

Book Recommendation Wrap-up Report

Recsys-3조 (INT로조)

팀 구성 : 김세훈T6020, 박시우T6060, 배준혁T6072, 백승빈T6075, 서동은T6076, 이현규T6142 총 6명

1. 프로젝트 개요

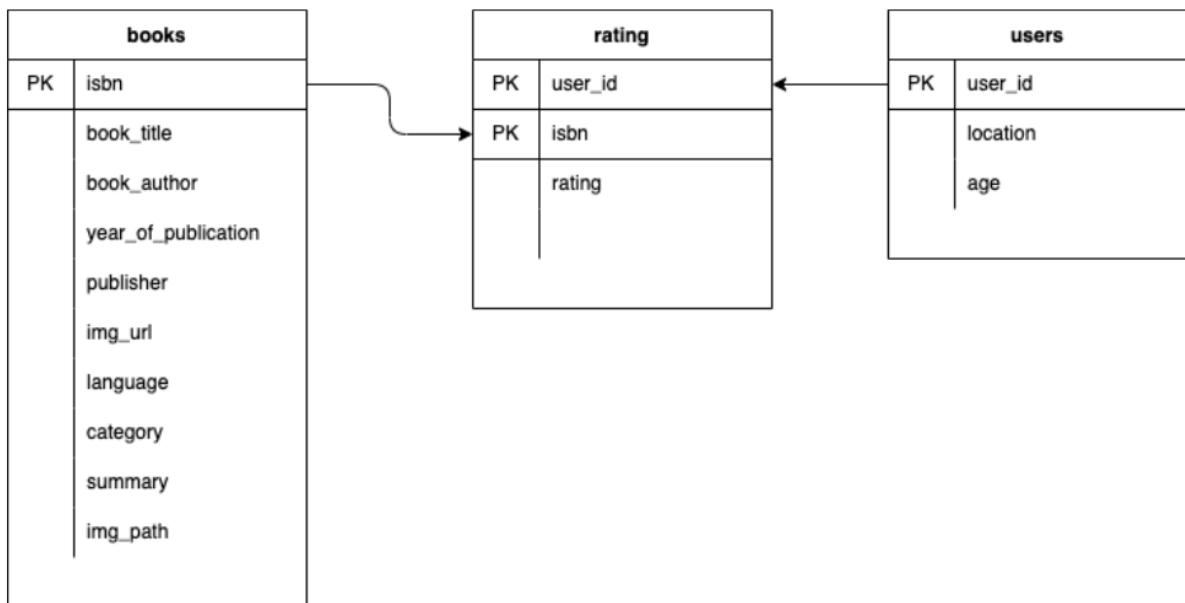
책과 관련된 정보와 소비자의 정보, 그리고 소비자가 실제로 부여한 평점, 총 3가지의 데이터 셋(users.csv, books.csv, train_ratings.csv)을 활용하여 각 사용자가 주어진 책에 대해 얼마나 평점을 부여할지에 대해 예측하는 것이다.

학습 데이터는 306,795건의 평점 데이터(`train_rating.csv`)이며, 149,570건의 책 정보 (`books.csv`) 및 68,092명의 고객 정보(`users.csv`) 또한 주어진다.

각각 데이터는 다음의 형태를 띄고 있다.

Input

- `train_ratings.csv` : 각 사용자가 책에 대해 평점을 매긴 내역
- `users.csv` : 사용자에 대한 정보
- `books.csv` : 책에 대한 정보
- `Image/` : 책 이미지



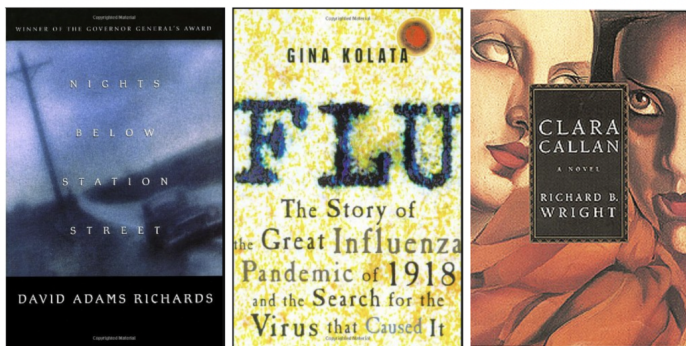
users.csv

user_id	location	age
8	timmins, ontario, canada	NaN
11400	ottawa, ontario, canada	49.0
11676	n/a, n/a, n/a	NaN
67544	toronto, ontario, canada	30.0
85526	victoria, british columbia, canada	36.0

books.csv

isbn	book_title	book_author	year_of_publication	publisher	img_url	language	category	summary	img_path
0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	en	['Actresses']	In a small town in Canada, Clara Callan reluct...	images/0002005018.01.THUMBZZZ.jpg
0060973129	Decision in Normandy	Carlo D'Este	1991.0	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	en	['1940-1949']	Here, for the first time in paperback, is an o...	images/0060973129.01.THUMBZZZ.jpg
0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999.0	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	en	['Medical']	Describes the great flu epidemic of 1918, an o...	images/0374157065.01.THUMBZZZ.jpg

image data



| train_ratings.csv

user_id	isbn	rating
8	0002005018	4
67544	0002005018	7
123629	0002005018	8
200273	0002005018	8
210926	0002005018	9

Output

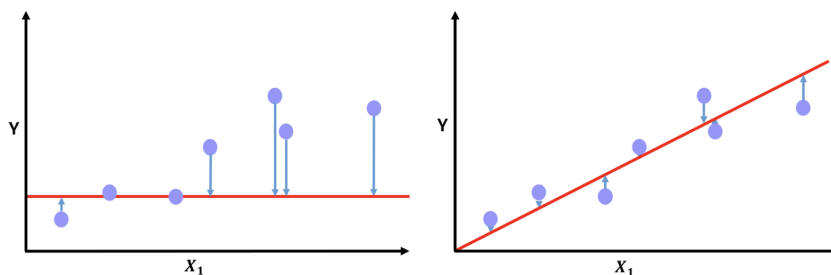
- test_ratings.csv 의 사용자가 주어진 책에 대해 매길 것이라고 예상하는 평점

1-1. 평가 데이터

- 26,167명의 사용자(user)가 52,000개의 책에 대해 남길 것으로 기대하는 76,699건의 평점(rating)을 예측한다.

1-2. 평가 지표

- 평점 예측에서 자주 사용되는 지표 중 하나인 RMSE (Root Mean Square Error)를 사용한다.
- 모델의 예측이 얼마나 잘못 되었는지를 수치로 나타내고 모델이 예측을 잘할수록 지표는 0 에 수렴한다.



RMSE Formula

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

1-3. 프로젝트 협업 전략

Git

- 해당 프로젝트의 협업툴로 Git을 사용한다.
- Git 관리 전략 중 GitHub Flow를 사용하며 Remote 중심 Branch 전략으로 배포의 중심이 되는 main 브랜치와 각각의 feature 브랜치로 구성하고 있다.

GitHub flow



<https://github.com/boostcampaitech6/level1-bookratingprediction-recsys-03>

Commit Convention

- **Issue**
 - 기능 추가 또는 테스트를 수행할 Task를 생성하고 수행시 새로운 브랜치를 만든다.
- **Pull requests**
 - 브랜치를 생성하여 작업이 완료되면 main 브랜치로 병합하는 과정에서 다른 팀원의 승인을 받는다.

Code Commit

- 구현한 기능에 대해 자유롭게 작성하되 머리말과 내용에 아래와 같은 형식을 따른다.
 - 머리말 : [(브랜치명 소문자)]
 - 내용 : #(이슈번호)

Header / Body / Footer

- 제목 머리말 : [(브랜치 이름 대문자)] (내용)
- 내용 : Issue 또는 Pull requests 템플릿에 따른다.

- 꼬리말 : Issue에 대한 Pull requests 처리시 Issue Tags란에 closed로 이슈 종료를 표시한다.

1-4. 서버 구성 및 개발 환경

서버 구성

- **AI Stage GPU Cloud 서버**
 - OS 및 버전 : Ubuntu 20.04.6 LTS
 - 성능 : V100 GPU

개발 환경

- 사용 언어 : Python
- 버전 정보 : 3.10.13
- 패키지 정보

```
numpy==1.26.2
pandas==2.1.4
matplotlib==3.8.2
scikit-learn==1.3.2
jupyter_client==8.6.0
jupyter_core==5.5.0
seaborn==0.13.0
torch==1.12.1+cu113
torchaudio==0.12.1+cu113
torchvision==0.13.1+cu113
```

2. 프로젝트 팀 구성 및 역할

2-1. 프로젝트 팀 구성

- 팀 이름 : INT로조
- 팀 인원 : 6명
- 팀 구성 : 김세훈T6020, 박시우T6060, 배준혁T6072, 백승빈T6075, 서동은T6076, 이현규T6142

2-2. 역할

김세훈 - EDA, 데이터 전처리, feature 특성별 중요도 파악, FM 모델 튜닝 및 성능 평가, WDN 하이퍼 파라미터 튜닝

박시우 - EDA, 데이터 전처리, 데이터셋 분할, 모델링, 하이퍼 파라미터 튜닝

배준혁 - EDA, 데이터 전처리, 깃허브 repo 관리, 앙상블, 하이퍼 파라미터 튜닝, 베이스라인 코드 보수

백승빈 - EDA, 데이터 전처리, 데이터셋 분할, 모델링, 하이퍼 파라미터 튜닝

서동은 - EDA, 데이터 전처리, 데이터셋 분할, 모델링, 튜닝

이현규 - EDA, 데이터 전처리, 하이퍼파라미터 튜닝, 모델링

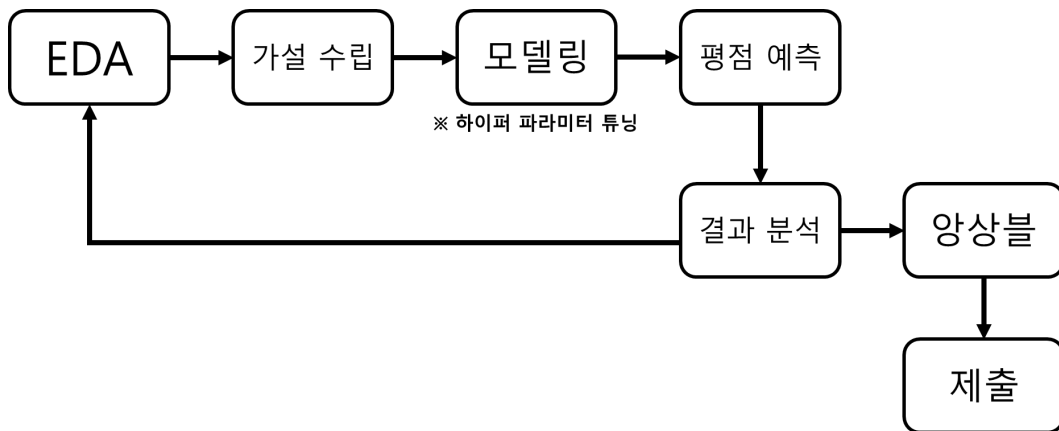
3. 프로젝트 수행 절차 및 방법

3-1. 프로젝트 타임라인

10	11	12	13	14	15	16
	강의수강				모델링	
			작업환경 세팅			
				EDA, 베이스모델 분석		
17	18	19	20	21	22	23
모델링			모델 튜닝			
			앙상블			

3-2. 프로젝트 파이프라인

- 프로젝트 진행은 아래와 같은 파이프라인으로 팀원 모두 end-to-end 방식으로 진행하였다.



- EDA → 가설 수립 → 모델링 → 평점 예측 → 결과 분석 → 앙상블 → 제출
 - 결과 분석 과정에서 만족하지 못한 결과가 나오면 EDA 과정부터 다시 진행했다.
 - 모델링 과정에서 하이퍼 파라미터 튜닝을 함께 진행하도록 한다.

3-3. 프로젝트 진행

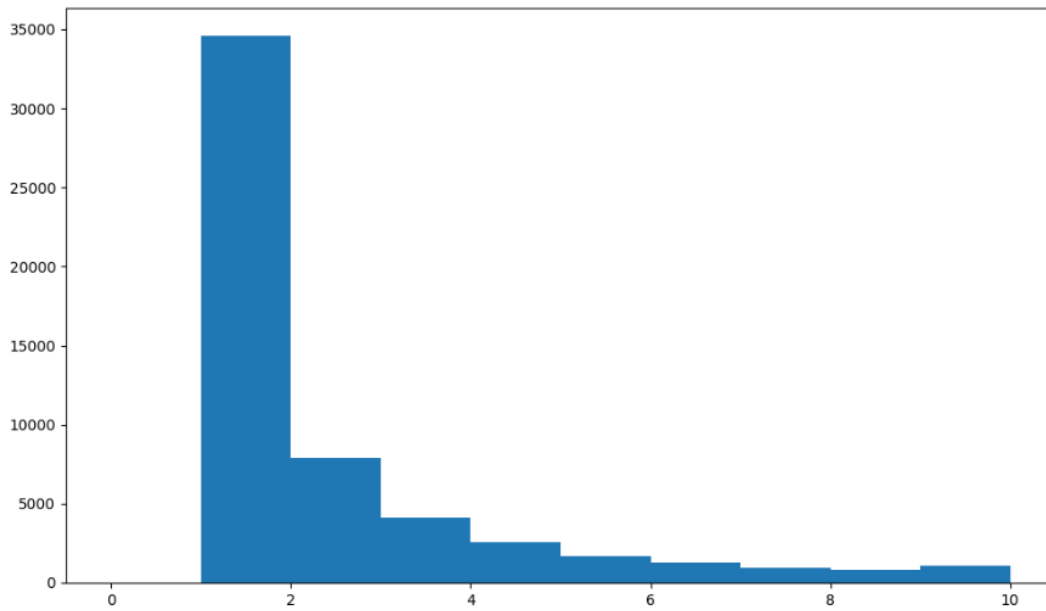
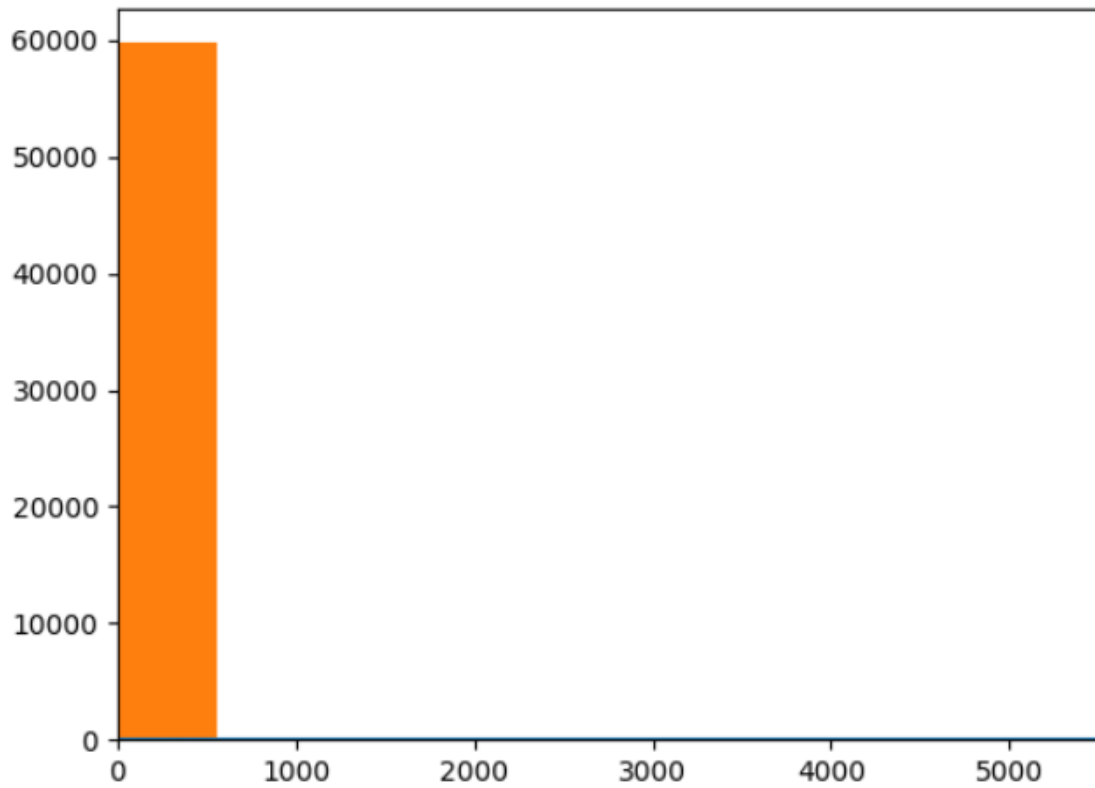
1. 각자 데이터 확인 후 생각하기에 좋은 결과를 낼 것 같은 모델을 사용한다.
2. 데이터 확인 후 효율적인 전처리 방법이 있다면 공유 및 사용한 모델의 점수를 공유한다.
3. 성능 좋은 모델들 위주로 앙상블을 진행하여 리더보드에 기록한다.
4. 앙상블이 끝난 데이터에 대해서도 다른 앙상블 결과와 앙상블 하여 결과를 확인한다.
5. 최대한 가장 좋은 모델 하이퍼 파라미터 튜닝 후 앙상블을 진행하도록 한다.

4. 프로젝트 수행 결과

4-1. 탐색적 데이터 분석(EDA) 및 전처리

유저의 평가 횟수 별 test_ratings.csv 등장 횟수

- 전체 유저에 대한 그래프(위)와 평가횟수 10회 이하인 유저에 대한 그래프(아래)
- 평가 횟수가 적은 유저의 비율이 높은 것을 확인할 수 있음

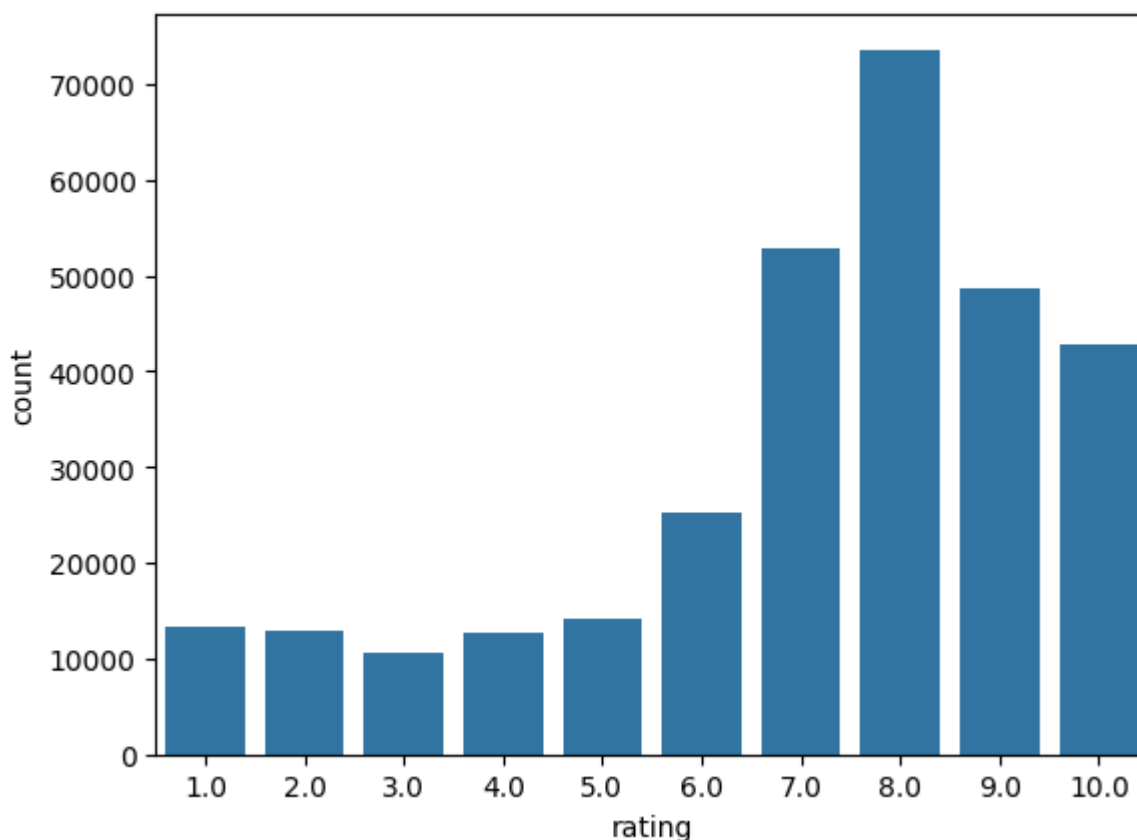


전체 test ratings.csv의 행 수: 76699

- 평가 5개 이하인 유저에 대한 예측: 22120개 (30%)
- 평가 1개 이하인 책에 대한 예측: 28799개 (30%)
- 평가 5개 이하인 책에 대한 예측: 44798개 (58%)

위의 결과를 바탕으로 long-tail 문제를 인식해서 두 그룹(head, tail)을 나누어서 각각을 잘 예측하는 모델을 만드는 것, 전반적으로 잘 학습하는 모델을 만드는 것 등을 시도했다.

train 데이터에서 book rating 개수



4-2. 모델 개요

baseline code

- FM, FFM, WDN, DeepCoNN, NCF, CNN_FM, DCN

이 중 하이퍼파라미터 튜닝 후 valid loss 기준으로 2.2 미만의 좋은 성능을 보인 모델은 NCF, DCN, CNN_FM이었습니다. NCF의 경우 overfitting 문제가 심한 편이어서 실제 test score는 좋지 않았다. 그래서 baseline code의 모델 중 최종적으로는 DCN, CNN_FM을 선택했다.

자체 활용 모델

1. LSTM_FM

- DeepCoNN 모델에서 텍스트 특징을 추출하는 부분을 기존 CNN에서 LSTM으로 바꾼 모델로 좋은 성능을 보였다.

2. MF

- surprise library에서 제공하는 것으로 좋은 성능을 보였다.

3. MLP_FM

- feature에 대해 context-aware으로 상호작용할 수 있는 FM 모델을 이용하여 MLP와 더해 새로운 모델을 생성해보았다.

4-3. 모델 선정 및 분석

실제 리더보드 제출 후의 스코어를 기준으로 위에서 좋은 성능을 보였다고 판단한 4가지 모델을 선정했다.

- MF, DCN, CNN_FM, LSTM_FM

4-5. 최종 제출

1. LSTM_FM 모델, MF 모델 앙상블
2. LSTM_FM 모델, MF 모델, DCN 모델 앙상블

두 최종 제출 모델의 리더보드 score는 2.1683으로 같다.

4-6. 리더보드 결과

- 순위 : 8위

순위	팀 이름	팀 멤버	RMSE ↕	제출 횟수	최종 제출
8 (-)	RecSys 3조		2.1683	59	6d

5. 자체 평가 의견

잘했던 점

- EDA 결과 등을 바탕으로 여러 시도를 해볼 수 있어서 좋았다.
- 평가 횟수, 평점 등을 기준으로 train 데이터를 나누어 모델을 따로 학습시킨 후 합쳐보았다.
- Text data는 CNN보다 RNN 모델이 잘 활용할 것이라는 아이디어로 DeepCoNN의 CNN 부분을 LSTM으로 바꿔서 활용해보았다.

- 평가 횟수, 평점 등을 기준으로 가중치를 달리 하는 새로운 앙상블 방법 등을 고안해보았다.

아쉬웠던 점

- 대체로 특별한 근거 없이 여러 모델을 단순히 테스트 해 보면서 하이퍼 파라미터를 튜닝하고, 앙상블도 단순 평균을 취했을 때 결과가 가장 좋았던 점이 아쉬웠다. 가설을 세우고 검증하는 것보다 실험을 반복하면서 결과를 얻고 그 원인을 분석하는 접근이 더 좋을 듯하다.
- 협업이 완전히 매끄럽지 않았던 점도 아쉬웠다. 깃허브 repo를 충분히 활용하지 못한 느낌이고, 각자가 뭘 해야 할지 명확하지 않았던 때도 있었다. 하나의 실험을 여러 명이 공유하지 않아서 개선의 여지가 있는 실험도 버려진 경우가 있는 듯하다.
- 실험 결과를 체계적으로 관리하지 않아서 실험 결과들로부터 도출할 수 있는 중요한 정보를 놓친 것 같다. wandb를 다음에는 팀에서 사용하여 각자의 실험 결과가 공유될 수 있도록 하면 좋을 것 같다.
- 프로젝트에서 어떤 모델을 활용할지, 어떻게 개선할지에 대해서만 주로 논의를 하다 보니 협업을 어떻게 해야 할지에 대한 논의가 제대로 이뤄지지 않아서 각자의 역량이 충분히 합쳐지지 못 했다.
- 팀원간 소통이 잘 되지 않아서 아쉬웠다.

배운 점

- Jupyter가 아닌 .py 파일로 코드로 PyTorch Template에 따라 구조를 짜는 것을 배웠다.
- 필요한 모델을 구현하여 baseline 코드에서 실행되게 직접 구현하는 경험을 쌓았다.

6. 개인 회고

김세훈

- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?
 1. XGBClassifier로 rating에 대한 feature의 중요도 확인
 - FM 모델을 기준으로 점수에 영향을 주는 Feature를 확인하기 위해 Feature 별 조합에 대한 성능을 직접 확인 해보았다.
 2. FM 모델과 MLP 모델을 활용한 새로운 FM + MLP 모델을 만들어 성능 테스트

- feature에 대해 context-aware으로 상호작용할 수 있는 FM 모델을 이용하여 MLP와 더해 새로운 모델을 생성해보았다.
3. FM 모델과 DCN 모델을 활용한 새로운 FM + DCN 모델을 만들어 성능 테스트
 - FM + MLP모델에 이어서 feature에 대해 깊게 학습하고 비선형 관계를 학습하기 효율적인 교차 학습 조합으로 좋은 성능을 발휘할 수 있지 않을까 하여 모델 생성해보았다.
 4. 언더 샘플링을 통해 점수가 낮은 분포 학습 진행
 - 현재 책 평점 분포는 점수가 낮은 분포는 충분한 데이터가 있지 않아 과적합이 생기는 것이 아닐까 하여 언더 샘플링을 통해 낮은 점수 분포를 채워보았다.
 5. K-fold Cross Validation 기능을 이용하여 모델 성능 확인
 - 현재 모델을 학습하기 위해 Train 데이터와 Test 데이터는 아래와 같이 8:2 비율로 학습을 진행하고 있다. 이 경우 Test 데이터에만 과적합 되는 것을 방지하여 좀 더 좋은 결과를 얻을 수 있지 않을까 하여 K-fold를 진행하게 되었다.
 6. 하이퍼 파라미터 튜닝
 - DCN 모델의 하이퍼 파라미터 튜닝을 진행하기 위해 run.py 파일을 따로 생성하여 파라미터 범위를 넣어 학습이 가능하게 만들어 진행하였다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
 - FM 모델과 MLP 모델을 활용하여 2.1888을 달성했다.
 - 점수 예측을 하기 위해 FM 모델 사용으로 context-aware으로 상호작용을 이끌어 내고 여러 레이어를 사용하여 feature간 특성을 이끌어 내어 평점을 예측해보았으나 그렇게 높은 결과를 얻어내지 못하였다.
 - EDA, 데이터 전처리, 모델 성능 평가, 하이퍼 파라미터 튜닝 등에 대한 여러 시도는 좋았으나, 무엇보다 중요한 것은 데이터 특성에 맞는 모델을 빨리 찾는 것이 중요하다는 것을 깨달았다.
 - 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
 - 협업 과정에서 GitHub를 제대로 사용하지 못한 점이 아쉬웠다. 그 이유는 두 가지가 있는데 먼저, 베이스 라인 코드를 기준으로 관리하려고 하였으나, 프로젝트 코드 저작권 문제로 Git에 다 Push할 수 없어 팀원들과 함께 소통하기에 어려움이 있었다. 두 번째로, 그리고 직접 서비스하여 버전 릴리즈를 하는 프로젝트가 아니기 때문에 각자의 코드를 공유함에 있어 버전을 통합으로 관리하는 것이 맞지 않았다. 이러한 점을 생각해 봤을 때 해당 프로젝트를 위해서는 일부 코드만 push 하고 코드가 중복되더라도 각자 폴더를 만들어서 Issue 관리로 팀원들의 진행 사항만 파악하는 것이 옳았을 듯 하다.

- 평점 예측에 사용되는 데이터의 특성을 제대로 파악하지 못한 점이 많이 아쉬웠다. 책 평점 예측은 평점에 영향을 미치는 각 feature의 분류를 통해 분류 문제를 잘 해결할 수 있는 모델이 무엇인지 찾고 그에 대해 성능을 높이기 위한 개선 방안으로 EDA와 하이퍼 파라미터 튜닝을 진행했어야 했다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?
 - 다음 프로젝트에서는 좀 더 정확하게 데이터 특성을 좀 더 깊이 있게 파악하여 문제를 가장 잘 해결할 수 있는 모델을 선정해 프로젝트를 진행할 예정이다. 이것을 해내기 위해 모델의 특성과 장단점을 잘 이해하고 이를 적용할 것이다.

박시우

1. 학습 목표

ai stages 2등 이내

2. 시도

이전 프로젝트 경험을 기반으로 봤을 때, 물론 모델을 선정하는 것도 중요하지만 데이터 전처리 방법을 바꿨을 때 성능에 많은 영향을 끼친다는 것을 알고 있었기 때문에 전처리를 바꿔보면서 어떤 게 최선일 지를 많이 고민하고자 했다.

3. 효과

전처리를 바꿔보면서 성능이 더 개선된 사항이 있었지만 생각보다 크게 개선되지는 않았다. 전처리 방법을 다르게 이것저것 해볼 수 있는 context data를 쓰는 모델 자체가 FM, FFM 밖에 없었는데 그 두 모델의 성능이 딱히 우수한 편이 아니었기 때문에 효과가 더 미미하게 느껴졌던 것 같다.

3. 한계점과 아쉬운 점, 시도하고 싶은 점

베이스라인에 제공된 모델 이외에도 새롭게 모델을 직접 만들어서 공부하는 방향으로 프로젝트를 진행하고 싶다는 생각이 있었는데 그렇게 시도하지 못했던 게 아쉬웠다.

맨 처음에 파라미터 튜닝을 해보면서 각 모델의 성능을 파악해보고 그 후 모델을 몇 개 선정하고 베이스라인 외의 모델을 뭐 쓸지 생각해보면서 프로젝트를 진행했으면 더 좋았을 거 같다.

다음에는 새로운 모델을 직접 만드는 것, 위에 언급했던 순서로 프로젝트를 진행하는 것을 꼭 시도하고 싶다.

배준혁

- 프로젝트 목표

이번 프로젝트의 개인 목표는 크게 2가지였다. 첫째는 협업 프로젝트의 경험을 쌓는 것, 둘째는 AI 개발 과정을 처음부터 끝까지 겪어보는 것. 그 외에 팀 목표로 여러 가지 가설을 세워 실험해보고, 결과를 해석하는 것과 최종 순위 2등 이내에 들기가 있었다.

개인 목표는 70% 정도 달성했다는 생각이 드는데, 팀 목표는 제대로 달성하지 못했다.

- 개인 목표 1

우선, 첫 개인 목표인 협업 프로젝트 경험 쌓기는 깃허브 공용 repo 관리를 통해 달성했다. 하지만 말 그대로 경험을 쌓았을 뿐, '깃허브를 잘 활용했는가?'를 생각해보면 그렇지 않다. 특강 때 배운 대로 이슈 템플릿, PR 템플릿 등을 만들고, branch를 따서 새로운 기능을 구현한 후 PR을 통해 main branch에 merge를 하긴 했다. 그러나 다들 깃허브 활용이 익숙하지 않다 보니 뒤로 갈수록 코드를 개인적으로 작성하고, 원활히 공유가 되지 않았다. 각자 자신이 할 작업을 이슈에 등록하고, 하나씩 해결하기로 했으나 이것도 흐지부지 되면서 잘 이뤄지지 않았다.

모두가 개인 작업에 집중하다 보니 팀의 협업 관리를 중점적으로 고민한 사람이 없었던 게 가장 큰 원인인 것 같다. 초반에는 내가 깃허브 repo 활용 방식을 제안하면서 나름 협업에 신경 쓰려고 했는데, 컨디션이 나빠진 것과 '나는 조장이 아니니까' 라는 이유들을 핑계 삼아 소홀히 하게 됐다. 솔직히 다음 프로젝트 때도 내가 주도적으로 협업을 이끌어 나갈 자신은 없지만, 최소한 어떻게 하면 협업을 더 잘할 수 있을까에 대한 고민은 충분히 해야겠다.

- 개인 목표 2

두번째 개인 목표인 AI 개발 과정 경험하기도 EDA, 모델 구현, 앙상블, 파라미터 튜닝 등을 해 보며 달성했다. EDA에서는 평가 수가 적은 유저, 책이 많음을 확인하여 팀원들에게 cold start 문제 해결에 대한 필요성을 알렸고, 베이스라인 코드에 없는 MF 모델을 구현해보았고, 평가 수나 평점 구간에 따라 각 모델의 가중치를 달리 하는 동적인 앙상블 함수도 구현했다. 특히 앙상블에 제일 노력을 많이 기울였는데, LSTM을 활용해 책의 summary 정보를 feature로 사용하는 모델과 MF 모델을 위주로 앙상블을 진행했다.

첫 앙상블은 cold start 문제 해결을 중점적으로 다뤘다. MF 모델은 각 유저와 책의 평가 데이터만을 활용하지만, LSTM 모델은 책 자체에 대한 정보를 추가적으로 활용하므로 cold start 문제에 더 강할 것이라 생각했다. 이를 바탕으로 평가 수가 적은 책에 대한 예측은 LSTM의 가중치를 높게, 평가 수가 많을 때는 MF의 가중치를 높게 반영해 앙상블을 했다. 하지만 실제로는 오히려 책의 평가 수가 많을수록 LSTM의 가중치를 높게 했을 때 결과가 더 잘 나와서 의아했다. summary에 대한 책 사이의 연관성이 떨어지거나, 평가 수가 적은 책이 너무 많아서 연관성을 충분히 학습하지 못한 것으로 추측된다.

또한 MF 모델은 전체 평점의 평균을 기준으로 편차를 예측하기에 평균과 멀리 떨어진 낮은 점수를 잘 예측하지 못할 것이라 생각했고, 실제로 대부분의 유저가 나쁘게 평가한 책에 대해 LSTM 모델이 MF 모델보다 더 낮은 점수를 주는 것을 확인했다. 이를 바탕으로

로 두 모델의 평균 예측 평점이 낮을 때는 LSTM 모델의 가중치를 높이고, 반대일 때는 MF 모델의 가중치를 높이는 식으로 앙상블을 시도했다.

이처럼 분석과 추론을 통해 앙상블을 시도했을 때 실제로 개별 모델들보다 예측 성능이 좋아지긴 했으나, 결과적으로는 단순히 두 모델에 1:1로 가중치를 줬을 때 가장 성능이 좋았다. 열심히 고민하며 시도한 방법보다 단순한 접근이 더 결과가 좋아서 좀 허무한 느낌도 들었다. 또한 앙상블에 3일 정도를 투자한 것보다 하이퍼 파라미터 튜닝에 하루를 투자한 결과가 더 좋았다. 다음 프로젝트에서는 여러 모델에 대한 테스트, 하이퍼 파라미터 튜닝 등 단순한 접근들을 먼저 충분히 시도하고 그 이후에 창의적인 접근을 해 봐야겠다.

앙상블 변형과 같은 실험적인 접근 외에, 하이퍼 파라미터 튜닝 과정에서 베이스라인 코드의 허점을 발견해 보완했던 점도 의미 있는 경험이었다. `weight_decay`를 조정해도 결과가 변하지 않는 것을 보고 `optimizer`를 선언할 때 `weight_decay`가 인자로 전달되지 않는 것을 발견 후 수정하였다. 그리고 `batch_size`를 줄일수록 `valid_loss`가 무조건적으로 좋아졌는데, 이 점에 의문을 갖고 고민함으로써 `valid batch size`가 평균 RMSE에 영향을 준다는 것을 깨달았다. 각 예측의 `error`가 그대로여도 `valid batch size`에 따라 값이 달라지기에 이 영향을 막기 위해 `train batch size`와 `valid batch size`를 분리했다.

- 팀 목표

팀 목표 중 다양한 실험과 결과 분석은 잘된 점도 있고, 아쉬운 점도 있다. 나는 상술한 것처럼 앙상블에 대해 여러 시도를 했고, 다른 조원들도 평가 수나 평점을 기준으로 데이터를 나눠 모델을 따로 학습 시켜 특정 조건의 데이터에 대해 잘 예측하는 모델을 노리는 등 다양한 시도를 했다. 하지만 대부분 결과가 좋지 않았고, 당장 눈에 보이는 성능 지표가 개선이 안 되다 보니 결과 분석보다는 새로운 접근을 시도하는 것에 더 치우쳤던 것 같다.

또 다른 목표였던 2등 내에 들기는 8등이라는 성적으로 마무리하면서 많이 아쉬운 결과를 냈다. 협업 및 소통 미숙, AI 및 데이터에 대한 이해 부족, 효율적이지 못한 접근 과정 등 여러 이유가 있겠으나, 아직 배우는 과정인 만큼 결과에 낙담하기 보다 그 과정에서 배운 것들에 더 집중하는 것이 중요하다고 생각한다.

결과적으로 성적은 좋지 않았으나, 그럼에도 프로젝트를 마친 후 다들 웃으면서 오랫동안 이야기를 나눴다. 프로젝트 이전부터 피어세션을 할 때 다양한 주제로 스몰 토크를 하면서 즐거운 분위기를 만들려고 노력해왔는데, 그 노력이 성과를 거둔 것 같다. 비록 팀으로서 성공적인 결과물을 내지는 못했고, 그 과정에서 업무적인 소통에 부족한 부분은 있었지만, 팀워크를 위한 커뮤니케이션은 충분히 성공적이었다.

백승빈

- 프로젝트를 진행하며 시도한 것

1. LSTM_FM

강의에서 DeepCoNN이 개발될 당시에 좋은 시퀀스 모델이 없어 CNN이 사용되었다는 이야기가 인상깊어 베이스라인 모델에서 CNN파트를 LSTM 모델을 사용하여 대체해보았다. 모델을 개발한 직후에는 Overfitting이 심했으나 모델의 hidden size와 layer개수를 줄여 모델의 단순화시킴으로써 Overfitting 문제를 해결할 수 있었다. 하이퍼 파라미터 튜닝 이후 train loss 1.921, validation loss 2.173의 결과를 얻을 수 있었다.

2. Hybrid model

EDA에서 rating개수가 5개 이하인 cold start 유저가 약 30%라는 인사이트가 있었다. cold start인 유저와 아닌 유저들로 데이터를 나누어 각 집단에 특화되도록 모델들을 학습시키고 그 결과를 합쳐보았다. 하지만 cold start 유저 집단의 학습이 개선되지 않았고 전체 데이터를 이용해 훈련한 모델의 결과보다 좋은 결과가 나오지 않았다. cold start 유저와 아닌 유저의 점수 분포가 유사한것이 이유가 아닐까 생각한다.

1차적으로 개발한 모델들의 결과를 분석한 결과 중에 모델들이 대체로 낮은 점수를 잘 예측하지 못한다는 분석이 있었다. 유저가 매긴 rating의 개수를 사용하는 대신 높은 점수와 낮은 점수의 데이터를 7:3과 3:7의 비율로 혼합하여 두 그룹으로 나누고 학습을 시켜보았다.

더 다양한 조합의 모델들을 시도해봤으면 다른 결과가 있었을지도 모르지만 큰 차도가 보이지 않는 곳에서 더 실험을 해보기는 시간적 여유가 부족했어서 아쉬움이 남는다.

3. BERT_FM

LSTM보다 대체적으로 좋은 성과를 보이는 BERT를 이용해보자는 의견이 있었다. 하지만 유저와 아이템의 summary feature를 얻기 위해 두 BERT모델을 연속적으로 돌리자 GPU 메모리 부족으로 에러가 발생했다. 더 연구하기에는 시간이 부족했고 또한 이미 텍스트를 임베딩하는 과정에서 BERT모델이 사용되는데 BERT를 다시 사용하는 것에 대한 이의가 제기되어 개발을 중단했다.

- 프로젝트를 진행하며 배운 것

인사이트가 꼭 결과로 이어지지는 않는다.

또한 구현중 발생한 여러 이슈를 해결하는 과정에서 내가 때때로 에러를 해결하는데에 집중한 나머지 다른 방법이 존재하는데도 보지못하는 경우가 발생했는데 특히나 팀 프로젝트에서는 시간관리가 중요한 만큼 에러를 해결할때 좀 더 시야를 넓히면 좋겠다는 생각이 들었다.

- 다음 프로젝트 때 개선할 점

이번 프로젝트에서 특히 프로젝트 진행 방식에 대해 아쉬움이 많이 남는데 인사이트를 통해 공통의 목표를 세우고 각자 개발을 하고 다시 결과에서 얻은 인사이트로 다음 목표를 설정하는 식으로 진행했다면 각자 개발을 하면서도 좀 더 유기적으로 진행할 수 있지 않았을까하는 생각이 든다. 또한 말로만 정보를 공유하는 대신 글이나 그래프로 각자의 작업과 결과를 공유했으면 더 정보교환이 수월했을것 같다.

서동은

- 반성할 점 :

1. EDA

- 데이터 시각화를 많이 활용하지 못 함.
- 상관관계 분석을 하지 않음
- 데이터 전처리에서 새로운 시도를 하지 않음

2. 베이스라인에 있는 모델들을 사용하는 것에 그침, 새로운 시도를 해보지 못 함

- 아쉬운 점:

1. 실험

long-tail을 해결하고자 하는 논문들이 많은 것으로 보아 나의 실험에 문제가 있었을 가능성이 농후함. 클러스터링 할 때 카테고리를 잘 사용했으면 좋았을 것 같고, 유저-아이템을 머지한 테이블을 사용해서 같이 클러스터링하면 좋았을 것 같음.

2. 뒤로 갈수록 열심히 하지 않았음. 끈기 있게 여러 시도를 해보았으면 좋았을 것 같다

3. 추천 시스템 라이브러리를 사용해보았어야 했다.

이현규

- 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

사용할 수 있는 모델을 사용해보려 노력하였음.

- 나는 어떤 방식으로 모델을 개선했는가?

여러 모델을 사용한 후 가장 성능이 좋은 모델을 고름 (왜 좋은 성능을 냈는지는 모름)

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

- 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

lv1 이기에 넘어가겠습니다. lv2에 계속

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

더이상 좋아지지 않는 점수에 해결방안을 찾지 못했음. 왜 성능이 좋은가에 대한 설명을 죽어도 하지 못한다는 자신에게 너무 아쉬움.

- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

개인적으로 많은 경험을 쌓아야 할 것으로 보임 (과거 캐글 대회 등)

- 느낀점.

추천이라는 것이 더욱 어렵다고 생각하게 됨.

아직 제대로 모르고 있는게 아닌가? 라는 생각으로 다시 정리를 해야하는 계기가 생김.

만약 실무였다고 가정한다면 나 자신은 제대로 일을 해결 할 수 있었을까 하는 생각이 듦.