영화 추천 시스템 플랫폼 구축

3조

Bit academy Big data Marketing Team

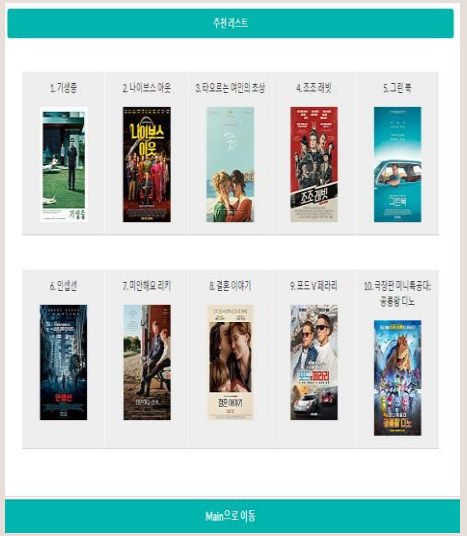
민 병혁 최 현수
권 태양

목차

1. 기획 단계

+

=



네이버 영화

네티즌 평점

네이버 영화

현재 상 영작

영화 추천 시스템

네티즌 평점에 의한 현재 상영 작 사용자 별 추천 시스템 구현

Bit Academy Big data Marketing Team

2. SYSTEM구성

Web Crawling , Analysis

Operation Data 적재

Web Service View

작업 별 조원 간 역할 분담제 진행

Bit Academy Big data Marketing Team

3. 구현 과정 (1) 데이터 수집 단계


```
#  영화 전체 리스트
from urllib.request import urlopen
from bs4 import BeautifulSoup

Main_url = "https://movie.naver.com/movie/running/current.nhn"
html = urlopen(Main_url)
main = BeautifulSoup(html, "html.parser", from_encoding="utf-8")
```

16,675,535	부산행	8	edwa****
16,675,537	붉은 돼지	10	fbtj****
16,675,538	남산의 부장들	10	p273****
16,675,539	월요일이 사라졌다	10	wlgn****
16,675,540	양자물리학	10	curb****
16,675,541	아포칼립토	10	kimk****
16,675,542	아포칼립토	10	koop****
16,675,543	남산의 부장들	9	hyuk****

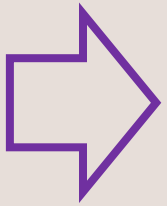
영화 제목 ,평점 , 사용자 이름에 대한 Crawling Code 작성

댓글 등록 순서를 Primary Key로 설정 후 영화, 평점, 유저 순으로 DB 적재

3. 구현 과정 (2) 데이터 전처리

	0	1	2
0	미운 오리 새끼	10	pig*****
1	부산행	8	edw*****
2	붉은 돼지	10	fbt*****
3	남산의 부장들	10	p27*****
4	월요일이 사라졌다	10	wlg*****
...	...	...	...
22732	터미네이터: 다크 페이트	6	sma*****
22733	조조 래빗	10	ytm*****
22734	기생충	9	017*****
22735	인셉션	10	who*****
22736	카센타	8	hia*****

22737 rows × 3 columns

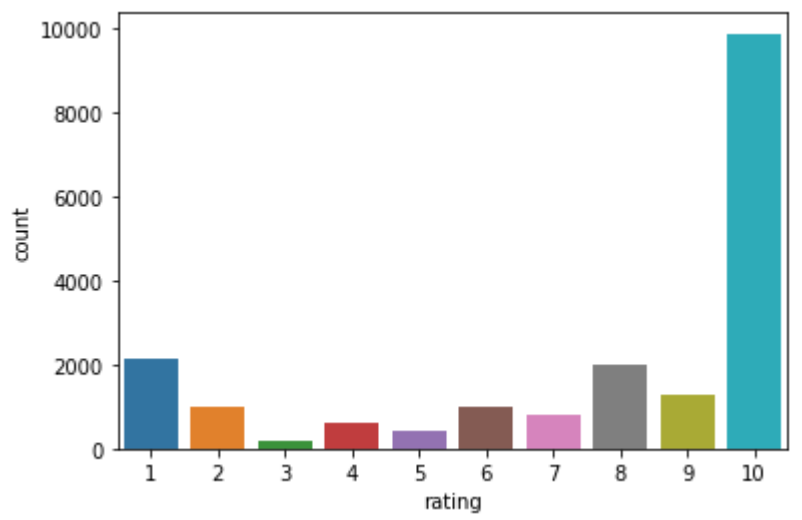


	userId	movieId	rating
0	pig	미운 오리 새끼	10
1	edw	부산행	8
2	fbt	붉은 돼지	10
3	p27	남산의 부장들	10
4	wlg	월요일이 사라졌다	10
...	...	...	...
22732	sma	터미네이터: 다크 페이트	6
22733	ytm	조조 래빗	10
22734	017	기생충	9
22735	who	인셉션	10
22736	hia	카센타	8

19221 rows × 3 columns

사용자 명이 일부 암호화되어 앞 3글자 가 동일시 한 사용자로 칭함.
이때 중복 값이 생기므로 처음 값을 제외한 나머지 값 제거.
Ex) pig*** / pig**** => pig

3. 구현 과정 (3) 데이터 탐색



- 주로 10점을 준 사용자가 대부분
- 다음으로 1점을 준 사용자가 많다.

```
len(df.userId.unique())
```

```
6224
```

사용자의 수 : 6224

```
len(df.movieId.unique())
```

```
2608
```

영화 수 : 2608

```
len(df) / (len(df.userId.unique()) * len(df.movieId.unique()))
```

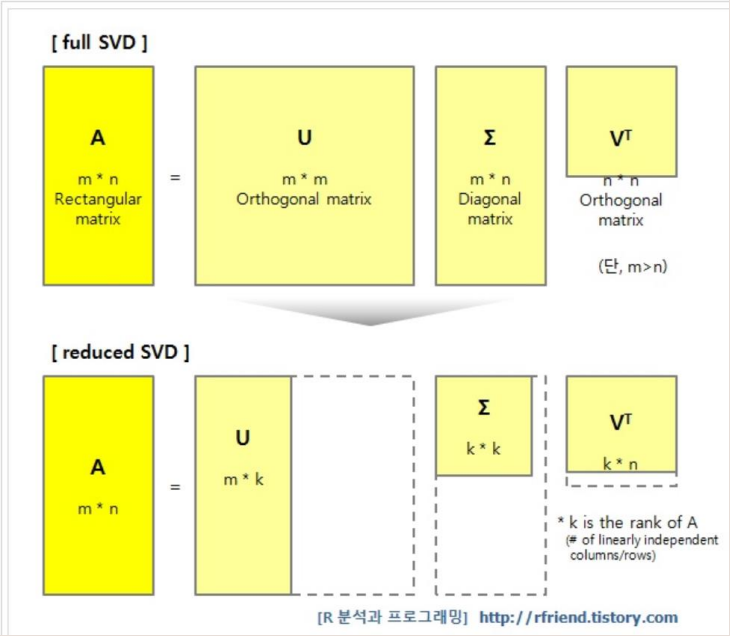
```
0.0011841284282492468
```

데이터 비율 : 데이터 수 / (유저 수 * 영화 수)

- 0.001 이므로 희소행렬임을 알 수 있다.

3. 구현 과정 (4) 분석 - 모델 선택

1) SVD
- 특이 값 분해 알고리즘
- 희소 행렬에 적합
- 사용자, 영화의 중요 특징을 이용해 예측하는 알고리즘



2) Baseline Only
- User와 Item의 Baseline을 이용한 평점 예측 알고리즘

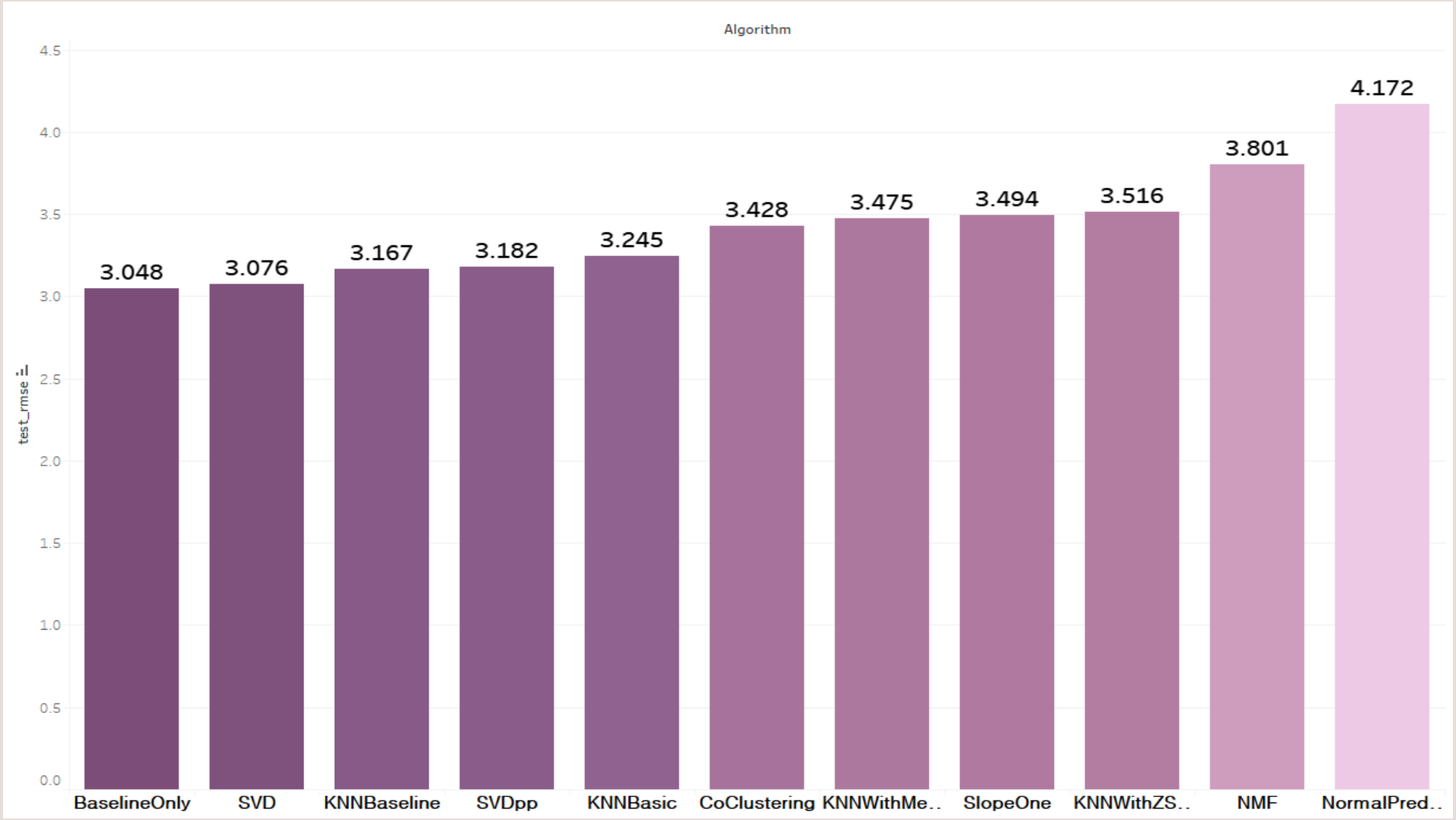
$$\hat{r}_{ui} = \mu + b_u + b_i$$

사용자 u가 I 라는 영화의 예측 값
= 평점 데이터 전체 평균
+ u라는 유저의 편향
+ i라는 영화의 편향

3) KNN Basic
- 사용자와 유사한 k명의 평점 데이터를 이용한 평점 예측 알고리즘

1. 사용자와 유사한 k명을 찾는다(사용자별 유사도 비교 – default : msd)
2. K명의 유사 사용자들의 영화 평점의 가중합을 하고 정규화(유사도가 높을수록 가중치 증가)

3. 구현 과정 (4) 분석 - 모델 성능 평가



- 모델성능평가 : RMSE

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Sum((예측값 – 실제값)^ 2)
의 값을 데이터 수로 나누고
루트를 씌운 값
:평균 오차에 대한 추정 값

RMSE를 기준으로 값이 낮은
두개의 모델 선택

- BaselineOnly
- SVD

3. 구현 과정 (4) 분석 - 모델 별 최적 파라미터 찾기

- BaselineOnly Default model RMSE : 3.048

```
# BaselineOnly 최적 파라미터 찾기
# defaults : als, 10, 15, 10
BaselineOnly_param_grid = {'bsl_options': {'method': ['als'],
                                           'n_epochs': [5,7,10],
                                           'reg_u' : [15,20,30,40],
                                           'reg_i': [5,7,10]
                                          }}
```

After

```
print(baselineonly.best_score['rmse'])
print(baselineonly.best_params['rmse'])
```

```
3.041295344607057
{'bsl_options': {'method': 'als', 'n_epochs': 5, 'reg_u': 15, 'reg_i': 5}}
```

- SVD Default model RMSE : 3.076

```
# SVD 최적 파라미터 찾기
# defaults : 20, 100, 0.005, 0.02
svd_param_grid = {'n_epochs': [10, 20, 30],
                  'n_factors' : [80,100,120],
                  'lr_all': [0.002, 0.005, 0.007],
                  'reg_all': [0.02, 0.05, 0.1, 0.15]}
```

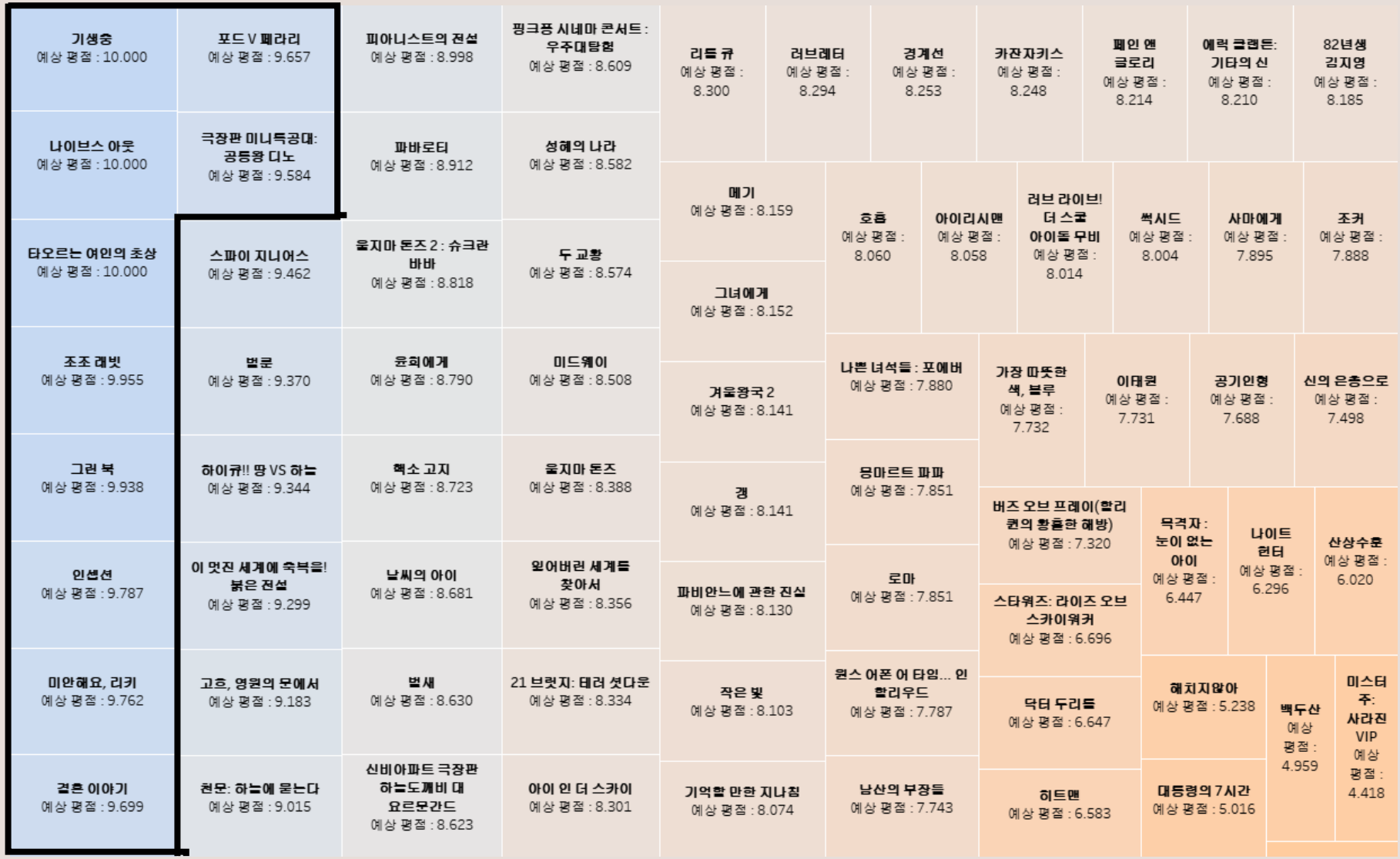
After

```
print(svd.best_score['rmse'])
print(svd.best_params['rmse'])
```

```
3.0691171868356366
{'n_epochs': 20, 'n_factors': 80, 'lr_all': 0.005, 'reg_all': 0.1}
```

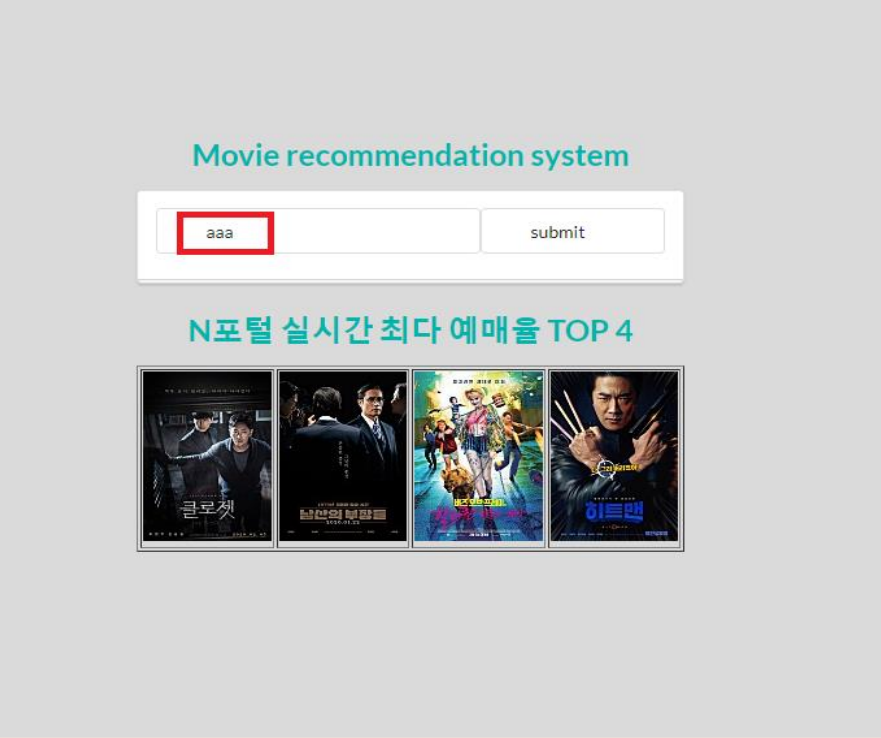
=> 최종 모델은 RMSE가 가장 낮은 값 3.041인 Baseline Only 선택(파라미터는 위와 동일)

3. 구현 과정 (4) 분석 - 모델 학습 및 산출 결과

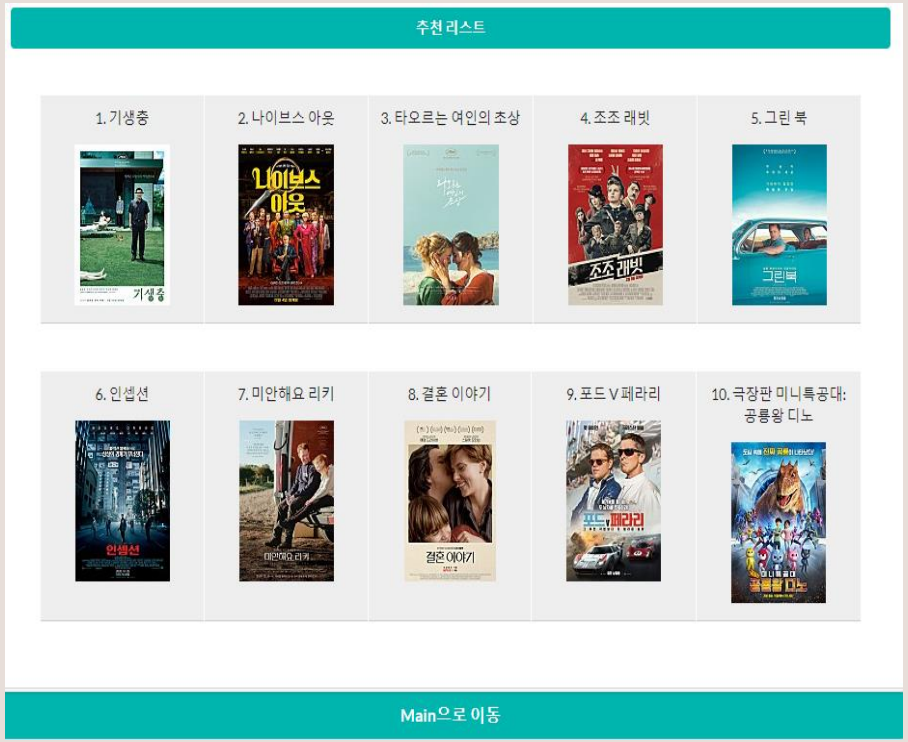


- 모델 학습 후 현재 상영 작들에 대한 예측 평점(사용자 : 'aaa')
- 가장 높은 평점인 왼쪽 위에서부터 가장 낮은 평점인 오른쪽 아래까지 내림차순 정렬
- 왼쪽 위에서부터 상위 10개를 추천

3. 구현 과정 (5) Web 화면 출력



Main 화면(사용자 입력 화면)



사용자 'aaa'를 위한 현재 상영작 10개 추천

감사합니다.

코드 참고

https://github.com/sunnight9507/Bit_Academy/tree/master/12%EC%A3%BC%EC%B0%A8/mini_project

Bit Academy Big data Marketing Team