

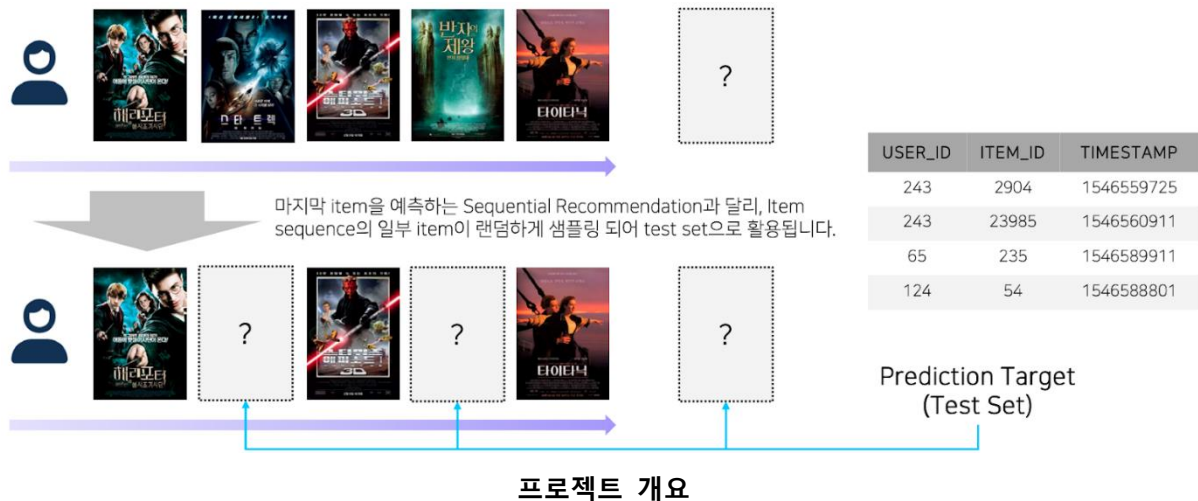
Movie Recommendation Competition Wrap up Report

Recsys-13 (Re?LU)

김원섭(T3044), 김진수(T3058), 민태원(T3080), 이상목(T3146), 조민재(T3204)

1. 프로젝트 개요

“사용자의 영화 시청 이력 데이터를 바탕으로 사용자가 다음에 시청할 영화 및 좋아할 영화를 예측”



대회에서는 implicit feedback 기반의 sequential recommendation 시나리오를 바탕으로 사용자의 time-orderd sequence 에서의 일부 item 이 누락(dropout)된 상황을 상정합니다. 이는 sequence 를 바탕으로 마지막 item 만을 예측하는 sequential recommendation 시나리오와 비교하여, 보다 복잡하며 실제와 비슷한 상황을 가정하고 있습니다. 또한, 여러가지 아이템 (영화)과 관련된 side-information 이 존재하기 때문에, 이것들을 어떤 식으로 효과적으로 활용할지도 중요한 포인트가 될 수 있습니다.

1.1 활용 장비 및 개발환경

OS : Ubuntu 18.04.5 LTS, Windows

IDE : VS Code

GPU : Tesla V100 *Boostcamp 로부터 제공받은 서버

주 사용 언어 : Python 3.8.5

Frameworks :Pytorch 1.10.2, implicit, etc.

Co-op tools : github, notion, slack

1.2 프로젝트 및 사용 데이터셋의 구조

-프로젝트

input	output
31360 명의 user 와 6807 가지 item 의 상호작용 데이터	user 별 10 가지 아이템의 추천(총 313600 행)

-데이터 셋

train
├── ML_item2attributes.json
├── directors.tsv
├── genres.tsv
├── titles.tsv
├── train_ratings.csv
├── writers.tsv
└── years.tsv

File	Description
ML_item2attributes.json	item 과 genre 의 mapping 데이터 Dict{str:list} 5,154,471 행
directors.tsv	영화 감독 데이터 item : int, director : str 5,905 행
genres.tsv	영화 장르 데이터 item : int, genre : int 15,934 행

titles.tsv	영화 제목 데이터 item : int, movie : str 6,799 행
train_ratings.csv	주 학습 데이터 user_id : int, item_id : int, timestamp : int 5,154,471 행
writers.tsv	영화 작가 데이터 item : int, writer : str 11,307 행
years.tsv	영화 개봉 년도 item : int, year : int 6,799 행

2. 프로젝트 팀 구성 및 역할

팀 구성 : 조민재, 김진수, 이상목, 민태원, 김원섭

팀원 개개인의 성장을 목표로 U stage 에서 학습했던 다양한 방법론들과 추가적으로 습득한 지식을 실험하였다. 이전 프로젝트에서는 각자 실험했던 내용들을 정리하지 않아 중복된 작업이 일부 발생하였기에, Notion 을 통해 각자의 실험내용 및 결과를 정리하고, Git 으로 서로의 코드를 참조해서 효율을 높이는 형태로 협업하였다.

3. 프로젝트 수행 절차 및 방법

효율적인 협업을 위해 커밋 메시지 컨벤션을 설정하고, 개인의 작업사항을 한눈에 파악하기 편하도록 Notion 페이지를 활용하였다. 다양한 모델 개선 아이디어 및 테크닉을 실험 후 성능을 기록하고, 이를 바탕으로 피어 세션에 토의하여 새로운 아이디어를 도출하였다.

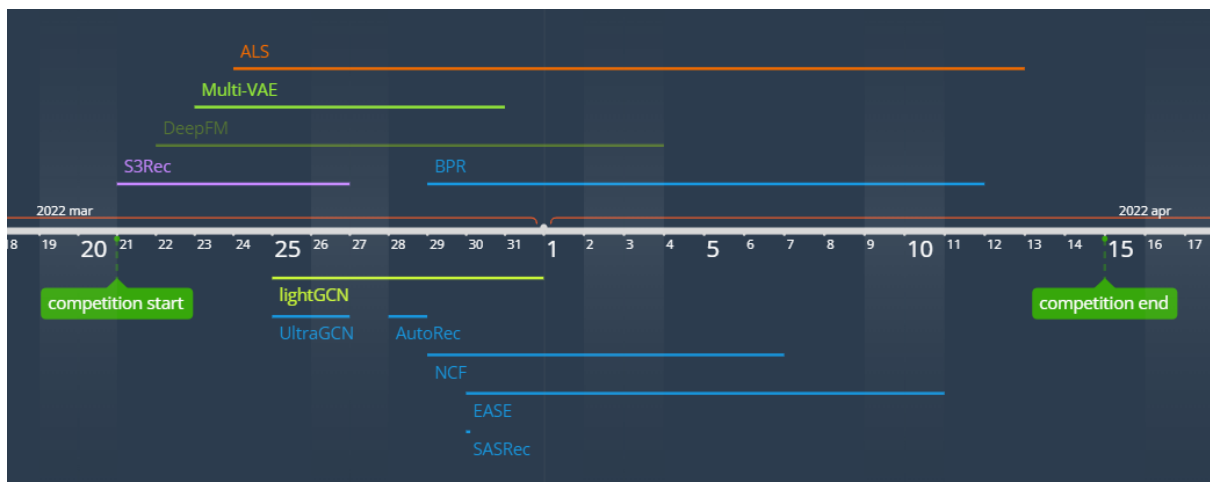
실험 기록
개인실험을 기록합니다.

- 실험: 편집하려는 실험에 마우스 커서를 놓고 **실험** 버튼을 클릭하여 실험을 기록할 수 있습니다.
- 진행 일시: 실험 '시작일'과 '종료일'을 기록해주세요.
- 진행 상태: 실험 및 product 진행 상태를 표시해주세요.
- 예상: 실험 예정입니다.
- 실험: 실험 중입니다.
- 완료: 실험을 완료하였습니다.
- 실험 결과: 실험 제출 결과를 표시해주세요.
- 새 실험: 점수 증가, 점수 감소, 점수 유지 중 하나를 선택해주세요.
- 점수: 제출 점수를 입력해주세요.
- 실험 결과 요약: 실험 제출 결과를 요약(정량적 평가)해주세요.
- 실험들을 하나하나 클릭하지 않아도 쉽게 결과를 확인하고 비교할 수 있습니다.

실험	실험자	진행 일시	진행 상태	실험 결과	리더보드	실험 결과 요약	git link	baseline
S3RecModel(Baseline)	이상목	March 21, 2022 → March 21, 2022	완료	새 실험	0.0884	baseline 점수 0.0884		0
DeepFM(Mission3)	전수 김	March 22, 2022 → March 23, 2022	완료	새 실험	0			-0.0884
MultiVAE	조민재	March 23, 2022 → March 24, 2022	완료	점수 감소	0.0224		https://github.com/boostcampai	-0.066
ALS	김원섭	March 24, 2022 → March 24, 2022	완료	점수 증가	0.1483	epoch 50	https://github.com/boostcampai	0.0599
MultiVAE+S3Rec(5개씩)	이상목	March 24, 2022 → March 24, 2022	완료	점수 증가	0.0963	navie 양상불 - 약간 개선		0.0079
MultiVAE(마스킹 추가)	이상목	March 24, 2022 → March 24, 2022	완료	점수 증가	0.0838	마스킹 추가, 200에폭		-0.0046
DeepFM(Input 추가)	전수 김	March 24, 2022 → March 25, 2022	완료	새 실험	0	연산량이 너무 많아 부적합한 것으로 판단됨	https://github.com/boostcampai	-0.0884
ALS + S3Rec (8.2)	김원섭	March 25, 2022 → March 25, 2022	완료	점수 감소	0.14			0.0516
lightGCN	조민재	March 25, 2022 → March 30, 2022	완료	새 실험	0.106	epoch 100	https://github.com/boostcampai	0.0176
UltraGCN	전수 김	March 25, 2022 → March 27, 2022	완료	점수 감소	0.004		https://github.com/boostcampai	-0.0844
MultiVAE(Train with All Data)	김원섭	March 26, 2022 → March 26, 2022	완료	점수 증가	0.1318	epoch 120	https://github.com/boostcampai	0.0434
MultiVAE + S3Rec(7-31)	이상목	March 26, 2022 → March 27, 2022	완료	점수 감소	0.1231	naive 양상불		0.1

Notion을 활용한 실험 기록 공유

대회 시작부터 역할을 나누어 EDA 및 Baseline code 분석을 실시하고, special mission 부터 시작하여 다양한 모델들을 가져와 적극 활용하였다.



대회 중 다양한 모델 실험 Timeline

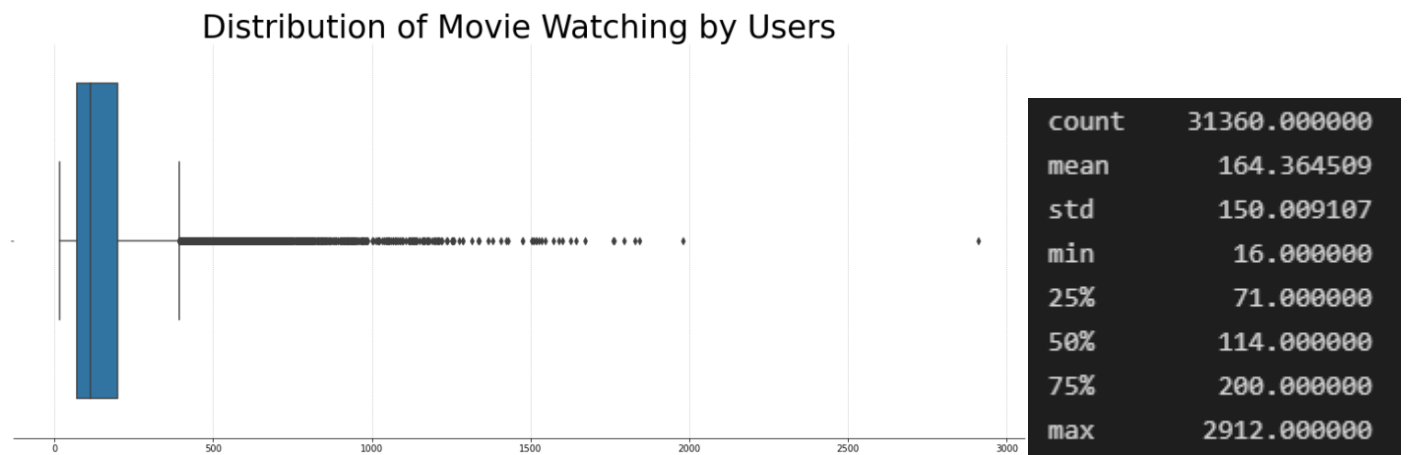
4. 프로젝트 수행 결과

4.1 EDA 및 데이터 전처리

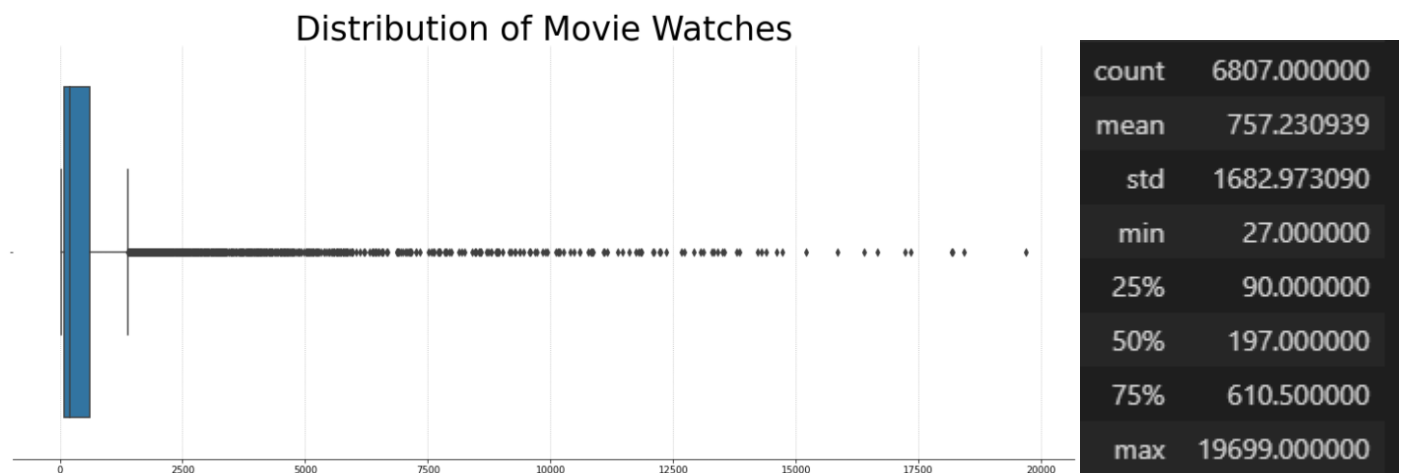
유저-영화 정보

user			item	year
29762	131174	[724, 830, 3247, 2116, 3745, 1620, 2082, 4639, ...]		[2008]
28765	126721	[6333, 1408, 44191, 431, 1287, 1882, 3252, 101...		[2010, 2012]
30394	133961	[319, 1416, 2428, 48385, 36517, 34319, 49530, ...]		[2007, 2008, 2009]
5254	22989	[3825, 8533, 85007, 99114, 1307, 1682, 64969, ...]		[2013, 2014, 2015]
632	2728	[1967, 308, 2788, 2083, 356, 318, 150, 590, 58...		[2008, 2011]
9366	40863	[2366, 308, 3261, 2747, 48385, 48780, 48394, 2...		[2007]
412	1776	[1407, 908, 2700, 4034, 1358, 543, 3471, 471, ...]		[2006]
21195	92892	[60, 2719, 374, 58559, 2053, 2808, 502, 2950, ...]		[2009, 2010, 2011, 2012]
30332	133701	[1285, 1244, 2528, 3476, 6350, 27773, 31658, 7...	[2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015]	
6605	28837	[1285, 333, 2278, 3108, 318, 50, 71745, 71535, ...]		[2009]

기존 데이터에 UNIX 시간으로 제공된 time 변수를 변환해 user 의 활동시기 추출



user 별 시청한 영화 수의 분포로 볼 때, 평균값이 164 로 대부분의 이용자가 200 편 정도의 영화를 본 것을 알 수 있다. 영화를 가장 많이 본 user 는 2912 개로 전체 영화의 약 42.7%를 차지한다. (이상치로 생각할 수 있음)



영화별 시청한 user 수 분포를 확인한 결과 일부의 영화는 많은 user 들이 본 것을 확인할 수 있다. 대부분의 영화는 1000 명 이하의 user 가 본 영화라고 생각할 수 있다.

영화 정보

item		title	year	genre	director	writer
0	318	Shawshank Redemption, The (1994)	1994.0	[Crime, Drama]	[nm0001104]	[nm0000175, nm0001104]
1	2571	Matrix, The (1999)	1999.0	[Action, Sci-Fi, Thriller]	[nm0905152, nm0905154]	[nm0905152, nm0905154]
2	2959	Fight Club (1999)	1999.0	[Action, Crime, Drama, Thriller]	[nm0000399]	[nm0657333, nm0880243]
3	296	Pulp Fiction (1994)	1994.0	[Comedy, Crime, Drama, Thriller]	[nm0000233]	[nm0000233, nm0000812]
4	356	Forrest Gump (1994)	1994.0	[Comedy, Drama, Romance, War]	[nm0000709]	[nm0744839]
...
6802	73106	American Pie Presents: The Book of Love (Ameri...	2009.0	[Comedy]	[nm0003289]	[nm0381221, nm0825738]
6803	109850	Need for Speed (2014)	2014.0	[Action, Crime, Drama]	[nm0915304]	[nm0309691]
6804	8605	Taxi 3 (2003)	2003.0	[Action, Comedy]	[nm0470443]	[nm0000108]
6805	3689	Porky's II: The Next Day (1983)	1983.0	[Comedy]	[nm0163706]	[nm0163706, nm0650276]
6806	8130	Girl Next Door, The (1999)	1999.0	[Documentary]	NaN	NaN

6807 rows x 6 columns

결측값 확인

```

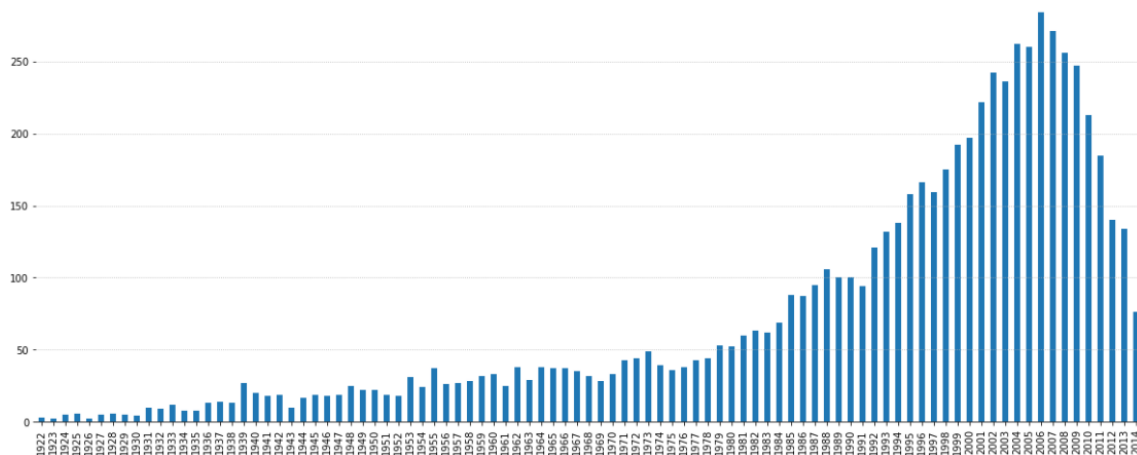
item      0
title     0
year      8
genre     0
director  1304
writer    1159

```

year(영화 개봉년도) : 영화의 연도 정보를 포함하는 title 파일을 이용해 결측 값 처리

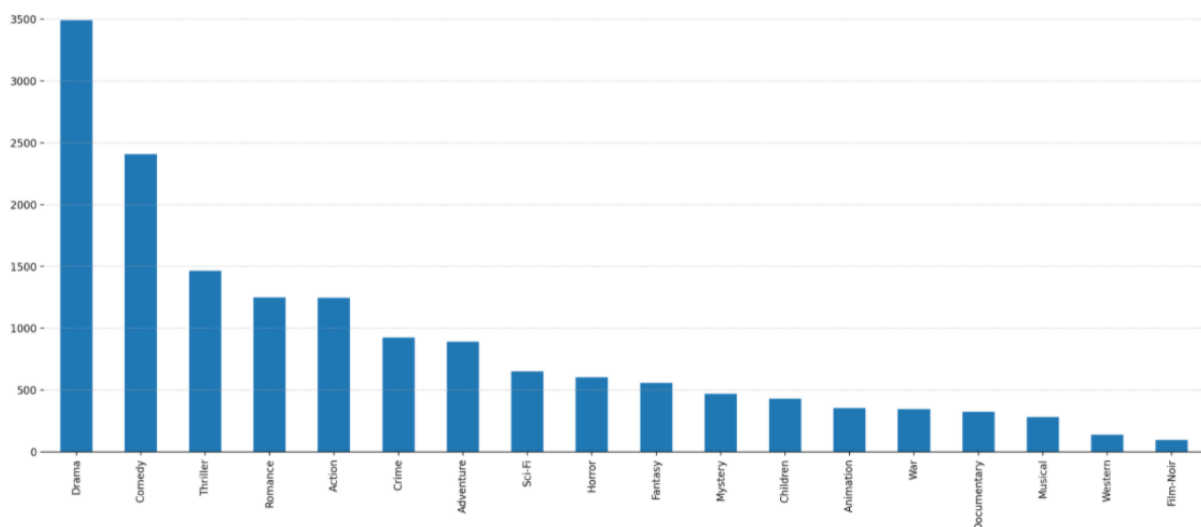
director(감독), writer(작가) : 결측 값이 많아 사용하기 적절하지 않음.

Number of movies released by year



영화들의 개봉연도는 1922 - 2014 년이지만 대부분의 영화가 1990 년도 이후에 개봉된 것을 볼 수 있다.

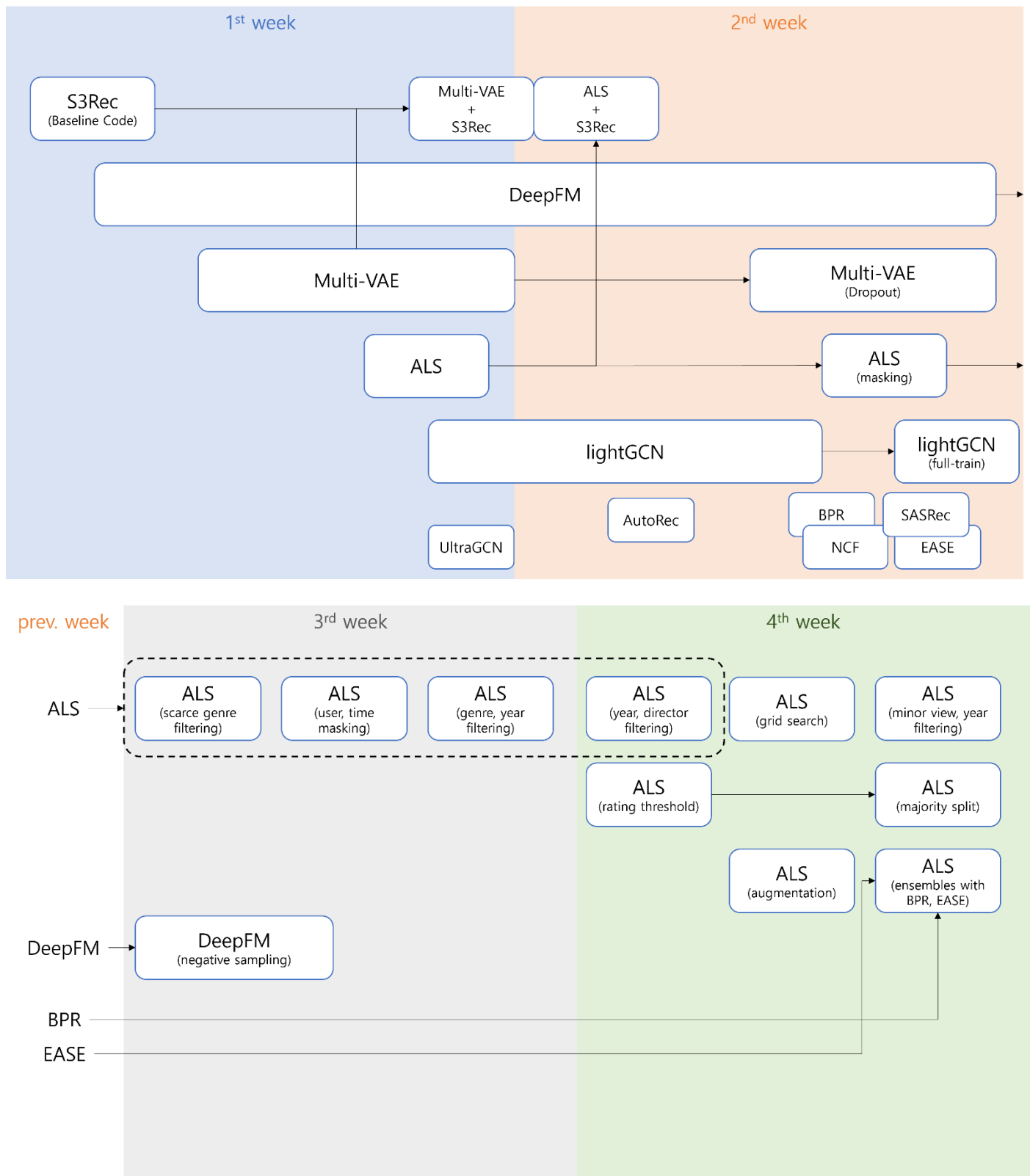
Number of Genres



영화의 장르는 Drama, Comedy 가 다른 장르에 비해 상대적으로 많은 것을 볼 수 있다.

4.2 활용한 모델과 방법, 그리고 결과

아래는 프로젝트 진행과정을 간단하게 도식화한 것이다.



S3 Rec : Baseline Code

Exp.	S3Rec	S3Rec + ALS
Description	Baseline code	ALS:S3Rec = 8:2 의 비율로 ensemble (bagging)
LB Score	0.0884	0.14

Deep FM

영화 추천 task에서는 영화 시청 이력 외에 다양한 side-information 이 존재함. 이러한 정보를 담을 수 있는 구조의 모델인 deepFM 을 활용하였다.

- negative sampling : refactoring 한 deepFM 을 바탕으로 pair-wise approach 에서 negative sampling 의 수가 실험 결과에 어떠한 영향을 미치는지 실험

Exp.	DeepFM	DeepFM(neg sampling)
Description	Implementation	negative sampling num = 1000
LB Score	0.0302	0.0423

Multi-VAE

VAE 에서 Latent vector 의 샘플링을 Gaussian distribution 이 아닌 multinomial distribution 으로 모델링하여, item 들이 한정된 user 의 예산을 놓고 경쟁하는 모습을 표현할 수 있도록 한다.

- MultiVAE + S3 Rec : VAE 는 어디까지나 interaction data 의 reconstruction 에 이용되므로, sequential 한 정보의 예측도 가능하지만 static 한 정보를 맞추는 것이 주 목적이다. 하여, 이를 보완할 수 있는 모델인 S3 Rec 의 출력과 ensemble 하는 방법을 활용해 보았다.

- Dropout : Multi-VAE 에 Dropout 비율을 조절하여 성능 변화를 측정해 보았다. Input layer 에만 적용하므로, 일종의 Data augmentation 으로 이해할 수 있고, 모델의 표현력을 늘리는 형태로 적용된다고 추정할 수 있다. Dropout 비율을 증가시킴에 따라 성능이 증가하는 모습을 보였다.

Exp.	MultiVAE	MultiVAE + S3Rec	MultiVAE(Dropout)
Description	implementation (200 epoch) 3/24	MultiVAE:S3Rec = 7:3 의 비율로 ensemble (bagging) 3/26	data 에 dropout 비율 조절하여 최적의 결과 출력 3/28
LB Score	0.0838	0.1231	0.0843

ALS

Implicit 라이브러리의 ALS 를 사용하였다

- ALS + S3Rec(8:2) : static 모델과 앙상블함으로써 sequential 모델의 단점을 극복하고자 함.

- Masking & Filtering : user 별 선호감독 Top3 으로 masking 하거나, user 가 본 영화들과 전혀 상관없는 장르의 영화, user 의 활동 년도 이후의 영화 + 시청수가 적은 영화 등 다양한 지표를 통한 masking & filtering

- ALS + EASE : EASE user-item 행렬과 ALS user-item 행렬을 정규화 후 더하여 추천

- rating threshold : 영화들의 시청자 수를 기반으로 major 영화와 minor 영화로 구분한 뒤 분류된 영화의 시청 비율을 통해 major user 와 minor user 를 구분. 한쪽에 추천을 하지 않음으로써 기본 ALS 모델이 어느 쪽에 더 적절한 추천을 제공하고 있는지 분석

- majority split : 영화들의 시청자 수를 기반으로 major 영화와 minor 영화로 구분한 뒤 분류된 영화의 시청 비율을 통해 major user 와 minor user 를 구분, 데이터를 split 하여 따로 학습을 시키고 각자 따로 추천

Exp.	ALS	ALS(Grid Search)	ALS(multi masking)	ALS(majority split)
Description	User-Item matrix 를 user 와 item vector 로 분해하여 추천에 활용한다	Grid Search 를 이용해 ALS 의 최적의 hyperparameter 를 탐색한다.	훈련데이터, 활동연도, 장르, 감독으로 masking	인기 영화를 주로보는 major user 와 비인기 영화를 주로 보는 minor user 를 개별 학습
LB Score	0.1483	0.1530	0.1484	0.1498

Light-GCN

Exp.	LightGCN	LightGCN(full)
Description	Implementation	Validation data 없이 모든 데이터로 Train
LB Score	0.1158	0.1178

EASE

Exp.	EASE	EASE (neg sampling)	ALS+EASE
Description	Implementation	neg sampling 후 MF 적용	ALS user-item 행렬 정규화 + EASE user-item 행렬 정규화 (soft-voting)
LB Score	0.1422	0.1123	0.1243

기타

Exp.	Autorec	BPR	UltraGCN
Description	implementation	random sampling 하여 feeding	implementation
LB Score	0.0006	0.0007	0.0040

- **SASRec** : 베이스 모델인 S3Rec 에서 pretrained 된 임베딩이 얼마나 효과가 있는지를 검정하고, SASRec 모델 자체만의 성능을 확인하기 위한 실험 진행

- **Neural Collaborative Filtering (NCF)**

: 기본적인 형태의 CF 모델인 ALS 에서의 결과가 잘 나오는 것을 확인함. 따라서 딥러닝을 적용한 cf 모델의 성능을 확인하기 위해 NCF 를 실험 -> 0.141(iter5)

: S3Rec 의 self-supervised learning 과 유사하게, pretrain 된 아이템 임베딩을 이용하여 NCF 모델을 학습 -> 0.134(iter5)

5. 자체 평가 의견

대회를 진행하며 전반적으로 추천시스템에 대한 이해도를 높일 수 있었다. 또한, 지난 대회에서 부족했던 협업을 보완하기 위해 다양한 방법을 제시하고 또 시도하였다. 대표적인 예시로 Notion 을 통해 각자 실험을 정리하고 기록하여 효율적인 의견공유를 할 수 있었다. 이로 인해 중복된 실험으로 시간을 낭비하지 않고 계획해둔 요한 모델들을 실험해 볼 수 있었고, 데이터 분석으로부터 생성한 가설을 검증해 볼 수 있었다. 비록 최종 결과는 조금 아쉬운 순위로 마무리했으나 충분한 실리를 챙길 수 있었던 대회였다. 이 후 대회에서는 팀원들 간 실험에 대해 코드리뷰를 진행할 계획이다.

6. 팀원 개인 회고

조민재(T_3204)

목표

사용자 - 영화 시청 기록 분석을 통한 영화 추천시스템 구현

개인 실험

- multi-VAE 를 대회 형식에 맞추어 refactoring
- lightGCN implementation 및 hyper parameter tuning
- 영화별 시청한 user 수를 기반으로 major movie 와 minor movie 를 구분, user 를 각 영화를 시청한 비중에 따라 major 한 영화를 선호하는 user 인지, minor 한 영화를 선호하는 user 인지 분리하여 분석하고자 함.
- 사용된 추천시스템 모델이 얼마나 user 취향을 잘 반영하고 있는지 분석한 후, 후처리를 통한 성능향상에 초점을 맞춤

회고

이번 대회에 가장 아쉬운 점이 있다면 추천 시스템에 임의의 정답을 두고 이를 예측하려는 문제였다는 점이다. 추천 시스템은 사용자의 경향성 혹은 아이템과의 상관관계를 분석하여 이를 바탕으로 다음 선택지를 추천해주는 시스템인데, 과거에 시청했던 기록의 일부를 정답으로 사용하여 기대했던 목적을 달성하기 힘들었기 때문이다. 이 점을 깨달은 뒤부터는 의욕이 조금 꺾인 것 같다. 그래도 성과가 있다면, 기본적인 cf 모델도 user-item 간의 상호작용으로부터 상당히 많은 정보를 뽑아낸다는 점을 알 수 있었다. 대회에 주어진 side information 을 사용하지 않고도 als 모델은 그 정보로부터 생길 수 있는 side effect 를 꽤 잘 표현하고 있었다. 이를 세세하게 조정하여 더 나은 추천을 제공함으로써 사용자의 선택을 이끌어낼 수 있을 것이다.

앞으로의 방향

모델로부터 단순히 결과를 끌어내는 것에 그치지 않고, 그 결과를 조작하여 더 나은 성능을 내는 과정에서 재미를 찾을 수 있었던 4 주였다. 내가 원하는 작업물에 적절한 활용법을 생각해낼 수 있을 만큼 추천시스템의 기본을 더 탄탄히 갖추고자 마음을 먹었다. 또한 이를 바탕으로 일상에서 다양한 비즈니스적 활용법을 구상해보고자 한다.

김진수(T_3058)

목표

추천시스템에 대한 이해도 향상 및 협업능력 향상

개인 실험

- 팀원들, 토론게시판 및 개인적인 EDA 를 통한 인사이트 발굴
- DeepFM, Autorec 같은 Special Mission Model 과 Github 에서 EASE, BPR, UltraGCN 모델 활용

- 팀에서 가장 잘 나왔던 모델인 ALS 에서 감독, 연도, 장르 필터링 실험,
- EASE 모델과 ALS 모델을 soft-voting 방식으로 앙상블 실험

좋았던 점

1. 협업 - 실험 표 작성

저번 대회에 비해 표를 정리하여 각자의 실험이 중복되는 과정이 생략되어 더욱 다양한 실험을 해볼 수 있어서 좋았다.

2. 팀원이 실험했던 모듈을 활용

깃으로 팀원들이 commit 한 작업물을 참고해보거나 앙상블같은 새로운 실험에 적용해 볼 수 있어서 좋았다.

아쉬웠던 점

1. 꼼꼼하지 못했던 모델 실험

: 실험에 최적화된 모델을 찾아보기 위해 다양한 모델을 실험하는 것은 좋았으나, 초반에 제대로 된 실험을 하지 못한 것이 아쉽다. 초반에 1~2 일만에 1 개의 모델을 실험하다보니 이런 상황이 나왔던 것 같다. 충분하지 못했던 실험이 결과적으로 의미가 없어졌고, 점수향상을 이루어 낼 수 없었다. 한 달에 시험해볼 모델은 4~5 개가 적당하다고 생각이 되므로, 다음 대회에서는 한주에 1~2 개를 목표로 실험해 봐야겠다.

2. 협업 - 코드리뷰

: 각자 실험했던 내용들이 중복되지 않아 많은 실험을 해볼 수 있었다. 하지만 서로 코드리뷰를 해보았다면 각자가 알지 못했던 부분을 잘 알 수 있었을 텐데 많이 이루어지지 못해서 아쉽다. 다음에 한다면 코드리뷰를 요청하는 시간이 필요하다고 생각된다.

민태원(T_3080)

목표

1. 데이터를 중점적으로 확인해 인사이트 찾기
2. 추천시스템에 사용되는 다양한 방법론을 익히고 상황에 맞게 적용할 수 있는 역량 향상
3. 효율적인 팀 프로젝트 진행을 위한 협업툴 적응

목표 달성을 위한 노력

- **문제에 대한 이해** : competition 에 제공 된 데이터의 설명으로는 중간 누락 데이터+미래 데이터를 예측하는 것이었는데 sequential 한 접근법이 의미 있을지에 대한 고민을 했고, 명확하게 어느 지점을 예측하는지 알 수 없기 때문에 static 한 방법에 초점을 맞춤.

결과) sequential model 보다는 static model 이 score 면에서 더 높게 나옴.

- **EDA 를 통해 다양한 가설 세우기** : 유저별 시청한 영화 수의 지표는 적당한 threshold 값을 조절하면서 영화를 많이보는 유저와 적게보는 유저를 나눠 다른 방식으로 추천을 해주는 방식을 사용해 봤고 영화별 시청한 유저수의 지표는 인기영화 추천, 비인기 영화 제거 와 같은 방법에 사용함. 후처리로 유저마다 활동기간을 고려해 미래 영화를 제거하는 방법을 사용.

결과) 인기있는 영화(유저들이 많이본 영화로 가정)를 추천 목록에 포함 했을 경우 점수 상승을 기대하기 어려움. 유저들이 많이 보지 않은 영화와 유저의 활동 시기를 고려해 후처리를 했을 경우 score 가 소폭 상승하는 것을 확인할 수 있었음.

- **다양한 방법론 과 model 탐색** : 추천시스템에서 자주 사용되는 CF, CB 에 대해 이해하고 해결할 문제, 데이터에 따라 적용할 수 있는 방법이 어느정도 정해져 있다는 것을 배움. 이 과정에서 문제 정의와 데이터에 대한 이해가 중요하다는 것을 알게됨. 추천시스템 model 을 구축해놓은 라이브러리나 github 를 탐색하며 여러 model 을 탐색해서 적용함.
- **적극적인 Github 사용** : 기존에는 github 를 단순히 저장공간으로 사용했다면, 이번 competition 에는 커밋 메시지 컨벤션을 사용했고, 유용하게 사용할 수 있는 코드를 서로 공유하며 이전보다 팀적인 활동을 진행함.

아쉬운 점

- 라이브러리를 사용하면서 model 을 테스트할 때, 이론적인 부분이나 model 에 대한 이해가 낮아서 제대로 활용하지 못한 것 같음. 또한, inference 부분은 구현이 필요했지만 구현을 하는데에는 아직 코딩적인 역량이 부족하다고 느낌.
- 데이터를 분석하는데에 많은 시간을 투자했지만 그렇다할 인사이트를 얻지 못한거 같음.
- 효율적인 실험을 위해 팀원들과 실험내용을 공유했지만 그 실험이 제대로 된 실험인지에 대한 생각을 해볼 필요가 있었음.
- competition 의 후반부로 갈 수록 쳐져가지고 마지막 주에는 의욕이 부족했던거 같음.
- wandb 와 같은 효율적인 팀워크를 위한 툴을 사용해보지 못함.

다음 프로젝트에서 시도해보거나 개선할 점

- 프로젝트와 이론을 함께하다보니 두가지를 함께 할 시간이 부족음. 다음 프로젝트에는 시간 관리를 효율적으로 하도록 노력하려고 함.
- 다른 팀들이 사용한 좋은 협업툴을 적용해보고 효율적인 방법을 모색.
- 팀원들과 실험 내용 공유를 좀 더 구체적으로 공유하고 잘못된 방향으로 실험을 한것이 아닌지 확인하는 절차가 필요하다고 생각함.

이상목(T_3146)

목표

- 데이터와 모델 고찰 및 성능변화 관측을 통한 추천시스템 이해

성장한 점

- **체계적인 실험 관리** : notion 을 통해 팀원 개인이 무엇을 하고 있고, 어떤 결과가 있었는지 인사이트를 효과적으로 공유할 수 있었다. 지난 프로젝트에서 개선하면 좋겠다고 고민했던 부분이었으며, 잘 적용되었다.
- **테스트 파이프라인 구축** : feature 와 label 의 구분이 모호한 상황이지만, 대회의 구성에 입각하여 테스트 파이프라인을 구축하여 실험의 성능을 간편하게 파악할 수 있었다. 또한 이 과정에서 argpartition 과 같은 고효율 알고리즘도 공부해 볼 수 있었다.
- **sparse & feature, label 이 혼재된 데이터 핸들링** : user-item matrix 로 표현 시 매우 희소해지는 데이터를 다루는 상황에 놓이면서, 어떻게 효과적으로 다룰지 계속 고민하게 되어 이에 숙련될 수 있는 계기가 되었다.

성장할 점

- 개인적으로 특히 관심을 가졌던 부분은 multimodal 한 side information 의 활용이었다. 하지만 이는 당연히 활용하는 방법이 복잡하고, 학습과 추론 속도 면에서 손해를 볼 수 밖에 없다. 게다가 성능 면에서도 ALS 같은 행렬 분해 알고리즘이 더 강세를 보여 활용하지 않게 되었다. 하지만 추가 정보를 활용할 수 있다면 더 좋은 성능을 기대할 수 있으리라는 생각에는 여전히 변화가 없다. 처음 추천시스템을 다루다 보니 덜 효과적인 방법으로 적용하여 이런 결과를 얻지 않았을까 생각한다. 해당 정보들도 올바르게 활용할 수 있도록 계속 공부하려 한다.
- 실험 측면에서는 인사이트를 공유하는데 있어 아주 성공적이라 방법이었다 생각하지만, 코드 측면에서는 공유가 조금 불편했다고 생각한다. 이 부분을 정교한 설계를 통해, 각자 실험을 인사이트 수준을 넘어 코드 모듈 수준으로 쉽게 활용할 수 있도록 구성해 보려 한다.
- NNI, MLflow 와 같은 자동화된 학습 및 테스트 툴을 활용해 보려 한다. pytorch 에 기반한 모델을 사용하지 않아 하이퍼 파라미터 튜닝을 직접 구현하여 활용했는데, 아무래도 직접 구현하기 보단 이를 활용해 보는 것이 더 효율적이었을 것이며, 통일성도 있고 한눈에 알아보기도 좋을 것이다. 모듈화와 자동화를 더 고민하여 전공자의 강점을 온전히 살릴 수 있도록 해보려 한다.

아쉬웠던 점

- 사실 대회를 진행하면서, '이게 정말 추천이 맞나?' 하는 의문을 계속 가졌던 것 같다. static+sequential 한 아이템의 예측이지만, 과연 유저가 원하는 추천이 이러한 형태를 원할 것인지 확신이 생기지 않는다. 최종 프로젝트를 진행할 시에는 AB 테스트와 같은 방법을 통해 유저 피드백을 직접 경험할 수 있는 방향으로 진행해보고 싶다.
- 추천 시스템은 빠른 시간 내에 유저에게 아이템을 보여 주는 것 역시 무엇보다 중요한 사항 중 하나라 생각된다. 대회이기에 성능을 높이기 위해 voting ensemble 과 같은 실 서비스에서 활용하기 어려운 방법도 시도하였지만, 추천 시스템을 실 서비스하기 위해서는 서빙 관점에서도 고민하며 진행하면 더 좋은 코드를 작성할 수 있지 않을까 생각한다.
- 여러 방향으로 EDA 를 시도하면서, 임베딩의 시각화가 잘 나타나지 않아 영화나 유저들 간 관계에 대한 인사이트를 발굴하지 못한 부분은 아쉬웠다. 특히 오피스 아워를 지나며 시계열

데이터란 점을 활용한 n-gram 인사이트는 매우 신선하게 다가왔다. 캐글 등지에서 다양한 데이터에 대한 EDA 를 구경하며 내공을 더 쌓을 필요가 있을 것 같다.

김원섭(T_3044)

목표

- (팀) 추천시스템에 대한 팀 전체의 기량을 향상시키기 위해, 다양한 심화 이론들에 대한 모델링을 각자 진행하고, 시도한 방법과 결과를 노션과 깃허브에 공유하는, 대회와 스터디의 장점을 융합한 형태의 협업을 진행한다.
- (개인) 추천시스템 관련 심화 논문을 읽고 이해하며, 이를 구현할 수 있는 능력을 함양하기 위해 나만의 모델링 코드를 구축하고, 다양한 실험 사이클(가설-검증)를 구성한다.

나는 내 목표를 달성하기 위해 무엇을 어떻게 했는가? (테마별 분류: EDA, CF/Context/Seq 모델)

- **EDA 및 데이터 분석 적용(필터링)**
 - 영화를 본 유저들의 취향(장르) 흐름(year 및 week 기준)을 살펴 본 결과, 유저들의 취향은 크게 변하지 않음(변하더라도 다시 회귀하는 성향을 발견).
= sequential 한 모델이 의미가 있을까? -> 기본적인 CF 모델부터 살펴봄.
 - 활동시기가 유저마다 다름을 확인.
= 유저별 활동시기 이후에 개봉한 영화들은 필터링시킴.
 - 영화추천 태스크의 흐름을 static/sequential 로 나누어 예측.
= static/seq 모델을 특정한 비율로 앙상블.
- **CF & 딥러닝 모델**
 - ALS 모델 구현 및 다양한 실험
 - Neural Collaborative Filtering 모델 구현 및 다양한 실험
 - Multi-VAE 모델 구현 및 다양한 실험
- **Context 모델**
 - deepFM 모델 구현 및 다양한 실험
- **Sequential 모델**
 - SASRec 모델 구현 및 다양한 실험

스스로 칭찬할 점은 무엇인가?

- 대회 초반, 대회 태스크를 먼저 고민하기 보다 팀 단위 협업을 어떻게 효율적으로 할 수 있는가를 먼저 고민하고, 좋은 팀 환경을 구축하려 노력했던 점을 칭찬하고 싶다.
- 필요한 태스크에 대해 정확한 문제 정의를 진행하기 위해 노력하였다.
- 추천시스템 모델들에 대한 이론을 추가로 공부한 뒤, 태스크별로 정리하는 작업을 가장 먼저 진행하여, 각 실험 태스크에 적절한 모델 설계를 진행하였다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 실험에 대해서는 노선을 통해 효율적으로 협업하였지만, 팀원들의 실험 코드에 대한 인사이트는 많이 공유받지 못했다고 생각된다.
- 문제 정의와 모델 설계 등 너무 세세하게 실험을 설계하였던 점은 스스로 칭찬할 점이지만, 동시에 한 실험에 너무 많은 자원(시간 등)을 투자했다는 점은 아쉽게 느껴진다.
- 실험 설계 과정에서 EDA 에 대한 비율이 조금 적었던 것 같다.
- 장기간 프로젝트에 대한 설계가 조금 미흡하여 팀 전체가 후반으로 갈수록 조금 느슨해졌던 것 같다.

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇인가?

- 다음 프로젝트에서는 깃 Issue 와 PR 을 적극적으로 사용하여 실험 코드에 대한 인사이트를 명확히 공유할 수 있도록 해야겠다.
- 팀 목표의 초점이 '점수 향상'에 맞춰진다면, 너무 세세한 실험 설계 보다는 빠른 가설-실험 사이클을 돌려보는 것도 좋은 대안일 수 있을 것 같다.
- 각 실험별로 충분한 데이터 분석이 필요할 것으로 보인다.