2021년 혁신성장 청년인재 집중양성 사업
인공지능 개발자 과정

사용자 평점을 이용한 영화 추천시스템 모형 개발

2조

김소희 · 김영주 · 유수진 · 유용빈

통계, 기계, 전자, 정보통신

목차

제1장 서론

제1절 프로젝트 개요

프로젝트 내용

주제	영화 추천시스템 모형개발
이유	추천시스템은 4차 산업 시대에서 필수적인 요소이다. 이미지 또는 영상과 같은 콘텐츠 추천, 쇼핑, 광고노출, 검색어노출 등 추천이 들어가지 않는 곳이 없다. 특히, 유튜브의 '알 수 없는 추천 알고리즘'은 나 자신보다 추천 알고리즘이 내 취향을 잘 안다는 느낌을 준다. 대체 어떠한 원리로 작동하는지 직접 추천시스템 모형을 개발하면서 몸소 느끼고자 한다.
목적	⍁ 탐색적 분석을 통하여 수집된 영화 데이터 이해 ⍁ 평가 지표 결과 정확도가 높은 추천 알고리즘 선정 ⍁ 사용자 맞춤 영화추천 및 평가
기간	2021.05.10 ~ 2021.06.07 (총29일)
분석범위	협업필터링과 하이브리드 알고리즘 중심으로 분석 수행
분석방법	recommenderlab 패키지 활용

프로젝트 진행계획

세 부 항 목	10	…	14	15	…	21	…	25	26	…	31	1	2	…	7
주제 선정															
기획서 제출															
자료수집/사전공부															
전처리/분석															
중간보고서 제출															
모델 생성/평가															
추천시스템 구현															
완료보고서 제출															

⿻ 개발환경 및 적용기술

◪ 프로젝트 관리 ⇨ Google Drive

◪ 코드 관리 ⇨ Google Drive

◪ 진행 장소 ⇨ 교육장, Zoom

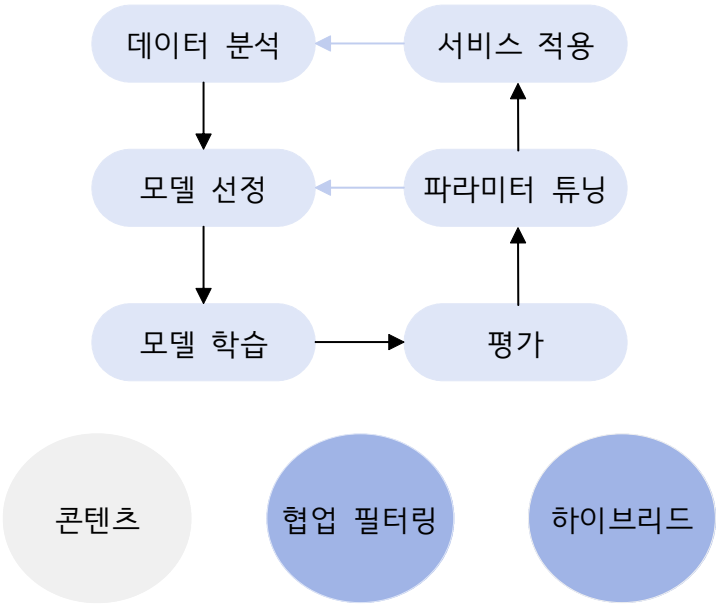
구분	소프트웨어		버전
Language	R		4.0.5
IDE	R-studio		1.4.1106
주요 Library	시각화	ggplot2	3.3.3
		wordcloud	2.6
	분석	recommenderlab	0.2-7
Web App	Shiny		1.6.0

제2절 추천시스템의 이해

개요

추천 시스템은 사용자(user)에게 상품(item)을 제안하는 소프트웨어 도구 이자 기술이다. 이러한 제안은 어떤 상품을 구매할지, 어떤 음악을 들을지 또는 어떤 온라인 뉴스를 읽을지와 같은 다양한 의사결정과 연관이 있다[1]. 다시말해, 대량의 정보를 이용하여 개인화된 정보 필터링 기술로, 특정사용자가 특정항목을 좋아할 것인지에 대해 예측하거나 특정사용자가 흥미있어할만한 N개 항목의 집합을 찾아내는데 사용된다[1].

추천시스템 프로세스



추천 모델 종류와 특징

Question. 유저가 좋아할 만한 아이템은 뭐지?
Answer. 추천 모델마다 정의가 다르다

추천시스템은 추천하고자 하는 목적에 따라 콘텐츠기반, 협업필터링 그리고 이 두 가지 방식을 조합한 하이브리드 방식이 있다[2]. 본 프로젝트에서는 협업필터링과 하이브리드 방식을 사용한다. 각각의 특징은 부록1을 참고.

사용된 추천 모델 종류

따라서 본 프로젝트에서 사용하는 추천 모델은 다음과 같다. 이웃 기반에서 사용자기반(UBCF)과 아이템 기반(IBCF)을, 모델 기반에서 특이값 분해(SVD)를, 하이브리드 방식은 Cascade 방식을 사용하였다.

구분	종류
협업필터링: 이웃기반	**UBCF, IBCF**
협업필터링: 모델기반	**SVD**
하이브리드	**Cascade**

추천 모델 평가 방법

정확도 평가지표	실서비스 만족도 평가지표
Precision/Recall	유저별 클릭수/CTR
F-score/AUROC	유저별 총 체류시간
nDCG/MRR	신규아이템 회전율

본 프로젝트에서는 실서비스 만족도는 평가할 수 없으므로 추천모델 정확도 평가지표를 통해 추천 모델을 평가하고자 한다.

평가 방법은 크게 데이터의 유형과 평가 목적에 따라서 나눌 수 있다. 데이터 유형이 연속형 데이터인 경우에는 예측 정확도로 평가하며, 범주형 데이터인 경우 분류 정확도로 평가한다. 추천 정확도는 아이템의 점수를 예측하는 알고리즘이 사용되고 실제 선호도와 예측 값의 차이로 계산하며 널리 사용되는 방법으로 Root Mean Squared Error(RMSE), Mean Average Error(MAE) 등이 있다. 분류 정확도는 선호도가 높을 것이라 예측한 상위 N개의 아이템에 대해 추천 성능을 평가할 때 사용되며 대표적으로 Precision, Recall, F1기법, Receiver operating characteristic(ROC) 등이 있다. 각각의 특징은 부록2을 참고.

제3절 국내·외 기술개발 현황

영화 대여 서비스인 넷플릭스의 경우, 대여되는 영화의 75%가 추천에 의해 이루어지고 있다.

인터넷 서점인 아마존에서는 매출의 35%가 추천에 의해 이루어지고 있다.

이용자들의 시청 시간 70%가 추천 알고리즘에 의한 결과이다.

아마존 등과 같은 전자상거래 업체, 유튜브, 애플 뮤직 등 콘텐츠 포털까지 추천 시스템을 통해 사용자의 취향을 이해하고 맞춤 상품과 콘텐츠를 제공해 조금이라도 오래 자기 사이트에 고객을 머무르게 하기 위해 전력을 기울이고 있다[6].

제2장 본론: 활용데이터

제1절 자료 소개

데이터는 movielense에서 제공하는 데이터를 활용하였다. 데이터는 영화의 장르와 관련된 정보, 평가 정보, 사용자가 남긴 태그 정보가 있다. 영화 데이터는 9742개 영화ID를 가지고 각 영화제목, 장르(특수문자 'I' 가 포함된 문자열)가 포함되어 있다. 평점 데이터는 100,836개 평가정보를 가지고 평가한 유저ID와 영화ID가 포함되어 있다. 태그 데이터는 3683개의 태그정보를 가지고 기록한 유저ID와 영화ID가 포함되어 있다. 본 연구는 협업필터링 중심의 모델을 만들 것이기 때문에 사용자의 평점 정보와 영화, 장르 데이터를 기반으로 모델링 할 것이다. 원본데이터는 다음과 같다.

영화데이터

	movieId	title	year	genres
1	1	Toy Story	1995	Adventure| Animation| Children| Comedy| Fantasy
2	2	Jumanji	1995	Adventure| Children| Fantasy
3	3	Grumpier Old Men	1995	Comedy| Romance
4	4	Waiting to Exhale	1995	Comedy| Drama| Romance
5	5	Father of the Bride Part II	1995	Comedy
6	6	Heat	1995	Action| Crime| Thriller

평점데이터

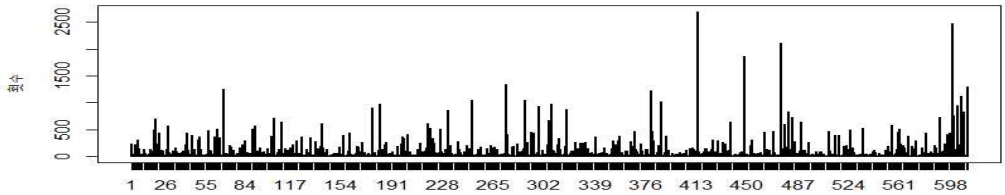
	userId	movieId	rating	timestamp
1	1	1	4	964982703
2	1	3	4	964981247
3	1	6	4	964982224
4	1	47	5	964983815
5	1	50	5	964982931
6	1	70	3	964982400

태그데이터

	userId	movieId	tag	timestamp
1	2	60756	funny	1445714994
2	2	60756	Highly quotable	1445714996
3	2	60756	will ferrell	1445714992
4	2	89774	Boxing story	1445715207
5	2	89774	MMA	1445715200
6	2	89774	Tom Hardy	1445715205

제2절 자료 탐색 및 전처리

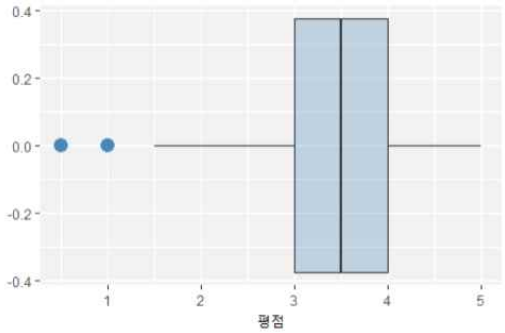
⧉ 자료 탐색: 평점



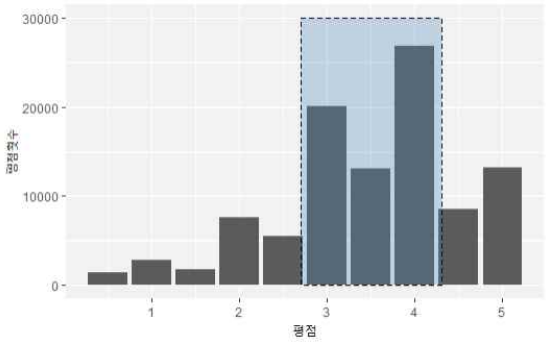
[그림 1] 유저별 평가 횟수

min(table(r$userId)) ⇨ 가장 적은 평가 횟수는 20회
max(table(r$userId)) ⇨ 가장 많은 평가 횟수는 2698회

평가를 많이한 414번(2698회), 448번(1846회), 474번(2108회), 599번(2478회) 사용자들에게 영향을 많이 받는 결과가 초래될 것으로 예상된다.

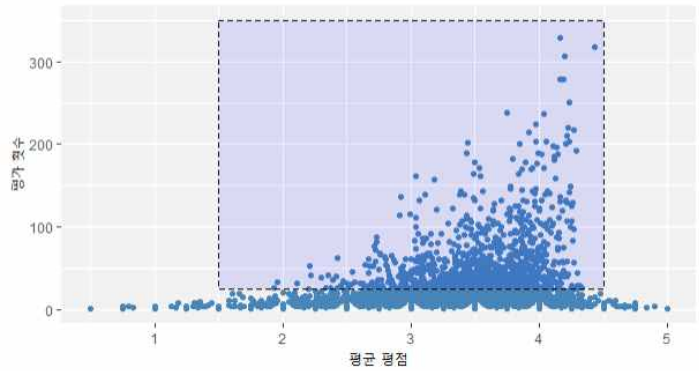


[그림 2] 평점 Box plot



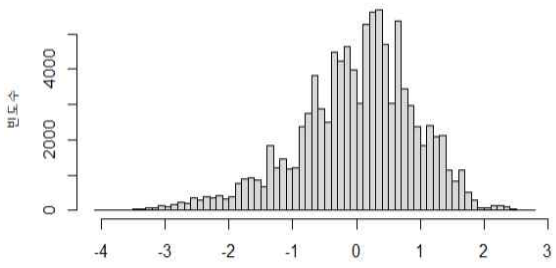
[그림 3] 유저 평점 분포도

평점의 분포를 확인한 결과, 3~4점 대의 분포가 가장 많았다. 대부분 3~4점 대에 몰려있기 때문에 0~1점대의 데이터의 이상치를 확인하였다.

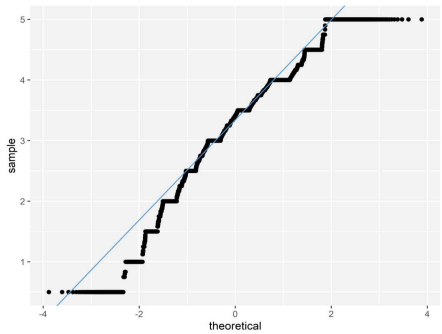


[그림 4] 영화 분포도

각 유저의 영화별 평균 평점 결과를 시각화한 그림이다. 이를 활용하여 표본(평가 수)이 적은 데이터를 제거하는 것에 활용한다. 일정 n이하의 데이터는 정확한 추천을 해주는 것에 방해 요인이 될 수 있기 때문이다.



[그림 5] 평점 정규화



[그림 6] 평점 q-q plot

평점에 대한 정규성을 확인하는 이유는 데이터가 한쪽으로 치우쳐진 데이터 분석시, 편향적인 결과가 나올 수 있기 때문이다. 평점 정규화의 목적은 데이터의 값이 너무 크거나 혹은 작은 경우에 모델 학습 과정에서 0으로 수렴하거나 무한으로 발산해버릴 수 있기 때문에 이를 해결하기 위함이다.

자료 탐색: 장르

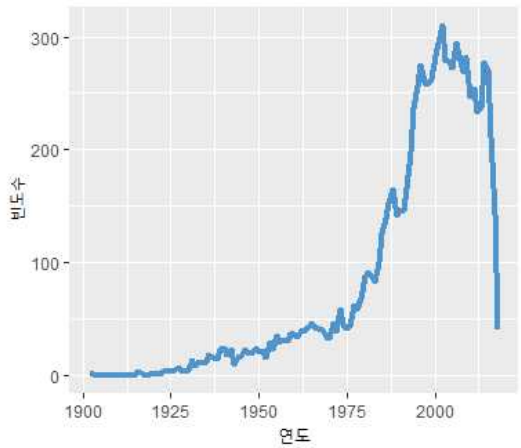
연도별 출시 장르 분포 확인을 통해 대중적인 영화의 분포와 평점이 특정 연도에 몰려있지 않은지 확인한다.

영화장르	장르빈도수	장르비율	누적비율
Drama	4361	19.7473%	19.75%
Comedy	3756	17.0078%	37.76%
Thriller	1894	8.5763%	45.33%
Action	1828	8.2775%	53.61%
Romance	1596	7.2270%	60.84%

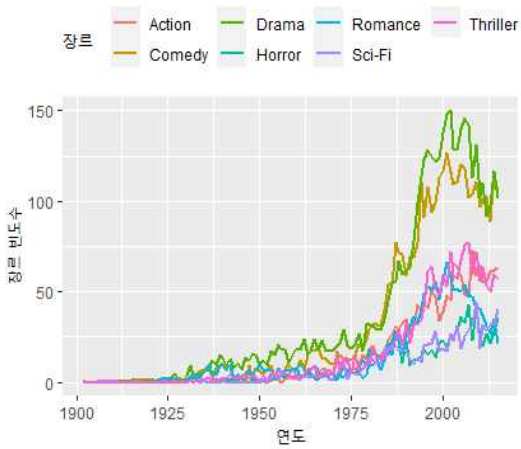
영화장르별 장르빈도수를 확인해본 결과 Drama, Comedy, Thriller, Action, Romance 순으로 많았다. 특히 Drama와 Comedy는 각각 비율이 19.74%, 17%이고 두 장르의 누적 비율이 45%에 달하여 상대적으로 다른 장르에 비해 많은 비율을 차지하고 있다.

연대	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000	2010
편수	10	33	136	197	278	391	499	1176	2207	2849	1915

수집된 movielens 데이터는 1902년부터 2018년까지의 영화데이터이다. 각 연대별 분포를 보면 매년 증가함에 따라 매년 영화의 수가 증가하고 있고 1970년대 이후로 영화 개봉수가 급격하게 증가한다. 2002년에 영화 개봉 수가 311편으로 가장 많았다.



[그림 7] 연도별 장르 빈도수

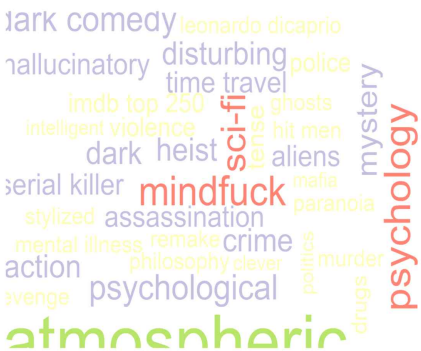


[그림 8] 연도별 장르 빈도수

1902년부터 2018년까지의 영화 데이터이다. 매년 영화의 수가 증가하고 있고 1975년 이후로 영화 개봉 빈도수가 급격하게 증가하고 있다. 특히, 2000년대에 진입하면서 영화 개봉 수가 가장 많았으며 2015년 이후로는 급격히 줄었다.

자료 탐색: 태그

[그림 9] 전체 태그 빈도



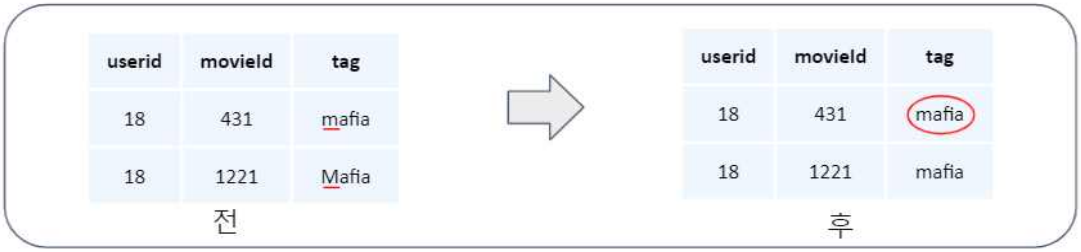
[그림 10] 장르별(Thriller) 태그 빈도

[그림 9]는 610명의 유저가 남긴 태그 내용을 텍스트마이닝을 통해 살펴보았다. atmospheric (분위기), thought-provoking(생각을 자극하는), superhero(슈퍼 히어로), funny등 앞서 살펴본 장르 비율에서 빈도가 높았던 장르와 유사하다.

[그림 10]은 장르빈도가 높았던 Thriller 장르의 영화들 태그빈도 내용이다. 전반적으로 태그 내용들이 전부 Thriller 라는 하나의 키워드로 함축될 수 있다고 해석이 되어 장르기반 추천모델과 태그기반 추천 모델의 차이점을 얻지 못 할 것이라 판단하였다. 따라서 본 프로젝트에서는 평점데이터와 장르데이터가 들어있는 movie, rating 데이터 셋을 사용하였다.

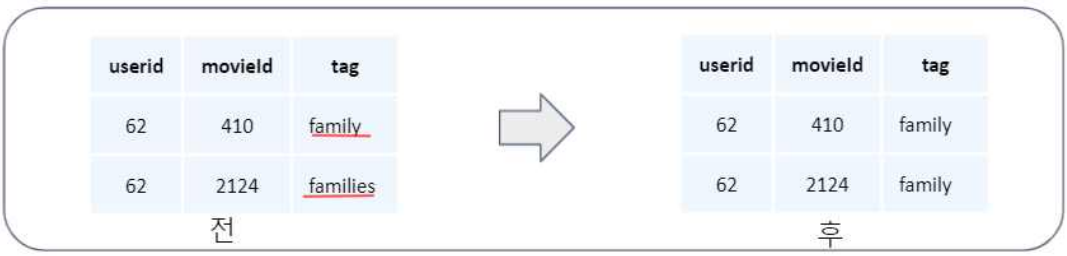
자료 전처리

➤ 현상: 영화의 태그가 대소문자 구분되어 중복 출력
➤ 방법: 모두 소문자로 변환



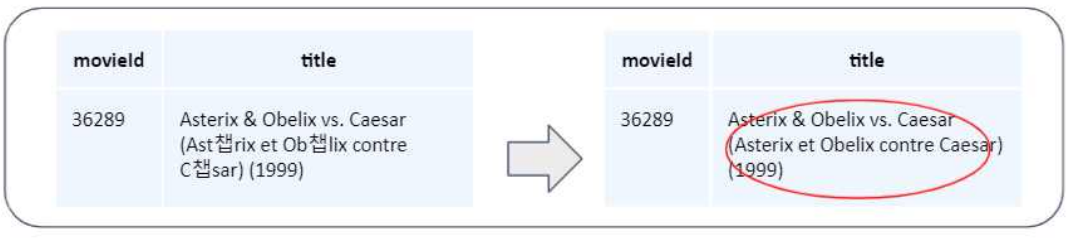
대소문자 구분 전처리

➤ 현상: 동일 태그에서 복수형, 단수형 존재
➤ 방법: 해당 행번호를 찾아 수정



명사 수일치

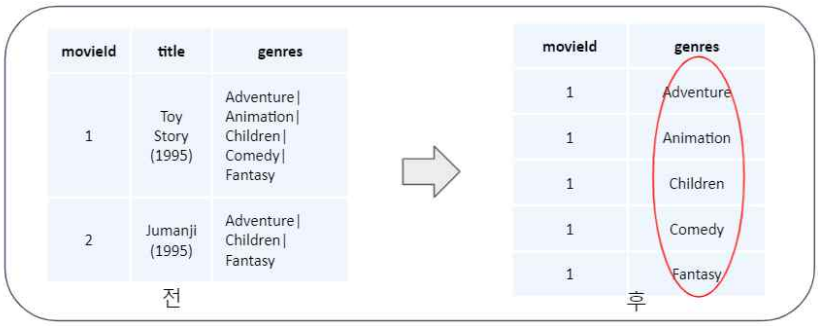
➤ 현상: 영화의 제목이 로마자, 특수문자로 출력
➤ 방법: 해당 행번호를 찾아 수정



특수문자, 로마자 전처리

➤ 현상: 장르가 분리되어 있지 않음

➤ 방법: movieId 기준으로 장르를 나열함



장르 벡터 분리 생성

➤ 모델링을 위해 원데이터를 2차원 Matrix 행렬 구조[1]로 데이터 병합

평점 기반 2차원 행렬구조로 변환하기 위한 데이터 셋

	userid	movieid	rating
1	1	1	4.5
2	1	4	3.0
3	1	5	1.0
4	1	10	3.0

	id	title	genres
1	1	Toy story	Adventure
2	2	Jumanji	Adventure
3	3	Grumpier Old man	Comedy

평점기반 2차원 행렬구조 변환 [행: userId, 열: title]

	Toy story	Jumanji	Grumpier Old man	Waiting to exhale	Father for the Bride
1	4.5	NA	NA	3.0	1.0
2	NA	5.0	NA	4.0	NA
3	NA	NA	3.5	3.0	NA
4	4.0	NA	NA	3.0	NA
5	NA	NA	3.5	3.5	NA

[1] Recommender 라이브러리에서 제공하는 realRatingMatrix의 구조를 따르기 위해 아래와 같이 변경이 필요하다.

⿻ 최종 데이터

(1) 사용자-평점 매트릭스

userId	X-man	Jumanju	Grumpier Old man	Waiting to exhale	Father for the Bride
1	5	0	0	3	1
2	0	5	0	4	0
3	0	0	3.5	3	0
4	0	0	0	3	0
5	0	0	3.5	3.5	0

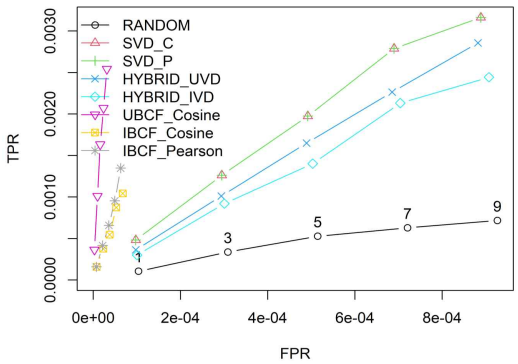
(2) 사용자-장르 매트릭스

userId	Action	Comedy	Drama	Fantasy	Horror
1	4.32	4.27	4.52	4.30	3.47
2	3.95	4.00	3.88	0.00	3.00
3	3.57	1.00	0.75	3.38	4.68
4	3.32	3.51	3.48	3.68	4.25
5	3.11	3.47	3.80	4.14	3.00

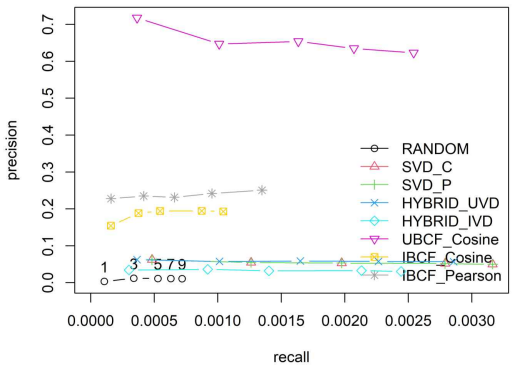
제3장 본론: 분석방법

제1절 알고리즘에 따른 모델 결과:
사용자-영화평점 데이터 & 사용자-장르평점 데이터

9742편의 영화들에 대해 610명의 사용자가 부여한 평가치 매트릭스를 준비한다. 610명의 고객들로부터 무작위로 80%인 488명의 사용자를 추출하여 학습데이터집합(Training Data Set)으로 사용하고, 20%인 122명의 사용자를 평가 데이터 집합(Test Data Set)으로 사용하였다.

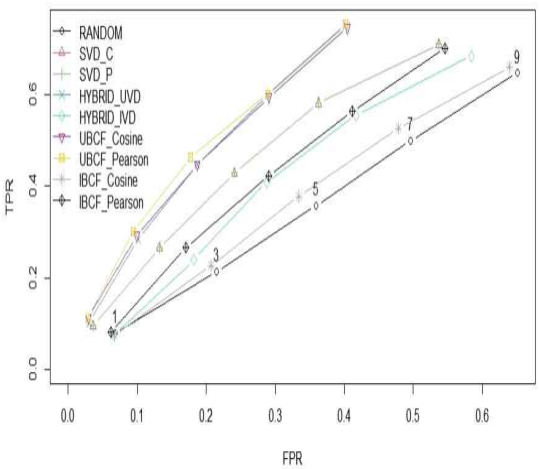


[사용자-영화평점] 알고리즘 간
ROC Curve

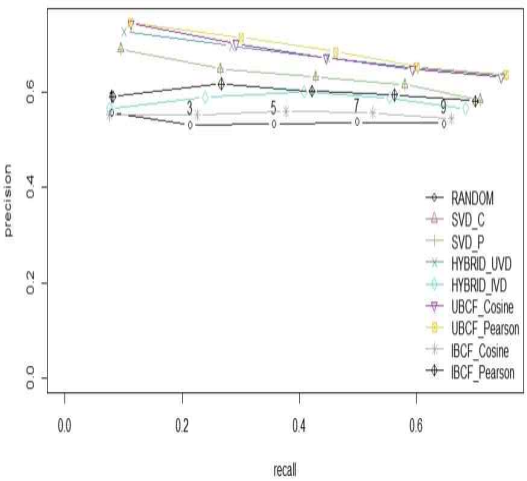


[사용자-영화평점] 알고리즘 간
Precision-Recall Curve

사용자-영화평점데이터에서 UBCF-Cosine 알고리즘의 ROC 곡선과 재현율 그래프를 통해 성능이 좋을 것이라 예측한다.



[사용자-장르평점] 알고리즘 별
ROC Curve



[사용자-장르평점] 알고리즘 별
Precision-Recall Curve

사용자-장르평점데이터에서 UBCF-Pearson 알고리즘의 ROC 곡선과 재현율그래프를 통해 성능이 좋을 것이라 예측한다.

F1 성능지표 비교

	평점	**장르**
SVD_C	0.004	0.458
SVD_P	0.004	0.458
HYBRID_IVD	0.003	0.486
HYBRID_UVD	0.003	0.430
UBCF_C	0.003	0.489
UBCF_P	-	0.497
IBCF_C	0.001	0.408
IBCF_P	0.001	0.445
RANDOM	0.001	0.395

F1 성능지표에서는 평점기반에서는 SVD가 평균 0.004로 상대적으로 높다. 장르기반에서는 UBCF_pearson이 평균 0.49로 상대적으로 0.01만큼 향상된 것을 확인할 수 있었다.

제2절 모델 성능 평가

◱ 사용자-영화평점 데이터 (예측: SVD-Cosine 성능우수)

알고리즘	RMSE	MSE	MAE
RANDOM	0.8508082	0.7859496	0.4829520
SVD	**0.4016936**	**0.2264247**	**0.1020443**
UBCF	0.8468345	0.7775405	0.6850284
IBCF	1.0924936	1.3545703	0.4066932

◱ 사용자-장르평점 데이터 (예측: UBCF-Pearson 성능우수)

알고리즘	RMSE	MSE	MAE
RANDOM	1.857840	3.745955	1.400783
SVD	1.560532	2.683299	1.175607
UBCF	**1.2039050**	**1.6453536**	**0.9078183**
IBCF	1.604917	2.807738	1.190940

RMSE는 오차의 양을 나타냄으로 낮을수록 예측의 정확도가 높다는 것을 의미한다. 사용자-평점 기반에서는 SVD-cosine Model의 RMSE값이 가장 작은 것을 확인할 수 있다. 즉, 본 프로젝트의 데이터에서는 Neighborhood Method인 코사인보다 Matrix factorization의 일종인 특이값 분해를 적용한 모델의 RMSE가 낮은 것을 확인할 수 있다.

사용자-장르 기반에서는 UBCF-pearson Model의 RMSE값이 가장 작았다. 예상외로 장르에 대해서는 SVD보다 사용자 기반 협업 필터링 방식의 추천성능이 더 높은 것을 확인할 수 있다.

제4장 결론

제1절 최종 모델

최종 모델

모델의 성능 평가결과 영화-평점, 장르-평점에서 성능이 좋은 모델이 다르게 도출됨을 확인하였다. 계획한 평가 기법에 따라 성능 지표인 ROC Curve, RMSE, MSE, MAE가 도출된다. 본 연구의 최종 결과 영화-평점은 SVD Model, 장르-평점은 UBCF Model이 성능 비교결과 가장 우수한 것으로 나타났다. 아래는 본 연구에서 모델링에 사용된 Parameter를 나타낸다.

Parameter	Value
유사도 행렬 크기	30
정규화 방법	Center

SVD: 영화-평점의 Model Parameter

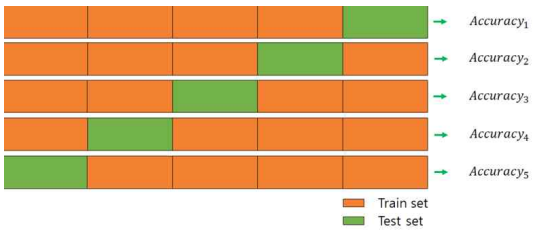
Parameter	Value
유사도 측정 방법	Pearson
유사도 행렬 크기	30
정규화 방법	Center

UBCF: 장르-평점의 Model Parameter

다음은 모델의 성능을 평가한 방법에 대해 기술한다. 제안하는 추천 시스템의 모델은 Cross-validation(교차 검증)을 사용하였다. 해당 방법은 기계 학습에서 데이터의 부족으로 인한 underfitting을 방지하기 위해 사용된다. 하지만 모의 실험 횟수가 증가 할수록 훈련/평가 시간이 오래 걸리는 특징이 있다. 해당 프로젝트에서 선정한 무비렌즈의 데이터는 전체 크기가 약 3.5MB로 제공하는 교육용 자료이다.주요 변수로는 k개의 fold를 지정하는 부분으로, k개의 subset으로 나누고 k번의 평가를 실행하게된다. 본 연구에서는 Cross-validation을 채택하였으며 관련 parameter는 아래와 같다.

Parameter	Value
평가기법	Cross-validation
모의 실험 횟수(k)	30
추천 영화 개수(n)	5

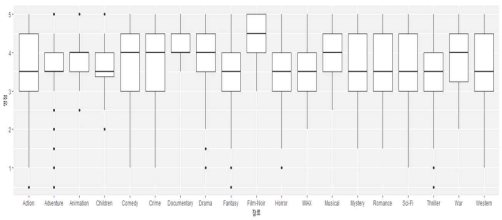
Cross-validation Parameter



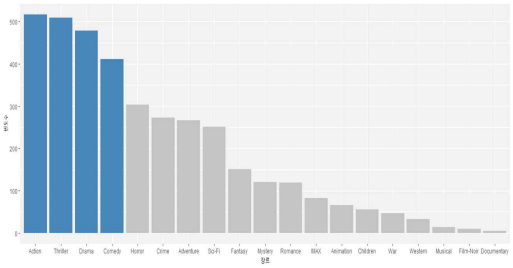
k-fold cross validation

추천 예시

최종 추천 모델이 특정 사용자에게 추천하는 영화가 얼마나 정확한지 예시를 검증하는 과정은 다음과 같다. 검증 과정으로는 전체 사용자 중 1302개의 평점을 남긴 사용자에 대하여 기초 통계 분석 후 예측 결과와 비교하였다. 평균 평점이 4점 이상인 장르는 "Film-Noir", "Documentary"이며, 시청 횟수가 300번 이상인 장르는 "Drama", "Thriller", "Action", "Comedy"이다. 610번 사용자가 시청한 영화 중 평점이 4 이상이고, 위 나열된 장르에 해당하는 영화를 기준으로 비교하여 추천 예시 결과를 확인한다.

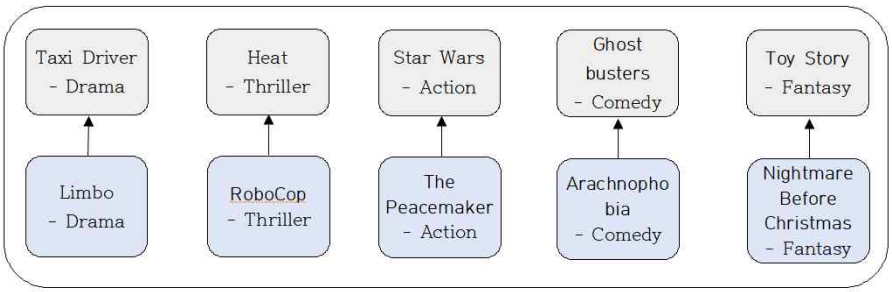


사용자 610번 시청 영화 평점의 Boxplot

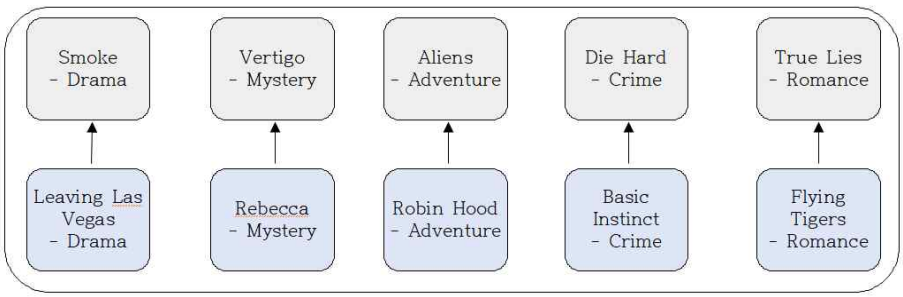


사용자 610번 시청 영화 장르의 빈도수

영화-평점 모델에서 추천한 영화 중 예측 평점이 높은 5개의 영화는 'Limbo', 'RoboCop', 'The Peacemaker', 'Arachnophobia', 'Nightmare Before Christmas'이다. 장르-평점 모델에서 추천한 장르는 "Drama", "Mystery", "Adventure", "Crime", "Romance"이며 해당 장르에 속하는 영화 중 평점이 4점 이상이면서, 최소 5사람 이상이 평점을 남긴 영화를 추천 영화로 반환한다.



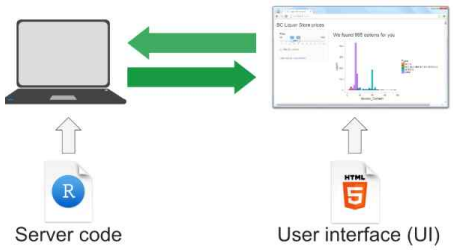
[영화 평점] 610번 시청 영화(상) <- SVD Model 추천 영화 목록(하)



[영화 평점] 610번 시청 영화(상) <- UBCF Model 추천 영화 목록(하)

추천시스템 구현

R shiny 개요



구성	기능
User Interface	Layout과 모양을 결정 Input, Output을 포함 Menu, Tab등의 내부 요소들을 정의
Server	Application Logic을 정의 Input~Output 전반적 과정을 포함
ShinyApp	UI 및 Server의 기능을 호출

R shiny는 R 사용자들에게 Interactive web app 제작을 가능하게하는 Package이다. Shiny Code로 HTML, CSS로 제작되는 웹앱을 동등하게 구현할 수 있다.

Web기반 영화 추천 시스템 배포

- 결과: 사용자에게 5개의 영화를 추천
- Model: 사용자-영화 평점기반 SVD Algorithm
- URL: bit.ly/3z8Udhi
- Source
- 참고자료: Movie Recommendation With Recommenderlab | STATWORX

구조	파일	기능
ShinyApp	app.R	UI, Server를 호출 Layout 요소를 정의
Server	data_server.R	입력된 평점들을 바탕으로 희소행렬을 생성 영화 추천 결과를 출력으로 반환
	ui_server.R	UI의 입력을 받아오는 기능을 정의
	load_data.R	movielens data set으로 생성한 SVD Algorithm Model을 정의

제2절 활용분야 및 전략제안

추천 시스템은 성능을 향상시키는 알고리즘을 찾는 것이 매우 중요하다고 판단할 수 있다.

아마존 등과 같은 전자상거래 업체, 유튜브, 애플 뮤직 등 콘텐츠 포털까지 추천 시스템을 통해 사용자의 취향을 이해하고 맞춤 상품과 콘텐츠를 제공해 조금이라도 오래 자기 사이트에 고객을 머무르게 하기 위해 전력을 기울이고 있다. [6]

사용자 측면에서는 선택적 고민을 해결해주고 만족도를 향상해 줄 것이다. 기업 측면에서는 고객을 유지, 이탈고객 감소, 수익창출의 효과가 있을 것으로 기대한다.

제3절 한계점

무비렌즈 데이터를 사용하면서 나타난 문제로는 2가지가 있었다.

1) 적은 양의 데이터로 인한 정보 부족

2) 최근 개봉된 영화의 평점 자료의 부족

협업 필터링 방식은 사용자-항목 평점 행렬의 희소성 문제를 고려할 필요가 있다. 이러한 과정에서 올 수 있는 문제점으로는 다음과 같다.

1) 적은 수의 평점을 가지는 항목에 대해 왜곡된 유사도가 계산되는 문제

2) 많은 평점을 가진 대중적인 항목에 대해서 편향적인 추천이 이루어질 수 있다는 문제

이런 문제를 해결하기 위해 데이터 탐색 부분에서 평점 분포를 살펴보고 이상치에 해당되는 평점을 제거하고 분포가 균일하도록 스케일 조정을 하였음에도 불구하고 성능이 향상되지 않았다.

참고문헌

[1] Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS), 22(1), 143-177.

[2] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3), 66-72.

[3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, (8), 30-37.

[4] Son, J., Kim, S. B., Kim, H., & Cho, S. (2015). Review and Analysis of Recommender Systems. Journal of Korean Institute of Industrial Engineers, 41(2), 185-208.

[5] 토크ON세미나, 추천시스템 분석 입문하기, Tacademy, 2021.

[6] 파이썬 머신러닝 완벽가이드, 위키북스, 권철민

부 록1

협업필터링은 방식에 따라 다시 이웃기반과 모델기반으로 나눌 수 있다. 이웃 기반은 사용자의 항목에 대한 평점을 예측하기 위해 시스템에 저장된 "사용자-아이템 간의 평점 정보"를 직접적으로 이용하며, 이는 다시 사용자기반과 아이템 기반 추천으로 나눌 수 있다. 모델 기반은 사용자-아이템 평점 정보를 이용하나 이를 직접적으로 이용하지 않고 예측 모델을 학습 시키는 데 사용한다. 행렬 인수분해를 이용한 잠재 요소 모델은 이러한 모델 기반 방식 중 하나로, 사용자와 항목들을 평점 정보로부터 잠재요소들의 벡터로 표현한다[3]. 대표적인 알고리즘은 다음과 같다.

(1) 협업필터링: 이웃기반

이웃기반	설명
사용자기반(UBCF)	**사용자기반 협력필터링은 사용자가 입력한 선호도 정보를 이용하여 사용자와 유사한 성향을 갖는 이웃 사용자를 선별한뒤, 선별된 이웃들이 공통적으로 선호하는 아이템을 사용자에게 추천해주는 방식**
아이템기반(IBCF)	**특정 아이템이 기준이 되어 사용자들에 의해 평가된 점수가 유사한 아이템을 이웃 아이템으로 선정한 다음, 이웃 아이템을 평가한 점수를 바탕으로 추천 대상 고객이 특정아이템에 대해 갖게 될 선호도를 예측하는 방식**

(2) 협업필터링: 모델기반

나이브베이즈, 군집화, 차원 축소(**특이값분해**, PCA, ALS) 등

(3) 하이브리드

콘텐츠 기반 방식과 협업 필터링 방식을 혼합한 추천 방식을 하이브리드 방식이라 한다. 하이브리드 방식은 정형화된 방식으로 있는 것이 아닌 각각이 가진 단점을 보완하고 장점을 취하는 다양한 방식이 시도되고 있다.

하이브리드 방식	설명
Weighted	여러 추천방식의 점수들이 하나의 추천을 생성하기 위해 합함
Switching	시스템이 현재 상황에 맞추어 추천 방식을 바꿈
Mixed	여러 다른 추천의 결과가 동시에 제시됨
Feature combination	여러 가지 다른 추천 데이터에서 가져온 특성들을 하나의 추천 알고리즘에 사용
Cascade	**한 추천 알고리즘이 다른 추천 알고리즘을 개선**
Feature augumentation	하나의 방식에서 나온 결과를 다른 방식의 특성으로 사용
Meta-level	하나의 추천 방식으로부터 학습된 모델이 다른 추천 방식의 입력값이 됨

부 록2

(1) 점수 예측 알고리즘의 평가방법

평가척도	설명
MSE	각각의 예측 점수와 실제 점수의 차이를 제곱한 후 이를 평균한 값
RMSE	MSE 값에 비하여 예측오차가 큰 관측치에 대해 상대적으로 적은 가중치를 부여한 값
MAE	오차의 절대값의 평균이며, 오차의 크기에 상관없이 모두 같은 가중치를 부여한 값

(2) 아이템 예측 알고리즘의 평가방법

평가척도	설명
Precision	(옳게 추천한 아이템 개수) / (추천한 전체 아이템 개수)
Recall	(옳게 추천한 아이템 개수) / (고객이 실제로 구매한 아이템 개수)
F-measure	2 / (1/Pre + 1/Rec)

추천 상품 수가 커질수록 recall 값은 증가하지만 precision 값은 감소하게 되며 이러한 상충관계까지 고려하였을 때 분류의 효율성을 평가하는 척도로 F-measure가 사용될 수 있다. F-measure 값이 '1'에 근접할수록 recall 값과 precision 값 모두 높다는 것을 의미하며, '0'에 근접할수록 둘 중 하나의 값이 상대적으로 낮은 값임을 의미한다.

부 록3

프로젝트 기여

주업무	기여자	검토자
기획	김소희	유수진
수집	김영주	유용빈
전처리	유용빈	김소희
분석	유용빈	유수진
시각화	유용빈	김영주
구현	김소희	유용빈
PPT	유수진	김소희
보고서	김영주	김소희
발표	김소희	김영주

부 록4

주차별 활동보고서(1주차)

<table>
<tr><td>과제참여
인원수</td><td colspan="2">4</td><td>회의참여
인원수</td><td colspan="2">4</td></tr>
<tr><td>활동일시
(회의 및
연구활동)</td><td colspan="2">1주차</td><td>장소</td><td colspan="2">교육장</td></tr>
<tr><td rowspan="8">참여자 명단</td><td>연번</td><td>소속(전공)</td><td>이름</td><td>서명</td><td>비고</td></tr>
<tr><td>1</td><td>통계학</td><td>김소희</td><td>김소희</td><td></td></tr>
<tr><td>2</td><td>기계공학</td><td>김영주</td><td>김영주</td><td></td></tr>
<tr><td>3</td><td>전자공학</td><td>유수진</td><td>유수진</td><td></td></tr>
<tr><td>4</td><td>정보통신공학</td><td>유용빈</td><td>유용빈</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>활동내용 및
회의
논의내용
기재</td><td colspan="5">주제선정후보
1. 신용카드 연체 예측(데이콘)
2. 따릉이와 기상데이터(날씨콘테스트공모전)
3. 시간대별 태양광 발전량 예측
4. 영화 분석 혹은 추천시스템

최종 주제선정 (과정에서 주제변경)
2. 따릉이와 기상데이터 ⇨ 5/14 영화 추천시스템
- 주제/목적/데이터수집

프로젝트 기획
- 머신러닝 프로젝트 흐름 파악</td></tr>
</table>

부 록5

주차별 활동보고서(2주차)

<table>
<tr><td>과제참여
인원수</td><td colspan="2">4</td><td>회의참여
인원수</td><td colspan="2">4</td></tr>
<tr><td>활동일시
(회의 및
연구활동)</td><td colspan="2">2주차</td><td>장소</td><td colspan="2">교육장, ZOOM</td></tr>
<tr><td rowspan="8">참여자 명단</td><td>연번</td><td>소속(전공)</td><td>이름</td><td>서명</td><td>비고</td></tr>
<tr><td>1</td><td>통계학</td><td>김소희</td><td>김소희</td><td></td></tr>
<tr><td>2</td><td>기계공학</td><td>김영주</td><td>김영주</td><td></td></tr>
<tr><td>3</td><td>전자공학</td><td>유수진</td><td>유수진</td><td></td></tr>
<tr><td>4</td><td>정보통신공학</td><td>유용빈</td><td>유용빈</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>활동내용 및
회의
논의내용
기재</td><td colspan="5">토의 결과 주제 변경 (따릉이 -> 영화추천시스템)

주제 변경이유 : 배운 데이터분석 내용을 기반으로 요즘 뜨고 있는 추천 알고리즘에 대한 지식공부와 머신러닝 스펙트럼을 넓히기 위해서

데이터 수집 : kaggle에서 제공하는 MovieLense 영화 데이터를 수집.
데이터 탐색 : 변수 탐색을 통해 대략적으로 수치형. 범주형으로 나눔,
해당 변수들에 이상치나 결측치 여부에 대한 탐색이후 전처리

추천시스템 이해 : 추천시스템에서 제공하는 알고리즘은 무엇인가?
추천이 이뤄지는 과정은 어떻게 되는지?
협업필터링이란 무엇인가?</td></tr>
</table>

부 록6

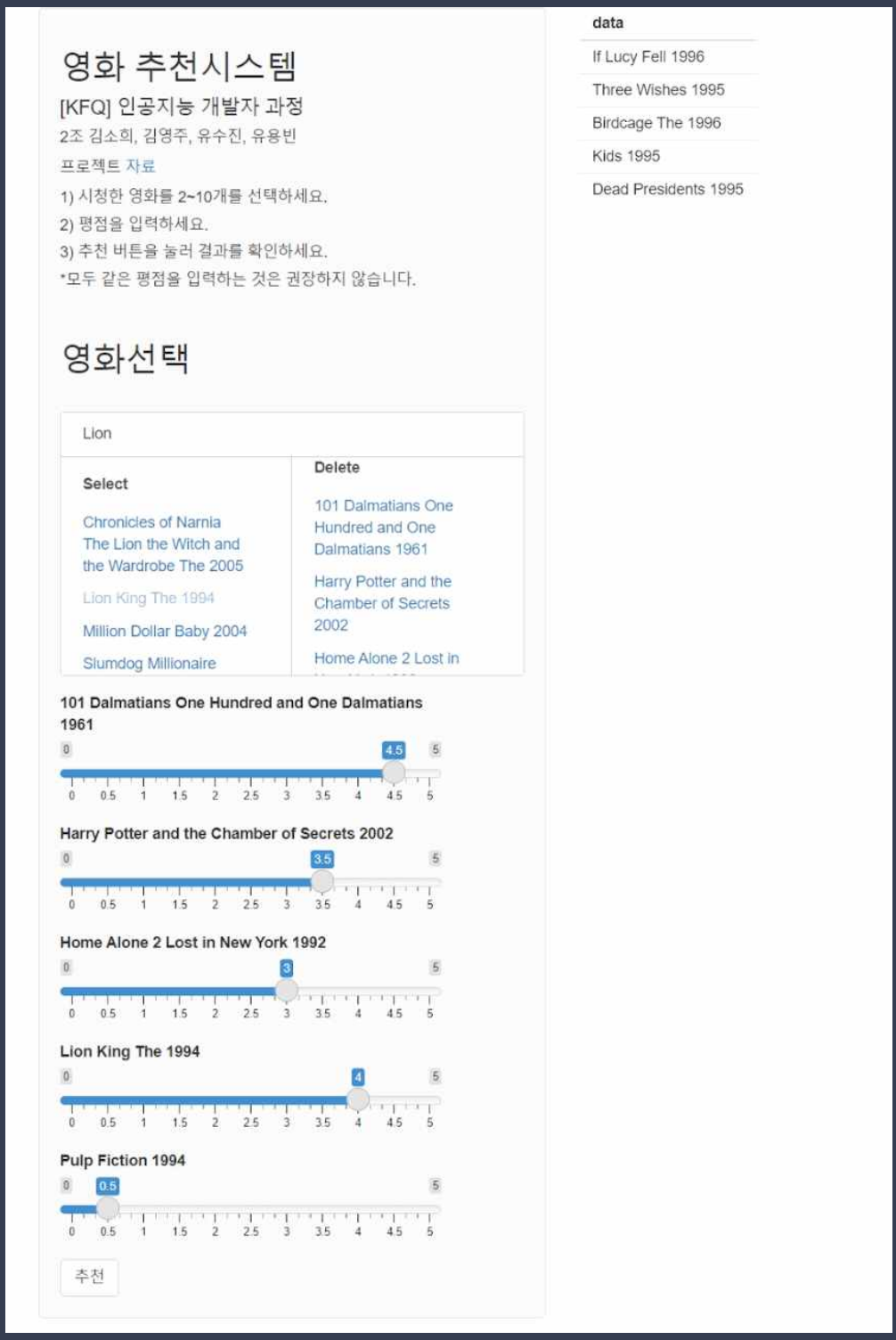
주차별 활동보고서(3주차)

<table>
<tr><td>과제참여
인원수</td><td colspan="2">4</td><td>회의참여
인원수</td><td colspan="2">4</td></tr>
<tr><td>활동일시
(회의 및
연구활동)</td><td colspan="2">3주차</td><td>장소</td><td colspan="2">교육장, ZOOM</td></tr>
<tr><td rowspan="8">참여자 명단</td><td>연번</td><td>소속(전공)</td><td>이름</td><td>서명</td><td>비고</td></tr>
<tr><td>1</td><td>통계학</td><td>김소희</td><td>김소희</td><td></td></tr>
<tr><td>2</td><td>기계공학</td><td>김영주</td><td>김영주</td><td></td></tr>
<tr><td>3</td><td>전자공학</td><td>유수진</td><td>유수진</td><td></td></tr>
<tr><td>4</td><td>정보통신공학</td><td>유용빈</td><td>유용빈</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>활동내용 및
회의
논의내용
기재</td><td colspan="5">지금까지 진행한 프로젝트를 마무리하며 추천 시스템에 사용되는 여러 가지 알고리즘 모델링을 적용해보았다. 적용해본 후 평가 과정을 거친 후 마지막으로 Wed으로 배포하는 작업을 진행하였다.

1. [장르] IBCF, UBCF, SVD 모델링 적용
2. [평점] IBCF, UBCF, SVD 모델링 평가
3. [장르] IBCF, UBCF, SVD 모델링 평가
4. [태그] 태그 텍스트마이닝 평가
5. 전개 계획 수립
6. 프로젝트 종료보고서 및 리뷰
7. Web 배포</td></tr>
</table>

부 록6

추천시스템 구현


영화 추천시스템
[KFQ] 인공지능 개발자 과정
2조 김소희, 김영주, 유수진, 유용빈
프로젝트 자료
1) 시청한 영화를 2~10개를 선택하세요.
2) 평점을 입력하세요.
3) 추천 버튼을 눌러 결과를 확인하세요.
*모두 같은 평점을 입력하는 것은 권장하지 않습니다.
영화선택
Lion
Select
Chronicles of Narnia The Lion the Witch and the Wardrobe The 2005
Lion King The 1994
Million Dollar Baby 2004
Slumdog Millionaire
Delete
101 Dalmatians One Hundred and One Dalmatians 1961
Harry Potter and the Chamber of Secrets 2002
Home Alone 2 Lost in
101 Dalmatians One Hundred and One Dalmatians 1961
Harry Potter and the Chamber of Secrets 2002
Home Alone 2 Lost in New York 1992
Lion King The 1994
Pulp Fiction 1994
추천
data
If Lucy Fell 1996
Three Wishes 1995
Birdcage The 1996
Kids 1995
Dead Presidents 1995