네이버 부스트캠프 AI Tech 5 기 Project 3 랩업리포트

RecSys Team 3 : 렉돌이
강찬미, 박동연, 서민석, 이준영, 주혜인

1. Team Wrap-up Report

1-1. 프로젝트 개요

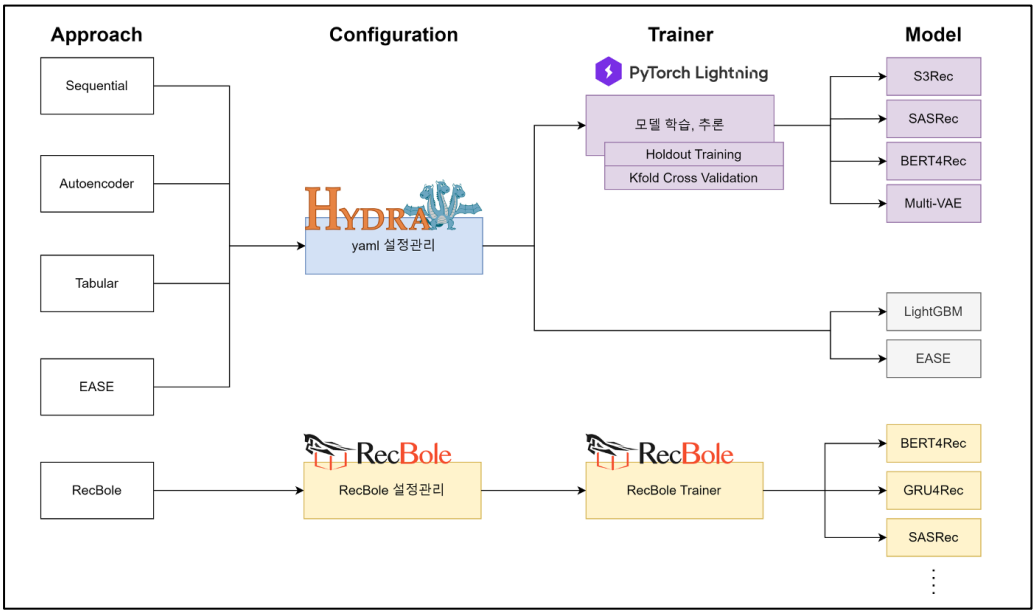
• 프로젝트 주제

- 일반적인 영화 추천 대회의 경우 유저의 영화 시청 이력 데이터를 바탕으로 다음에 시청할 영화 및 좋아할 영화를 예측함
- 본 대회에서는 실제와 비슷한 상황을 가정하기 위해 마지막 영화 뿐만 아니라 중간 시청 이력이 누락된 상황을 상정함
- 또한 여러가지 영화와 관련된 Side-information 이 존재하기 때문에 이것들을 효과적으로 활용하는 것이 중요함

• 활용 장비 및 재료

- **서버 스펙**: AI Stage GPU (Tesla V100-SXM2)
- **협업 툴**: Github / GatherTown / Zoom / Notion
- **기술 스택**: Python / Pytorch / Pytorch Lightning / Wandb / Hydra / Scikit-learn / RecBole / Ray Tune

• 프로젝트 구조



• 사용 데이터의 구조

- 유저-영화간 상호작용에 대한 train_ratings.csv 파일과 영화의 side information 해당하는 5 개 파일이 존재

파일명	설명
train_ratings.csv	사용자의 고유번호, 영화의 고유번호, 상호작용이 발생한 시간(초)으로 구성
titles.tsv	영화의 고유번호, 영화 제목으로 구성
years.tsv	영화의 고유번호, 영화 발매 년도로 구성
directors.tsv	영화의 고유번호, 감독의 고유번호로 구성
genres.tsv	영화의 고유번호, 장르로 구성
writers.tsv	영화의 고유번호, 작가의 고유번호로 구성

1-2. 프로젝트 팀 구성 및 역할

이름	역할
강찬미	Sequential 베이스라인 구축, BERT4Rec 구현
박동연	EDA 및 전처리, Tabular 베이스라인 구현, EASE 구현, Ensemble 구현
서민석	EDA, Autoencoder 베이스라인 구축, Multi-VAE 구현 및 HPO
이준영	Sequential 베이스라인 구축, S3Rec 구현
주혜인	EDA, RecBole + Ray Tune 베이스라인 구축

1-3. 프로젝트 수행 절차 및 방법

• 팀 목표

- **새로운 베이스라인 작성** : 제공된 베이스라인을 참고하여 우리 팀만의 자체적인 베이스라인 구축
- **스프린트 방식 도입** : 전반적인 계획 수립을 통한 체계적인 진행을 위해 스프린트 방식 도입
- **적극적인 Github 도입** : 이전보다 적극적인 issue 사용 및 PR 을 통한 코드 리뷰 활성화
- **다양한 Tool 사용** : Pytorch Lightning, Hydra, Github action, RecBole, Ray Tune 등과 같은 다양한 tool 경험

• 프로젝트 협업 문화

노션을 사용한 효율적인 협업

- **고민 해결 일지:** 고민 해결 일지에 프로젝트 관련하여 필요하다고 생각하는 것을 공유하고 해결하는 과정을 문서화하였음
- **칸반 보드**: 노션에 Jira 를 토대로 애자일 방법론을 적용하여 칸반보드를 제작해 협업을 수행하였음
- **스프린트 단위 자체 팀 회고 진행**: 스프린트 단위로 자체 팀 회고를 진행하여 매주 목표와 각 작업들의 완료 여부 및 계획 공유

적극적인 Git 활용

- **Git Convention**: commit 메시지, github flow 전략 도입
- **Pre-commit 활용**: black 포매터를 이용해 코드 스타일 통일
- **Github issue 활용**: 작업할 목록을 issue 에 정리
- **Github Action 을 이용한 자동 테스트**: pytest 를 이용한 모델 러닝 테스트

• **프로젝트 타임라인**

스프린트 1	스프린트 2	스프린트 3	스프린트 4	스프린트 5
5/30 ~ 6/6	6/6 ~ 6/10	6/11 ~ 6/14	6/15 ~ 6/19	6/19 ~ 6/22
프로젝트 초기 세팅				
EDA				
Sequential 베이스라인				
	데이터 전처리			
	Multi-VAE 모델 구현			
		Tabular 베이스라인		
		RecBole 베이스라인		
		BERT4Rec 모델 구현		
			EASE 모델 구현	
				하이퍼파라미터 튜닝
				Ensemble

- **Sprint 1 (5/30~6/6)** : 프로젝트 초기 세팅, EDA Sequential 베이스라인 구축
- **Sprint 2 (6/6~6/10)** : EDA, 데이터 전처리, Sequential 베이스라인 구축, Multi-VAE 모델 구현
- **Sprint 3 (6/11~6/14)** : Tabular, RecBole, Sequential 베이스라인 구축, BERT4Rec 모델 구현, Multi-VAE 모델 구현
- **Sprint 4 (6/15~6/19)** : RecBole 베이스라인 구축, EASE 모델 구현, BERT4Rec 모델 구현
- **Sprint 5 (6/19~6/22)** : Ensemble 구현 및 실험, 하이퍼파라미터 튜닝

1-4. 프로젝트 수행 결과

1. EDA

- 유저: 313,360 명/영화: 6807 개/총 데이터: 5,154,417
- 영화 제목(titles.tsv)의 데이터에서 제목 뒤에 개봉 연도가 숫자 형태로 기입되어 있음 -> 연도(years.tsv) 데이터와 거의 일치하는 것을 확인
- 학습 데이터의 연도 값에 결측치가 있음을 확인 (약 1800개)
- 유저들은 평균 164개의 영화를 평가하였고, 최소 16개, 최대 2912개를 평가하였음
- 가장 많이 본 장르는 Drama, Comedy, Action, Thriller

2. 데이터 전처리

① 연도 데이터의 결측치 채우기 (years.tsv)

- Side information 중에서 기존의 year.tsv를 그대로 사용하여 train_rating.tsv와 병합하였을 때, year.tsv에 없는 연도가 있어 약 1800개의 결측치가 생김
- 영화 제목 데이터(titles.tsv)에 연도가 기입되어 있는 것을 문자열 처리를 통해 추출하여 years.tsv에 있는 결측치를 채우도록 하였음
- 총 8개의 영화에 대한 연도 정보가 제공되지 않음으로써 생기던 1800개의 연도 결측치를, 위의 방법을 사용함으로써 연도 결측치를 0개로 줄일 수 있었음

② 대표 작가 및 대표 감독 선정하기 (writers.tsv, directors.tsv)

- 본 데이터에서는 하나의 영화에 대해서 여러 명의 작가와 여러 명의 감독 정보가 연결되어 있었음
- 모든 작가와 감독 정보를 사용한다면 데이터의 차원이 너무 커져 차원의 저주가 발생할 수 있음. 또한 모든 작가, 감독 정보보다는 유명한 대표 작가와 대표 감독만 주로 유저의 선택에 영향을 미쳤을 것이라 추정
- 각 영화에 대해서 가장 평가 유저 수가 많은 대표 작가와 대표 감독만 남기도록 하였음
- 2989명의 작가는 1916명으로, 1340명의 감독은 1275명으로 줄여서 label encoding을 거쳐서 사용하도록 하였음

③ 장르 (genres.tsv)

- 장르 정보는 총 18가지로 다른 side information에 비해 그 종류 수가 비교적 적은 편에 속함
- one-hot encoding을 적용하여 하나의 벡터로 표현함

3. 모델링 1: General Recommendation

- 유저와 아이템의 상호작용을 모델링, 이때, 상호작용한 순서는 고려하지 않으며 유저의 일반적인 선호를 예측
- 대회 test 데이터 셋에 존재하는 '특성 시점 이전에 랜덤으로 dropout된 아이템'에 대한 예측을 목표로 함

3-1. General Recommendation 모델 개요

Model	특징
Multi-VAE	Variational Autoencoder 기반의 CF 모델, multinomial likelihood 와 annealing 기법을 사용함
EASE	Autoencoder 기반의 user-free 모델로, 희소성이 높은 데이터와 cold start problem 에 강인함
LightGBM	트리 기반 모델로 leaf-wise 방식을 사용하여 학습속도가 빠르다는 장점을 가짐

4. 모델링 2: Sequential Recommendation

- 유저가 아이템들과 상호작용한 순서를 고려하여 모델링, 유저가 다음에 상호작용할 아이템을 예측
- 대회 test 데이터 셋에 존재하는 '특정 시점 이후에 상호작용한 아이템'에 대한 예측을 목표로 함

4-1. Sequential Recommendation 모델 개요

Model	특징
GRU4Rec	GRU 를 사용하여 item sequence 의 순차적인 패턴을 학습함
BERT4Rec	BERT 를 적용한 모델로 SASRec 과 달리 양방향으로 학습할 수 있어 좋은 성능을 보일 것이라 기대함
S3Rec	Self-supervised learning 을 통해 Side information 과 item sequence 정보를 pretrain 함
S3Rec + SASRec	S3Rec 으로 pretrain 된 item embedding 을 초기 가중치로 SASRec 을 finetune 함
S3Rec + BERT4Rec	S3Rec 으로 pretrain 된 item embedding 을 초기 가중치로 BERT4Rec 을 finetune 함
SRGNN	GNN 계열의 모델로 아이템 사이의 복잡한 전환을 정확하게 표현함

5. 앙상블 전략 수립

- 각 모델 및 방법론이 추론하는 방식은 서로 다르기 때문에, 앙상블을 통해 서로의 취약점을 보완해줄 수 있을 것이라 가정
- 앙상블 전략은 우선순위 기반을 통해 이루어졌음
  - 0 순위: 여러 추론 파일에 공통적으로 등장한 아이템
  - 1~N순위: 각 방법론이 생성한 추론파일에 Public Score를 토대로 휴리스틱하게 우선순위를 부여
- 우선순위에 따라서 유저별로 10개의 아이템이 선정될 때까지 순차적으로 아이템을 선정 및 삽입

6. 최종 솔루션 모델

- Public Score 를 기준으로 높은 성능을 보였던 EASE 와 Multi-VAE 를 앙상블한 모델과 단일 EASE 모델을 최종 솔루션 모델로 선정함
- **EASE**: Public Score: 0.1600 → Private Score: 0.1600
- **EASE + Multi-VAE**: Public Score: 0.1560 → Private Score: 0.1578

1-5. 자체 평가 의견

[배운 점]

- **모듈화의 중요성**: 주어진 베이스라인모델을 Pytorch lightning 으로 구현하는 과정에서 pretrain 과 finetune 두 모델이 서로 합쳐져 있어 이를 분리할 필요성을 느껴 두 모델을 모듈화하였음. 그 결과로 메모리와 작동 시간 측면에서 더 효율적으로 학습할 수 있게 됨. 이로부터 모듈화의 중요성과 이점을 느낄 수 있었음
- **Input/Output 을 명료하게 정리하는 것의 중요성**: Tabular 베이스라인을 구축하는 과정에서 Input/Output 의 형태를 명확하게 정의하지 않아 방법론 및 모델 탐색 과정에 있어 혼란과 실패를 겪었음. 이로부터 Input/Output 을 명료하게 설정하는 것이 실제 모델을 탐색하고 구현하는 과정에서 중요한 역할임을 깨닫게 됨

[아쉬운 점]

- **공통 지표의 부재**: 구현한 여러 모델에서 공통적으로 성능의 비교할 수 있는 지표가 없었던 점, 이로 인해 public score 만을 기준으로 성능을 평가하게 된 것
- **앙상블 전략**: 객관적인 방법이 아닌 실험적 방법을 통해 앙상블을 진행한 것, 다양한 앙상블 방법을 시도해보지 못했던 것
- **모델 구현 소모 시간**: 베이스라인 구축 및 각 모델 구현 과정에서 예상보다 많은 시간을 소모하게 되었고 이로 인해 다양한 실험, 방식을 시도해보지 못한 것

2. Personal Wrap-up Report

2-1. 강찬미_T5009

1.내 학습목표를 달성하기 위해 한 노력

- **베이스라인 구축** : 이번 프로젝트의 가장 주된 목표였으며 Pytorch lightning 을 통해 주어진 베이스라인을 구축하였다. 기존 베이스라인의 구조가 lightning 을 적용하기 어려운 구조로 되어 있어 이를 pretrain, finetune 부분으로 나누어서 주어진 config option 에 따라 개별적으로 동작하도록 코드를 작성하였다. 이 과정에서 모듈화와 최적화에 대한 고민을 해볼 수 있었고 Pytorch lightning 구조에 대한 이해를 높일 수 있었다.

2.내가 모델을 개선하기 위해 한 노력

- **BERT4Rec 추가** : 기존 S3Rec 에서는 SASRec 을 통해 finetune 을 하고 있어 BERT4Rec 모델로도 학습할 수 있도록 BERT4Rec 을 추가하였다.

3.내가 한 행동의 결과로 달성한 것 및 얻은 깨달음

- **task 에 적합한 모델 탐색의 필요성**: 베이스라인 모델인 S3Rec 을 lightning 으로 구현하는 과정에서 꽤나 많은 시간을 소모하였다. 시간을 많이 투자하여서 진행하였던 이유는 더 다양한 정보를 학습에 사용한다는 점과 베이스라인이라는 점이었다. 하지만 예상과는 달리 좋은 성능을 보여주지 못했고 최종 제출 또한 다른 모델로 하였다. 생각해보면 지난 2 번의 대회에서도 베이스라인이 아닌 모델들이 더 좋은 성능을 보여주었는데 이번에는 왜 그렇게 베이스라인이 잘 나올 것이라 확신하였는지 모르겠다. 이 과정을 거치며 대회에 대한 이해를 바탕으로 적합한 모델을 찾아보는 과정을 먼저 갖는 것이 중요함을 깨닫게 되었다.

4.내가 새롭게 시도한 변화와 효과

- **모듈화**: 베이스라인 구축하는 과정에서 모듈화를 진행하며 어떻게 얼마나 분리하는 것이 좋은지 고민해볼 수 있었다. 그 결과 pretrain 과 finetune 을 하나의 main 에서 설정만 바꾸어 돌릴 수 있게 되었다,
- **최적화**: 학습 과정에서 병목이 걸리는 부분을 확인하고 이를 해결하기 위해 다양한 방법을 시도해보았다. 그 결과 기존 에폭마다 8 분정도 걸리던 학습 시간을 약 4 분으로 줄일 수 있었다.

5.마주한 한계와 아쉬웠던 점

- **구축 과정에서의 시간 소모**: 실제 구현하는 시간보다 어떻게 구조를 짜야하는지 고민하는데 많은 시간을 보냈던 것이 아쉽다. 실질적으로 고민을 하고 작성한 구조에서도 새롭게 예상치 못한 오류를 발견했기 때문에 우선 " 나중에 이런 경우가 생기면 어떻게 하지? " 라는 고민은 조금 접어두고 궁극적으로 해결해야 하는 문제에 대해서만 고민하는 것이 중요한 것 같다.

6.다음 프로젝트에서 시도해볼 것

- **크롤링을 통한 데이터 수집**: 크롤링에 대해 알고 있었지만 직접 코드를 짜서 크롤링 해본 경험이 없어 다음 프로젝트에서 도전해보고 싶다.
- **프론트엔드 담당**: 프론트관련 공부를 해본 적이 없어 다음 프로젝트에서 기회가 있다면 공부하며 경험을 쌓아보고 싶다.
- **유저 피드백 반영**: 실제 서비스를 하게 된다면, 유저의 피드백을 통해 모델을 지속적으로 개선시키는 것이 중요하다 생각해 기회가 된다면 꼭 시도해보고 싶다.

2-2. 박동연_T5080

1. 내 학습목표를 달성하기 위해 한 노력

- **EDA & 데이터 전처리:** 데이터를 분석함으로써 다양한 아이디어나 인사이트를 발견하고 발전하고자 했다. 그래서 이번에는 EDA 팀에 자발적으로 참여하여 적극적으로 팀원들과 아이디어를 공유하였고, 이를 토대로 약 3~4 가지의 전처리를 수행하는 코드를 작성했다.
- **배운 것과 공부한 것을 충분히 활용하기**: 지금까지 배운 것들과 깨우친 것들을 프로젝트 과정속에서 충분히 활용하고자 노력했다. 그래서 팀원들에 비해 비교적 숙지가 덜 된 Tree model 에 대해서 좀더 공부하여 Tabular Approach 에서 활용하였고, 커리큘럼 내 강의에서 언급되었던 EASE 모델을 개인적으로 조금 더 공부한 뒤 직접 모델을 구현 및 추가하였다.

2. 내가 모델을 개선하기 위해 한 노력

- **데이터 전처리**: Tabular Approach 에서 Tree Model 을 사용할 때 성능을 올리기 위해서는 적절한 전처리와 다양한 Feature Engineering 의 역할이 중요하다. 따라서 특정 컬럼 내 결측치 처리, 특정 컬럼들에 대한 대표값을 산출하는 등의 전처리 과정을 추가하였다.

3. 내가 한 행동의 결과로 달성한 것 및 얻은 깨달음

- **모델과 데이터의 Input/Output을 정확하게 정의하기**: Tabular Baseline을 구축할 때 Input/Output 형태가 명료하게 정의되지 않아서 많은 혼란과 실험적인 단계를 거치게 되었던 것 같다. 즉, Intput/Output을 명확하게 설정해야 그 중간 단계의 역할을 수행하는 방법론과 모델을 잘 정의 및 탐색할 수 있다는 것을 깨닫게 되었다.

4. 내가 새롭게 시도한 변화와 효과

- **Tabular Approach & Tree Model:** 이전 프로젝트에서는 Sequential Approach & DL model 을 주로 담당했었다. 그러다보니 이번에는 딥러닝 모델에 비해 가볍지만 여전히 좋은 성능을 내며, 실무에서도 잘 사용한다는 Tabular Approach & Tree model 에 대해서 궁금해졌다. 따라서 이번 프로젝트에서는 다양한 categorical side information 을 활용할 수 있을 것이라고도 생각하여 Tabular Approach 와 Tree model 을 사용해보았다.
- **모델 구현:** EASE 모델이 본 대회의 태스크에 적합하다고 판단하여 모델을 추가하고자 했고, 이 때 비교적 간단한 형태라고 생각하여 직접 구현해보게 되었다. 물론 하나부터 열까지 모두 다 스스로의 힘으로 한 것은 아니었고, 다른 많은 자료를 참고했지만, 구현을 끝까지 해내었고 단일 모델로서 성능도 꽤나 괜찮게 나오는 것을 확인해서 뿌듯했다

5. 마주한 한계와 아쉬웠던 점

- **전처리 적용 여부에 대한 성능 확인:** 목표했던 EDA 와 전처리 파트에 적극적으로 참여했지만, 정작 이러한 전처리 프로세스를 적용했을 때와 그렇지 않았을 때에 대한 성능 향상 여부를 확인하지 못했다. 성능확인까지 했어야 전처리 단계에서 세운 가설이 들어맞는지 검증할 수 있었는데, 그러지 못한 점이 아쉽다.

6. 다음 프로젝트에서 시도해볼 것

- **지속적인 실험 및 검증과 업데이트:** 이번 프로젝트에서는 직접 문제를 정의하고 데이터를 확보 및 모델을 선정 하기 때문에 각각의 요소를 변경하는 것이 다양한 방면으로 영향을 미칠 것으로 추정된다. 따라서 우리가 최적화하고자 하는 지표를 잘 선정하여, 어떤 실험과 변경을 하는지에 따라 이 지표가 어떻게 바뀌는지 잘 추적하여 최적화하고 싶다.
- **BigQuery, AirFlow 등의 프레임워크 적절히 사용하기:** 최종 프로젝트에서는 직접 수집한 다양한 형태의 대용량 데이터를 사용하게 되는데, 이를 잘 저장, 처리, 업데이트할 수 있는 적절한 프레임워크를 사용해보고 싶다.

2-3. 서민석_T5102

1. 내 학습목표를 달성하기 위해 한 노력

- **Autoencoder 기반 CF 방법론**: Autoencoder 기반 CF 방법론을 이해하고 적용해보는 것을 시도했다. 여러 방법론 중 Multi-VAE 모델을 중심으로 유저의 아이템에 대한 선호를 모델링하기 위한 데이터 처리, 학습 및 추론 코드를 작성하였다.
- **Pytorch Lightning**: Pytorch Lightning 프레임워크를 적용해보는 것을 시도했다. 이번 프로젝트에서는 구현 자체에 집중하느라 Lightning 기반의 프레임워크의 장점을 온전히 느끼지는 못했지만, 모델 아키텍쳐와 학습 루프를 분리하여 모듈화하는 부분은 실험 과정에서 다른 모델을 추가하거나 작성한 코드를 유지 관리하기에 용이하다는 것은 확실히 느낄 수 있었다.

2. 내가 모델을 개선하기 위해 한 노력

- **모델 최적화**: 실험을 통해 Multi-VAE 모델을 최적화했다. 구체적으로 annealing 파라미터(최댓값, 수렴시기 조정), 모델 크기(레이어 수, 노드 수), 활성화 함수, 검증데이터 분할 방식 등을 최적화 하였으며 이를 통해 recall@10 기준 17.4%(0.1119->0.1314)로 성능을 향상시켰다.

3. 내가 새롭게 시도한 변화와 효과

- **딥러닝 논문 이해 및 구현**: Multi-VAE 는 논문 저자가 공개한 공식 구현체 코드가 존재하지 않았다. 따라서, 논문을 이해하고 직접 코드를 작성했다. 다른 사람이 작성한 코드와 내가 작성한 코드를 비교하고, 논문에 반하는 내용이나 오류, 그리고 예외 상황은 없는지 고민하는 과정을 통해 모델링 역량을 성장 시킬 수 있었다. 또한 논문 내용을 그대로 적용하지 않고 대회 task 에 맞춰 검증 데이터셋 분할 방식과 하이퍼파라미터를 수정하여 리더보드 성적을 향상시킬 수 있었다.

4. 마주한 한계와 아쉬웠던 점

- **개발 실력과 계획**: 프로젝트 초반 계획은 먼저 Multi-VAE 구현 및 실험을 빠르게 끝내고, sequential 모델들을 구현할 계획이었다. 그러나 Multi-VAE 구현과정에서 autoencoder 기반 CF 모델에 적합한 데이터 입출력 형태와, 검증방법 그리고 VAE 모델 아키텍쳐를 이해도를 높이는 것 자체에 많은 시간이 소요되어 결국 다른 모델들은 적용해보지 못했다. 실력과 맞지 않게 무리한 계획을 세웠던 부분이 문제였다고 생각한다. 이를 해결하기 위해 개발 실력을 향상시켜 개발 주기를 짧게 하는 것, 현재 실력을 인지하고 적합한 계획을 세우는 것을 시도할 것이다.

5. 다음 프로젝트에서 시도해볼 것

- **Representation 추출 방법론에 대한 고민**: '이미지와 유사한 노래를 추천한다.' 라는 문제를 풀기위해, '어떤 방법론을 사용하여 이미지의 representation 을 추출하는 것이 가장 적절할까?'에 대한 고민을 해보고 싶다.
- **FAISS 기반 Vector DB 구축**: 이미지 벡터를 FAISS 기반 Vector DB 에 저장하고, 쿼리 이미지 벡터가 입력됐을 때 가장 유사한 k 개의 벡터를 출력하는 벡터 유사도 기반 검색 시스템을 구축해보고 싶다.
- **클라우드 서비스 플랫폼 경험**: AWS 나 GCP 같은 클라우드 서비스 플랫폼에서 제공하는 도구들을 프로젝트에 활용해보는 경험을 해보고 싶다.

2-4. 주혜인_T5208

1. 내 학습목표를 달성하기 위해 한 노력

- **새로운 툴 경험하기**: RecBole & Ray Tune 베이스라인 작성하기
- **회고 및 기록**: 데일리 회고 하기 & RecBole tutorial 작성해두기

2. 내가 모델을 개선하기 위해 한 노력

- **sequential 정보 활용**: 영화 평가 이력이 순서가 있는 sequential data 라는 점을 활용하기 위해 지난 대회에서 활용하지 못했던 SRGNN 모델을 활용했다.

3. 내가 새롭게 시도한 변화와 효과

- **라이브러리의 Github 깊게 탐색하기**: recbole 에서 numpy 관련 오류가 해결되지 않아 익명의 준영님과 함께 recbole github 을 살펴봤다. 이전에도 관련 이슈가 있어서 새로운 버전에 반영되지 않은 PR 을 참고해 나도 수정할 수 있었다. 이 경험 이후로 Ray tune 에서 directory 관련 오류가 났을때도 github 을 통해 오류의 원인을 파악할 수 있었다. 또 한번 문제해결력에 도움이 되는 좋은 경험을 한 것 같다.
- **RecBole 사용하기:** RecBole 라이브러리는 여러 추천 모델을 사용할 수 있어 편리하기 때문에 활용했고 베이스라인 완성 후에는 다양한 모델들을 config 에 모델명을 수정하는 것만으로도 실험해볼 수 있었다.
- **Ray Tune 도입**: 베이스라인 완성 시기가 꽤 늦어졌다고 생각해서 짧은 시간내에 hyperparameter 를 효율적으로 탐색하기 위해 ray tune 을 활용했다.

4. 마주한 한계와 아쉬웠던 점

- **충분하지 않은 실험**: RecBole 베이스라인을 완성한 시기가 늦어져서 충분한 실험을 하지 못한 것 같다.
- **EDA & 파생변수 생성**: 이번 프로젝트에서 가장 큰 목표가 파생변수를 생성하는 것이었지만, 많은 사용자들의 선호도를 시각화하는 것이 어려웠고 그 결과 새로운 파생변수를 만들어내지 못해서 아쉽다.
- **validation set** 구축: 자체 validation score 와 public score 의 차이가 꽤 크게 났었다. validation set 을 구축하는 로직이 잘못되었던 것 같아서 이 점을 개선하지 못한 것이 아쉽다.

5. 다음 프로젝트에서 시도해볼 것

- **EDA를 바탕으로 파생변수 만들기**

  : 최종 프로젝트에서는 플레이리스트 관련 데이터를 활용하는데, 아직 모델을 구체적으로 정해두지 않아서 플레이리스트의 인기도나 유형 관련해서 파생변수를 만들 수 있으면 좋은 영향을 줄 수 있을 것 같다.

- **문제를 정의하고 해결하는 과정을 논리적으로 설계하고 기록하기**

  : 이력서 피드백 특강을 듣고 생각하게 되었는데, 협업중심의 회고에서 벗어나 프로젝트 관련 내용을 통해 느낀점을 작성하려면 문제해결과정을 잘 기록해두는게 좋을 것 같다.

2-5. 이준영_T5158

1. 내 학습목표를 달성하기 위해 한 노력

- **Hold 병렬 학습**: 병렬학습을 위해 기존 holdout trainer를 재사용, 독립적인 fold trainer를 구현했다

2. 내가 모델을 개선하기 위해 한 노력

- **다양한 Side information 활용**: S3Rec 모델의 4가지 loss를 중, attribute를 활용하는 즉, side information을 활용하는 부분을 모듈로 분리하였고, 다양한 side information을 모델에 학습시켜볼 수 있었다.
- **속도 개선:** 전처리된 학습 데이터를 미리 준비해 학습시 같은 전처리 과정을 반복하지 않도록 하였다. S3Rec의 Pretrain 과정에서는 1에폭당 11분이 걸리던 것을 4분까지 줄였다.

3. 이번 프로젝트에서 도움이 되었던 것 및 깨달음

- **Lightning Profiler**: Lightning으로 도입하면서 학습 속도가 저하되는 이슈가 있었는데 lightning에서 제공해주는 profiler기능을 사용해 hydra config부분에서 병목이 생긴다는 점을 알게 되었다.

4. 다음 프로젝트에서 시도해볼 것

- **모듈화와 Lightning도입**: 이번 대회에서 개발했던 S3Rec모델은 pretrain과 finetune두개의 방식에 있어 pytorch lightning을 도입하기 어려웠다. 때문에 다양한 설계 시도를 해보았고, 이 끝내 pytorch lightning에 S3Rec을 구현하는데 성공시켰다.

5. 마주한 한계와 아쉬웠던 점

- **낮은 cv score 점수**: S3Rec 모델에 공을 들였음에도 불구하고 성적은 매우 좋지 않았다. 이는 아마도 데이터 전처리를 잘 못 수행했기 때문으로 추정되는데, 모델에 대한 이해가 부족한 것도 있지만 구조가 다소 복잡해 알기가 어려웠다. 과잉 모듈화를 한 것이 아닐까 의심이 된다.
- **병렬화 도입 무산과 하드웨어의 한계**: 이미 S3Rec과 SASRec모델이 많이 무거워 holdout으로 해도 학습 리소스를 많이 잡아먹고 있었다. 학습 속도를 위해 메모리를 더 사용하고 있었던 것도 하드웨어에 부담이 되었다. 결국, 이런저런 이유로 병렬화는 포기하게 되었는데 어쩌면 이것이 1gpu 1model을 지향하는 lightning의 이유일 수 있겠다.
- **Lightning에 대한 이해 부족**: 이번 대회에서 처음 Lightning을 이용해 baseline을 구축해 보았는데, 쉽지 않았다. 나의 잘못된 이해로 팀원의 발목을 잡는 일도 있었고 에러를 잡을 때도 시간이 오래 걸렸다.
- **부족한 실험**: baseline 구축에 많은 너무 시간을 들여 실험을 많이 해보지 못했다.

6. 다음 프로젝트에서 시도해볼 것

- **Product Serving**: 다음 프로젝트에선 Product Serving 기술을 많이 활용해야 할 필요성이 있다. 이 점을 기회로 잘 활용해 다양한 것을 해볼 의향이 있다.
- **데이터 관련 로직 컴포넌트 분리:** 모델 학습 과정을 프로파일링 해보면, 생각보다 데이터 처리에서 정말 많은 시간을 잡아먹는다. 때문에 데이터 처리는 별도의 컴포넌트로 분리해야 할 것 같다.
- **도커 활용:** 기다리고 기다렸던 도커 혹은 컨테이너를 사용해 볼 수 있을 것 같다. 물론, GCP의 각종 기능들을 활용한다면 안 쓰는 쪽으로 갈 수도 있겠다.