

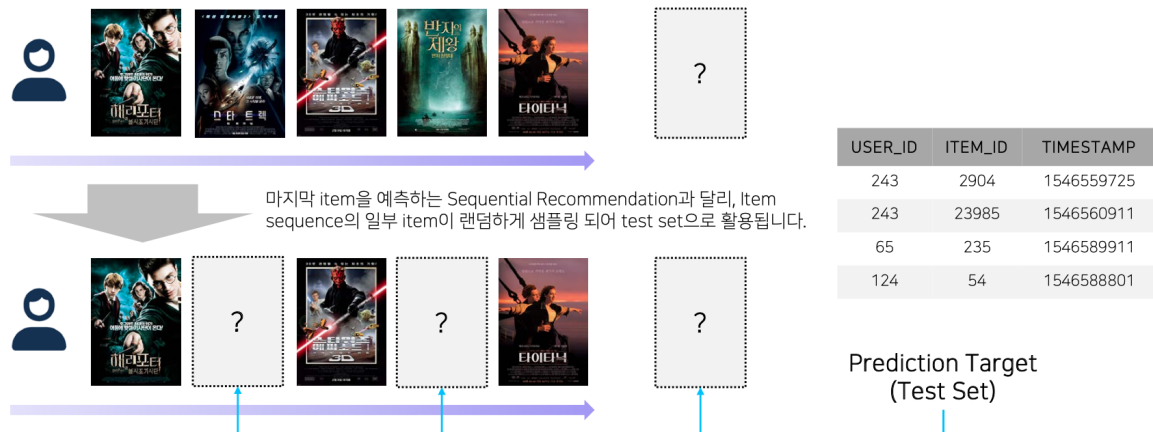


# Movie Recommendation Wrap-up Report

RecSys-2조(R\_AL\_SE) | 김수진, 김창영, 박승아, 전민서, 한대희, 한예본

## 프로젝트 요약

- 주제 : 사용자의 영화 시청 이력 데이터를 바탕으로 사용자가 다음에 시청할 영화 및 좋아할 영화를 예측
- 소개



- implicit feedback으로 전처리된 MovieLens 데이터 활용
- implicit feedback 기반의 sequential recommendation 시나리오를 바탕으로 사용자의 time-ordered sequence에서 일부 item이 누락된 상황을 상정
- 사용자의 순차적인 이력(timestamp)과 implicit feedback을 고려한다는 점에서, 추천시스템 입문 강의 등에서 자주 소개되는 explicit feedback 기반 협업 필터링 문제와 차별화
- Task : 사용자가 "시청한" 영화 중 마지막 영화를 포함하여 일부를 제거 한 후, 이를 예측하고, 그 중 맞춘 개수를 평가 점수로서 사용한다.

## 프로젝트 일정 및 협업 방식

- 프로젝트 일정
  - 1) 프로젝트 개발환경 구축
  - 2) EDA를 통한 데이터 구조 파악
  - 3) 가설 기반 모델링
  - 4) Ensemble(Hard Voting)을 통한 최종 결과물 제작

- 협업 방식

#### Github 기반 작업

- 이슈 기반 작업
- 데일리 스크럼/피어세션을 활용해 코드 리뷰 후 병합
- GitHub Kanban Board를 활용한 프로젝트 일정 관리

#### 노션 기반 실험일지 작성

- 실험 일지 작성을 통한 원활한 결과 공유 도모

**MovieRec 실험 일지**

#	Submit	Model	Public	Description	요지
1		Baseline	0.0324	code 처음 다룬 범용인 있는 sample_submission 파일	박승아   서울 산업공학과
4		Baseline 3SRec	0.0877	Pretrain 후 train에 inference	박승아   서울 산업공학과
5		Baseline SASRec	0.0810	Pretrain 없이 train에 inference	박승아   서울 산업공학과
6		Baseline SASRec	0.0754	미션4	김창영
10		Recbole EASE	0.1294	Quick Start, 0.8: 0.1: 0.1, weight 250, seed = 2020, infer = 로컬	한예본 민서 전
11		Recbole EASE	0.1593	Quick Start, 0.999: 0.001, weight 250, seed = 2020, infer = 로컬	민서 전 수진 김
12		Recbole EASE	0.1594	Quick Start, 1: 0: 0, weight 250, seed = 2020, infer = 로컬	한예본 수진 김
13		Recbole EASE	0.1594	Quick Start, 1: 0: 0, weight 250, seed = 2020, infer = 로컬	민서 전 수진 김
14		Recbole EASE	0.1594	Quick Start, 1: 0: 0, weight 500, seed = 2020, infer = 로컬	한예본 수진 김
15		Recbole EASE	0.1592	weight 1000, 1:0:0, seeds=2050, seed = 2020, infer = 로컬	한예본 민서 전
16		Baseline 3SRec	0.0844	inference할 때 10, 20, 30, 40, 50번째 item embedding 사용	
26		MultiVAE	0.1261	epochs=10, log_interval=100, lr=0.001, wd=0.0	DAEHHEE
28		MultiVAE	0.1381	epochs=50, log_interval=100, lr=0.001, wd=0.0	DAEHHEE
31		EASE	0.1599	lambda = 300, prefer_rating = 1	수진 김
32, 33		EASE	0.16	lambda = 300 && 500, prefer_rating = 0.9	수진 김
38		Recbole ADMMSLM	0.1475		민서 전
39		Hard Voting	0.1607	28, 32, 38	박승아   서울 산업공학과
40		Recbole ADMMSLM	0.1569	0.999: 0.001: 0	
41			0.0880		김창영
42		Recbole LightGCN	0.0888	LightGCN, [0.9999, 0.0001, 0], [lr@10, 0.1267], [lr@10, 0.0524], [lr@10, 0.0596], [lr@10, 0.1267], [precision@10, 0.0127]	
43		AutoRec	0.1313	num_epochs 50, batch_size: 256, lr: 0.005	DAEHHEE
44		AutoRec	0.1362	num_epochs 200, batch_size: 256, lr: 0.005	DAEHHEE
45		Hard Voting	0.16	output_file = ['J22_EASE10.csv', 'EASE20.csv', 'J40_ADMMSLM10.csv', 'ADMMSLM20.csv', 'J28_MultiVAE10.csv', 'MultiVAE20.csv', 'JFM20.csv', 'JALS20.csv']	

## 프로젝트 팀 구성 및 역할

전체 : 데이터 EDA

김수진 : Baseline 기반 3SRec/SASRec/BERT4Rec 실험, EASE/VASP baseline 구현, Recbole Framework 코드 분석 및 inference

김창영 : NBCF, ALS 구현, CAR모델 학습 템플릿 작성 및 실험

박승아 : RecBole\_GNN 기반 그래프 모델 실험, 앙상블코드 작성

전민서 : RecBole 기반 모델 실험 및 베이스라인 코드 작성, Bert4Rec\_Random\_query 구현, Negative Sampling

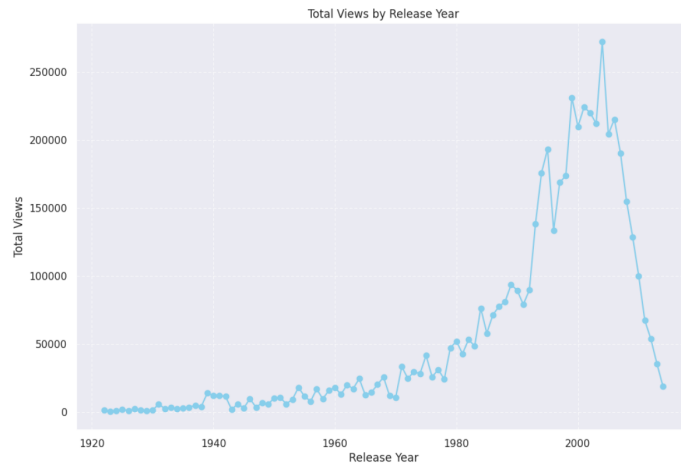
한대희 : Negative Sampling, DeepFM, AutoRec, MultiVAE

한예본 : RecBole 기반 모델 실험, BERT4Rec, Negative Sampling, SASRec/BERT4Rec query 방식 수정 시도

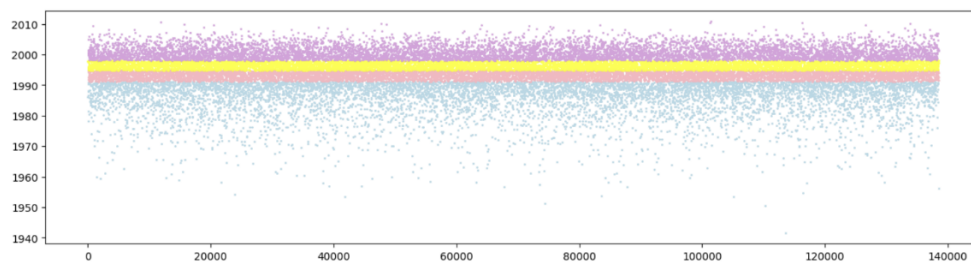
## 프로젝트 수행 결과

### 1. Data EDA

- 연도별 영화 출시 추세



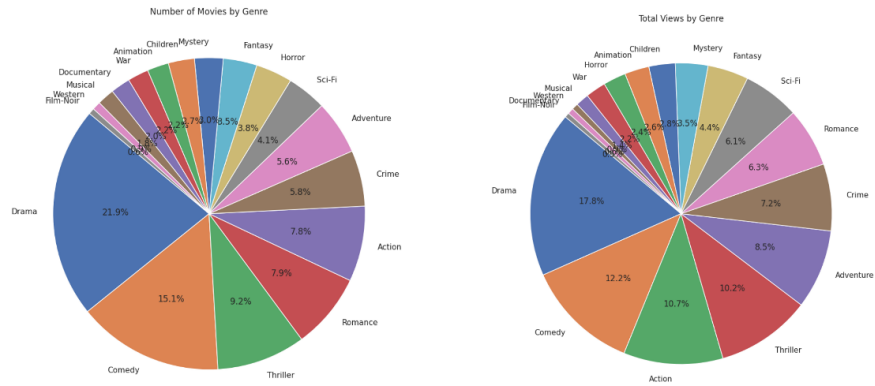
- 유저별 상호작용한 영화들의 평균 개봉 연도



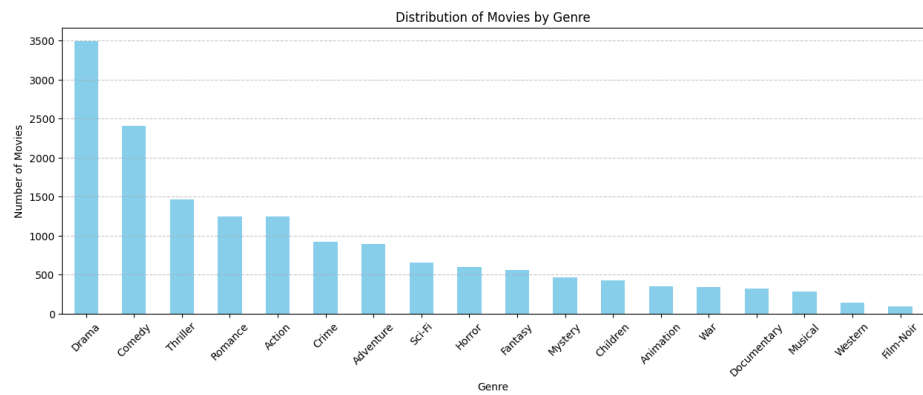
- 각 연도별 장르의 출시 회수를 시각화



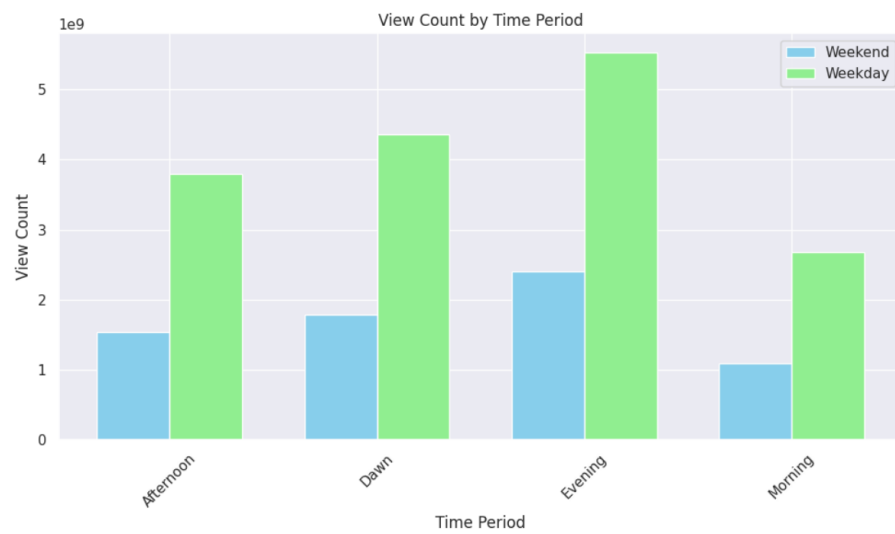
- 영화별 장르의 출시 비율/시청횟수별 장르의 비율 시각화



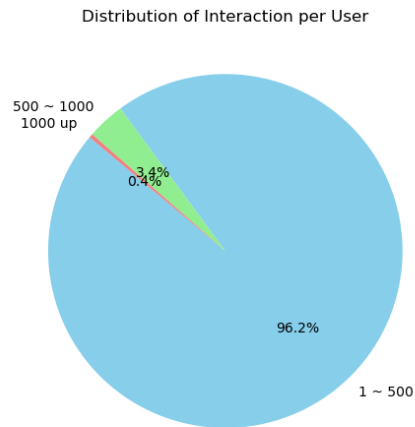
- 각 장르별 영화 분포 시각화



- 유저-아이템 상호작용 시간 구간별 시각화



- 유저별 아이템 상호작용 비율 시각화



## 2. 실험 모델

### • 이웃기반CF

유저별 아이템 상호작용을 vector로 보고 각 유저간의 코사인 유사도를 계산하여 가장 유사한 유저가 시청한 아이템 추천

### • 행렬분해기반

**SVD:** 유저-아이템 행렬을  $U, \Sigma, V^T$ 로 분해후 축소된 데이터를 활용하여 해당 사용자가 상호작용하지 않은 영화의 implicit을 예측하는 방식

**ALS:** 유저행렬과 아이템 행렬을 차례대로 최적화, 대용량 데이터 병렬처리로 빠른 학습 속도가 장점

### • Context-Aware

**FFM:** 특성이 속한 여러 개의 필드를 latent factor를 고려하여 특성 간의 상호작용을 학습

**DeepFM :** FM과 Deep 네트워크가 병렬로 결합된 구조로, FM은 낮은 차원의 특징 상호 작용을 모델링하고, Deep 네트워크는 고차원 특징 상호 작용과 비선형성을 포착합니다.

- 사용자가 시청한 영화를 긍정적 사례(Positive instances)로 간주하고, 해당 영화에 대한 평가를 1로 처리했습니다. 그리고 사용자가 시청하지 않은 영화에 대해서는 각 사용자별로 부정적 사례(Negative instances)를 샘플링하여 모델링을 수행했습니다.

### • Sequential

**SASRec:** Self-Attention을 적용하여 long-term과 short-term dynamics를 효과적으로 반영, 병렬적인 계산

**Bert4Rec:** 양방향 Self-Attention을 적용하여 과거와 미래 정보를 모두 학습에 사용

- 랜덤하게 sequence 내부의 데이터를 마스킹 한 후 해당 masking을 맞추는 작업을 수행하고 최종적으로는 마지막 sequence를 query하는 방식으로 성능을 판별하는데 이번 대회 Task에서는 last query가 아닌, 랜덤한 지점을 query 하는 대회였기 때문에 Task와 다르지 않았나 생각이 든다.

- 위 현상을 개선하기 위해서는 모델 inference 과정에서 마지막 위치의 sequence를 query하는것이 아닌 중간에 일정한 지점에 대해 query하는 방식으로 진행했어야 한다고 생각한다.

## • 선형모델

**EASE:** hidden layer를 사용하지 않고(shallow), closed-form의 목적함수로 autoencoder처럼 입력을 출력으로 복원시키는 과정을 통해 학습하는 선형모델

- Recall@10의 private score가 0.1602로, 단일 모델로는 가장 좋은 성능을 보였다.
- EASE 모델은  $X'X$ 의 역행렬로 답을 구하는데, 그러면  $Ux$ 였던 행렬의 크기가  $Ix$ 로 바뀌게 된다. 즉, item의 수에 비해 user의 수가 충분히 많다면,  $X$ 보다  $X'X$ 가 더 dense한 행렬이다.
- 이번 대회의 데이터는 유저 31360명과 아이템 6807개, 그리고 sparsity가 약 97.5%이었는데, 이렇게 되면  $X$ 보다  $X'X$ 가 더 dense한 행렬이 되어, 데이터의 sparsity를 어느 정도 보완해주었던 것으로 보인다.

**ADMM SLIM:** 아이템 간의 유사도를 계산하는 SLIM 모델을 최적화하기 위해 ADMM(제약 조건이 있는 최적화 문제를 해결하기 위해 부분 문제를 번갈아가며 해결하는 수치 계산 알고리즘)을 활용

- Recall@10의 private score가 0.1552로 EASE 다음으로 좋은 성능을 보였다.
- ADMM을 활용하여 최적화 문제를 푸는 과정에서 아이템 간의 유사도를 계산하고, 정규화와 제약 조건을 조절하여 추천 정확도를 개선한다. 이를 통해 사용자의 수와 독립적인 훈련 시간으로 훈련시간을 줄이고, 대규모 사용자 데이터에 효과적으로 확장할 수 있는 장점을 가지고 있다.
- EASE 모델과 마찬가지로 간단한 알고리즘을 통해 해를 구하는 방식이며, SLIM 모델의 경우 sparsity가 어느 정도 보장되어 있을 때 더 좋은 성능을 보인다는 점에서, 이번 대회 데이터에서 좋은 성능을 보인 것으로 추측한다.

## • Graph

**SimGCL:** 사용자-아이템 유사성과 그래프 구조를 사용하는 contrastive learning 기반 모델

- 실험 결과, valid set에 대해 0.0768의 recall@10 값을 보였다. 타 모델에 비해 저조한 성적을 보여 추천 결과는 제출하지 않았다.

**XSimGCL:** SimGCL을 향상시킨 모델로, SimGCL과는 달리 extreme similarity 기법을 사용하고 더 복잡한 모델 구조를 가지고 있어 더 높은 계산 비용이 필요

- valid set에 대해 0.1939의 recall@10 값을 보이며 SimGCL의 결과를 크게 상회하였으나 public recall@10이 0.1280으로 나왔다. 학습 데이터 자체가 크기가 작아 이러한 문제가 발생하였을 수 있으며 학습 데이터의 크기에 비해 모델이 매우 복잡했을 가능성도 있다.

**LightGCL:** constrastive learning과 graph augmentation을 결합해 사용자-아이템 유사성을 효과적으로 학습하는 모델

- 높은 정확도와 효율적인 학습 속도를 가진 장점이 있으나 모델 구조가 복잡하다. valid set에 대해 0.1267이라는 recall@10 결과를 보였으나 public recall@10은 0.0888로 저조한 성적을 보였다. 과적합이 발생한 이유는 XSimGCL과 같으리라 판단하였다.

## • AutoEncoder

**Mult-VAE :** Variational Autoencoder(VAE)를 기반으로 한 협업 필터링(Collaborative filtering ) 모델이다. 다항 분포를 사용하여 사용자-아이템 상호 작용 데이터를 모델링하고, 잠재적인 사용자와 아이템 표현을 학습한다.

- argparse를 사용하여 설정한 하이퍼파라미터인 lr(learning rate), wd(weight decay), batch\_size 등을 조절하여 성능을 개선하였습니다. public recall@10이 0.1378 으로 어느정도 높은 성적을 보였다.

**RecVAE:** Mult-VAE에 새로운 prior distribution,  $\beta$ -vae 논문에서 활용한  $\beta$  hyperparameter, 학습과정에서 alternating update의 세 가지 주요 아이디어를 도입하였다.

- Recall@10 private score가 0.1383이었는데, hyperparameter tuning을 하였다면 더 좋은 성능을 보였을 것으로 예상된다.

**CDAE :** 원본 Interaction Matrix X 와, Reconstruct 된 매트릭스의 차이를 학습하는 모델로, 라그랑주 제약 조건을 이용하는 모델이다.

**VASP:** VAE와 EASE를 결합한 모델로 VAE의 비선형 표현 학습과 EASE의 효율적인 선형 모델링을 동시 활용

- VASP는 EASE와 VAE가 결합한 구조이기 때문에 단일 모델에 비해 복잡도가 증가해 학습해야 할 파라미터 수가 많아짐, 이에 맞는 데이터가 부족했고 파라미터의 최적화가 어려워 성능이 떨어졌다고 추측한다.
- EASE와 MultiVAE를 단순히 앙상블 했을 때는 모델 간의 다양성을 활용하여 성능이 향상되었으나 VASP와 같이 모델을 직접 결합하는 방식은 우리의 태스크에서 각 모델의 장점을 최대화하지 못했다고 생각된다.

### 3. 최종 모델 및 앙상블 결과

#### • 최종 모델 결과

Model	Public Recall@10	Private Recall@10	Model	Public Recall@10	Private Recall@10
NBCF	0.0401	0.0390	Mult-VAE	0.1378	0.1402
SVD	-	-	RecVAE	0.1362	0.1383
ALS	0.1255	0.1266	CDAE	0.1431	0.1450
FFM	0.0988	0.0992	VASP	0.1119	0.1141
DeepFM	-	-			
SASRec	0.0816	0.0733			
BERT4Rec	0.0877	0.0845			
EASE	0.1600	0.1602			
ADMMSLIM	0.1569	0.1552			
SimGCL	-	-			
XSimGCL	0.1280	0.1313			
LightGCL	0.0888	0.0909			

- 실험한 단일 모델들의 결과는 위 표와 같다. EASE가 0.1600의 recall@10을 보이며 가장 좋은 성능을 나타내었고 다음으로 ADMMSLIM이 0.1569로 뒤를 이었다. 또한 CDAE, Mult-VAE, RecVAE에서 좋은 결과를 얻어 이를 앙상블 후보로 활용하였다.

#### • 앙상블(Hard Voting) 결과







Ensemble	Public Recall@10	Private Recall@10
EASE10, EASE20, ADMMSLIM10, ADMMSLIM20, CDAE, MultVAE10, MultiVAE20, RecVAE10	0.1627	0.1619
EASE10, EASE20, ADMMSLIM10, ADMMSLIM20, MultVAE10, MultiVAE20, RecVAE10	0.1626	0.1629

- 앙상블은 Hard Voting 방식으로 하였다. 결과물 별 반영 비율을 다르게 조정할 수 있었으나 파일 이름 별로 1표씩 Voting하게끔 하였다.
- EASE, ADMMSLIM, CDAE, Mult-VAE, RecVAE 총 5개의 모델을 활용하였다. 이 중 EASE, ADMMSLIM, Mult-VAE는 Top10 뿐만 아니라 Top20도 활용하였는데 Top10에 랭크된 아이템들에 가중치를 주기 위함이었다. 또한, 결과값이 미묘히 작아 Top 10 추천에 랭크되지 않은 경우도 반영할 수 있

도록 Top20 내역도 사용하였다. Top20을 사용했을 때 recall@10 값이 저하된 경우는 Top20 리스트를 제외하였다(CDAE, RecVAE).

- **최종 순위**

- Public score 기준 0.1627로 3위였으나 Private score에서 0.1629를 얻어 최종 2위로 마무리하였다.

2 (1 ▲)	RecSys_02조	     	0.1629	58	5d
------------	------------	---	--------	----	----

#### 4. 자체 평가 의견

- **잘했던 점**

깃허브를 이용한 이슈기반 협업을 원활히 진행하였다.  
여러 모델들을 사용하였다.  
Config를 사용하여 코드 관리가 깔끔하게 되었다.

- **시도했으나 잘 되지 않았던 것들**

GNN 기반 모델에 side information을 넣으려 했으나 RecBolt 프레임워크 사용 미숙으로 적용하지 못하였다.  
side information을 고려한 모델들의 성능이 다소 아쉬웠다.

- **아쉬웠던 점**

RecBolt\_GNN에 sequential graph 모델도 있었는데 활용해보지 못해 아쉬움이 있다.  
CAR 모델에 EDA를 통해 얻은 결과들을 유의미한 성능개선으로 연결시키지 못한 부분이 아쉽다.  
하이퍼파라미터 튜닝을 적극적으로 하지 않은 점이 아쉽다.  
렉볼을 사용한 것은 좋았으나, 모델이 확실히 어떻게 동작 하는지 이해하지 못한 점은 아쉽다.  
Wandb를 사용하지 않았다.  
Sequential model을 대회 task의 특징을 고려하여 개선하고자 하였으나 다양한 아이디어에 비해 이를 모두 구현하지 못하여 좋은 성능을 내지 못하였다.

- **프로젝트를 통해 배운 점 또는 시사점**

Recbole이라는 좋은 추천시스템 라이브러리가 있다.  
추천시스템은 좋은 모델도 중요하지만, 데이터 셋에 맞는 모델을 선정하는 것이 제일 중요하다.



## 1. 학습 목표 달성 방법

- **팀과 개인의 학습 목표:** end-to-end 모델 구현, 태스크에 맞는 논리적인 모델 설계, github 활용 등 협업 및 소통 능력 향상
- **개인 학습 측면의 접근:** Baseline 분석 후 원하는 부분 수정 및 구현으로 다양한 실험 시도, 모델 설계부터 제출물 생성까지 프로젝트의 workflow 를 경험
- **공동 학습의 중요성:** 분업 후 맡은 부분 요약 정리 후 리뷰, 각자 모델 실험 후 실험 일지 업데이트로 원활한 소통

## 2. 모델 개선 방법

- Baseline BPR loss function 구현 및 튜닝 실험, Recbole Framework 를 활용해 좋은 성능을 내는 모델을 찾아 튜닝 시도, EASE, VASP Baseline 구현 후 실험, 우리에게 주어진 데이터와 태스크에 가장 적합한 모델이 EASE 였으며 단일 모델로 가장 좋은 성능을 냄. 사용자와 상호작용이 있을 때 값을 1 로 설정하지 않고 0.8~0.9 로 설정했을 때 성능 향상이 있었음

## 3. 달성한 결과와 깨달음

- 태스크에 맞는 모델이 있고 이 과정에서 문제 정의/가설 설정/접근 방식 설정이 필요한 역량을 깨달음, 모델 성능이 잘 나왔을 때 왜 그런지 분석하는 과정에서 수학적 역량이 중요함을 다시 한번 느낌
- 대회에서는 시간이 한정적이기 때문에 효율적인 업무 수행이 필요하고 RecBole 같은 Framework 를 활용하면 빠르게 진행할 수 있어 사용하는 방법을 익혀놓으면 도움이 됨, 해당 framework 로 EASE 모델의 성능이 좋음을 확인해서 따로 Baseline 을 구현해 사용자 선호도 점수를 다르게 적용하는 실험을 진행
- 협업할 때 각자 어떤 부분이 어떻게 진행되는지 기록하고 확인할 수 있어서 수월하게 진행할 수 있었음, 협업 시 다른 팀원이 나의 작업물을 참고할 때 잘 이해할 수 있도록 기록하는 습관이 필요하고 프로젝트를 통해 소통 능력을 키울 수 있었음

## 4. 마주한 한계와 아쉬웠던 점

- 실제 ML20M 에서 성능이 1 위인 VASP 가 우리 태스크에서도 좋은 성능을 낼 줄 알았는데 적용해보니 성능이 좋지 않았고 이 과정에서 태스크에 맞는 모델 분석의 필요성을 느낌, RecBole 에 customize model 을 만들 수 있는데 Framework 가 익숙하지 않아 활용하지 못한 부분이 아쉬움, Side info 가 들어간 모델들이 전반적으로 성능이 다 안 좋았는데 대회 진행 중에는 분석할 시간이 없어 해당 부분에 집중하지 못했음

## 5. 미래 프로젝트를 위한 새로운 시도

- RecBole framework 를 익혀서 다양한 custom model 을 구상해 실험을 진행해보고 싶음
- EDA, FE 를 통해 데이터를 분석하고 시사하는 바를 명확하게 파악한 뒤 사용하고자 하는 모델이 적합한지 정의하는 과정을 체계화하고 싶음

# 김창영 T6042

## 학습 목표

- Context-Aware Recommendation 모델 공부 및 활용해보기
- EDA를 통한 인사이트 도출과 이를 활용한 성능 개선
- Recbole 라이브러리 사용해보기

## 프로젝트에서 시도한 것들

이전 프로젝트와 동일하게 깃허브를 활용해서 협업을 진행했다. 문제 사항 혹은 앞으로 진행할 방향성, EDA 등을 이슈화 하여 이슈 기반으로 브랜치를 생성하여 작업 후 PR하여 최종적으로 팀원들의 코드 리뷰를 거쳐 메인브랜치에 병합하는 식으로 협업을 하였다.

### 1. EDA

연도별 영화 출시 추세를 꺾은선 그래프로 시각화, 각 연도별 장르의 출시 횟수를 히트맵으로 시각화, 전체 영화 장르별 비율과 시청회수별 장르의 비율을 시각화했다. 영화별 작가, 감독의 분포와 시청회수별 작가, 감독의 분포를 시각화하여 비교해 보았다. 평점을 준시간(UNIX형태로 되어있음)을 일반적인 yyyy-mm-dd형태로 변환하여 이를 기반으로 주간/주말, 새벽/오전/오후/저녁별로 시각화를 해보았다.

### 2. 모델링 및 ML pipeline(+WandB 적용) 설계

팀내에서 미션코드5(Bert4Rec) 분석을 맡아서 간단한 논문 리뷰와 제공된 코드를 바탕으로 모듈화 및 학습 파이프라인을 설계했다.

추천대회이기때문에 기초적인 모델인 이웃기반협업필터링(NBCF)을 설계해보았다. 코사인 유사도 기준으로 유저와 가장 비슷한 유저를 탐색해서 추천하게끔 구현했다.

또한 행렬분해기반 추천모델인 ALS기반 MF를 구현했다.

EDA를 통해 확인한 context 정보를 Context-Aware Recommendation(CAR)모델에 활용하기 위해 FM, FFM, DeepFM 모델을 구현해보고 모듈화하여 학습 파이프라인을 설계하여 실험을 진행했다.

### 3. Recbole 사용

Recbole라이브러리를 활용해서 다양한 추천모델들을 경험해보았다.

## 느낌 점

지금까지는 모델기반추천시스템만을 사용해왔는데 이번엔 이웃기반협업필터링 즉, 메모리기반추천을 적용해보았다. 실제 적용 후 모든 유저의 추천결과를 뽑는데 30시간가까이 걸렸다. 메모리기반 추천의 한계인 많은 연산량에 따른 오랜 추천시간을 느껴볼 수 있었으며, 왜 모델기반추천이 각광받는지 느껴볼 수 있었다.

추천결과를 뽑아내는데 메모리기반 추천은 당연히 오래걸리지만, 모델기반추천모델들도 결과 추론시 5분이상의 시간이 걸렸다. 이를 통해 Candidate Retriever 모델을 통해 한 번 후보군을 필터링하고 Ranking모델로 정렬하는 2 stage 모델을 적용해보면 어떨까라는 생각이 든다.

이전 프로젝트에서 사용해본 적 없는 모델들을 적용해볼 수 있는 좋은 기회였다. 특히 이번 대회는 s3Rec 베이스라인만 주어졌기 때문에 다른 모델들은 직접 구현해야했다. 바닥부터 추론까지 end-to-end로 파이프라인을 설계하는 능력의 중요성을 느꼈다.

## 한계/교훈을 바탕으로 다음 프로젝트에서 시도해 볼 내용

CAR모델에 EDA를 바탕으로 진행한 전처리를 통해 성능 개선을 시도해봤지만 CAR 모델 자체가 전반적으로 높은 성능을 보이지 않았다. 이번 대회는 AE계열의 모델들이 높은 성능을 보였다. 마스터님께서 이번대회 데이터가 AE에서 학습하는 representation과 잘 맞기 때문이라고 말씀하셨다. 추후 CAR모델 논문에서 사용된 데이터셋과 이번대회 데이터셋을 비교해보면 좋을 것 같다.

### 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

저번 Deep Knowledge Tracing 프로젝트와는 달리, 이번 Movie Recommendation 프로젝트에서의 첫 번째 목표는 먼저 잘 구성된 추천 시스템 프레임워크를 활용해보는 것이었다. 이를 위해 PyTorch 기반의 RecBole\_GNN Github를 clone하여 Graph 기반의 모델의 종류를 접하고 신속히 학습, 추론하며 우리가 가진 데이터셋에 적합한 모델을 찾기 위해 노력하였다. 또한 해당 프레임워크를 팀원들도 쉽게 활용할 수 있도록 recbole\_gnn에 불필요한 내용들을 정리한 후 repository에 필요한 코드만 Push 하였다.

더불어 이번 프로젝트의 두 번째 목표는 논문과 그 논문의 근간이 되는 코드를 비교하며 이해하는 것이었다. Mult-VAE, SASRec, Bert4Rec 등의 모델을 강의와 논문을 통해 학습한 후, 이들의 이론과 코드를 함께 확인하며 아이디어를 구현해 나가는 방법을 익혔다.

### 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

이전과 비교해, 적극적으로 다른 연구자들이 공개한 추천시스템 프레임워크를 활용하였다. 코드 구현이 약해 항상 구현하고 디버깅하는 데 시간이 너무 오래 걸렸는데, 이러한 시간을 줄여 Movie Recommendation 프로젝트 수행과 최종 프로젝트 아이디어 구상을 동시에 진행할 수 있었다. 다양한 프레임워크를 통해 프로젝트를 생산성 있게 수행하였다.

### 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

그래프 기반 모델은 다른 딥러닝 모델들에 비해 아주 고전적인 모델만 접하였다. 따라서 최근에 등장한 그래프 기반 모델의 이론적 내용을 공부해 우리의 데이터셋에 맞는지 안 맞는지 확인하기에는 시간이 부족하였다. 그러므로 무작정 recbole\_gnn github에 있는 전체 모델을 모두 학습시켜 보았는데, 모델마다의 특징을 기억하고 유리한 모델을 선택하는 가설 기반의 효율적 과정을 거치지 못한 점이 아쉬웠다.

### 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

최종 프로젝트에서는 우리가 가진 데이터 셋의 특성을 파악하고 가설을 세워 모델을 만드는 과정을 거치고 싶다. 또한, 데이터 셋의 한계로 recbole\_gnn의 general model만을 활용해보았는데 sequential이나 social 모델을 써보는 경험을 해보고 싶다.

그리고 그동안 MAB 등의 방법은 학습만 진행하고 level1, level2 프로젝트에서 활용해본 적이 없는데 이를 활용하는 방법을 찾아 시도해보고 싶다.

## 전민서

- 가) 프로젝트 과정 속의 나의 학습 목표
  - A. TASK에 맞도록 기존 Model을 '수정'하는 방향으로 진행하기.
    - i. Task에 맞게 설계된 모델을 사용하는 것이 아닌, 나의 TASK에 맞도록.
  - B. 공개된 오픈소스를 이용하여, 최대한 많은 모델을 사용해보기.
    - i. 처음부터 공부 하는 것도 중요하지만, 어떠한 모델이 나의 상황에 맞는지 살펴보고, 성능이 좋았던 모델 특성의 교집합을 찾아 데이터셋의 특징을 예측.
  - C. 철저한 가설 - 검증 방식으로 이유가 있는 행동 하기.
  - D. 재사용 가능한 코드 작성
- 나) 학습 목표를 이루기 위해 노력한 점들
  - A. Recbole을 팀원들이 사용할 수 있도록 코드 작성
  - B. 모델을 Task에 맞도록 customizing
- 다) 배운 것들, 변화한 점
  - A. config파일을 사용하여 실험 관리를 용이하게 하였다.
  - B. 내가 풀려는 문제의 정확한 정의 및 해결 방안을 스스로 구상할 수 있게 되었다.
  - C. 논문을 읽고, 해당 모델이 어느 부분에 강점이 있으며, 실제 적용시 문제점을 어느정도 예측할 수 있게 되었다.
  - D. Implicit Feedback의 단점과 한계점을 알 수 있게 되었다.
  - E. Negative Sampling의 방식 및 주의점을 알게 되었다.
    - i. 적절한 Negative Sampling은 성능을 향상시키지만, 과하거나 부적절한 방식은 역효과를 낸다.
    - ii. 이미, 데이터셋의 분포에 따른 Negative sampling방법론들이 많이 존재한다.
  - F. 데이터셋의 라벨 분포에 따른 모델 성능차이를 이해 하였다.
    - i. Negative Feedback과 Positive Feedback유무에 따른 성능 차이 체험
  - G. sequence모델(Bert4Rec)이라도, 어느 부분에서 Query하는지에 따라 성능이 다르다.
  - H. RecSys는, NLP와 CV에 비해 훨씬 데이터셋의 영향력이 큰 분야가 맞다고 생각한다. 해당 분야들처럼 '성능이 좋은 모델'이 아닌 '나의 상황에 맞고' and '성능이 좋은' 모델이 중요하다고 생각한다. 즉, 다른 분야에 비해 사용자의 모델링 실력이 더 강조되며, 딥러닝뿐만 아닌 강화학습, 고전 머신러닝, 수학적 기법등 많은 방식 중 어떤 방식이 나의 상황에 잘 맞을지는 그 누구도 모르는 것이라고 생각한다. 단순하게 요즘 '딥러닝'이 좋으니까, 라는 생각으로 해당 분야만 공부할 것이 아니라 추천시스템 전반을 아우르는 공부를 할 필요성을 느꼈다.
- 라) 마주친 한계
  - A. Recbole을 이용해서 성능이 좋은 모델을 찾으려다 보니, 모델을 정확히 이해하지 못하고 튜닝하는 불상사가 일어났다.
  - B. 프레임워크를 분석하는 과정에서 코드 이해를 명확하게 하지 못했다.
  - C. 성능이 좋은 모델을 발견했음에도 불구하고, 내가 원하는 방식으로 모델을 수정하려는 과정에서 나의 가설을 모델에 적용하는데 많은 어려움이 있었다.
  - D. 데이터를 다루는 과정에서, 과감하게 특정 부분을 제거하거나, augment시키는데 지식이 부족하다.
- 마) 다음 프로젝트에서 시도해볼 것
  - A. Product Serving 적용하기
  - B. Multi Armed Bandit 사용해보기

# Movie Recommendation Wrap Up Report

한대회\_T6179

## 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- Task 에 적합한 모델과 데이터 전처리를 위해 체계적인 EDA 를 수행하여 데이터셋을 심층적으로 이해하고, 관련된 인사이트를 공유하였습니다.
- 고전적인 추천 모델부터 최신 모델들 이해도를 더욱 향상시키기 위해 Dive into Deep Learning Recommender Systems 파트를 학습하고 다양한 모델들을 실험하였습니다.
- Notion 을 활용하여 각 실험의 결과와 결론을 정리하여 추후에 참고할 수 있도록 체계적으로 관리하였으며, 이를 팀원들과 공유함으로써 프로젝트의 효율성을 크게 향상시켰습니다.

## 나는 어떤 방식으로 모델을 개선했는가?

- 데이터는 유저의 평가 기록인 Implicit feedback data 로, 사용자가 시청한 영화를 Positive instances 로 간주했습니다. 이때, 각 유저가 시청하지 않은 영화를 Negative instances 로 취급하기 위해 Negative instances sampling 을 수행했습니다. 이 과정에서 유저가 시청하지 않은 영화를 단순히 랜덤으로 추출하는것이 아니라, 각 유저의 선호하는 영화 장르와 선호하지 않는 영화 장르를 고려하여 Negative 데이터를 샘플링하여 성능을 향상 시켰습니다.
- 또한 성능이 좋았던 EASE 모델을 활용하여 유저의 TopK 샘플링을 수행했습니다. 이 과정에서 예상 랭킹이 가장 낮은 후보들을 Negative 데이터로 처리하여 모델의 예측 성능을 향상시켰습니다.

## 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- 배치사이즈와 드롭아웃 레이트를 조절하여 개선한 Autoencoder 를 기반으로 한 Multi-VAE 모델을 앙상블하여 최종적으로 2 등을 차지하는 데 기여했습니다. 이 과정에서 단순히 top-10 후보들을 예측하는 것이 아니라, 성능이 좋았던 모델들을 기준으로 top-20 후보들을 생성하여 voting 앙상블을 진행하여 전체 성능을 향상시켰습니다.
- 이러한 경험을 통해 프로젝트 전체의 성공에 기여할 수 있다는 것을 깨달았으며, 프로젝트에서 팀원들과의 협력과 모델 개발 능력이 상당히 중요하다는 것을 명확히 이해하게 되었습니다.

## 마주한 한계 혹은 아쉬웠던 점은 무엇이며, 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- General 한 모델들의 성능은 어느정도 준수 하였지만, sequential 모델, side information 을 고려한 모델들의 성능이 다소 아쉬웠습니다. 다음 프로젝트 에서는 sequential 데이터와 side information 을 고려한 모델들에 대해 집중적으로 연구하고, 이를 효과적으로 활용하는 방법을 탐구할 것입니다. 또한 다양한 모델들을 비교하고 최적의 모델을 선택하는 과정에 더 많은 노력을 기울일 것입니다.
- Explicit feedback 기반 추천에 비교하여 Implicit feedback 기반 추천 에 대한 경험이 부족했었습니다. Level 3 프로젝트에서 Implicit feedback 데이터를 다루는 만큼 Implicit feedback 기반 추천에 대한 이해를 더 깊이 파고들고, 관련된 모델과 알고리즘에 대한 학습을 강화하여 효과적인 추천 시스템을 구축할 것입니다..

### [프로젝트 과정 속의 학습 목표]

- Recbole 라이브러리를 공부하고, 대회에서 활용할 수 있도록 구현해보자.
- BERT4Rec 논문을 공부하고, 이를 대회의 task에 맞게 구현해보자.
- General, sequential, context-aware 모델을 모두 활용하여 일반화 성능을 높이자.

### [학습 목표를 이루기 위해 노력한 점들]

#### - Recbole 라이브러리를 활용한 다양한 모델 실험

팀 내 개인 미션으로 Recbole 라이브러리를 맡게 되어 Recbole 라이브러리의 구조와 사용법을 정리하여 공유하였다. 이를 계기로 이번 대회에서 Recbole을 활용한 다양한 실험을 하는 것을 맡게 되었다. 대회 데이터를 Recbole에 맞게 수정하는 것부터, 직접 생성한 config를 적용하여 main.py에서 학습한 후, 제출 형식에 맞게 inference하는 코드를 만들었다. 대회 데이터로 Recbole을 활용할 수 있게 구현한 후에는 그 안의 다양한 모델들에 대해 공부하고, task에 맞는 모델들을 실험해보면서 더 빠르고 효율적으로 실험할 수 있었다.

#### - BERT4Rec 모델 구현

이번 대회는 sequence 내부에 랜덤하게 10개의 영화가 비어있고, 이를 맞추어야 했기 때문에 BERT4Rec의 학습 방법이 적합하다고 판단했다. 함께 BERT4Rec을 맡은 팀원과 논문 스터디를 한 뒤에 그 내용을 정리하고, 대회의 task를 고려하여 구현을 시도하였다. task에 맞는 결과를 만들기 위해 모델의 구조뿐만 아니라 실험 과정 전반의 내용에 집중하여 공부하고, 이를 적극 활용하는 경험을 해볼 수 있었다.

#### - 다양한 negative sampling 시도

Implicit data를 가지고 DeepFM과 같은 모델을 실험하기 위해서는 negative sampling이 필요함을 알게 되어 다양한 negative sampling의 방법을 알아보고, 이를 시도해보았다. 특히 가장 성능이 좋았던 모델에서 TopK를 반대로 뽑아보는 시도를 했을 때, 유의미한 결과가 나왔다는 점이 기억에 남는다.

#### - 깃허브 이슈 기반 협업

팀원들과 깃허브 활용에 대한 규칙을 정하고, 이를 지키면서 깃 사용에 익숙해질 수 있었다. 특히, 이슈를 생성하고, 칸반 보드를 활용하여 자신이 진행 상황과 앞으로의 계획을 공유하면서 더욱 효율적으로 프로젝트를 진행할 수 있었다.

### [프로젝트를 통해 배운 점과 변화한 점]

#### - 다양한 모델 공부

Recbole 라이브러리를 활용하면서 새로운 모델에 대해 많이 알 수 있었다. 특히 이번 대회에서 interaction만을 사용하는 general model이 좋은 성능을 보이면서, EASE와 ADMM SLIM과 같이 간단하지만 sparse한 데이터에 강한 모델을 알게 되고, 이를 활용하기 위해 공부하면서 배울 수 있었다.

#### - 협업을 통해 일의 효율과 동시의 나의 능력을 키울 수 있다는 점

깃허브를 통한 협업을 통해 프로젝트 진행 속도를 높인 것뿐만 아니라, 팀원들과 함께 모델을 공부하고, 구현하면서 혼자서 공부할 때보다 훨씬 빠르게 인공지능 지식을 습득할 수 있었다.

### [프로젝트 중 마주친 한계]

- BERT4Rec 모델 구현까지는 완성하였으나, task에 맞게 더 좋은 성능을 낼 수 있도록 수정하는 시간이 부족하였다. user마다 sequence의 길이가 달라 중간에 임의로 빠져있을 item을 추측할 때 어느 부분에서 inference해야 할지에 대한 고민을 충분히 하지 못했다.
- 더 적절한 negative sampling 방법을 찾지 못하였다.
- Recbole에서 wandb를 활용하지 못하고 대회 기간이 끝난 점이 아쉽다.

### [다음 프로젝트에서 시도해볼 것]

- 최종 프로젝트에서는 지난 3개의 대회형 프로젝트에서 배운 내용을 바탕으로 실제의 데이터를 활용하여 추천을 시도하자.
- Product Serving 기술을 공부하여 이를 활용할 수 있도록 하자.