Movie Recommendation Wrap-up Report

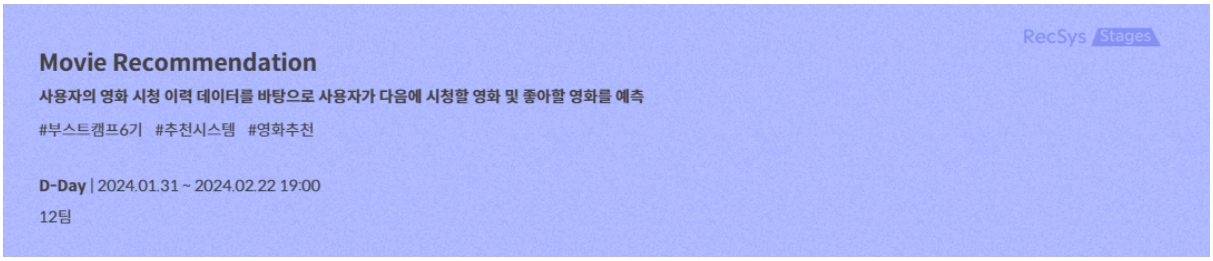
RecSys 4조 파이팅해야조

팀원 : 김세훈 김시윤 문찬우 배건우 이승준

1. 프로젝트 요약

본 프로젝트는 사용자의 영화 시청 이력을 기반으로 사용자가 다음에 시청할 영화 및 선호할 영화를 예측하는 추천 시스템 모델을 제작하는 것에 목적이 있다. 데이터를 다양한 유형으로 파악하고자 ADMMSLIM, EASE, CDAE, DeepFM, EASE, lightgcn, MultiDAE, MultiVAE, sasrec, 과 같은 여러 모델들을 사용하였다. 이 모델들의 장점을 취합하기 위해 이를 앙상블하여 최종적인 결과를 도출하였다. 본 프로젝트의 결과 각각의 모델에 대해 3:3:1:1:1:1:1의 weight을 두어 Hard Voting 앙상블을 진행하였다. 실험 결과 Recall@10 가 **0.1632**로 가장 성능이 높았으며 최종적으로 이를 제출하였다.

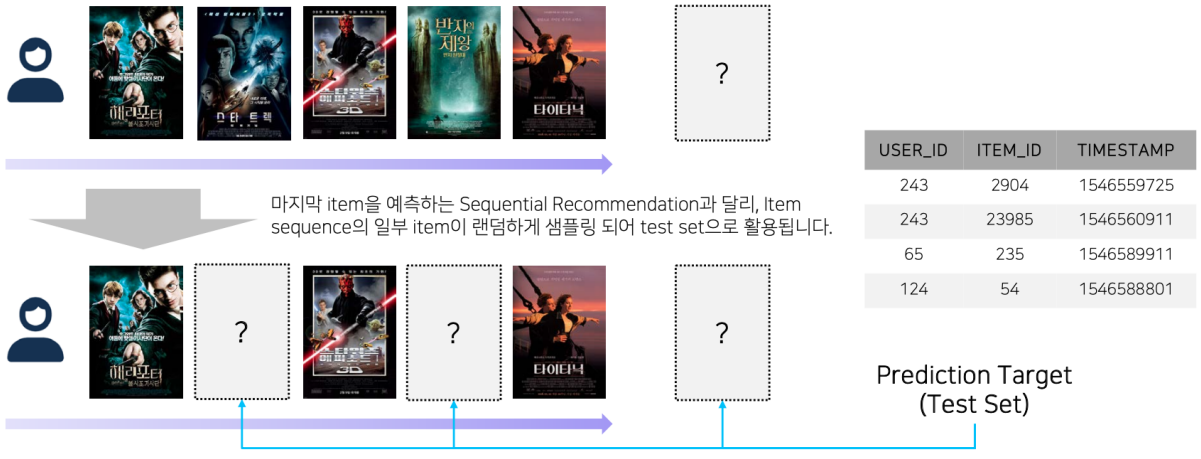
2. 프로젝트 개요



본 프로젝트는 사용자의 영화 시청 이력 데이터를 바탕으로 사용자가 다음에 시청할 영화 및 좋아할 영화를 인공지능을 통해 예측하는 프로젝트이다.

해당 프로젝트에서는 implicit feedback 기반의 sequential recommendation 시나리오를 바탕으로 사용자의 time-ordred sequence에서 일부 item이 누락(dropout)된 상황을 상정한다. 이는 sequence를 바탕으로 마지막 item만을 예측하는 sequential recommendation 시나리오와 비교하여, 보다 복잡하며 실제와 비슷한 상황을 가정하고 있다.

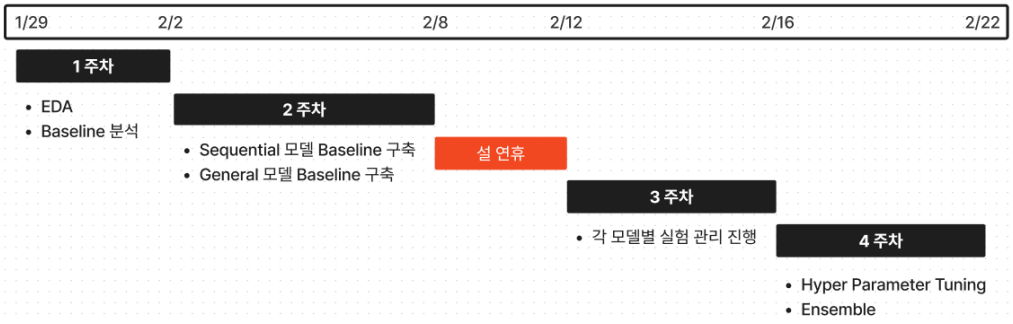
본 프로젝트를 통해 실제 현업에서 일어나는 데이터의 누락 등과 같은 유사한 환경에서 사용자의 행동을 예측하는 task를 수행할 수 있으며, 이를 통해 추천시스템의 다양한 상황에서 각 데이터와 모델을 어떤 관점에서 바라보아야 할 것인지 알 수 있는 계기가 될 것으로 기대된다. 추가적으로 이를 통해 플랫폼을 사용하는 유저에 대한 개인화된 추천시스템을 만듦으로써 비즈니스적인 가치를 증대할 수 있을 것으로 예상할 수 있다.



USER_ID	ITEM_ID	TIMESTAMP
243	2904	1546559725
243	23985	1546560911
65	235	1546589911
124	54	1546588801

3. 프로젝트 수행 절차 및 방법

3-1. 프로젝트 일정 및 타임라인



3-2. 서버 구성 및 환경

- 서버 정보 : AI Stages GPU V100 서버
- 버전 정보 : Python 3.10.13
- 패키지 정보

```
numpy==1.22.2
pandas==1.4.1
python-dateutil==2.8.2
pytz==2021.3
recbole==1.2.0
scipy==1.8.0
six==1.16.0
torch==1.10.2
tqdm==4.62.3
typing_extensions==4.1.1
```

3-3. 프로젝트 팀 구성 및 역할

공통	**EDA, Hyper Parameter Tuning, Git Management, Recbole**
김세훈	MultiDAE, MultiVAE Baseline 구축 및 실험, Ensemble
김시윤	RecBole 실험환경 세팅(기본 환경, inference), ease, lightgcn, recvae, deepfm 모델 실험, 앙상블 진행
문찬우	모델 성능 확인, 모델간의 유사도 확인 및 Ensemble
배건우	서버환경 구축, 베이스라인 구축
이승준	SASRec Baseline 구축 및 실험, RecBole을 활용한 EASE, ADMM-SLIM, CDAE, GRU4Rec 모델 실험

3-4. 협업 전략

Notion

- 프로젝트 진행 시 이슈 및 일정 공유
- 환경 세팅, 프로젝트와 관련하여 학습한 내용 공유

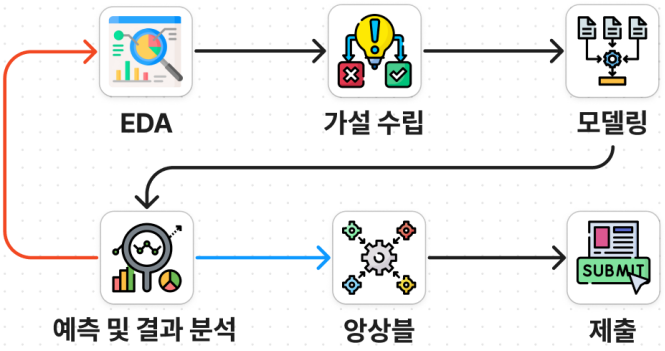
Github

- Baseline 코드 공유 및 코드 관리

Weights & Biases

- 각자 모델 실험 후 wandb를 통해 모델의 시각화 결과 공유

3-5. 프로젝트 파이프라인



4. 프로젝트 수행 결과

4-1. EDA

데이터셋의 구조는 다음과 같이 이루어져 있다.

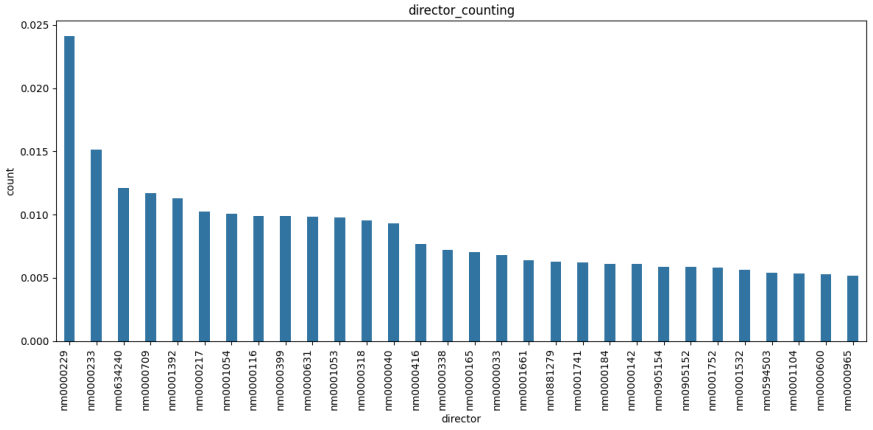
```
train
├── Ml_item2attributes.json
├── directors.tsv
├── genres.tsv
├── titles.tsv
├── train_ratings.csv
├── writers.tsv
└── years.tsv
```

train_ratings.csv

- 주 학습 데이터로 userid, itemid, timestamp로 구성되어있으며, 총 **5,154,471**의 행으로 이루어졌다.
- `userid` : 총 **31,360** 명의 유저의 userid가 존재
- `itemid` : 총 **6,807** 건의 영화의 itemid가 존재
- `timestamp` : 유저가 영화를 시청한 시간 이력
  - timestamp의 경우 영화를 시청한 시간이 아닌 movie lens data를 참고하였기 때문에 유저가 영화에 평점을 매긴 시간이라고 할 수 있다.

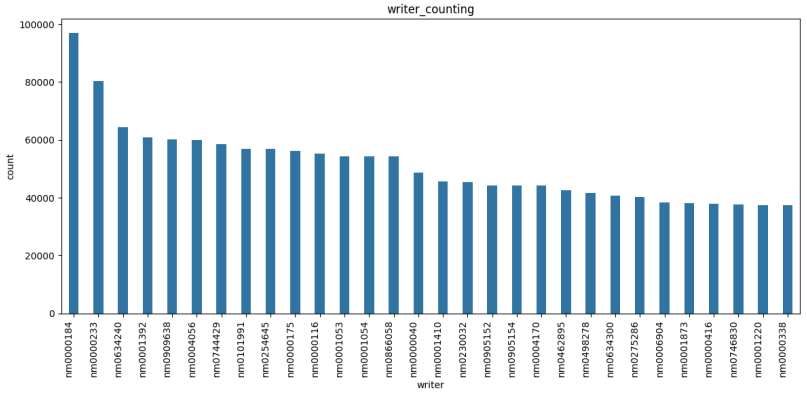
directors.tsv

- 영화별 감독에 대한 자료로, 총 **5,905**개의 행으로 이루어져있다.
- `item` : 영화의 itemid
- `director` : 영화 감독의 이름
  - nm0000005와 같이 암호화 처리
  - 각 영화감독이 만든 작품의 선택의 비율은 다음과 같다.



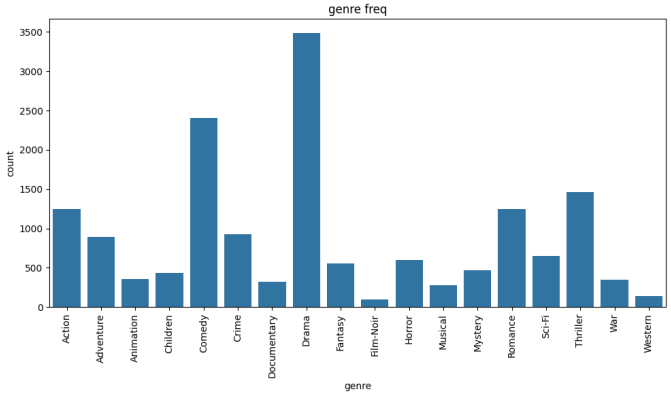
writers.tsv

- 영화별 작가에 대한 자료로, 총 **11,307**개의 행으로 이루어져있다.
- `item` : 영화의 itemid
- `writer` : 영화 작가의 이름
  - nm0000005와 같이 암호화 처리
  - 각 영화작가별 선택된 데이터는 다음과 같다.



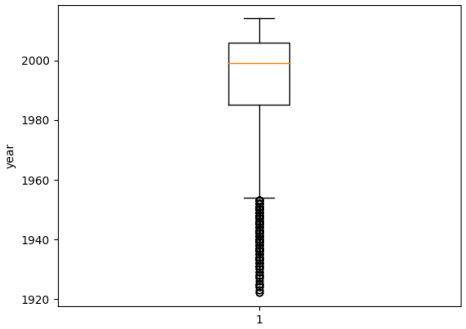
genres.tsv

- 영화의 장르 (한 영화에 여러 장르가 포함될 수 있음)에 대한 자료로, 총 **15,934**개의 행, 총 **18개의 장르**로 이루어져 있다.
- `item` : 영화의 itemid
- `genre` : 영화의 장르명
  - 장르별 frequency를 확인한 결과, drama가 가장 높은 것으로 나타났다.



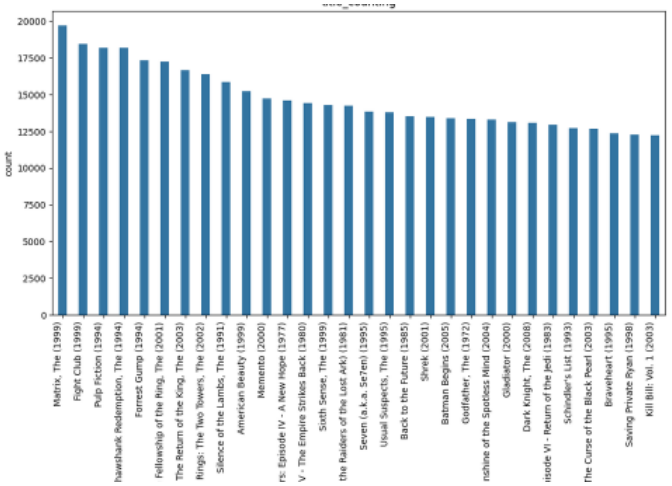
years.tsv

- 영화의 개봉년도에 대한 자료로, 총 **6,799**개의 행으로 이루어져있다.
- `item` : 영화의 itemid
- `year` : 영화의 개봉년도
  - 1985년부터 2006년까지 총 50%의 데이터가 있으며, 1985년 이하의 영화는 약 25%인 것으로 나타났다.



titles.tsv

- 영화의 제목에 대한 자료로, 총 **6,807**개의 행으로 이루어져 있다.
- `item` : 영화의 itemid
- `title` : 영화의 타이틀
  - 각 영화의 타이틀 별 frequency는 다음과 같다.



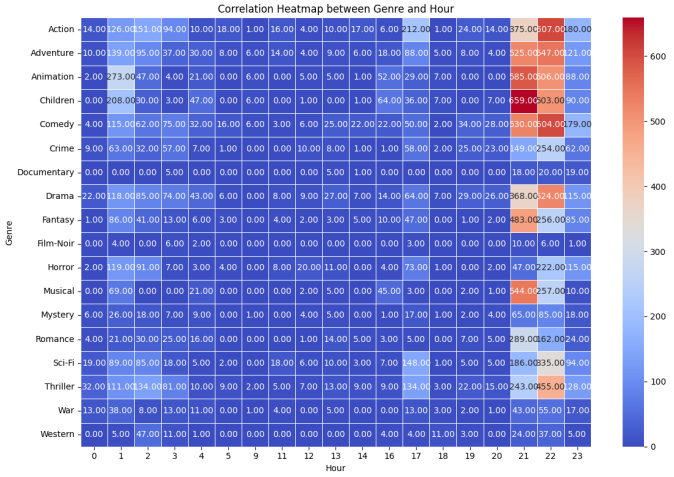
Ml_item2attributes.json

- 전처리에 의해 생성된 데이터로 item과 genre를 매핑한 후 다음과 같이 json형식으로 처리

```
{
   "1":[8,12,13,5,9],
   "2":[8,13,9],
   "3":[5,6],
...
}
```

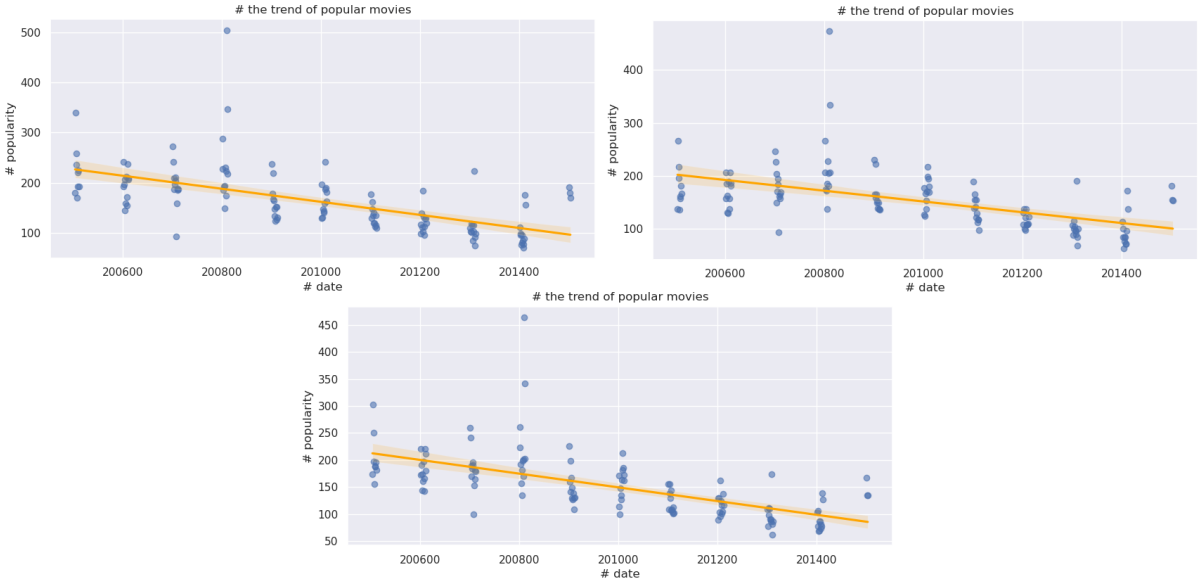
장르에 따른 유저의 영화 관람 경향성

- 각 장르별 영화를 관람하는 시간 등에 차이가 있을 것이다라는 가정을 바탕으로 분석을 진행
- 이에 대해 차이가 존재한다면 유저가 영화를 관람하고자 하는 시간을 고려했을 때 효과적으로 영화를 추천할 수 있을 것이라 기대
- 가설
  - h0 : 시간별 영화 상영 횟수 및 장르에 차이가 없을 것이다.
  - h1 : 시간별 영화 상영 횟수 및 장르에 차이가 있을 것이다.
- 분석 결과
  - 시간대에 따른 영화 관람 경향성에 차이가 뚜렷하게 존재하지 않음을 확인하였다.
  - 이는 aistage에서 기술한 것과 다르게 영화를 관람한 시간이 아니라 평점을 매긴 시간이기 때문에 본 가설 자체가 성립되지 않음을 확인할 수 있었다.



시간이 지남에 따른 인기 경향성

- 인기 있었던 영화가 시간이 지나도 여전히 인기 있는지 알아보기 위해 분석 진행
- 가설 : 당시 인기 있는 영화라도 시간이 지남에 따라 인기가 떨어질 것이다.
- 분석 결과 : 데이터 분석 결과 3편의 영화 모두 시간이 지남에 따라 인기도가 우하향으로 떨어지는 것을 알 수 있다.



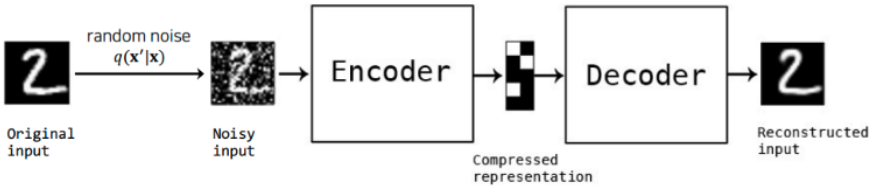
4-2. 모델링

General

데이터의 일반적인 특징을 파악하기 위해 General 모델에 대한 실험을 진행

MultiDAE

- Auto-Encoder 기반의 협업필터링이며, K-dimensional latent representation 샘플링 z와 noise를 추가한 뒤에 decoder로 복원하는 모델이다.
- MultiVAE와 차이점은 MultiDAE에서는 noise만 사용하여 decoder로 복원하는 모델이다.



선정 이유

- MultiVAE와 다르게 입력에 noise만 사용하여 좀 더 단순하게 상호작용을 예측할 것이다.

예측 결과

- Baseline MultiDAE 구축 및 결과 → **0.1344**
- Parameter Tuning으로 hidden dimension을 [600, 1800] 으로 학습한 결과 → **0.1371**

결과 분석

- 해당 데이터셋에서는 user, item을 입력으로 noise만 추가하였으나 좋은 성능을 보였다.

EASE

- EASE(Embarrassingly Shallow Autoencoders for Sparse Data)는 추천시스템의 협업 필터링과정에서 적은 은닉층을 가진 모델이 높은 추천 정확도를 가지고 있는 것에 입각하여 극단적으로 hidden layer를 제거한 선형모델이다.

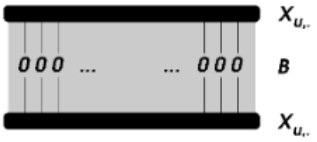


Figure 1: The self-similarity of each item is constrained to zero between the input and output layers.

선정 이유

- EASE 모델은 특히 sparse한 데이터를 처리하는데 특화되어있기 때문에 가지고 있는 data의 sparsity를 고려했을 때 효과적인 패턴에 대한 학습을 진행할 수 있을 것이라 판단하여 본 모델을 사용하였다.

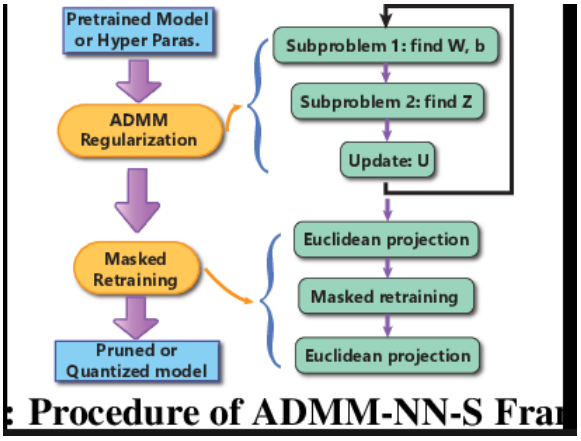
예측 결과

- Baseline구축 결과 → **0.1450**
- Parameter Tuning결과 → **0.1595**
- reg_weight : 500, CV : Leave one out

결과 분석

- 본 프로젝트에서는 학습에 사용된 데이터를 단순하다고 가정하였으며, 이에 맞게 최대한 단순한 상호작용을 파악하는 모델로써 본 모델을 사용하였다. 그 결과 가장 좋은 성능을 보였으며, 이는 곧 가설에 맞는 모델을 적절하게 사용하여 다음과 같은 성능을 보이는 것으로 해석할 수 있다.

ADMMSLIM



- ADMMSLIM은 EASE와 마찬가지로 데이터의 sparsity를 처리하기 적합한 모델로써 주로 사용된다고 할 수 있다. 최적화 문제에 있어 작은 부분으로 문제를 분할하는 ADMM과 아이템 간의 유사성을 나타내는 가중치 행렬을 학습하기 위한 방식인 SLIM의 결합으로 해당하는 제약 조건을 학습하는 방식으로 진행된다.

선정 이유

- 본 데이터의 크기를 대규모라고 가정하였으며, 이에 대해 대규모의 데이터를 작은 부분으로 각각 나누어 처리하는 방식이 최적해를 찾는 데 유용할 것이라는 가정하에 본 모델을 사용하였다.

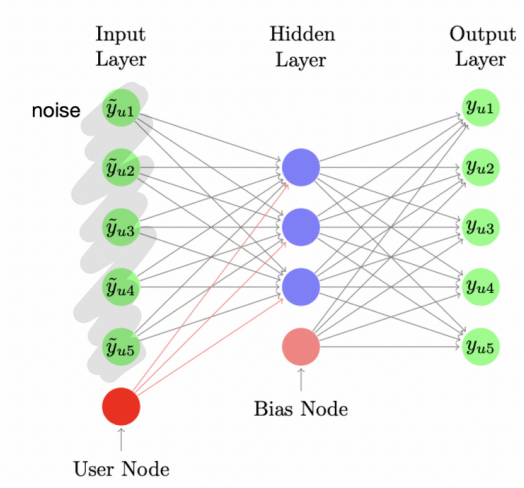
예측 결과

- 최종적으로 하이퍼파라미터 튜닝 후 성능은 Recall@10 기준 **0.1563**으로 나타났다. EASE와 유사한 수준으로 성능이 보여지는 것을 확인할 수 있었다.

결과 분석

- sparsity가 낮은 데이터의 특성에 맞는 모델이기 때문에 좋은 성능을 보이는 것으로 해석할 수 있다.

CDAE



선정 이유

- CDAE(Collaborative Denoising Auto-Encoders for Top-N Recommender Systems)는 DAE에 Collaborative Filtering을 적용한 모델이다.

- CDAE는 유저-아이템 간의 interaction을 아이템에 대한 rating이 아닌 preference를 학습시키는다는 점에서 DAE와 차이점을 보인다고 할 수 있다.
- 이를 통해 유저와 영화에 대한 interaction에서 각 유저의 해당 영화에 대한 선호도를 바탕으로 확률을 계산하는 것이 유용한 예측을 보일 것이라고 가정하여 본 모델을 사용하였다.

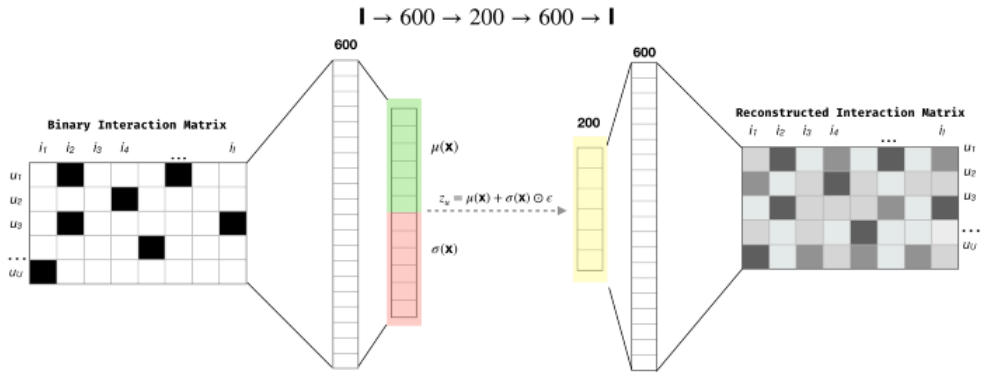
예측 결과

- 최종 성능은 Recall@10 기준 **0.1318**을 보였다.

결과 분석

- 본 모델의 결과 유저-아이템 간의 상호작용에 대한 선호도에 대한 확률을 확인할 수 있었다. 하지만 기존의 모델에서 확인하였던 상호작용보다 상대적으로 낮은 recall 점수를 기록하였는데, 이는 유저와 아이템간의 상호작용이 지나치게 단순하였기 때문에 모델에 대한 학습이 적절하게 진행되지 않았기에 다음과 같은 결과를 보이는 것으로 사료된다.

MultiVAE



선정 이유

- MultiVAE는 user-item interaction matrix를 multinomial distribution이라고 가정하고 학습시키는 모델로 VAE의 구조에 다음과 같은 Loss를 사용하는 것으로 정의된다.

$$\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u) \stackrel{c}{=} \sum_i x_{ui} \log \pi_i(\mathbf{z}_u).$$

- MultiVAE는 multi class classification을 위한 CE Loss를 사용하며, probability vector인 reconstruct x 에 대한 softmax결과 $\pi i(zu)$를 reconstruct x vector와 곱함으로써 확률값을 구하는 구조로 추천 시스템에서 top-N ranking에 적합한 모델이기 때문에 본 과제에서 유용하게 사용될 것으로 기대하였다.

예측 결과

- Baseline MultiVAE 구축 및 결과 → **0.1304**

결과 분석

- Parameter Tuning으로 뚜렷한 성능 향상을 보이지 않았는데, 이는 주어진 데이터의 조건을 종합하였을 때 학습을 하기에는 각 데이터의 구조가 단순하다고 판단했고 데이터의 특성이 multinomial distribution에 맞지 않는 것으로 해석할 수 있다.

RecVAE

선정 이유

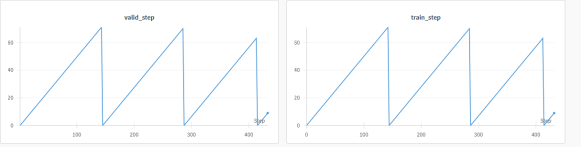
- RecVAE는 Autoencoder Multi VAE의 regularization의 구현 방식을 다음과 같이 변형하여 이전 epoch의 parameter를 고려하도록 하였다. 이를 통해 parameter smoothing의 효과 및 기존 모델의 과적합을 방지하기 위해 본 모델을 사용하고자 하였다.

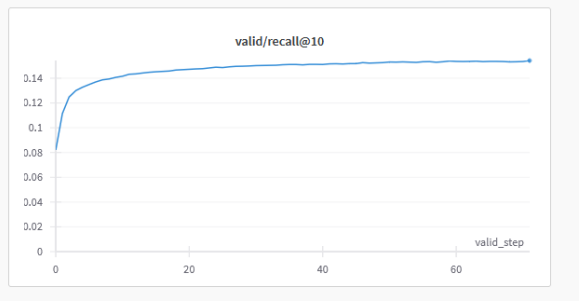
$$p(z|\phi_{\text{old}}, x) = \alpha\mathcal{N}(z|0, I) + (1-\alpha)q_{\phi_{\text{old}}}(z|x)$$

예측 결과

- 실험결과 recall@10 기준 **0.1507**(private), **0.1321**(public)으로 MultiDAE보다 높은 점수를 기록하는 것을 확인하였다.
- 이를 바탕으로 모델의 최적화를 진행한 결과는 **0.1356**로 나타으며 학습과정은 다음과 같다.

```
'hidden_dim': 600,
'latent_dim' : 250,
'dropout_rate' : 0.5,
'gamma' : 0.005,
'beta' : 0.2,
'not_alternating' : True,
'e_num_epochs' : 2,
'd_num_epochs' : 1,
'lr' : 0.0006036802931022732,
'batch_size' : 1024,
'num_epochs' : 300,
'num_workers' : 2,
'valid_samples' : 10,
'seed' : 42,
```





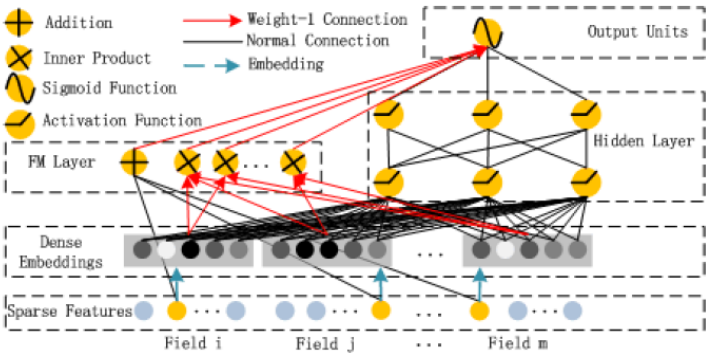
결과 분석

- RecVAE는 기존의 Autoencoder에서 제기되는 과적합과 smoothing 문제를 효과적으로 해결한 모델로써 사용되었으며, 논문에 제시한 바와 같이 이러한 문제들을 해결하였기에 다른 AE모델보다 더 좋은 성능을 보이는 것으로 해석할 수 있다.

Context-Aware

유저-아이템 간의 상호작용 데이터 이외에도 아이템 데이터의 메타 데이터를 활용한 모델에 대한 실험을 진행

DeepFM



선정 이유

- 유저-아이템의 상호작용을 확인하는 FM과 일반적인 아이템 간의 상호작용을 확인하는 Wide모델을 동시에 고려하여 high order feature interaction 및 low order feature interaction을 모두 고려할 수 있는 모델의 특징을 이용하고자 하였다.

예측 결과

- 모델을 이용하여 실험한 결과, recall@10 기준 **0.0880**로 나타났다.

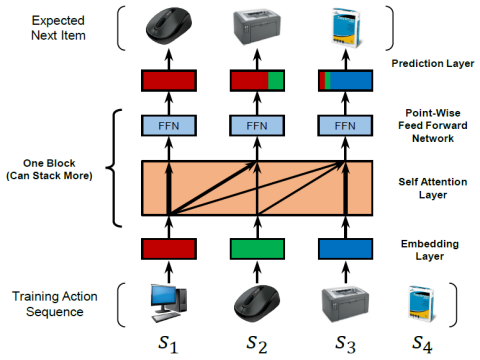
결과 분석

- 예상과 달리 유저와 아이템간의 추가적인 side information이 효율적으로 작동하지 못했다고 보여진다.

Sequence

시간이나 순서에 따라 연속적으로 나타나는 종류의 데이터를 처리하고 마지막 데이터를 예측하기 위해 실험 진행

SASRec



선정 이유

- 프로젝트의 test 데이터인 마지막 데이터를 예측하기 위해 데이터의 순차적인 정보를 반영한 모델에 대한 실험을 진행했다.
- Sequence 모델의 특징 중에서도 다음 결과는 이전 결과에 가장 영향을 많이 받을 수 있도록 Self-Attention Layer에서 구현된다.
- 본 대회에서 모델 성능 평가에 쓰이는 테스트 데이터의 조건 중 한 가지 조건인 마지막 결과를 도출하는 task에서 좋은 성능을 보일 것이라 예상하여 사용했다.

예측 결과

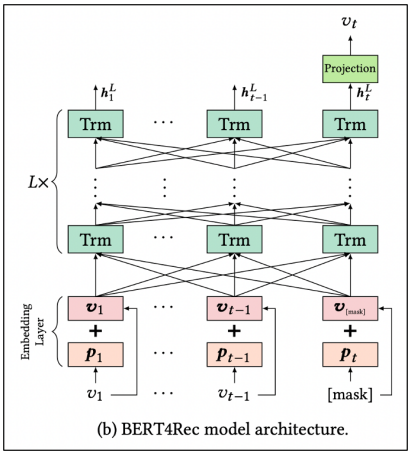
- 하이퍼파라미터 튜닝을 마친 모델의 최종 결과는 recall@10 기준 **0.949**로 나타났다.

결과 분석

- General model에 비해 현저히 떨어지는 성능인데 테스트 데이터로 사용된 마지막 데이터의 비중이 sequence 중간의 비중보다 낮기 때문인 것으로 해석할 수 있다.

BERT4Rec

- BERT4Rec은 다음 item을 예측하기 보다는 주변 context를 파악하는 cloze task를 통해 학습한다.
- cloze task는 유저의 행동 시퀀스에 대해 [Mask]라는 토큰을 사용하여 앞 뒤 정보로부터 [Mask]의 양방향성 정보를 파악할 수 있도록 하는 모델이다.



(b) BERT4Rec model architecture.

선정 이유

- SASRec이 단방향 예측에 있어 좋은 성능을 보이나 해당 대회에서는 양방향 예측이 더 유의미한 성능을 보일 것이라는 판단으로 BERT4Rec에 대한 실험을 진행했다.

예측 결과

- parameter tuning으로 Local 기준 **0.13 → 0.16** 으로 약 19%의 성능 향상이 있었다.
- 최종적인 public 제출 결과는 recall@10 기준 **0.0782**로 나타났다. 이는 모델의 학습 결과 overfitting이 나타난 것으로 해석할 수 있다.

결과 분석

- GRU4Rec이나 SASRec과 달리 유저의 행동 패턴을 바탕으로 squencial한 추천을 해주기 때문에 sequence 중간에 예측해야 하는 아이템에 대해서도 다른 sequence model에 비해 비교적 좋은 성능을 보였다고 해석할 수 있다.

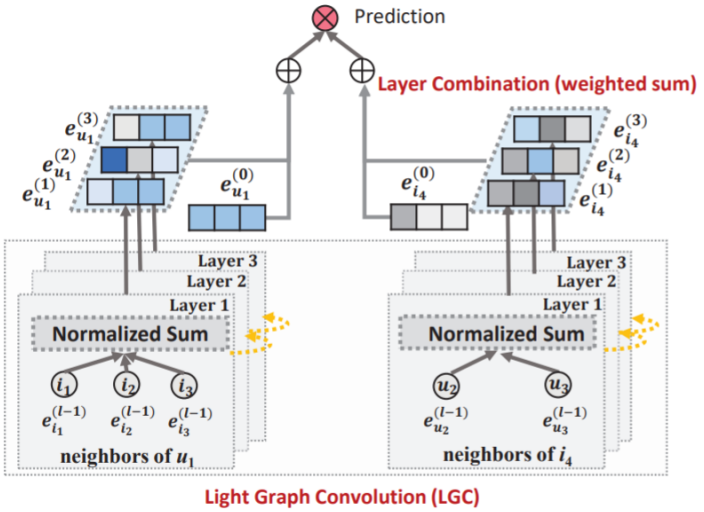
Graph

유저와 아이템이 상호작용하는 일반적인 특징을 파악하기 위해 실험을 진행

LightGCN

선정 이유

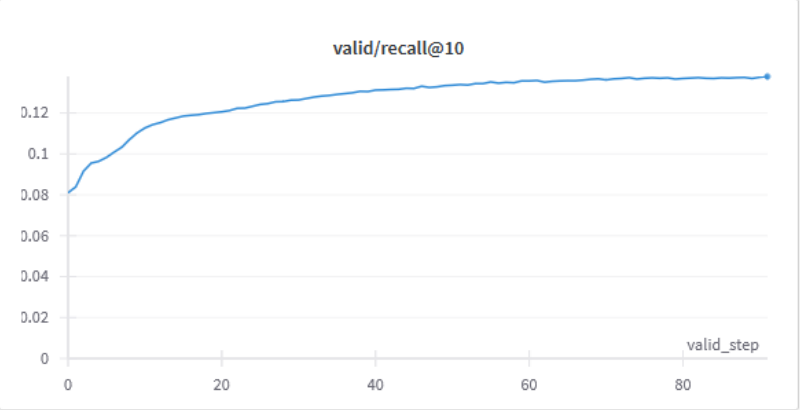
- LightGCN은 사용자와 아이템간의 상호작용 관계를 예측하는 모델로, userid, itemid 및 시간을 나타내는 time에 따른 영화를 볼 확률로써 나타내었다.



Light Graph Convolution (LGC)

예측 결과

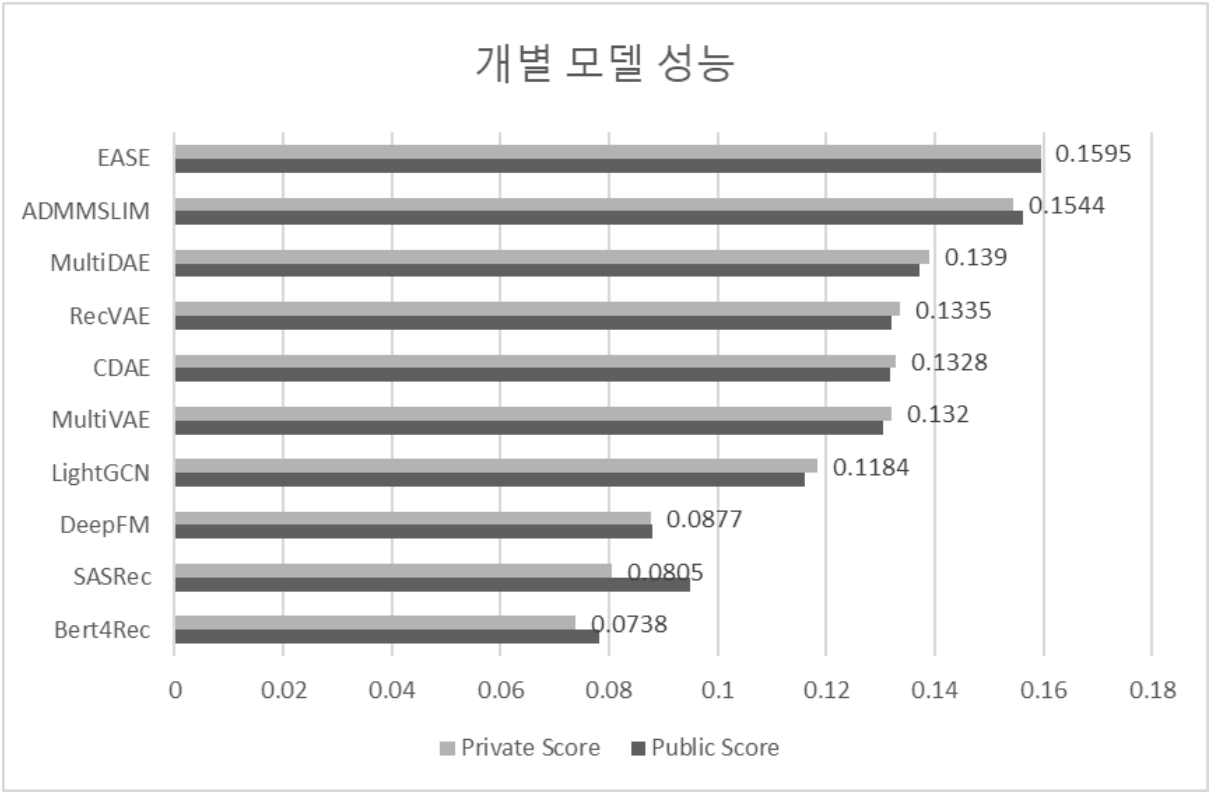
- baseline의 결과 aistages public **0.1160**으로 나타났다. baseline에 대해서는 성능의 테스트를 위해 epoch를 대폭줄여 학습시켰기 때문에 추가적인 epoch와 하이퍼파라미터에 대한 튜닝을 진행하였다. 그 결과 local 기준 0.1359에서 **0.1379**로 증가하는 것을 확인할 수 있었다.
- 최종결과 public에서는 **0.1180**로 확인할 수 있었다. 이는 private에서의 다소의 overfitting이 있기 때문에 다음과 같이 최종결과가 감소된 형태로 나타난 것으로 해석할 수 있다.



결과 분석

- 본 모델은 유저와 영화 item간의 상호작용을 그래프의 관계로써 확인하고자 하였다. 그 결과 유저와 아이템간의 뚜렷한 상호작용이 존재하지 않기 때문에 결과를 예측함에 있어 상대적으로 좋지 않음을 확인할 수 있었다. 또한 사용된 데이터의 시간의 특성이 학습을 진행하는데 있어 유의하게 작용하지 않음을 추가적으로 확인할 수 있었다.

4-3. 앙상블



Type	Model	Public Score	Private Score	
General	EASE	0.1595	0.1595	(reg_weight: 500.0, LS = test only)
	ADMMSLIM	0.1563	0.1544	CV (Leave one out)
	CDAE	0.1318	0.1328	
	MultiVAE	0.1304	0.1320	
	MultiDAE	0.1371	0.1390	
	RecVAE	0.1321	0.1335	
Context-Aware	DeepFM	0.0880	0.0877	
Sequence	SASRec	0.0949	0.0805	(batch size = 256)
	Bert4Rec	0.0782	0.0738	
Graph	LightGCN	0.1160	0.1184	(epoch=90, batch_size=2048, lr=0.001)

Hard Voting 앙상블

- 개별 모델의 성능을 바탕으로 임의의 가중치를 설정하여, 모델별로 item(ranking에 상관없이)에 가중치를 주었다.

  이 중 user별로 가장 높은 가중치를 가진 10개의 item을 선별하는 방식을 사용하였다.
- EASE, ADMMSLIM, CDAE, MultiVAE, MultiDAE, DeepFM, SASRec, LightGCN, Bert4Rec에 대한 가중치
  1. 9개의 모델에 대해 전부 같은 비율로 가중치를 부여 , Public Score: **0.1608** / Private Score: **0.1586**
  2. 높은 성능이 나온 EASE, ADMMSLIM 모델에 4의 가중치를 부여하고 나머지 모델에 1의 가중치를 부여 , Public Score: **0.162** / Private Score: **0.1619**
  3. EASE, ADMMSLIM 모델에 가중치를 낮춰 3의 가중치를 부여하고 나머지 모델에 1의 가중치를 부여 , Public Score: **0.1632** / Private Score: **0.1623**
- 모델 간의 item 추출을 기반으로 한 유사도

	admmslim	cdae	deepfm	ease	lightgcn	sasrec
admmslim.csv	1	42.84%	26.73%	69.89%	34.83%	11.07%
cdae.csv	42.84%	1	32.51%	48.13%	44.87%	14.01%
deepfm.csv	26.73%	32.51%	1	27.27%	31.64%	11.69%
ease.csv	69.89%	48.13%	27.27%	1	38.60%	12.03%
lightgcn.csv	34.83%	44.87%	31.64%	38.60%	1	13.57%
sasrec.csv	11.07%	14.01%	11.69%	12.03%	13.57%	1

- 높은 성능이 나온 EASE와 ADMMSLIM 두 모델의 예측을 바탕으로 예측하지 못한 아이템을 특성이 다른 모델들로부터 보완하는 형식으로 앙상블을 진행하였다.

4-4. 최종 제출

- Hard Voting 앙상블 : EASE**(3)**, ADMMSLIM**(3)**, CDAE**(1)**, MultiVAE**(1)**, MultiDAE**(1)**, DeepFM**(1)**, SASRec**(1)**, LightGCN**(1)**, Bert4Rec**(1)**

Public Score	Private Score
0.1632	0.1623

5. 자체 평가 의견

- **잘했던 점**
  - RecBole 라이브러리를 활용하여 데이터에 맞는 모델을 빠르게 탐색해볼 수 있었다.
  - Hard Voting ensemble을 통해 성능 향상을 확인할 수 있었다.
- **시도했으나 잘 되지 않았던 것들**
  - 기존에 존재하는 모델을 Custom하는 것을 시도했으나, 데이터에 대해서 제대로 파악하지 못해 성공하지 못했다.
  - Ranking을 기반으로 한 개선된 Hard Voting 방식의 ensemble을 구현하려고 했으나, 모델의 inference 부분에서 적용하지 못해 실패했다.
  - Wandb Sweep을 적용하려고 했으나 모델의 학습 시간이 너무 오래 걸리는 문제로 인해 제한된 시간 이내로 완료할 수가 없었기 때문에 적용하지 못했다.
- **아쉬웠던 점들**
  - 직접 모델링을 진행해보지 못하고 기존에 있던 모델 구조를 그대로 사용했던 것이 아쉽다.
- **프로젝트를 통해 배운 점 또는 시사점**
  - top-k를 구하기 위해 각 아이템에 대한 prediction을 모두 계산해야 하는 것은 복잡한 과정이 필요하다.
  - inference를 토대로 모델이 어떻게 예측했는지에 대한 EDA도 중요하다는 것을 느꼈다. 모델이 예측한 top-k가 확률이 낮은 경우가 있었는데 해당 데이터의 경우에는 어떤 특성을 가지고 있고, 이를 토대로 모델이 데이터의 어떤 부분을 예측하지 못하는지 확인해볼 필요가 있다.

6. 개인 회고

김시윤

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

- EDA 결과 주어진 자료를 가지고 feature engineering을 진행하는 것은 어렵다고 판단하였다. 따라서 주어진 데이터를 바라보는 관점을 모델로써 바라보기 위해 Recbole baseline을 구축하고 이에 대한 실험을 진행하였다. 그 결과, 여러 관점을 가진 모델들을 조합하여 최적의 조합을 찾아낼 수 있었다. 이를 통해 일반적으로는 데이터가 주어진 상황에 맞게 모델을 사용하지만 그렇지 않은 경우에는 어떻게 데이터를 바라보아야 하고, 그 결과를 해석하는가에 대해 다른 시각으로 생각해 봐야 할 때가 있음을 알게 되었다.
- 추가적으로 만든 코드와 출처등을 공유함으로써 내가 만든 코드와 더불어 더 발전된 형태로 실험을 진행할 수 있도록 함으로써 더 나은 결과를 얻을 수 있게 한 것 같다고 생각한다.

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

- 한 가지의 feature의 관점에만 국한하지 않고, 다양한 모델들을 살펴보며 각 모델의 특징을 고려하여 이를 최종적으로 적용하고자 하였다. 하나의 모델에 대한 튜닝을 진행하여서 0.11, 0.08등에 그쳤던 모델링의 결과를 다른 모델들과 앙상블 하였을 때 **0.16**으로 더 좋은 결과를 얻어내었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- EDA에서 몇 가지 가설을 제시했지만, 그것들이 모두 fit하지 않았다. 더 좋은 가설을 생각하지 못하고 안 나왔으니 더 이상 생각하지 않고 넘기는 안일한 생각을 한 것이 아쉬웠다. Recbole에 대한 베이스라인 코드를 만들긴 했지만 그것이 최적의 형태가 아니었던 것 같다. 내가 만든 코드를 모두가 효율적으로 사용하지 못했던 것이 아쉬웠던 것 같다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- baseline 코드를 설계할 때 class와 arg parser에 대한 사용법에 능숙해져서 간단한 구조로 모두가 효율적으로 사용할 수 있도록 코드를 만들어야겠다. EDA에 있어서도 조금 더 여유를 갖고 꼼꼼하게, 모든 데이터의 의미를 파악하도록 더 살펴보아야겠다.

김세훈

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

- **MultiVAE, MultiDAE Baseline 구축 :** 이번 프로젝트에서 Autoencoder를 활용한 협업 필터링 Baseline을 구축하였다. Data Preprocess, DataLoader, Model Load, Train, Inference 모두 순차적으로 구현하였으며, 이 과정에서 모델이 학습을 진행하는 전 과정을 역할에 맞게 클래스화 하여 알아보기 쉽게 구현하였고, 팀원들도 이를 활용해 함께 모델을 테스트 해볼 수 있었다.
- **Ensemble :** Hard Voting 앙상블에 대해 다음과 같은 아이디어를 구현했다. 만약 Top-k개에 대해 모델 A, B, C 각 3:2:2의 비율이 주어졌을 때, 먼저 A, B, C 모두 공통적으로 예측한 아이템을 채택하고, 나머지 아이템에 대해서는 A, B, C를 주어진 비율대로 앙상블을 실시했다. 그 결과, **0.1601**의 점수를 기록했으나 아쉽게도 점수가 낮아 아이디어가 채택되지 못했다. 점수가 낮았던 이유는 앙상블 데이터가 Top-k개라 하더라도 1~k등이 어떤 아이템인지 알 수 없어 잘못 앙상블 되었을 것으로 보인다.
- **Recbole 라이브러리 사용 :** 이 대회를 통해 Recbole 라이브러리를 활용하여 간단한 코드 작성만으로 모델을 효과적으로 학습할 수 있음을 확인할 수 있었다. 또한, Recbole을 이용하면 적합한 모델을 빠르고 손쉽게 탐색하여 실무에도 바로 적용할 수 있을 것으로 기대된다.

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

- MultiVAE, MultiDAE Baseline 구축 과정에서 모델의 파라미터를 arguments가 아닌 config 파일로 따로 저장하여 불러오는 방식으로 코드 관리 용이성이 높아졌으나, Weights & Biases의 Sweep을 이용한 실험관리 측면에서는 조금 불편하다는 단점이 있었다.
- Top-k라는 예측값을 뽑기 위한 코드를 작성하기 위해 모델 구조에 대한 이해도를 높일 수 있었다.
- 점수 예측과는 다르게 Top-k를 예측하기 위해서 보다 많은 모델이 필요하다는 것을 실험을 통해 알 수 있었으며, 그에 대해 팀원들과 의사 소통하여 여러 아이디어를 나눌 수 있었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 앙상블 진행 과정에서 모델이 예측한 아이템에 대해 순위를 매기지 않은 것이 아쉬웠다. 아이템에 대해 순위를 매겼었다면 좀 더 좋은 점수를 예측할 수 있었을 것이다.
- 이번 대회에서는 모델을 구현하는 것 보다 Recbole 라이브러리에 좀 더 많이 의존했다. 물론, 여러 모델을 적용해볼 수 있는 장점이 었지만, Sequence 모델을 많이 다루지 못한 것이 아쉽다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 최종 프로젝트에서 데이터 특성을 빠르게 파악하여 그에 맞는 적절한 모델을 선택해서 모델의 성능을 높일 수 있도록 할 것이다.
- 여러 모델을 앙상블 한 것이 성능이 잘 나오는 것처럼 팀원들과 앙상블하여 최종프로젝트에서 좋은 결과물을 만들 수 있도록 노력할 것이다.

문찬우

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

- 다른 팀원들에 비해 모델을 통한 직접적인 성능 향상은 이뤄내지 못했지만, 여러가지 앙상블 방법을 연구해보고 실험하여 이를 통해 최종적인 성능 향상을 이뤄냈다. 훈련 데이터의 전체 유저에 대한 각각의 10개의 추천을 바탕으로 Soft Voting을 구현하려 했으나, 여러 모델의 랭킹 및 probablity를 뽑는 과정에 어려움이 있어 여러 모델의 구조적인 성능과 각 유저별 item의 자카드 유사도를 구하는 방식을 통해, 모델에 가중치를 주어 Hard Voting 앙상블을 구현하여 Public 스코어 Recall@10 기준 0.1595에서 0.1632로 약 2.3%의 성능 향상을 이뤄냈다.

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

- 모델의 input부터 output까지 어떠한 구조로 데이터가 처리되어 결과가 나오는지 구체적인 흐름을 파악하고, 이를 코드를 통하여 직접 구현해 보았다. 여러가지 모델의 구조나 원리의 차이점을 활용하여 이를 하나로 합쳐보려는 노력을 하였고, 모델의 최종적인 결과 향상을 위해 이전 보다 다양한 앙상블 방법을 진행해 보았다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 서버의 환경 설정에 있어서 전보다, 시간 소모를 많이 하였다. 모델링에 있어서는 성능 위주의 모델을 찾는 것에만 집중하여, 다양한 시도를 해보지 못하였다. 여러가지 모델을 결과 부분에서 합치는 것이 아니라 학습 단계의 weight값을 갱신하는 부분에서 합쳐보려는 1차적인 목표를 구현하지 못한 것도 매우 아쉬운 것 중 하나이다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 좋은 모델을 만들기 위해서는 구체적인 EDA가 필요하다는 것을 다른 팀들의 발표를 보고 느꼈다. 너무 라이브러리나 패키지에만 의존하지 않고, 내가 사용하는 데이터에 최적화 된 모델을 직접 구현 할 수 있는 능력을 길러야겠다고 생각했다.

배건우

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

- 팀원들과 버전들을 맞추기 위하여 baseline base코드를 자동화 시켰습니다. 그래서 코드만 복붙하면 환경이 구성되고 github에서 버전충돌도 없게 하였습니다.
- 베이스 python script에 코드들을 자동화 시켜서 팀원들이 쉽게 작업할 수 있도록 하였습니다.

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

- 전 대회에서는 구현한 모델을 베이스 python script에 녹이지 못하고 Jupyter Notebook를 사용한 것이 너무 아쉬웠습니다. 그래서 이번 대회에서는 베이스 python script에 구현한 코드를 녹이는 것이 목표였는데 성공하였습니다. python script에 익숙해져서 큰 프로젝트를 효율적으로 다뤄볼 수 있었습니다.
- 전 대회에서 hyper parameter tunning에 시간을 많이 할애하여 시간이 너무 아깝다고 느꼈습니다. 그래서 이번에 hyper parameter tunning을 최적화 하는 도구를 활용하고 싶었는데 적용에 성공하였습니다. 이 시간을 아껴 모델을 더 많이 공부할 수 있었습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 제가 하나하나 python script를 짠 것이 아니고 베이스 python script를 수정하는 것에서 그친 것이 아쉬웠습니다.
- python script에 익숙하지 않다보니 코드 치는데 시간이 많이 흘렀습니다. 그래서 모델 성능을 높이는 것에는 많은 노력을 하지 못했습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 베이스 python script 제가 수정한 부분 말고도 비효율적인 부분이 많았습니다. 다음에는 새로 custom하게 코드를 구성하여 저희 입맛에 맞춰보려고 합니다.
- python script에 더욱 익숙해져 모델 성능을 높이는 것에도 노력을 하려고 합니다.

이승준

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

- recbole을 통해 다양한 모델을 실험해보면서 데이터에 잘 맞는 모델을 빠르게 탐색할 수 있었고 빠르게 가설을 세워볼 수 있었다. 이전 대회에서 모델 하나에 대해서 심도 있게 탐구한 나머지 성능에 대해서는 성과를 내지 못했었는데 빠르게 다양한 모델을 실험해보면서 성능 부분에서 좋은 성과를 낼 수 있었던 것 같다.
- SOTA 모델이 항상 좋은 것이 아니고, 복잡한 모델이라고 해서 항상 좋은 것이 아니라는 것도 알 수 있었다. 실제로 s3Rec이나 wide & deep과 같은 복잡한 모델은 좋은 성능을 내지 못했는데 이는 학습에 필요한 충분한 데이터가 존재했을 때 효과를 본다는 것을 알게 되었다. 학습할 파라미터가 많은 만큼 충분한 데이터가 필요하다는 것을 알게 되었고 데이터의 규모에 따라서 때로는 단순한 모델이 오히려 좋은 성능을 보인다는 것을 알게 되었다.

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

- SASRec의 Baseline을 구축해봤다. 밑바닥부터 한 것은 아니지만 데이터와 파라미터의 흐름에 따라서 실험하고자 하는 모델을 모듈화 하는 데 성공할 수 있었다. 비록 모듈화의 본래 목적인 모델을 커스텀하는 것은 실패했지만 모듈화를 통한 실험 관리와 코드 공유의 효율성에 대해서 느낄 수 있었다.
- 다양한 모델을 빠르게 시도하면서 주어진 데이터에 맞는 모델을 빠르게 탐색할 수 있었다. EDA를 통해 데이터의 특성을 파악하고 이를 통해 적절한 모델을 찾는 방법도 있지만, 다양한 모델을 빠르게 실험해보면서 데이터에 맞는 모델을 탐색하는 방법도 있다는 것을 알게 되었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 앙상블을 위한 코드를 작성하는 데 어려움이 있었다. Soft voting을 위한 방법을 알고는 있지만 구현하는 데 있어 부족함 때문에 코드를 완성하지 못한 것이 아쉽다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 코드를 구현하는 데 있어서 어려움을 겪었던 것 같다. EDA 및 시각화 부분에서 머릿속에 그려지는 부분을 빠르게 구현하는 데 어려움을 겪어서 pandas와 시각화하는 툴을 다루는 실력을 고도화 하기 위해 노력할 것이다. 그리고 파이썬 전반의 코드를 구현하는 실력을 키워서 베이스라인 구현 및 앙상블 구현 등 코드를 구현하는데 빠르게 실험하기 위한 실력을 갖출 것이다.
- 구축한 베이스라인을 바탕으로 코드를 고도화 해볼 것이다. 모델을 커스텀하는 것부터 밑바닥부터 베이스라인 구축하는 부분까지 전 과정을 스스로 구현해볼 것이다.