아름다운  
예상 ★4.2

예상 별점이 높은 작품 추천



영화 평가, 코멘트 남기기, 커뮤니티

# 요구 명세

## User Requirement

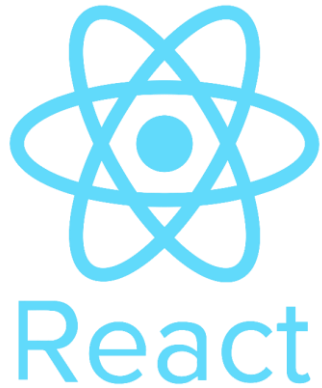
1. 사용자가 원하는 영화를 검색하여 찾을 수 있다.
2. 로그인 이 가능하고 리뷰와 평점을 작성하여 저장 할 수 있다.
3. 작성한 평점을 바탕으로 사용자에게 맞는 영화를 추천해줄 수 있다.

## System Requirement

1. URL에 알맞은 페이지를 전송한다.
2. 영화 정보 Api에서 요구한 영화의 상세정보 등을 올바르게 불러온다.
3. 로그인, 리뷰 등의 정보를 DB에 올바르게 작성하고 불러 올 수 있다.
4. 목적에 맞는 머신러닝 Api 를 작성하고 활용 할 수 있다.

# 소프트웨어 아키텍처

Client - server



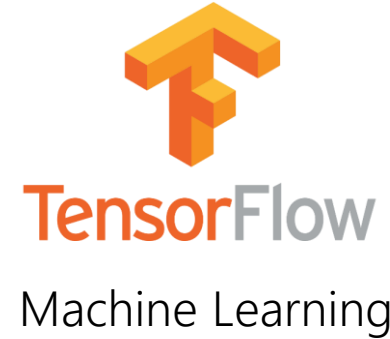
Client



Framework



Server



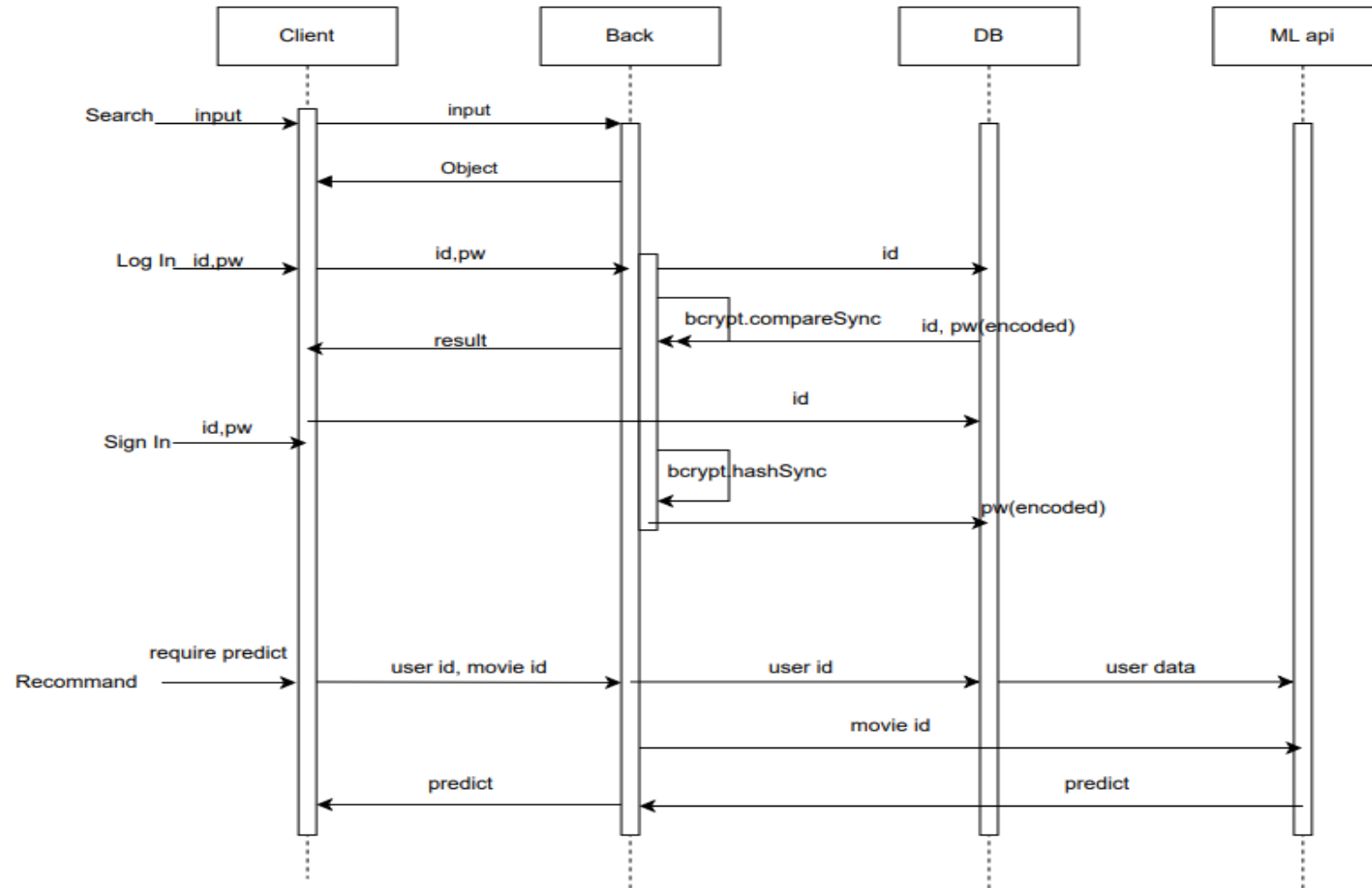
Data Base



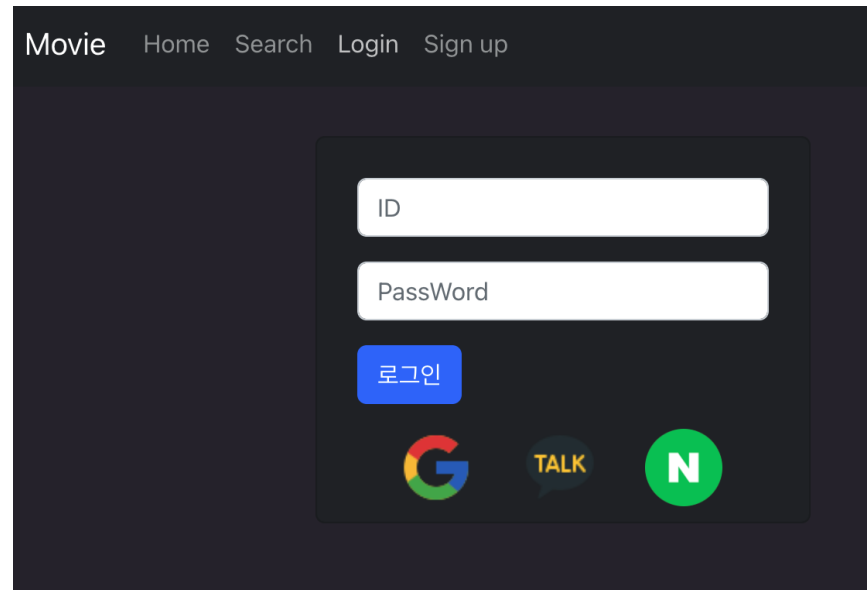
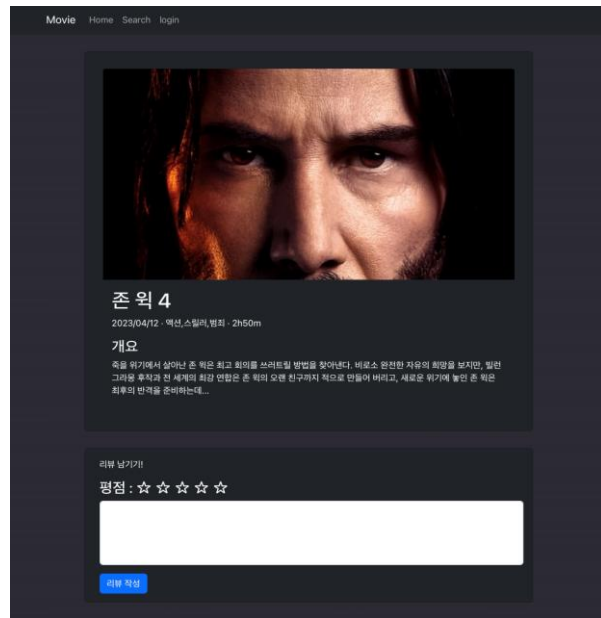
OAuth

OMDB API

# Sequence Diagram



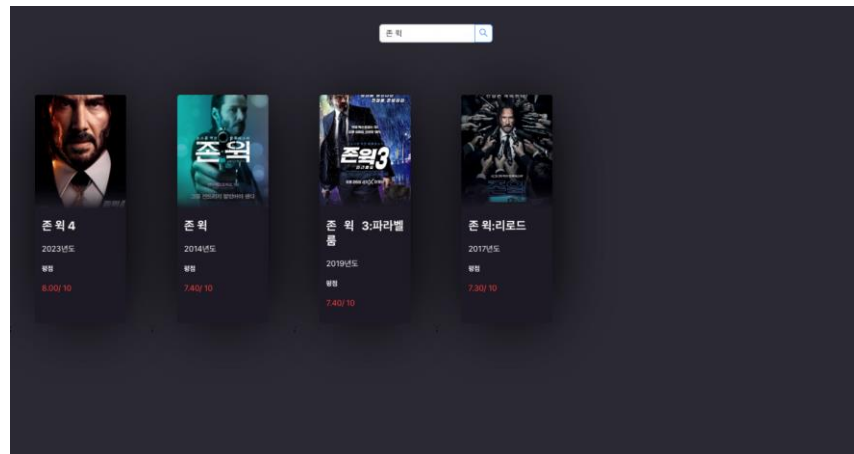
# 진행상황



## 1학기 완료사항

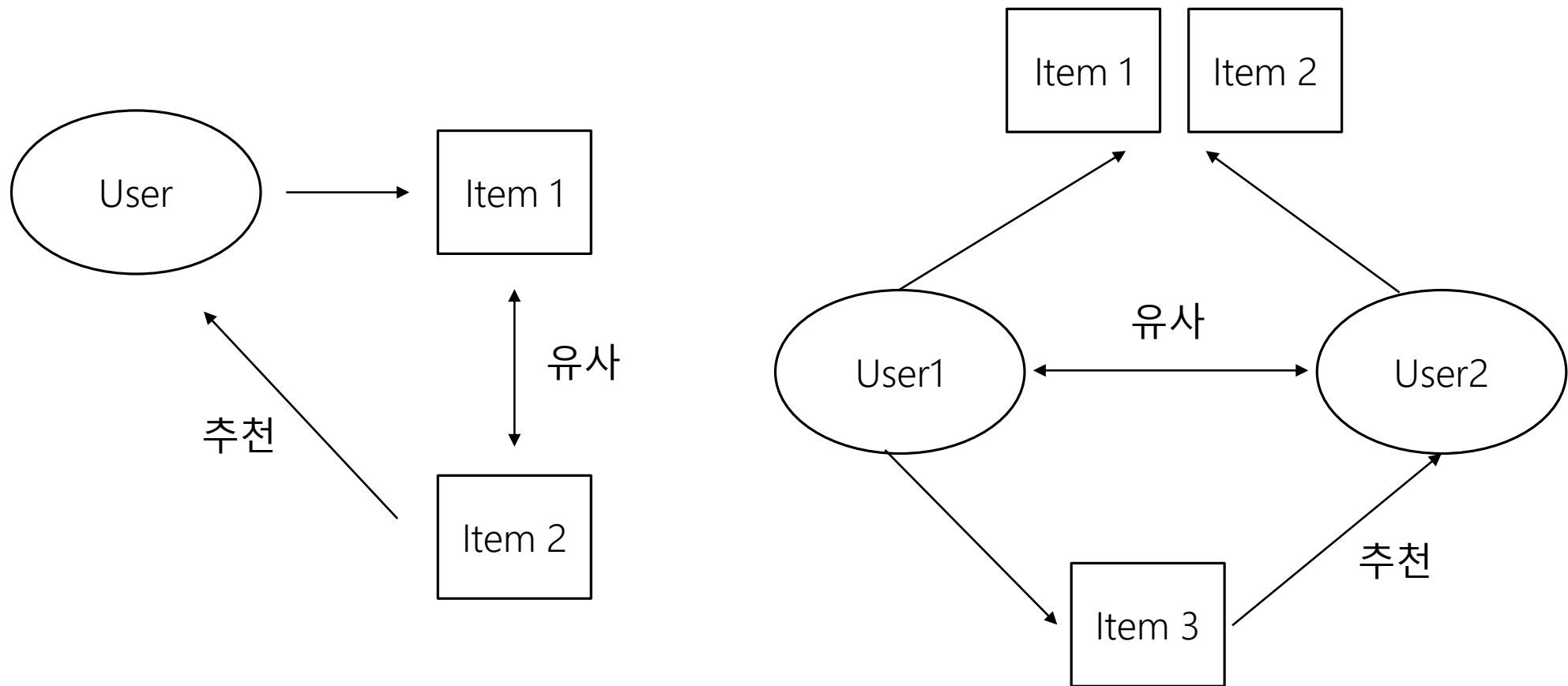
### 사이트 전반 기능 구현

- 영화 디테일페이지
- 영화 검색
- 사용자 로그인



# 진행상황

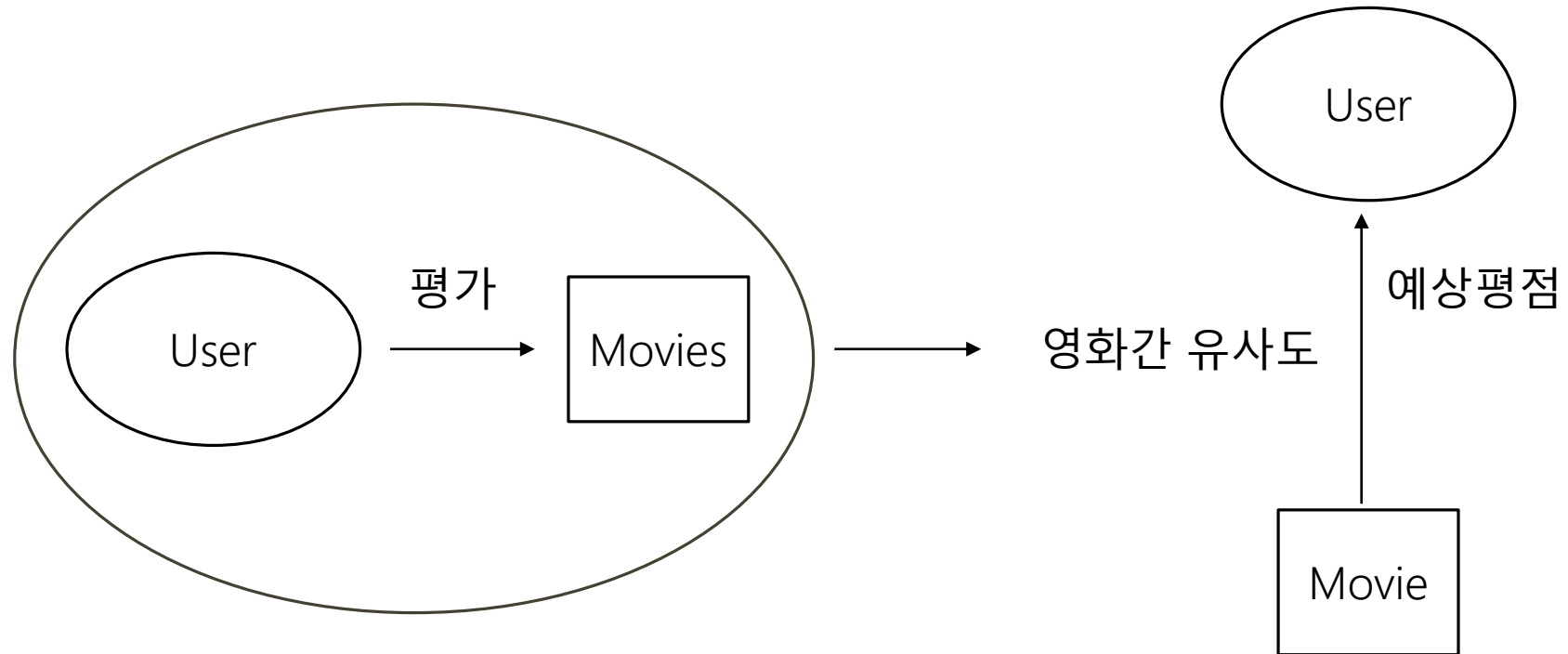
## 컨텐츠 기반 필터링 vs 협업 필터링





# 진행상황

## 개인화된 아이템 기반 추천



## 진행상황

## 데이터 정제

# Movielens dataset

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

[illegible]

# 진행상황

```
item_sim = cosine_similarity(ratings_matrix_T, ratings_matrix_T)
```

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*ba i
title										
'71 (2014)	1.0	0.000000	0.000000	0.0	0.0	0.0	0.000000	0.0	0.141653	
'Hellboy': The Seeds of Creation (2004)	0.0	1.000000	0.707107	0.0	0.0	0.0	0.000000	0.0	0.000000	
'Round Midnight (1986)	0.0	0.707107	1.000000	0.0	0.0	0.0	0.176777	0.0	0.000000	

사용자 - 평점 간 분포를 통해  
영화간 코사인 유사도 산출

# 진행상황

## 최근접 이웃 협업 필터링

```
def predict_rating_topsim(ratings_arr, item_sim_arr, n=30):
    pred = np.zeros(ratings_arr.shape)

    for col in range(ratings_arr.shape[1]):
        top_n_items = [np.argsort(item_sim_arr[:, col])[:-n-1:-1]]
        for row in range(ratings_arr.shape[0]):
            pred[row, col] = item_sim_arr[col, :][top_n_items].dot(ratings_arr[row, :][top_n_items].T)
            pred[row, col] /= np.sum(np.abs(item_sim_arr[col, :][top_n_items]))
    return pred
```

	pred_score
title	
Twelve Monkeys (a.k.a. 12 Monkeys) (1995)	3.949761
Aliens (1986)	3.822976
Alien³ (a.k.a. Alien 3) (1992)	3.813187
Say Anything... (1989)	3.705513
Sixteen Candles (1984)	3.576744
Terminator 2: Judgment Day (1991)	3.570628
Breakfast Club, The (1985)	3.532973
Die Hard (1988)	3.522424
Stand by Me (1986)	3.424591
'burbs, The (1989)	3.404946

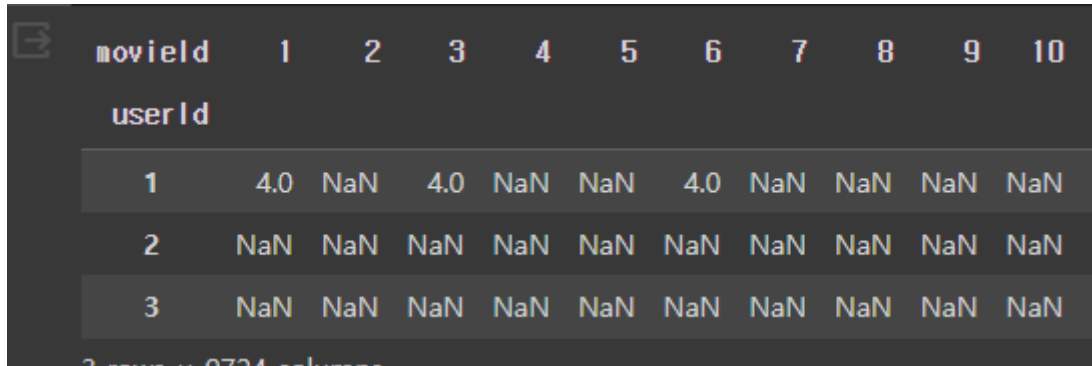
산출해낸 영화간 유사도 기반으로  
개인화된 평점 학습

해당 영화의 Top n의 유사도를 가진  
영화에 대하여 유사도벡터와 실제평점  
벡터의 코사인유사도계산.  
이때, n은 최근접 이웃 범위 계수

사용자가 평점을 부여한 영화에  
대해서만 MSE를 구하고 최소화

# 진행상황

## Validation dataset의 부재



movieId	1	2	3	4	5	6	7	8	9	10
userId										
1	4.0	NaN	4.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

User는 대부분의 영화에 평점을 매기지 않음.  
따라서 진행했던 학습에서는 Validation dataset 을  
분리하더라도 Validation이 불가능

이로 인해 위 모델이 잘 작동하는지 확인하기가  
어렵다.

다른 방식의 모델, 데이터셋, 혹은 Validation 필요

# 예상결과

- 여러가지 학습을 통해 가장 정확한 모델을 작성 한다.
- 파이썬으로 작성된 모델을 Flask와 같은 웹서버 프레임워크를 통하여 api로 이용한다.
- 작성된 api를 통하여 영화추천 페이지와 Detail 페이지에 예상평점을 출력한다.

# 향후 계획

- 10월 3째주 – 모델 완성
- 10월 4째주 – 웹 서버 프레임워크를 통하여 api로 제공
- 11월 – 완성된 api를 통하여 웹사이트 완성



**감사합니다**

---