



[Recsys 8조 EXIT] Movie Recommendation Wrap Up Report

1.1 프로젝트 개요

RecSys Stages

Movie Recommendation

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

#부스트캠프5기 #추천시스템 #영화추천

D - 1 | 2023.05.31 ~ 2023.06.22 19:00

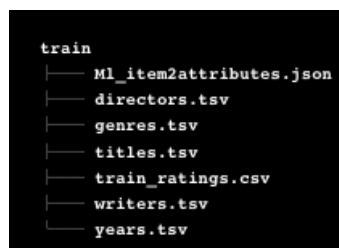
23팀

프로젝트 주제

- 사용자의 영화 시청 이력 데이터 (implicit feedback) 를 바탕으로 사용자가 다음에 시청할 영화 및 좋아할 영화를 예측하는 대회

데이터 개요 및 구조

- 데이터 셋



- Ml_item2attributes.json
 - 전처리에 의해 생성된 데이터(item과 genre의 mapping 데이터)
- directors.tsv
 - 영화별 감독
- genres.tsv
 - 영화 장르 (한 영화에 여러 장르가 포함될 수 있음)
- titles.tsv
 - 영화 제목
- train_ratings.csv
 - 주 학습 데이터, userid, itemid, timestamp(초)로 구성

- writers.tsv
 - 영화 작가
- years.tsv
 - 영화 개봉년도

실험 환경 및 사용한 툴

- (팀 구성 및 컴퓨팅 환경) 5인 1팀, 인당 V100 서버를 VSCode와 SSH로 연결하여 사용
- (협업 환경) GitHub, Notion
- (의사 소통) Slack, Zoom

1.2 프로젝트 팀 구성 및 역할

김지우_T5063	CDAE 모델 제작 및 실험, Ensemble
박수현_T5085	SASRec 모델 제작 및 실험, Ensemble
석예림_T5110	RecVAE 모델 제작 및 실험, Ensemble
임소영_T5172	EDA, EASE 모델 제작 및 실험
전중원_T5185	lightGCN 모델 제작 및 실험, Ensemble

1.3 프로젝트 수행 절차 및 방법

1.3.1 Auto-Encoder

EASE

모델 선정 이유

- Sparse 한 데이터에 robust 한 EASE 를 도입하여 최적의 성능 도출
- Closed Form 으로 구현되어 모델 구현 및 실행 속도가 빠름

프로젝트 수행 결과

- 하이퍼파라미터인 lambda 값과 matrix를 채우는 value 값을 적절히 조절하여 성능 향상
 - Lambda : 300 / value : 0.9 → 0.1600
 - Lambda : 300 / value : 1.0 → 0.1599
 - Lambda : 300 / value : 0.8 → 0.1598
 - Lambda : 150 / value : 0.9 → 0.1592
 - Lambda : 500 / value : 1.0 → 0.1422

RecVAE

모델 선정 이유

- CF 모델 중 Auto encoder 기반 Top-N 추천에 최적화 된 모델
- 정규 분포를 가정하여 잠재 변수를 샘플링 하며, 잠재 공간에서의 연속적인 특징 표현
- Encoder, Decoder간의 번갈아 가며 업데이트 하는 Alternating Update를 통해 상호작용을 재구성하는 능력 향상

Dataset 구성

- user_id의 인덱스를 랜덤하게 섞고, 데이터 세트의 마지막 일부 사용자를 validation set으로 구성

프로젝트 수행 결과

- Baseline RecVAE 구현 → public score 기준 **0.1307**
- 기존의 랜덤하게 10개 선택하여 validation set 구성에서 최소 item 수 보다 적은 item과 user를 가진 데이터를 제외하고 랜덤하게 validation set 구성 → public score 기준 **0.1324**
- optuna 통해서 얻은 hyperparameter로 모델 학습 → public score 기준 **0.1268**
- 마지막 epoch의 모델 inference에서 가장 성능이 좋은 epoch의 모델 inference로 변경 → public score 기준 **0.1447**

CDAE

모델 선정 이유

- Noise를 추가한 user-item interaction을 원래의 interaction으로 복원하는 과정에서 latent representation을 학습
- Input Layer에 User Node를 추가하여 User Bias를 모델링
- 다른 CF 모델들의 일반화된 형태이면서 동시에 높은 유연성을 가지는 모델

Dataset 구성

- 각 user_id별로 interaction이 있었던 10개의 아이템을 random하게 sampling하는 방식으로 validation set 구성

프로젝트 수행 결과

- Training
 - 원래 CDAE 모델은 부분적으로 corrupted된 user-item interaction으로부터 latent representation을 학습하여 data point를 복원하기 위해 하나의 hidden layer를 사용함
 - 성능 개선을 위해 hidden layer의 수를 늘리거나 corruption probability, mapping function, latent dimension 등을 바꿔가면서 다양한 실험 진행
 - mapping function의 경우, identity mapping보다는 sigmoid나 tanh function을 적용해 **non-linearity**를 반영하도록 했을때, 성능 향상을 보임
 - corruption probability의 경우, corruption probability를 **높일수록** 성능 향상을 보임 → Input에 noise를 추가하여 모델이 **robust한 representation**을 학습하도록 함
 - 원래의 CDAE 모델은 하나의 hidden layer만 가지는 형태로 구성되어 있지만, neural network structure를 깊게 쌓아보는 시도를 함 → hidden layer가 4 이상이 되면 성능 하락을 보임. hidden layer의 수가 3까지는 각 layer의 latent dimension에 따라 다른 양상을 보임

- latent dimension의 경우, hidden layer의 수에 따라서 다른 양상을 보임 → hidden layer를 하나만 두었을때는 k = 50보다 k = 300일때 높은 성능을 보였지만, hidden layer를 여러개 두었을때는 각 layer마다 다른 latent dimension의 개수를 지정할때 더 높은 성능을 보임
- validation set recall@10 : **0.17658**
- Public Score
 - mapping function, corruption probability, latent dimension 변경, hidden layer 추가로 **public score** 기준 **0.1088**에서 **0.1157**으로 성능 향상

1.3.2 Transformer

SASRec

모델 선정 이유

- Public/Private dataset의 경우 각 유저의 마지막 아이템 뿐만 아니라 Sequence 내부에 아이템 또한 샘플링 되었기 때문에 Static Model과 Sequential Model의 앙상블로 해당 대회를 진행하기를 희망했음
- 대부분의 팀원 분들께서 AutoEncoder 기반의 Static Model들을 맡아주신 만큼, 최종 앙상블을 위해 Sequential Model의 공백을 메우고자 Sequential Modeling을 맡음
- Sequential Model로 SASRec을 선택한 이유는, Max_Seq_Len가 300이상에서 성능이 잘 나오는 이번 대회 데이터 특성 때문에 Long Term Dependency에 대응할 수 있는 Transformer 계열의 모델을 선택함
- 최종적으로 GTN(Gated Transformer Networks)와 SASRec의 성능을 비교하였고, 성능이 더 좋은 SASRec을 선택함

Dataset 구성

- TrainSet
 - input : user_sequence*[-max_seq_len-3 : -3]
 - output : user_sequence*[-max_seq_len-2 : -2]
- ValidationSet
 - input : user_sequence*[-max_seq_len-2 : -2]
 - output : user_sequence*[-2]
- TestSet
 - input : user_sequence*[-max_seq_len-1 : -1]
 - output : user_sequence*[-1]

*user_sequence : 각 유저의 영화시청 기록

프로젝트 수행결과

수행절차	Recall@10(Public LB)
(06/19) Baseline SASRec 설계 및 max_seq_len = 300	0.0898
(06/20) 클래스 내부에 선언된 객체(필드)의 Type 변경 - 해당 객체의 Type을 List에서 ModuleList로 변경하여 해당 객체가 원활히 학습되도록 오류를 수정함	0.0898 → 0.0954

수행절차	Recall@10(Public LB)
(06/20) Multi-Head Attention Structure 경량화 + Activation function 변경 (ReLU → GeLU) + Attention Score Dropout 추가 - 베이스라인 코드에서 제공된 SASRec과 비교하기 위해 유사하게 수정했는데, 모델 파라미터 수는 적으면서 성능은 유사해서 채택함 - 변경 전/후의 Valid/Test Recall@10의 성능만 확인했으며, Public LB를 확인하지는 않음	≈ 0.0954 (not checked in LB)
(06/21) Data Augmentation 추가 - 어떠한 유저의 경우 시청기록이 max_seq_len이 훨씬 넘는 경우가 있어 모두 학습 데이터로 활용되기를 희망했음 - 각 유저의 모든 영화시청 기록을 고려하여 Data Augmentation을 진행함	0.0954 → 0.0867
(06/21) Data Augmentation 제거 및 hidden_size = 256 설정	0.0867 → 0.0998
(06/21) Negative Sampling Logic 변경 - 베이스라인 코드에서 제공된 Negative Sampling의 경우 데이터에서 언급된 6807개의 아이템 이외에 대해서도 샘플링이 이루어지기 때문에 수정함	0.0998 → 0.1230
(06/22) 수정된 Data Augmentation + Hyper Parameters Tuning + Overshooting 문제 대응 - 각 유저의 모든 영화시청 기록을 고려한 Data Augmentation의 경우 성능이 하락했기 때문에 Recent Past만 고려한 수정된 Data Augmentation을 진행함 - Weight_decay = 0으로 설정되어 있어 발생한 Overshooting 문제에 대응하도록 lr = 0.001로 학습한 모델을 Pretrain Model 삼아 lr = 0.0001로 재학습 시킴	0.1230 → 0.1311
(06/22) Seed Ensemble - Weight Initialization에 따라 성능이 크게 바뀌는 것을 포착해 Seed 값을 바꿔 다양한 성능을 가지는 모델들을 만들고, 이러한 모델들을 앙상블 하려고 시도했으나 시간 내에 마무리 하지 못함	not completed..

1.3.3 GCN

lightGCN

모델 선정 이유

- 다른 팀원들이 AE, transformer 기반의 모델을 맡았기 때문에 그래프 기반의 모델을 활용하여 다른 모델은 포착하지 못하는 부분을 lightGCN을 통해 포착해보고 싶었음

구현 방식

- using recBole library
 - recBole library를 활용하여 이미 구현되어 있는 lightGCN을 활용

data preprocessing

- user와 item의 값이 동일한 값이 존재하기 때문에 모델이 user와 item을 명확히 구분하지 못할것이라 생각
- user와 item의 값을 겹치지 않는 새로운 값으로 할당
- recBole에서 원하는 데이터 입력 형식이 있어 원하는 방식으로 변경

프로젝트 수행 결과

- 학습
 - validation_set recall@10: 0.1273

- test_set recall@10: 0.1386
- 제출
 - recall@10: 0.0022

실패 원인 분석

recBolt library 사용법 미숙	recBolt library의 train부분까지는 원활하게 진행되었지만 inference부분에서 값이 원하는 형태로 나오지 않음
user와 item의 인덱싱	그래프 기반 모델의 학습시 패딩을 고려하지 못하여 인덱싱이 제대로 매칭이 되지 않았음

1.3.4 성능 고도화

Optuna

RecVAE

- latent dimension, not_alternating, learning rate, epoch, batch size 등 optuna를 활용해 모델의 학습에 필요한 hyperparameter tuning 진행
 - 적절한 범위를 설정하지 못하여 학습시간 오래 걸림
 - 고정 시킬 수 있는 hyperparameter도 tuning을 시도해 보아 성능 향상에 실패
- 기존의 baseline 보다 높은 성능을 보이지 않아 optuna를 통해 얻은 hyperparameter 사용하지 않음

CDAE

- corruption probability, latent dimension, learning rate, batch size 등의 hyperparameter를 optuna를 이용해서 tuning 진행
 - baseline보다 높은 성능을 보이지 않아서 최종적으로 사용하지 않음
 - 1) 적절한 hyperparameter의 범위를 지정하지 못함
 - 2) hidden layer의 수와 latent dimension 간의 상관관계가 있는데 모든 layer의 latent dimension을 동일하게 지정하니 적절한 hyperparameter를 지정하는데 실패함

Ensemble

- CF 모델의 추천 결과와 Sequential Recommendation의 추천 결과에 차이가 있기 때문에 앙상블을 통해 성능을 높이고자 함

Version 1

- 각각의 모델별로 가중치를 입력받음
 - 이때 가중치는 자연수이며 합이 10
- 그 후 각각의 모델의 각각의 user 별로 상위에서 가중치만큼 추출하여 한 파일로 합침
 - 예) sasRec, EASE, recVAE에 가중치를 각각 5, 4, 1을 부여
 - sasRec의 결과물 csv파일에서 user별로 상위 5개의 item을 추출
 - EASE의 결과물 csv파일에서 user별로 상위 4개의 item을 추출
 - recVAE의 결과물 csv파일에서 user별로 상위 1개의 item을 추출
 - 각각의 추출물을 합쳐서 하나의 csv파일로 저장하여 앙상블 완료

- 만약 첫번째 csv이후의 csv파일에서 추출하다 앞에서 이미 추출했던 item이라면 그 아이템은 건너뛰고 다음의 아이템 추출

양상블 결과

EASE	sasRec	recVAE	recall@10 public → private
4	5	1	0.1746 → 0.1612
5	4	1	0.1747 → 0.1627

- EASE 단일 모델의 결과: 0.16
- sasRec 단일 모델의 결과: 0.1311
- recVAE 단일 모델의 결과: 0.1447
- 양상블 후 public score값이 0.1747까지 상승

Version 2

- 각각의 모델 별로 가중치 입력
 - 가중치는 float이며 합은 1
- 각 모델의 결과값에 user별로 rank를 10부터 1까지 부여
- EASE 모델의 결과와 SASRec 모델의 결과에서 동일한 item을 추출
 - EASE 모델과 RecVAE 모델은 AutoEncoder 기반 모델로 50~60% 정도 결과 값 동일하여 EASE 모델과 SASRec 모델에서만 동일한 item 추출
- 동일한 item을 제외한 나머지에 대해 가중치와 weight를 곱하여 (10 - 동일한 item 갯수)만큼 추출
- 만약, 동일한 item이 하나도 없다면, 가장 성능이 좋았던 EASE모델에서 전부 추출

양상블 결과

EASE	SASRec	RecVAE	recall@10 public → private
0.4	0.4	0.2	0.1684 → 0.1618

Version 3

- 고려한 모델들의 출력에 대해 공통으로 존재하는 아이템들은 모두 추천리스트에 넣어줌
- (10 - 공동 아이템들 갯수)에 대해서는 고려하고 있는 모델들의 출력을 번갈아 탐색하면서 추천리스트에 없는 높은 스코어의 아이템을 추천리스트로 넣어줌으로써 유저마다 10개의 추천리스트를 완성함

양상블 결과

고려한 모델들	recall@10 public → private
EASE, SASRec	0.1784 → 0.1642

1.4 프로젝트 수행 결과

public 2등

- Recall@10 (0.1784)

순위	팀 이름	팀 멤버	Recall@10 ↕	제출 횟수	최종 제출
2 (-)	RecSys_08조		0.1784	29	4d
1	RecSys_05조		0.1864	47	4d
2	RecSys_08조		0.1784	29	4d

private 4등

- Recall@10 (0.1642)

순위	팀 이름	팀 멤버	Recall@10 ↕	제출 횟수	최종 제출
4 (2 ↕)	RecSys_08조		0.1642	29	4d
1	RecSys_05조		0.1714	47	4d
2	RecSys_10조		0.1663	26	4d
3	RecSys_04조		0.1654	80	4d
4	RecSys_08조		0.1642	29	4d

1.5 자체 평가 의견

잘했던 점

- 지우 : 논문을 읽고, 어떻게 모델을 개선해나갈 수 있을지 아이디어를 얻어서 실험해보는 과정을 경험한 것이 좋았음
- 증원 : 멤버들간의 분업이 효율적으로 이루어져서 한정적인 시간내에 좋은 결과를 만들어낸것 같음
- 예림 : 다양한 앙상블 기법에 대해 팀원들 간 의견을 공유하고, 실험 해 보면서 효율적인 결과를 도출해 낼 수 있었음
- 소영 : 각자 철학을 가지고 모델을 선택한 뒤, 최적의 결과를 도출하기 위해 여러 가지 앙상블 방식을 도입한 것이 좋았음
- 수현 : 부족한 시간에 대응하고자 빠르게 Baseline Model 설계 및 Valid/Test Set을 설정해서 신속한 실험 진행했음

아쉬웠던 점

- 지우 : public score와 valid score가 align이 안되었을때, 원인 분석이 충분히 이루어지지 못했음
- 증원 : 그래프 기반의 모델이 결과를 냈으면 좀 더 다양한 방면의 모델들의 앙상블을 통해 결과의 향상을 이뤄 낼 수 있었을것 같은데 결과를 내지 못해 아쉬웠음
- 예림 : optuna를 통해 최적의 hyper parameter를 구하고 싶었지만, 범위 설정 및 변동될 hyper parameter를 적절하게 지정해 주지 않아서 예상했던 성능보다 충분히 나오지 않았음

- 소영 : 프로젝트와 대회를 병행하다보니, 대회에 온전히 집중할 수 있는 시간이 매우 적어 타이트한 일정으로 진행되었다. 다양한 모델을 실험해보지 못한 것이 조금 아쉬움
- 수현 : 시간 부족으로 어느정도 구체화했던 아이디어를 적용해보지 못해 너무 아쉬움

배운점 & 시사점

- 이론적으로 성능향상이 보장되는 앙상블은 적극적으로 시도해야겠다.
- 하이퍼 파라미터를 잘 설정하면 최종적으로 성능 향상을 이끌어 낼 수 있다.
- Top-K Recommendation에서 Negative Sampling을 활용하면 좋은 성능을 이끌어 낼 수 있다.

2.1 개인회고

김지우_T5063

- 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?
 - 논문을 읽고 모델의 개념 및 구성요소를 이해하고, 어떤 방식으로 성능을 개선할 수 있을지 아이디어를 얻어서 다양한 실험을 해봤다.
 - 모델의 성능이 잘 나오지 않았을 때, 무엇이 원인인지 분석해서 문제를 해결했다.
 - 라이브러리를 사용하는 것이 아니라, pytorch로 짜여진 코드를 이해하고, 직접 성능을 높이기 위해 모델의 구조를 변경하거나 data를 handling하는 부분의 코드를 작성했다.
 - Recommendation Task의 다양한 앙상블 방법에 대해 고민하고, 아이디어를 제시했다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
 - 논문을 읽으면서 해당 모델이 어떤 강점을 가지고 있고, 왜 이전 모델에 비해서 성능이 개선 되었는지 이해할 수 있게 되었다.
 - 딥러닝 모델의 경우, 하이퍼파라미터에 크게 영향을 받기 때문에 tuning에 많은 신경을 써야하고, 또 optuna를 사용한다고하더라도 범위를 지정하는데 신중해야한다는 사실을 깨달았다.
 - Sequential Recommendation과 Static Recommendation의 결과가 상이하기 때문에 두 모델을 적절히 앙상블하는게 성능을 높일 수 있는 키포인트라는 점을 알게 되었다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
 - 주어진 Meta Data(Context Information & Feature Information)를 충분히 활용하지 못했다.
 - 시간이 부족해서 실험 과정을 꼼꼼히 기록하지 못한 것이 아쉬웠다.
 - public score와 valid score가 align이 안되었을때, 원인 분석이 충분히 이루어지지 못했다.
 - 다양한 모델을 실험해보지 못한 것이 아쉽다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?
 - Auto-Encoder 계열의 모델에 user와 item feature나 contextual information을 결합하는 형태도 고려해보고 싶다.
 - 어떤 시도를 했을때 어느 정도의 성능 향상이 있었는지 상세하게 기록하면 좋을 것 같다.

- public score와 valid score가 왜 차이가 나는지 원인을 상세하게 파악하고, 개선 사항이 있을지 생각해봐야겠다.
- 다양한 모델을 앙상블하는게 Recommendation Task에서는 중요하기 때문에, 다양한 모델을 고려해서 실험해봐야겠다.

박수현_T5085

- 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?
 - 부족한 시간에 대응하고자 빠르게 Baseline Model 설계 및 Valid/Test Set을 설정해서 신속한 실험 진행했다.
 - 모델 성능이 나오지 않는 상황에서 많은 고민 끝에 오류 및 개선사항을 포착해 문제를 해결했다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
 - 클래스 내부에 선언된 객체의 Type을 List에서 ModuleList로 수정해 오류를 해결하면서 Pytorch를 기계적으로 프로그래밍 해왔던 점을 깨달았는데, 내부의 철학을 이해할 수 있도록 노력해야겠다.
 - 학습과정에서 Overshooting 현상을 발견하면서 weight_decay = 0 인 것을 알게 되었다. 여전히 하이퍼파라미터의 정확한 기능들을 숙지 못한 점을 반성해야겠다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
 - Side Information을 모델에 반영하고 싶어 적용 방법을 어느정도 구체화 한 상황에서 시간 부족으로 적용하지 못한 것이 아쉽다.
 - Seed 값에 따라 모델의 성능이 많이 다른 점을 발견해 Seed Ensemble을 시도했으나 시간 내에 마무리하지 못했다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?
 - 여전히 뇌피셜로 근거 없는 결정을 내리는 잘못된 습관을 가지고 있는데, Data Driven Decision을 내릴 수 있도록 노력해야겠다.
 - Torch를 활용한 프로젝트를 기계적으로 프로그래밍을 하는 습관을 내부 철학을 따져보면서 구현할 수 있도록 고쳐봐야겠다.
 - 계속 미루고 있는 WandB를 꼭 활용해보고 싶다.

석예림_T5110

- 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?
 - Top-N 추천 대회에 적합한 모델을 선택하고, 논문을 통해 모델의 구조 및 이해도를 높였다.
 - 구현된 모델의 코드에서 성능을 높힐 수 있는 부분을 찾고, validation set의 구성을 실험하며 성능을 높여나갔다.
 - 다양한 ensemble 기법을 고안하고 구현하며 성능 향상을 위해 노력하였다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
 - validation set 구성을 변경, 마지막 epoch의 모델에서 가장 성능이 좋은 epoch의 모델을 inference하는 등 다양하게 성능을 높일 수 있는 방법이 존재함을 알 수 있었다.
 - optuna로 최적의 hyperparameter를 찾기위해 범위 지정이 중요한 것을 깨달았다.

- Static 모델과 Sequential 모델 모두 단일 모델로 성능이 괜찮았지만, Ensemble시 공통된 결과가 user 별 없는 경우도 존재했다. 따라서, ensemble시 어떤 전략을 취할 지가 중요했다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**
 - CF 기반의 모델이다 보니 대회에서 주어진 movie에 관한 정보 데이터를 활용하지 못한 점이 아쉬웠다.
 - 모델의 구조도 변경하여 성능 향상을 시도했지만, 완벽하게 습득한 모델이 아니라 한계가 있었다.
 - Optuna를 활용하여 최적의 hyperparameter를 찾으려고 노력했지만, 범위 지정 실패로 성능이 더 낮게 나왔다.
- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
 - 데이터 특성 별 적절한 모델을 선택할 수 있게 모델에 대해 충분한 이해도를 바탕으로 모델을 사용해야겠다.
 - 빠른 실험도 좋지만, 기록하는 습관을 가져 중복되는 실험을 하지 않도록 해야겠다.

임소영_T5172

- **나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?**
 - 이번 대회 역시, 모델에 대한 이해도를 높이는 것이 학습 목표였다. 논문을 읽으며 모델이 고안된 배경과 구현 방식에 대해 자세히 알 수 있었다.
 - 논문 중, loss function 을 multinomial 한 것으로 변경하면 성능이 더 좋을 것이라는 것이 언급되어있어서 변경할 수 있는 방법에 대해 모색했다.
 - FFM_DCN 모델 구현을 위해 Dataset, DataLoader, Model 을 구성하려 노력하였다.
- **내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?**
 - Hidden layer 가 없어도 좋은 성능을 내는 것을 알게되니 layer 가 깊을수록 무작정 성능이 좋을 것이라는 생각이 잘못되었음을 깨달았다. 역시 데이터와 실험을 통해 가설을 증명해내는 과정이 중요함을 다시 한번 깨달을 수 있었다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**
 - Loss function 을 multinomial 로 바꿀 수 있는 방법을 모색하던 중, 수학 지식의 한계로 인해 결과를 도출하지 못한 것이 매우 아쉬웠다.
 - 앞서 말했듯, 시간이 부족하여 FFM_DCN 모델을 완성하지 못한 것이 매우 아쉬웠다. User-item matrix 만 이용하여 Closed form 으로 예측하는 EASE 와는 달리, FFM_DCN 모델은 전반적인 프로세스를 모두 구현해야했는데 모델에 대한 이해도가 낮다보니 하루만에 구현하는 것이 불가능했다.
 - 하이퍼파라미터가 1개뿐이라 적절한 수치를 찾기 위해 optuna 를 활용하기보다는 직관적으로 생각하고 대입한 것이 아쉬웠다.
- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
 - 다음 프로젝트는 우리 팀에서 직접 모은 데이터를 이용하여 추천해주는 프로세스이다보니, 모델에 대한 이해도를 높이고 특정 데이터에 최적의 성능을 내는 모델이 무엇인지 자세히 공부해야겠다. Sparse 한 데이

터에는 어떤 모델이 좋은 성능을 내는지, sequential 한 데이터에는 어떤 모델이 좋은 성능을 내는지 등 sota 모델에 대한 정리가 필요할 것 같다.

전증원_T5185

- 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?
 - lightGCN을 사용하기 위해 이미 구현된 라이브러리를 탐색했다.
 - 라이브러리의 사용법이 구체적으로 정리되어 있지 않아 소스코드를 보며 코드를 탐색하였다.
 - 값이 범주형범수로 주어지는 경우의 양상블을 하기위해 다양한 방면으로 고민했다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
 - 라이브러리를 사용하여 결과내기에 실패하였다. 그 후로 라이브러리를 사용하여 모델을 적용하려 해도 모델의 특성, 모델이 원하는 데이터의 종류등을 기본은 알아야한다는 것을 깨달았다.
 - 양상블을 통해 recall@10이 0.16에서 0.1747까지 올라가는 경험을 하였다. 양상블이 각 모델들의 단점을 보완해준다는 깨달음을 얻었다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
 - 라이브러리를 통해 모델을 사용하려 해도 모델에 대한 충분한 이해가 있어야 했는데, 모델에 대한 충분한 이해가 없이 사용하려다 그래프 기반의 모델이 패딩을 갖는다는 사실을 알지 못하여 인덱싱에서 실수가 있었다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?
 - 다음 프로젝트 부터는 모델의 특성을 제대로 파악한 후 사용해야겠다는 생각을 하게 되었다.