

Movie Recommendation

Recsys-7조(X10)

곽정무, 박준하, 박태지, 배현우, 신경호, 이효준

1. 프로젝트 개요

1-1. 개요

본 프로젝트는 약 510만 개의 데이터를 포함한 변형된 MovieLens 데이터를 기반으로 추천 시스템 알고리즘을 개발하는 대회이다. 사용자의 영화 시청 이력과 implicit feedback을 바탕으로 사용자가 다음에 시청할 영화 및 시청할 가능성이 높은 영화를 예측하는 것을 목표로 한다.

데이터셋은 시간 순서에 따라 일부 아이템이 누락된 상황을 가정하였으며, 영화의 장르, 감독, 작가와 같은 다양한 side-information이 포함되어 있다. 해당 프로젝트에서는 단순한 순차적 추천을 넘어, 누락된 데이터와 시간적 요소를 동시에 고려한다.

1-2. 환경

협업 환경: GitHub

의사 소통: Zoom, Slack, Jira, Confluence

2. 프로젝트 팀 구성 및 역할

공통	EDA 및 랩업 리포트 작성
곽정무	PM, Researcher, Sequential 모델 담당
박준하	Researcher, Baseline, AutoEncoder 계열 모델 담당
박태지	Sequential 모델 실험, RecBole 실험
배현우	Engineer, Baseline, 결과 분석
신경호	LGBMRanker, RecBole 실험
이효준	Confluence 구축, 결과 분석, 랩업리포트 관리

3. 프로젝트 수행 절차

3-1. Timeline

(1주차) 수업 수강 및 EDA

(2주차) 아이디어 회의 및 모델링

(3주차) 결과분석 및 앙상블



그림 1. 프로젝트 Timeline

3-2. 협업 과정

- [1] 회의를 진행하기 전 각자 이야기하고 싶은 내용을 간단하게 Notion에 기록한다.
- [2] 매일 데일리스크럼 및 회의를 통해 아이디어를 구체화하여 Confluence에 기록한다.
- [3] Jira를 이용하여 기간 및 task를 할당한다.
- [4] 자율적으로 Zoom을 이용하여 진행상황을 공유하거나 발표를 진행한다.

4. 프로젝트 수행 결과

4-1. EDA

이번 대회에서 제공된 데이터는 MovieLens 데이터를 가공한 데이터로 유저와 아이템의 비율을 고려했을 때는 ML-20M데이터에 가까웠다. 유저와 아이템의 상호작용 비율의 경우 2.65%로 ML-20M과 ML-1M의 평균에 근사했으며 추천시스템의 데이터 중에서는 dense한 편에 속한다고 볼 수 있었다. 이에 대한 원인을 파악하기위한 EDA 진행결과 상호작용이 가장 적은 유저의 평가 빈도가 16회로 상호작용이 매우 적은 유저들이 필터링됐다는 것을 확인할 수 있었다.

표 1. MovieLens Data와 대회 Data

	ML-20M	ML-1M	대회 Data
# of users	136,677	6400	31360
# of items	20,108	3900	6807
# of interactions	20M	1M	5.1M
% of interactions	0.36%	4.27%	2.65%
Date range	1995 - 2015	2000 - 2003	2005 - 2015

그리고 장르, 감독, 작가의 경우 다중값을 가지는 피쳐로 작가의 경우 한 영화의 작가가 24명인 경우도 있었다. 장르에 대해서는 18개의 고유값을 가지므로 처리에 어려움이 덜했으나 감독과 작가의 경우 약 1300명, 2900명의 고유값을 가져 데이터 처리에 큰 어려움이 있었다.

표 2. 감독, 작가, 개봉연도 피쳐 특징

	Unique	IsNull	Item unique
Director	1399	301445	1304
Writer	3838	312692	1159
Year	93	1832	8

또한 감독, 작가, 연도의 경우 모든 영화에 대한 데이터를 보유하지 않아 결측치를 포함하고 있었다. 반면 연도의 경우 제목에 있는 연도를 활용해 결측치를 처리했다.

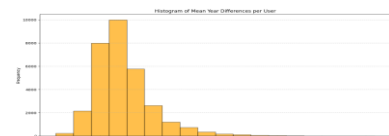


그림 2. 전체 데이터 기준 평가연도와 개봉연도의 차

마지막으로 EDA를 통해 평가연도와 개봉연도의 차이를 확인할 수 있었다. 전체데이터의 유저별 평균 평가연도와 개봉연도의 차와 유저별 마지막 5개 데이터의 유저별 평균 평가연도와 개봉연도의 차를 비교했을 때, 해당 유저의 마지막 평가에 가까울수록 최신영화에 대한 평이 더 많은 것을 확인할 수 있었다.

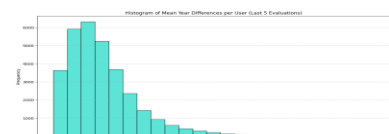


그림 3. 유저별 마지막 5개 데이터 기준 평가연도와 개봉연도의 차

4-2. Train Test Split

Dataset을 구분하는 것은 학습 결과에 큰 영향을 끼치는 부분이다. 이번 대회는 보편적으로 사용되는 MovieLens 데이터를 재가공하여 사용자마다 Sequential 데이터와 Random으로 Masking된 데이터를 추론하는 것이 목적이었다. 위와 같은 특성을 고려하여 우리는 유저별 시청이력의 Timestamp의 마지막 1개와 랜덤으로 추출한 데이터 2개를 사용하여 데이터셋을 분리하였다.

초반에는 단일 모델을 사용하였기에 위와 같은 분할 전략을 선택하였으나 프로젝트를 진행하면서 Static 모델과 Sequential 모델을 앙상블하기 위하여 위 방법을 분리하여 Static 모델에는 Random 2개의 데이터, Sequential 모델에는 Leave-One-Out을 이용하여 마지막 1개의 데이터를 추론하는 전략으로 변경하였다.

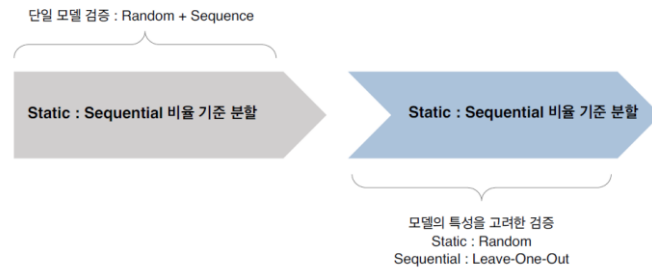


그림 4. Train Test Split 전략

4-3. 모델 실험 및 결과

1) 시퀀셜 모델

시퀀셜 모델의 경우 이번 대회의 데이터셋에서 유저별 평균 sequence length 가 160 정도였기에, RNN기반의 GRU4Rec보다는 long term dependence를 반영하는 트랜스포머 기반의 SASRec과 BERT4Rec을 고려하였다. SASRec과 Bert4Rec을 고려할 때 가장 큰 차이점은 SASRec은 과거로부터 unidirectional하게 학습하고 BERT4Rec의 경우 마스킹된 아이템을 복원하기위해 bidirectional하게 학습한다는 것이다.

2023년에 나온 두 편의 논문에서[1][2], Li와 Klenitskiy는 두 모델의 가장 큰 차이는 loss function이라고 지적했다. SASRec의 경우 기존의 논문 설정 상 positive sample 한 개에 대하여

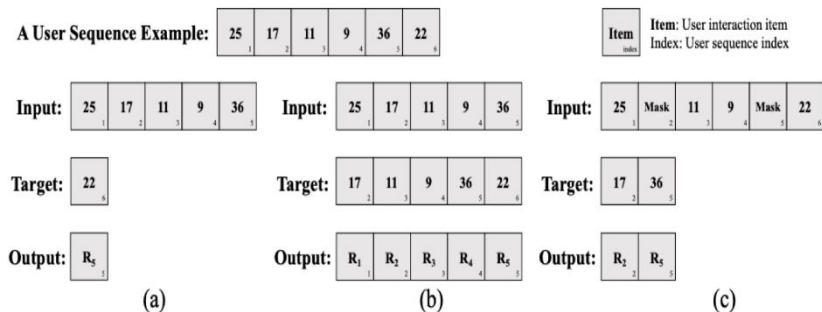


그림 5. SASRec과 BERT4Rec의 학습 방법

negative sample 하나를 두고 binary cross entropy loss를 사용한다. 반면에 BERT4Rec의 경우 마스킹한 아이템에 대해 가능한 모든 아이템을 고려한 cross entropy를 사용한다.

그림 5에서 논문의
예시를 가져왔는데
(b)가 SASRec의 학습
방식이고 (c)가
BERT4Rec의 학습 방

식이다. SASRec의 경우 모든 timestamp에서 loss를 계산하는 방식이기 때문에 negative sampling을 하지 않고 BERT4Rec과 똑같이 cross entropy loss를 적용할 경우 BERT4Rec 보다 성능이 좋다고 주장했고, 학습 단계와 예측 단계의 프로세스가 맞아 task 적으로도 더 적합하다는 주장이 설득력이 있어 SASRec을 sequential 모델로 최종 선정하였다.

표 3은 leave one out 평가 방식에서 SASRec의 recall@10 변화 과정이다. 가장 먼저 적용한 부분은 max sequence length이다. 원래 베이스라인의 모델은 max sequence length를 50으로 제한했는데 대회 데이터셋의 sequence가 평균적으로 긴 편이었기 때문에 200으로 늘렸을 때 소폭의 성능 향상이 있었다.

표 3. ML Dataset Benchmark (Leave one out)

Dataset	Metrics	SASRec	BERT4Rec _{hf}	BERT4Rec-VAE	SASRec
ML-1M	HIT@10	0.2089	0.2594	0.2512	0.2849(36.38%)
	NDCG@10	0.1119	0.1409	0.1387	0.1642(46.74%)
ML-20M	HIT@10	0.1437	0.2378	0.1558	0.2176(51.43%)
	NDCG@10	0.0716	0.1310	0.0892	0.1214(69.55%)

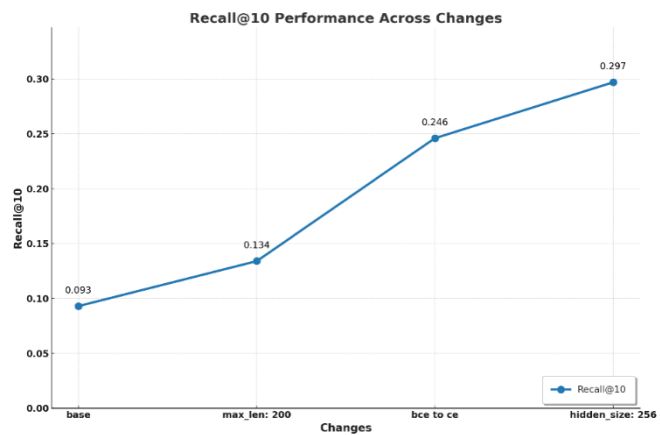


그림 6. SASRec의 recall@10 변화 과정

가장 큰 성능 향상은 논문에서 주장한 바와 같이 네거티브 샘플링을 통한 binary cross entropy loss에서 cross entropy loss로 바꿨을 때 일어났다. 최종적으로 아이템의 개수가 6000개가 넘기 때문에 이를 반영해 hidden size를 64에서 256으로 올렸고 그 결과 recall@10이 0.297으로 향상되었다.

2) 스테틱 모델

Static 모델의 경우, AutoEncoder 계열의 모델들이 상대적으로 높은 성능을 보였다. 데이터가 매우 dense하기 때문에 user-item matrix를 압축하고 복원하는데 있어 참조할 정보가 더 많기 때문이라고 추측하였다.

실험해본 AutoEncoder 계열의 모델은 대표적으로 Multi-VAE, SVD-AE, RecVAE, EASE가 있었다. 이 중 MultiVAE의 경우에는 [3]이 지정한 대로 denoising 메커니즘을 추가로 구현하였다.

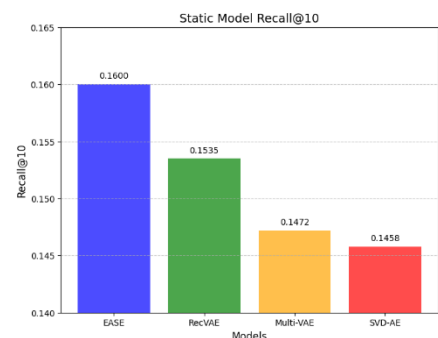


그림 7. Public data에서의 recall@10

그 중 가장 높은 성능을 보여준 모델은 EASE였으며, 해당 모델은 public data 기준 recall@10에서 0.1600의 성능을 기록하였다. 잘 정제된 대회용 데이터라 노이즈가 적어 간단한 모델인 EASE가 잘 동작하는 것으로 보였다.

또 [4]에 따르면, EASE 모델은 다른 AutoEncoder 계열의 모델에 비해 더 개인화된 추천을 잘 수행한다고 주장하였기에 이를 EASE가 잘 동작하는 근거라 판단하였다. 따라서 이 가설을 확인해보기 위해 각 모델이 주어진 데이터셋 기준 가장 많이 시청된 300개의 영화를 얼마나 추천하는지 분석했다. 그 결과 표에서 볼 수 있듯이, 오히려 EASE가 모든 auto encoder 계열의 모델 중 가장 많이 인기도 기반의 추천을 한다는 사실이 확인되었다. 논문의 내용과 불일치하는 결과였으며, 이 또한 데이터셋의 특수성으로 인한 것으로 추측됐다.

표 4. 모델별 Top300 인기영화 추천 비율

	EASE	SVD-AE	Multi-VAE	RecVAE
추천 비율	0.8520	0.8422	0.7974	0.7772

정리하자면 static부분의 최종 모델로 public data에서 가장 높은 성능을 보이는 EASE를 선정하였다. 다른 AutoEncoder 모델들과 앙상블 하려는 시도도 있었으나, 유의미한 성능 향상은 일어나지 않았다. 같은 계열의 모델이라 앙상블의 의미가 희석된 것이라 판단하였으며, 실제로 모델들의 공통 추천 비율을 확인해본 결과 추천 중 약 60%가 공통되었음을 알 수 있었다.

3) 앙상블 모델

Static 한 모델과 Sequential한 모델 팀을 분리해서 프로젝트를 진행했지만 추천된 아이템들이 얼마나 상이할지에 대해서는 결과를 확인하기 전까지는 확신할 수 없었다. 따라서, 공통으로 추천된 아이템이 데이터셋에서 얼마나 차지하는지 확인했고 공통 추천 비율이 약 5%에 불과했다.

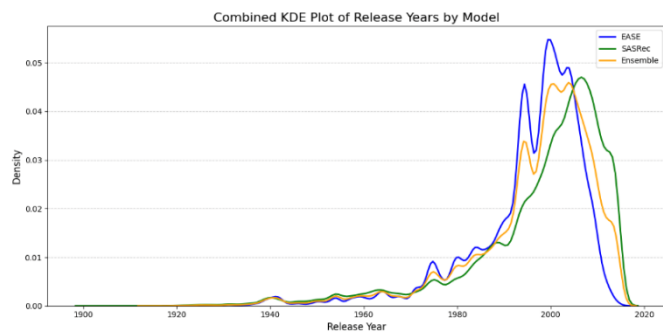


그림 8. EASE, SASRec, 앙상블 모델 개봉연도

추천한 영화들의 개봉 시기에 있어서도 SASRec이 최신 영화를 추천하는 비율이 높았고 이는 EDA 분석결과와 부합해 두 모델이 각기 목표한 의도를 반영했다 확신하고 앙상블을 진행했다.

표 5, 6. EASE, SASRec, 앙상블 모델 인기도 분석 및 아이템 다양성 분석

	EASE	SASRec	Ensemble
추천 비율	0.8520	0.3748	0.6398

Model	Coverage
EASE	0.2724
SASRec	0.9379
Ensemble	0.8626

인기도와 아이템 다양성 분석에 있어서 제한적으로 영화를 추천했던 EASE 와 달리 SASRec은 다양한 영화를 추천했고 공통된 아이템을 먼저 추가하고 EASE와 SASRec이 약 5:5 정도로 추천하게 하였을 때 앙상블 모델 또한 다양성에 있어 향상되었고 public score에서 우수한 성적을 거뒀으나 private score에서 점수가 크게 하락하였다.

다음은 private score 기준 점수 변화인데 sequential한 데이터가 public에 비해 적어서인지 EASE를 기반으로한 추천을 약 6개 SASRec을 기반으로한 추천을 약 4개 넣었을 때 가장 좋은 성능을 보였다.

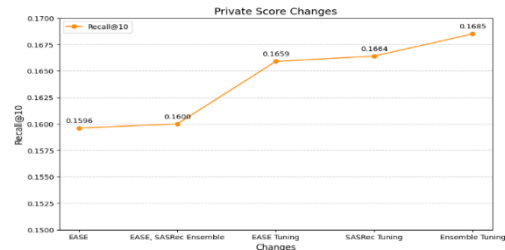


그림 9. Private Score 변화 과정

5. 자체 평가

이번 프로젝트를 진행하면서 관련 논문과 자료를 면밀히 리서치하여 데이터셋에 적합하게 적용하기 위해 노력하였다. 활발한 회의를 통해 팀원들 간의 원활한 의견 교류가 이루어졌으며, 이를 바탕으로 다양한 모델을 시도하였다. 또한, 모델 결과를 후분석하며 추천의 질을 심도 있게 분석하고, 앙상블 기법을 적용하여 성능 향상을 도모하였다.

그러나 아쉬운 점도 있었다. 활발한 회의를 통해 여러 가설을 세웠지만, 그것들을 실험으로 검증하지 못한 부분이 많았고, Side-Inf를 충분히 활용하지 못한 점이 있다. 모델의 다양성은 확보하였지만, 각각의 모델에 대한 완결성이 부족하여 최적의 성능을 이끌어내지 못한 부분도 있었다.

이번 경험을 통해 가설의 실험적 검증과 Side-Info의 적극적인 활용, 그리고 모델의 완성도를 높이는 것이 중요하다는 교훈을 얻었다.

References

- [1] Improving sequential recommendation models with an enhanced loss function", Li et al., 2023
- [2] Turning Dross into gold loss: is BERT4Rec really better than SASRec?", Klenitskiy et al., 2023
- [3] RecVAE: a New Variational Autoencoder for Top-N Recommendations with Implicit Feedback, Ilya Shenbin et al., 2019
- [4] Embarrassingly Shallow Autoencoders for Sparse Data, Harald Steck, 2019

개인회고 (곽정무_T7504)

이번 대회는 개인적으로 정말 다이내믹 했다. Sequential 모델을 담당해 많은 논문을 리서치하며 대회와 관련한 흥미로운 논문들을 적용했을 때 public score 상 많은 성능 향상으로 이어졌고 압도적인 점수차를 유지하며 나름 뿌듯했으나, private score 가 공개되었을 때 sequential 정답의 비율이 public 보다 낮아 점수 하락의 폭이 큰 걸 확인하고 실망감이 컸다.

하지만, 부스트캠프에서 대회는 성장을 위한 촉진제일뿐 점수가 본인이 얼마나 성장했는지에 대한 절대적 지표가 아닌만큼 이번 대회는 만족할만한 성과를 거뒀다고 생각한다. 특히나, 최근 recommender system 도메인 내에서 sequential recommendation 논문들이 많이 출간된만큼 대회를 위해 논문들을 리서치하며 최신 트렌드에 대해 어렴풋이 감을 잡을 수 있었고 대회의 데이터셋을 고려해 모델의 특성에 따라 어떻게 적용해 볼 수 있을까 고민하는 시간을 가지며 많이 성장함을 느꼈다.

Sequential recommendation 중에서는 SASRec, BERT4Rec, CL4SRec 모델이 기억에 남는데 SASRec 은 팀 차원 랩업 리포트에서 다뤘으니 넘어가도록 하고 유저의 과거로부터 현재까지의 시간 상 기록이 정말로 시간에 따라 정렬이 된 건지 의문을 던지며 마스킹된 아이템의 전 시점과 후 시점의 맥락을 고려하는 BERT4Rec 이나 contrastive learning 을 통해 유저만의 고유한 representation 을 어떻게 잘 학습할 수 있을까에 대한 고민이 담겨있는 CL4SRec 은 이번 대회에서 sequential 한 아이템보다 static 한 아이템 예측에 적합하지 않을까 하며 적용을 시도하고 실패했지만 두 모델은 학습하고자 하는 목표가 명확하고 개인에 특화된 추천에 적합하기에 추후에도 계속 시도해볼만한 모델이라고 생각한다.

이번 대회는 기존의 대회와 다르게 리서쳐 역할에 집중했기 때문에 PM 역할을 맡기는 하였으나 비중이 내려간 측면이 있었다. 프로젝트 기간 중에 회의 막바지에는 Jira 화면 공유를 통해 팀 내에서 해결해야 하는 이슈를 실시간으로 추가했고 역할도 배정해 팀 내에서 해결해야 되는 부분을 명확하게 전달하였다. Jira 에서 타임라인을 지속적으로 설정해 팀 차원에서 데드라인에 맞춰 체계적으로 진행하려 하였으나 이 부분에 대해서는 미흡했고 대회 진행 양상이 예상과는 달라 어쩔 수 없었지만, 여전히 개선해야 되는 부분이라고 생각된다.

프로젝트를 진행하면서 스스로에게 아쉬웠던 부분은 아무래도 구현에서 굉장히 아쉬웠다. 논문을 읽고 이해하는데에는 전혀 문제가 없었지만 이해한 부분을 코드로 어떻게 옮겨볼지에 대해서는 바로 떠오르지 않았고, 이는 예상보다 더 많은 시간을 소요하게 만들

거나 심지어 잘못된 구현으로 제대로 된 성능이 나오지 않아 스스로에게 실망감을 느낄 때도 있었다.

따라서, 앞으로 한 달 정도 마지막 프로젝트 전 여유 시간이 어느 정도 있어 이 기간 동안 과거 과제를 참조해 마지막 대회에서 논문 리서치를 할 때 구현에 대한 감이 올 수 있을 정도로 실력이 향상되는게 목적이지만 쉽지 않다고 느끼지만 해내고 싶다.

아직 기업 해커톤을 앞두고 있지만 이번 대회는 지금까지 진행한 네 개의 대회 중 가장 기억에 남을만한 대회였다. 대회를 위해서 논문을 하루에 4 편 이상 보고 새벽에 잠을 켜 팀원들과 함께 대회에 대한 아이디어를 공유하며 어떻게 발전시킬 수 있나 같이 고민했던 경험은 정말 소중한데 마지막 기업 해커톤도 이번 프로젝트와 같이 열정적으로 팀원과 함께 진행하고 싶은 열망이 강하다.

개인회고 (박준하_T7523)

이번 프로젝트에서 가장 중점을 둔 목표는 다양한 논문을 읽고 이를 직접 구현하는 것이었다.

처음으로 시도한 모델은 그래프 기반의 모델이었다. 그래프 기반 모델 중 대표적인 모델인 LightGCN 을 우선 구현해보았는데, 생각보다 낮은 성능을 기록하였다. 추측컨데, 데이터가 매우 dense 해 LightGCN 의 smoothing 이 오히려 임베딩의 학습을 방해하는 것으로 판단된다. 실제로 실험 결과 layer 를 줄일수록 더 나은 성능을 보였으며, layer 가 없는 기본적인 MF 모델이 성능이 가장 높음을 확인할 수 있었다. 이로부터 모델의 성능은 데이터에 따라 크게 달라질 수 있으며 모든 경우에 적용 가능한 완벽한 모델은 없다는 사실을 실감할 수 있었다.

그래프 기반 모델에서 한차례 실패를 겪은 후, 다른 조원들로부터 EASE 모델이 매우 높은 성능을 내는 것을 확인하였다. 그렇기에 EASE 와 동일한, auto encoder 기반의 다양한 모델들을 구현해보았다. 대표적으로 Multi-VAE 와 Multi-DAE, SVD-AE, RecVAE 을 각각 구현해보았는데 아쉽게도 EASE 의 성능에 미치지 못하였다. 하지만 이에 그치지 않고 성능이 상대적으로 뛰어나면서 튜닝 가능한 RecVAE 모델의 파라미터와 구조를 직접 수정해보았으며 그 결과 약간의 성능향상을 이룰 수 있었다. 특히 유의미한 성능향상은 입력에 normal distribution 에서 sampling 한 noise 를 추가로 넣어 모델이 denoising 역할에 더 집중하도록 한 것에서 일어났다. 이렇게 직접 모델을 수정해보는 과정은 모델에 대해 깊이 생각해볼 수 있었던 의미있는 경험이었다.

이번 대회에서 느낀 아쉬운 점은 시간과 지식 부족으로 인해 원하는 논문을 전부 구현하지 못했다는 점이다. 일례로 이번 데이터에서 denoising 메커니즘이 잘 동작하는 것으로 보아 Blurring-Sharpening Process Model(BSPM)도 잘 동작할 것으로 추측했지만 시간 부족으로 인해 접근하지 못했으며, 이 점이 매우 아쉬웠다. 추후에 시간을 들여 공부하고 직접 구현해야겠다고 다짐하였다.

이번 프로젝트는 단순히 모델을 구현하는 데 그치지 않고, 모델의 본질을 이해하며 개선하는 의미 있는 경험이었다. 이러한 경험을 발판 삼아 앞으로 더 깊이 있는 연구와 실험을 진행하고자 한다.

개인회고 (박태지_T7524)

이번 프로젝트를 진행하며 리서치를 통해 논문을 읽고, 관련 GitHub 자료를 참고하여 데이터와 문제에 적합한 모델을 구현하려고 시도를 하였다. 허나, 지식 부족으로 인해 원하는 만큼의 성능을 끌어내지 못했다. 특히 최신 논문을 읽고 이를 실제 프로젝트에 적용하려 했지만, 구현 과정에서 실패한 점은 아쉬움으로 남았다. 이는 최신 기술을 다루기 위한 기초적인 이해와 실력을 더 탄탄히 해야겠다는 필요성을 절실하게 느끼게 해준 계기가 되었다.

과거 회고에서 한 모델에만 집착하지 않고 다양한 모델을 시도하겠다고 다짐했던 만큼 이번 프로젝트에서는 이를 실천하려고 노력했지만, 생각만큼 끈기 있게 임하지 못했던 점이 아쉽다. 한 모델에서 성능이 기대에 미치지 못하면 그 원인을 분석하고 발전시키려는 노력을 하기 보다는, "이 모델은 성능이 잘 안 나오니 더 이상 시도할 필요가 없다"고 스스로 단정 짓고 넘어간 경우가 있었다. 이를 통해서 한 모델에 대해 깊이 고민하고 개선점을 탐구하는 습관을 들여야겠다는 중요한 교훈을 얻었다

프로젝트 기간 동안 팀끼리 Zoom 을 지속적으로 켜두고 팀원들과 작업하며 의견을 실시간으로 공유한 점은 소통의 질을 높이는 데 크게 기여했다고 생각한다. 작업 도중 새로 발견한 인사이트나 유의미한 정보를 즉각적으로 공유하고 피드백을 받을 수 있었기에 팀워크와 협업의 장점을 충분히 발휘할 수 있었다.

앞으로 다가올 다음 대회를 준비하며, 이번 프로젝트의 경험을 바탕으로 부족한 점을 보완하려고 한다. 최신 논문을 더 깊이 이해하고 구현 능력을 키우기 위해 기초적인 부분부터 다시 공부하며 실력을 쌓을 계획이다. 또한, 한 모델에 대해 더 끈기 있게 탐구하고 개선점을 찾아 발전시키는 태도를 기르려 한다. 이번 프로젝트는 부족함을 돌아보게 함과 동시에 성장의 가능성을 발견한 소중한 경험이었다고 생각한다.

개인회고 (배현우_T7525)

추천시스템을 다루는 처음 프로젝트로서 배운 점도 많았지만, 그보다도 어려운 점들이 많음을 깨달았다.

정형데이터를 활용하는 ML 프로젝트와 비슷하게 생각하여 베이스라인 코드를 짜보겠다 했지만 전혀 다른 느낌이었어서 레퍼런스 코드를 보는 날들이 많았다. 여러 코드들을 참고하며 실력이 많이 늘었지만 여전히 여러 모델을 하나의 구조에 녹여내는 것에 큰 어려움을 겪었고 인풋으로 들어갈 데이터의 형태와 모델에서 아웃풋으로 나오는 정보의 형태에 관해 고민이 많았다.

생각보다 상호작용 데이터 외에 추가로 컨텍스트 정보를 활용하기가 어려워서 이번 프로젝트 동안 모델 중심으로 진행하게 되었는데 여러 모델의 작동 구조를 알아보며 아이디어 등 배운 점이 많았고 여러 시도를 해본 것이 좋았던 것 같다. 그러나 동시에 많은 모델을 시도해보려다 보니 각 모델에 대해 깊이 있게 탐구를 진행하지는 못했다.

추천시스템에 대해 어떤 도메인인지 알 수 있던 좋은 프로젝트였고 우리의 데이터셋, 상황에 맞게 모델을 수정하여 성능을 향상시켰던 경험이 소중한 것 같다. 실험으로 이어지지 못한 가설들에 미련이 남고 sparse 한 데이터였다면 시도했던 모델들 중에 어떤 것이 잘 작동하였을지가 궁금증으로 남는다.

개인회고 (신경호_T7530)

약 4 개월간 많은 강의를 수강하고 프로젝트를 진행하면서 다양한 지식을 축적하였다고 생각했지만 제대로 된 기반을 다지지 않은 채로 무작정 쌓아 올렸다는 생각이 들었다.

모든 프로젝트에서 중요한 점을 고르라고 하면 베이스라인을 구축하는 것으로 생각한다. 이전의 베이스라인 구조가 너무 이상적인 나머지 그것을 따라 하려고 고민하고 구현하는데 시간을 투자하였지만, 진척이 없었던 것이 마음에 많이 걸린다.

다양한 모델을 적용하려고 시도하였지만, 성공적으로 마치지 못했다. FM 모델을 적용하려고 코드를 작성하였지만 제대로 동작하지 않았고, 이를 보완하기 위하여 negative sample 을 구현하여서 추가로 입력하였지만, 성능에 변화가 없어서 embedding 을 구현하는 과정에서 오류가 있었던 것으로 생각하였다. 또한, LGBM 을 이용한 모델을 구현하였는데 이 역시 성능이 너무 낮게 나와서 데이터 처리에 문제가 있었던 것으로 생각한다.

차후에는 RecBole 을 이용하여 BERT4Rec, ADMMSLIP 을 이용하여 준수한 결과물을 내었지만, 라이브러리를 끌어다가 쓴 것이라 정작 실력 향상에는 큰 도움이 되지 못하였다는 생각이 든다. 나중에 내가 필요한 부분을 논문에서 찾아서 튜닝을 해야하는 상황을 고려하고 있기에 pytorch 에 대한 복습과 지금까지 읽은 논문을 다시 한번 분석하는 시간을 가져야 할 것 같다.

마지막 프로젝트까지 남은 시간에 serving 에 관한 강의를 들으면서 모델에 대한 복습을 진행할 것이다. 또한 구현 측면의 능력뿐만이 아닌 베이스라인을 구축하는 등 구조적인 지식을 키워서 지금보다 개인적 또는 팀적으로 도움이 되는 한 명의 일원이 될 것이다.

개인 회고 (이효준_T7545)

이번 프로젝트는 해커톤 이전 마지막 프로젝트였기에 부스트캠프 내에서 학습한 내용들을 최대한 활용하는 것을 목표로 설정했다. 특히 추천시스템 모델에 대한 강의와 과제를 참고하여 모델을 구현하는 것을 최우선 과제로 두었으며 다른 캠퍼들이나 멘토님, 마스터님들이 사용했던 분석기법이나 논문리뷰를 통해 얻은 지식을 응용하고자 노력했다.

프로젝트 초기 베이스라인 코드 개발 파트를 담당하며 가장 많이 고민한 부분은 train과 validation 데이터셋을 최대한 ground truth와 유사하게 구분하는 것이었다. 이를 위해 random하게 masking된 데이터와 유저별 마지막 sequence에서 제외된 데이터의 비율에 대해 분석하며 최적의 비율을 찾고자 노력했다.

또 EDA를 통해 유저와 아이템의 특성을 살피고자 노력했다. 각 평점 데이터들을 결측치 유무의 모든 조합에 따라 분석하며 결측치가 유저와 아이템의 상호작용에 어떤 영향을 미치는지 자세히 살펴보고 유저별 마지막 n개 데이터의 아이템별 인기도, 평가 간격 등을 분석하며 유저의 특성을 분석했다.

먼저 train valid split을 적절하게 진행하여 public score와 유사한 검증 성능을 보이는 비율을 찾을 수 있었고 이를 통해 보다 효과적인 실험을 진행할 수 있었다.

또한 유저와 아이템의 특성과 상호작용의 특성에 대한 심층적인 분석을 통해서 상호작용에 대한 각 피쳐들의 영향을 파악함으로 side-information으로 활용할 피쳐를 선정하고 그를 활용할 방법에 대한 인사이트를 얻을 수 있었다.

이전 프로젝트들과 비교하여 새롭게 시도한 업무는 모델링에 참여한 것이다. 모델 구현을 진행하고 side-information을 활용한 실험을 진행한 결과, Negative Sampling부터 Loss까지 고려할 부분이 많으나 dense한 데이터의 특성으로 인해 그 성능에 아쉬움이 있음을 확인하고 팀 실험 범위를 축소하여 실험 시간을 대폭 축소할 수 있었다.

또한 추천 결과의 후분석을 진행한 것 역시 새로운 시도였다. 이를 통해서도 효과적인 앙상블 기법을 선정할 수 있었으며 모델의 특성과 inductive bias를 보다 확실하게 분석할 수 있었다.

이번 프로젝트에서는 모델 구현에서의 어려움이 가장 컸다. 특히 side-information를 활용하는 모델의 구현에서 다중값과 고유값이 많으며 결측치 또한 많은 컬럼의 경우 적절한 데이터 처리가 어렵다는 것을 알 수 있었고 적절한 Negative Sampling과 Loss의 설정 역시 까다로운 요소이자 중요한 요소임을 알 수 있었다. 그리고 관련 도메인의 현업 경험이 있는 멘토님과의 미팅을 통해 도메인 지식과 프로젝트 경험의 중요성에 대해서도 다시금 깨닫게 되었다.

각자 파트를 나누어서 실험을 진행하고 실시간으로 코드를 공유하면서 코드를 개선해 나간 점은 굉장히 좋았으나 개별적으로 모델 구현을 진행하다보니 파이썬 파일을 활용한 베이스라인을 짜기에 어려움이 있었다.

이번에 마주한 가장 큰 한계는 대회와 현업의 간극이었다. 현업에서는 side-information을 활용할겠지만 이번 프로젝트에서는 그를 활용하지 못했고 더 나아가 Netflix의 썸네일 개인화처럼 이미지나 다른 유형의 데이터를 활용한 멀티모달 모델 또한 사용할 수 없었다. 또한 실질적인 추천보다는 예측에 가까운 문제였다는 점과 유저의 시청시점이나 시청시간이 아닌 평가시점 데이터를 제공받은 것도 아쉬움이 남았다.

다음 프로젝트에서는 모델의 inductive bias를 보다 면밀히 분석하여 데이터에 최적화된 모델을 선정하고, 효과적인 모델 구현을 위해 노력하겠습니다. 또한 단순히 모델을 구현하는 데 그치지 않고 문제 상황에 맞게 모델을 유연하게 변형하여 최적의 해결 방안을 도출하는 것을 목표로 할 것입니다.