네이버 부스트캠프 AI Tech 5기 Project 1 랩업리포트

Team 1 : 일로 와봐
강찬미_T5009, 박동연_T5080, 서민석_T5102, 이준영_T5158, 주혜인_T5208

1. Team Wrap-up Report

1-1. 프로젝트 개요

프로젝트 주제

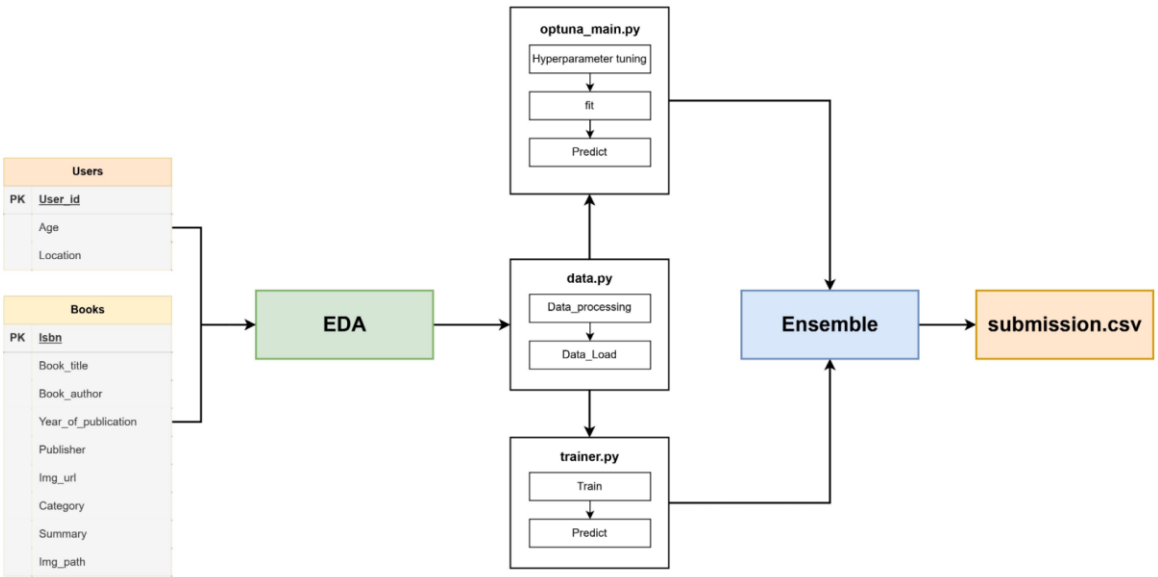
영상, 기사 같은 다른 콘텐츠와는 달리 책은 한 권을 다 읽기까지 적지 않은 시간을 필요로 하고 구매를 위해 이용할 수 있는 정보가 한정적이라 소비자들은 책을 고르는데 신중을 가하게 된다.

그렇기에 이번 프로젝트에서는 소비자들의 책 구매에 도움을 줄 수 있도록 책, 소비자의 정보를 바탕으로 소비자가 특정 책에 줄 평점을 예측하도록 한다.

활용 장비 및 재료

- 서버 스펙 : AI Stage GPU (Tesla V100-SXM2)
- 협업 툴 : Github / GatherTown / Zoom / Notion / Google Drive
- 기술 스택 : Python / Pytorch / VScode / Wandb / Optuna / Scikit-learn

프로젝트 구조 및 사용 데이터셋의 구조도



1-2. 프로젝트 팀 구성 및 역할

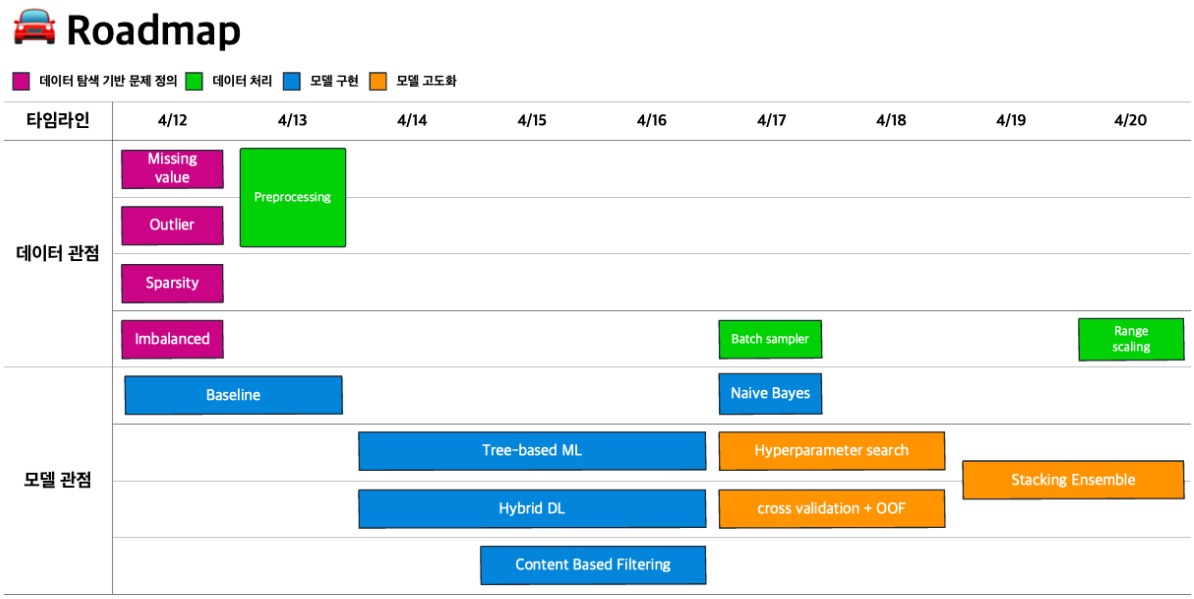
이름	역할
강찬미	Age 결측치 처리 ML 모델(Catboost, lgbm) gridserchCV 및 oof 구현 (ML 모델, FM, FFM) 최적화
박동연	상위 카테고리 필드 생성 및 결측치 처리 3layer CNN_FM 구현 (DeepCoNN, WDN, CNN_FM) 최적화
서민석	(weighted random sampler, cross validataion, OOF, range scaling) 구현 (NCF, DCN) 최적화
이준영	(Wandb sweep, stacking) 구현 (FFM + DCN, DeepCoNN + CNN) 구현 및 최적화 CNN_FM context data 입력 구현
주혜인	Age 결측치 처리 Early Stopping, ML 모델(Catboost, lgbm, NB) gridsearchCV 및 optuna, oof 구현 (WDN, ML 모델) 최적화

1-3. 프로젝트 수행 절차 및 방법

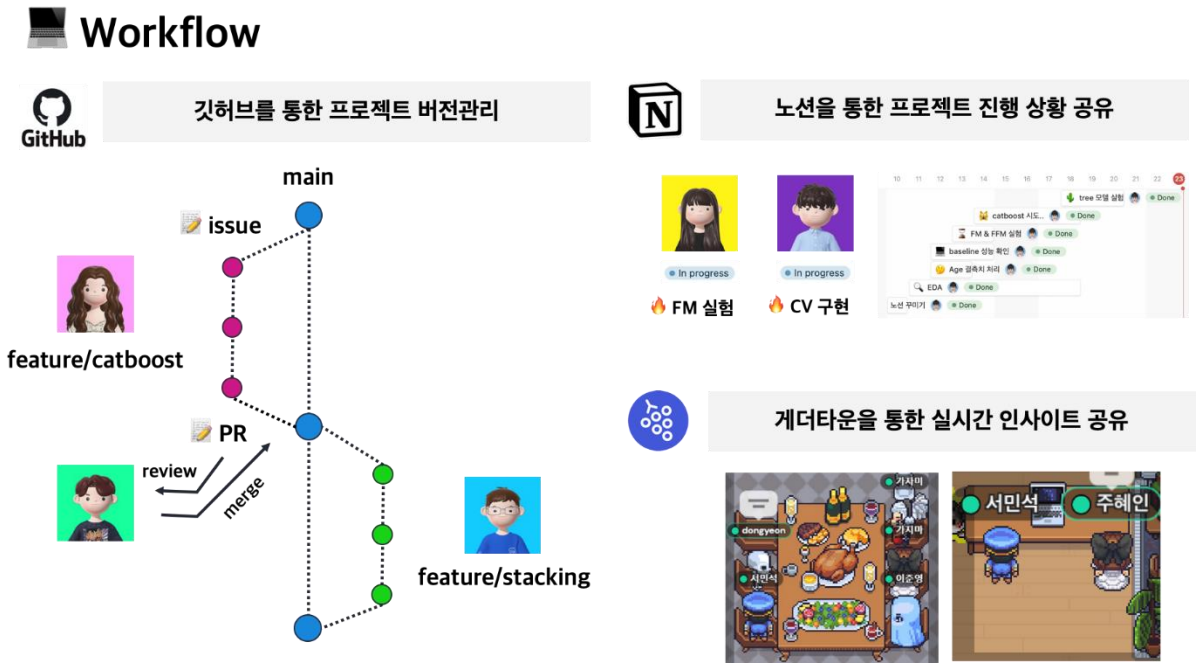
팀 목표 설정

- 배운 내용을 프로젝트에 적용해보기
- 우리 팀에 적합한 협업 문화 고민해보기 - 지속가능한 협업 문화 수립하기

프로젝트 수행 과정

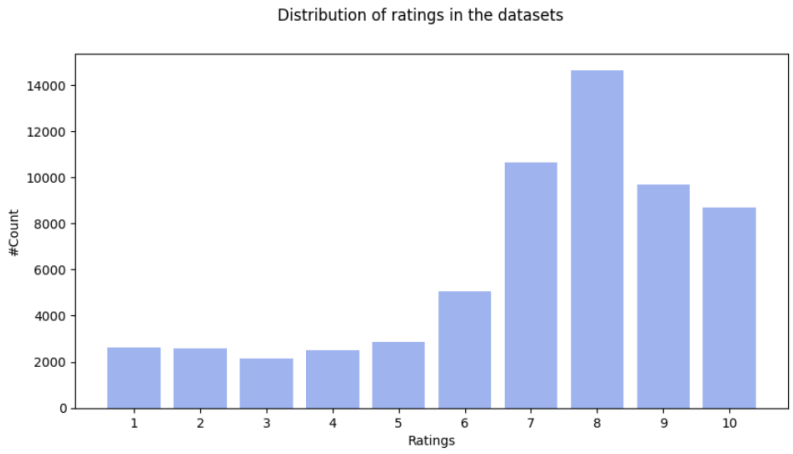


프로젝트 협업 문화



1-4. 프로젝트 수행 결과

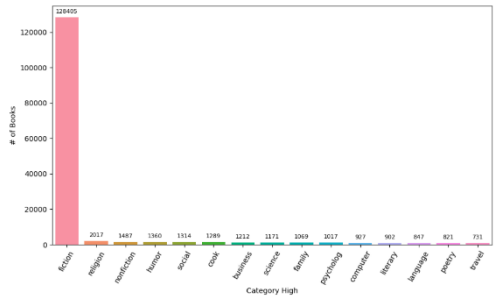
1. 탐색적 데이터 분석 (EDA)



- user 의 age 컬럼과 book 의 category, language, summary 컬럼에서 40%이상의 결측치가 확인됨
- book 의 publisher 컬럼에서 이름이 잘못 표기된 이상치가 있음
- 데이터의 rating 분포를 분석하였을 때, 7~10 사이의 평점이 가장 많이 분포해있는 것을 알 수 있음
- 1~6 에도 비교적 적지만 여전히 rating 이 존재하는 것을 확인할 수 있음
- 이 외의 자세한 내용은 부록 3-1, 3-2, 3-3 에 첨부

2. 데이터 전처리

2-1. Category 결측치 전처리



휴리스틱한 방식으로 뽑아낸 상위 카테고리 43 개를 활용

4105 개의 하위 카테고리를 쓰는 것보단, 43 개의 상위 카테고리를 쓰는게 일반화 및 예측에 더 좋을것으로 생각됨

동일한 책 이름이지만 카테고리가 결측치가 아닌 것들을 찾아서 결측치를 채움 → 0.02%까지 줄임

2-2. Age 결측치 전처리

- Age 필드 내에 결측치를 채우기 위해서 다양한 방법을 적용
- **Normal Random, Total Zero**, Total Average, County Average, Uniform Random, KNN, Stratified
- 각 방법 별 구체적인 설명은 부록 3-5 에 기재

2-3. 그 외 다양한 전처리

- location split: 하나의 필드 값을 ","로 구분하여 city, state, ···, country 필드로 구분하여 저장
- api 활용: 외부 api 를 활용하여 location 결측값 값을 3%까지 줄였지만 대회 치팅이라고 판단하여 제거

3. 모델 개요

Model	특징
3layer CNN context FM	- CNN layer로 이미지 vector를 학습한 뒤 user, book context vector와 결합해 FM으로 학습하는 모델 - CNN의 layer 수를 2에서 3으로 높임으로써 더 낮은 loss에 더 적은 수의 epoch으로 빠르게 도달
FFDCN	- FFM과 DCN을 병합한 하이브리드 모델 - FFM의 출력과 DCN의 출력을 concat한 후, linear layer의 출력으로 평점을 예측
DeepCoNN_CNN	- DeepCoNN모델의 FM layer에 이미지 vector를 추가한 모델 - DeepCoNN보다 약간 성능이 좋아짐

※ 자세한 모델 별 설명은 부록 3-4 에 정리

4. 모델 선정 및 분석

4-1. 최종 모델 선정

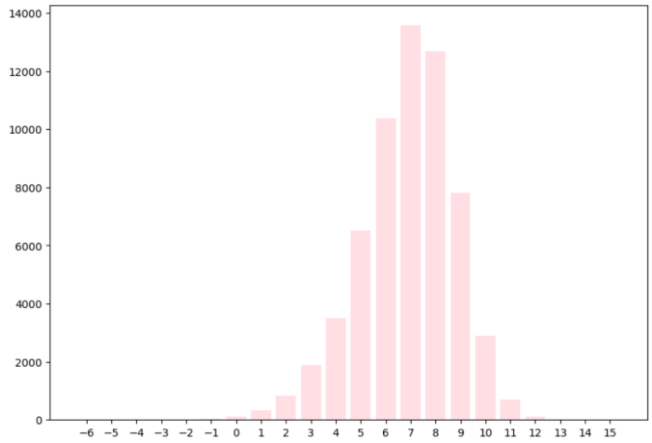
Baseline Model	Custom Model	Tree based Model
- FFM - DCN - NCF - DeepCoNN	- 3 layer CNN context FM - cv 3layer CNN context FM - FFDCN - DeepCoNN_CNN	- Catboost (Top1, 2 Rated) - LightGBM

baseline 에서 제공된 모델들을 포함하여, 다양한 방식으로 개조한 모델과 새로운 모델들을 합하여 총 11 개의 모델들을 Stacking 앙상블을 하였음

4-2. 최종 모델 선정을 위한 분석 및 검증 절차

1. 각 모델에 대해서 **파인 튜닝** 과정을 거침
2. 파인 튜닝 후 일반적으로 좋은 성능을 보인 **모델들을 기준에 따라 선정**
   Q. 범주형 데이터 활용에 적합한 Tree 모델인가?
   Q. 파인튜닝을 통해 val_loss 가 2.1 대에 진입한 바가 있는가?
3. CatBoost, CNN FM, DeepCoNN, FFDCN, CatBoost 선정
4. 위 모델들을 **다양한 경우의 수로 조합**하여 성능 추이 확인
5. 다양한 모델들을 **하나씩 추가 및 제출해가며 성능 추이 확인**
6. 가장 좋은 성능을 낼 수 있는 모델들을 **최종 앙상블** → oof(out-of-fold)를 적용한 경우와 적용하지 않은 경우, 두 가지를 최종 제출 모델로 선정

4-3. Range scaling



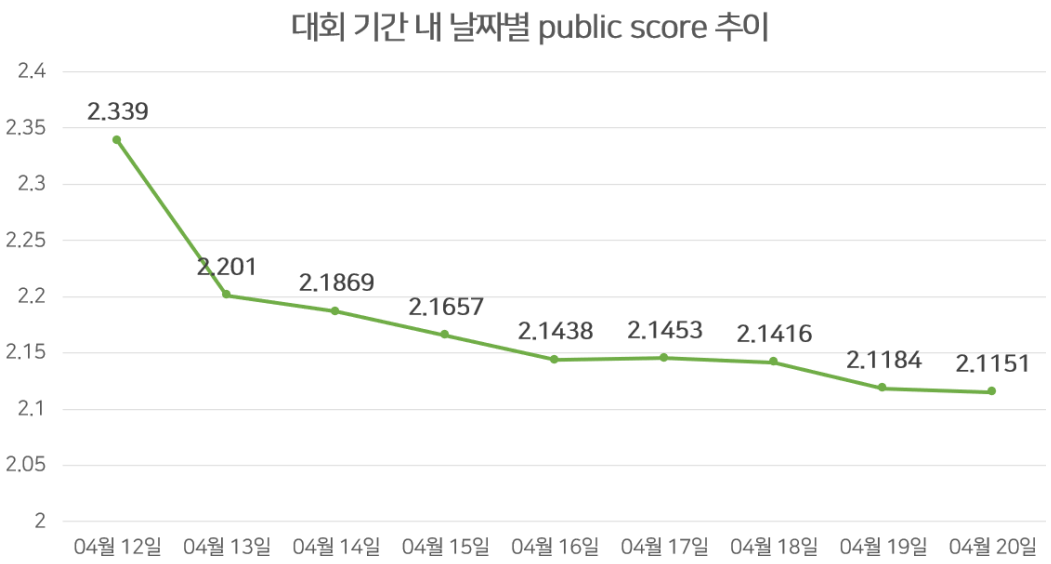
- 평점은 1 점에서 10 점 사이로 예측되어야 함
- 최종 예측값에서 1 미만인 값은 1 로, 10 초과인 값은 10 으로 변환하는 사후 처리를 수행함

5. 프로젝트 결과

5-1. 각 모델의 최고 성능 (RMSE Loss 기준)

Model	tra_loss	val_loss
FFM	-	2.435
DCN	1.657	2.346
NCF	1.606	2.35
DeepCoNN	1.919	2.181
3 layer CNN context FM	1.977	2.161
FFDCN	-	2.169
DeepCoNN_CNN	1.812	2.179
CatBoost	-	2.139
LightGBM	-	2.190

5-2. 대회 기간 내 날짜 별 public score 추이



5-3. 앙상블한 최종 모델의 성능

val score	public score	final socre (public+private)
2.1214	2.1151	2.1077

- 최종 제출한 모델의 팀에서 내부적으로 구현한 val_score 은 2.1214
- 대회 홈페이지에 submit 파일을 제출한 후에 확인한 public score 는 2.1151 (당시 기준 5 등)
- 대회 종료 이후에 공개된 final_score 는 2.1077 로, loss 가 약 0.0074 만큼 줄어듦 (최종 3 등)

1-5. 자체 평가 의견

[목표 달성 정도]

배운 내용 적용하기

- 주어진 baseline code 이외에도 강의에서 배운 모델과 성능을 위한 여러 기능을 직접 구현함
- 결과적으로 최종 리더보드상 3 위라는 좋은 성과를 거둠

지속가능한 협업 문화 수립하기

- 팀원 모두가 git convention 을 준수하여 프로젝트의 완성도를 높임
- 체계적인 문서화를 위한 노션 활용

[잘했던 점]

- 실험과정 : wandb 를 통한 실험 결과 로깅과 sweep & optuna 를 활용한 하이퍼파라미터 탐색 자동화
- 일반화 : cv score 를 통한 모델 평가 일반화와 oof prediction 을 통한 예측값 일반화
- 데이터파악 : 지속적인 EDA 를 통해 얻은 인사이트를 모델링에 적용함
- 의사소통 : 게더타운을 활용한 실시간 의사소통

[시도했으나 잘 되지 않았던 것들]

- 결측치 대체 : 다양한 시도에도 불구하고 성능에 직결되는 방법을 찾지 못함
- ML model 의 온전한 모듈화를 하지 못함
- 전이학습을 시도했지만 과적합으로 이어짐
- cold – start 문제를 해결하기 위한 시도를 했지만 성능을 개선하지 못함

[아쉬웠던 점들]

- 문서화 : 노션 페이지의 효율적인 활용
- 계획 수립 : 장기적 관점에서의 명확한 목표 수립 미약
- 결과 파일 관리 : 반복적인 실험으로 누적된 파일의 효율적 관리
- 프로젝트 수행 과정 : 파생변수 생성과 hyperparameter search tool 의 효율적인 사용

2. Personal Wrap-up Report

2-1. 강찬미_T5009

1. 내 학습목표를 달성하기 위해 한 노력

- **익숙해지기** : 대회도 실제 데이터에 추천 모델을 적용시키는 것도 처음이기 때문에 전반적인 진행과정이 많이 낯설고 어려웠다. 그렇기에 baseline 코드를 살펴보고 팀원들의 진행 상황을 참고하며 최대한 이해하고 익숙해지며 팀의 흐름을 놓치지 않으려 노력했다.
- **소통하기** : 평소 질문을 쉽게 하는 편이 아니지만 프로젝트를 진행하는 동안은 막히는 부분이 있거나 궁금한 부분이 있으면 주저하지 않고 바로 팀원들과 문제 상황을 공유하고 그에 대한 이야기를 나누며 해결하려 했다.

2. 내가 모델을 개선하기 위해 한 노력

- **다양한 피처 전처리 방식 조합** : Category(2-way), Age(4-way) 총 8 개의 전처리 방식이 존재하여 그 중 최적의 조합를 찾는 실험을 진행했다.
- **다양한 피처 조합** : 피처 중요도를 확인한 후, 중요도가 작은 피처 중 일부를 삭제하며 다양한 피처 조합을 시도하였다.
- **파라미터 튜닝** : Optuna 를 통해 Tree 모델 하이퍼 파라미터를 튜닝하였다.

3. 내가 한 행동의 결과로 달성한 것 및 얻은 깨달음

- **다양한 시도의 중요성** : 모델마다 좋은 성능을 보이는 피처, 전처리 조합이 다르며, 구현할 수 있는 범위 내에서 다양한 시도를 해보는 것이 중요하다는 것을 깨닫게 되었다.

4. 내가 새롭게 시도한 변화와 효과

- **모듈화** : tree 모델 gridsearchCV 를 구현하며 baseline 코드를 참고하여 모듈화를 시도했다. 부족한 부분이 많은 코드이지만, 유기적으로 잘 동작하는 것을 보며 뿌듯했고 해당 과정을 통해 모듈화에 대한 이해도를 높일 수 있었다.
- **git 을 통한 협업** : 초반에는 실수로 main 이 잘못될 것을 걱정하며 소극적으로 사용했으나 git 에 익숙한 팀원들의 도움으로 여러 기능을 경험해보며 git 을 조금 더 능숙하게 사용하게 되었다. git 을 제대로 사용한다면 협업시에 굉장히 유용함을 깨닫게 되었다.

5. 마주한 한계와 아쉬웠던 점

- **낮은 이론 이해도** : 다양한 모델과 방법론에 대한 이해도가 낮아 팀원들이 공유해주는 내용을 완벽하게 이해하지 못하는 순간이 많았다.
- **낮은 효율성** :모듈화 경험이 없어 구현시에 시간을 많이 낭비한 것 같다는 생각이 든다. 조금 더 빨리 구현을 했다면 모델을 최적화할 시간이 더 많았을 것이며 조금 더 좋은 결과를 낼 수 있었겠다는 아쉬움이 남는다.
- **리더보드 성적** : 프로젝트를 시작할 때는 분명 리더보드에 연연하지 않고 가능한 많은 것을 시도하고 경험해보자고 다짐했는데 리더보드 점수가 나타나고 계속해서 순위가 바뀌면서 성적에 신경쓰기 시작했다는 점이 아쉬웠다

6. 다음 프로젝트에서 시도해볼 것

- **모델 구조 수정** : 다른 팀원들이 이번 프로젝트에서 한 것처럼, 레이어를 추가하거나 모델을 결합해 하이브리드 모델을 만들어 보고 싶다.
- **환경구축** : 이번의 경우 모델을 추가하고 학습하는 것만으로 충분히 벅차 다른 부분을 건들여보지 못했지만 다음에는 학습 환경 세팅, 협업 환경 세팅과 같은 부분을 경험해보고 싶다.

2-2. 박동연_T5080

1. 내 학습목표를 달성하기 위해 한 노력

- 이전 기수 및 다른 대회의 회고 자료, 강의 자료 등을 **적극적으로 참고**하여 본 프로젝트에 **반영 및 팀원에게 공유**하기
- 부드러운 팀 분위기와 협업을 위해, 팀원들의 의견을 **적극적으로 경청하고 공감 및 수용, 개선할 수 있는 의견 전달**하기

2. 내가 모델을 개선하기 위해 한 노력

- **전이학습:** baseline 에서 제공되었던 CNN_FM 에서 CNN 을 저명한 모델로 교체하고, 성능을 높이고자 하였다. 하지만 과적합으로 인해 오히려 성능이 떨어졌다.
- **모델 구조 변경:** CNN_FM 에서 기본 2 layer CNN 을 3 layer CNN 으로 변경하고자 하였고, 그 결과 기존 모델보다 빠르게 더 낮은 loss 에 도달할 수 있었다.
- **파인 튜닝:** DeepCoNN, CNN_FM, WDN 등 다양한 모델에 대해서 파인 튜닝을 통해서 최적화를 시도했다.

3. 내가 한 행동의 결과로 달성한 것 및 얻은 깨달음

- **알고 있는 것과 해본 것은 다르다**: 머리로 알고있는 다양한 모델과 성능 개선의 기법이 있지만, 이를 실제로 구현 및 도입해본 경험이 많지 않았다. 그래서 코드로 구현했을 때 어떤 오류와 허점과 실수가 있었는지 잘 알지 못한 것이 아쉽다. 이제는 머리로 알고 있는 것들을 코드로 구현하여 실체화해보는 것을 연습하도록 해야겠다.

4. 내가 새롭게 시도한 변화와 효과

- **실험의 논리성과 타당성 확인**: 각종 실험의 내용을 배경 및 가정, 가설, 결과 및 분석 내용을 하나의 문서로 정리함으로써 각 실험과 행위에 대한 논리성과 타당성을 쉽게 확인할 수 있어서 좋았고, 이를 통해 어떤 점을 개선해볼 수 있을 지 비교적 쉽게 파악할 수 있었다.
- **보다 적극적인 Git 활용**: Git 의 적극적인 활용이 절실한 상황이었고, 특강 때 배운 것들을 포함해서 pr 리뷰도 최대한 많이하려고 노력했다. 프로젝트 이전에 비해서 확실히 Git 과 더 친숙해졌고, Git 이 무서운게 아니라 오히려 고마운 도구라는 것에 대해서 공감하게 되었다.

5. 마주한 한계와 아쉬웠던 점

- **리더보드 순위**: 리더보드 순위를 신경쓰지 않기로 해놓고서는, 신경을 많이 쓰다보니까 이를 올리기 위해서 공부한 것을 복습 및 도입해보는 것 보다 성능 개선을 위한 파인 튜닝 실험 위주의 시간을 보냈던 것이 아쉽다.
- **체력적 한계**: 체력적인 한계로 인해 더 시도 및 도입해볼 것들을 하지 못했던 것이 아쉽다. 운동을 안하고 앉아서 공부하는 것보다, 잠깐이라도 운동을 다녀오는게 더 장기적으로 오래, 많이 공부할 수 있는 길인 것 같다.

6. 다음 프로젝트에서 시도해볼 것

- **프로젝트 측면 : 모델 및 파일 버전 관리**: 프로젝트를 할 수록 다양한 버전의 모델과 파일들이 생겨나 이를 구분하고 찾는 것이 어려워졌다. 따라서 이를 자동화 및 간소화할 수 있는 방안을 탐색하여 다음 프로젝트에 적용해보고자 한다.
- **개인적인 측면 : 기능 구현**: 이번 프로젝트에서는 팀원들이 만들어준 기능을 자주 사용하면서도, 이를 직접 내가 구현하여 참여해볼 생각과 시간을 가지지 못했던 것이 아쉽다. 따라서 다음 프로젝트에서는 기능 구현에 좀 더 적극적으로 참여하고자 한다!

2-3. 서민석_T5102

1. 내 학습목표를 달성하기 위해 한 노력

- **실험환경 구축**: 개인적인 목표는 좋은 실험환경을 구축하는 것이었음. 프로젝트 시작 이전 부터 pytorch template 과 weight & bias 를 공부하면서 미리 관련된 고민을 해왔기 때문에 기초 실험환경을 빠르게 구축할 수 있었음. 이를 통해 팀원들이 실험과정를 모니터링하고 실험결과를 관리할 수 있도록 함. 또한, cross validation 기반의 팀 내부 score 을 도입하여 팀원들 모두가 같은 기준을 통해 의사결정을 내릴 수 있도록 함. 결과적으로 팀원들이 더 많은 trial and error 를 경험할 수 있었고 이를 통해 모두가 만족할만한 좋은 결과를 낼 수 있었음

2. 내가 모델을 개선하기 위해 한 노력

- **Batch sampler**: EDA 를 통해 baseline 모델들이 대체적으로 평균값(7~8 점)으로 예측하는 경향이 존재하는 것을 파악함. weighted random sampler 를 구현하여 batch 를 생성할 때 모든 평점에서 균등하게 샘플링될 수 있도록 가중치를 조정함. 샘플링을 진행한 단일 모델의 성능은 오히려 떨어짐. 샘플링을 진행하지 않은 모델의 예측값과 앙상블했을 때 성능 향상을 보임. 샘플링을 통해 극단값의 데이터를 더 학습한 모델의 예측값이 앙상블 과정에서 기존 모델의 예측값과 가중합되면서 유효한 시너지 효과를 낸 것으로 해석됨

3. 내가 새롭게 시도한 변화와 효과

- **Git**: git 을 적극적으로 사용함. 9 일 동안 issue 11 개와 PR 20 개를 작성하면서 git 을 더 이상 무서움이 대상이 아닌 유용한 도구로 인식하게 되었으며, 내 나름대로의 git 사용 프로세스를 정립할 수 있었음

4. 마주한 한계와 아쉬웠던 점

- **Content Based Filtering**: cold start 유저들 대상으로는 텍스트 임베딩을 통한 유사도 기반의 Content Based Filtering 을 통해 평점을 예측하는 로직을 제안하고 실험까지 진행했음. 하지만 실험결과가 만족스럽지 않았고, 모듈화 과정에 시간이 많이 소요될 것 같아서 main branch 에 올리지 못하고 포기했던 점이 아쉬웠음. 내 아이디어를 main branch 올릴 수 있는 형태까지 구현할 수 있도록 개발 실력을 향상시킬 계획임. 아이템이나 유저를 더 효과적으로 표현할 수 있는 representation 에 대한 고민도 더 해보고 싶음
- **CS 기초 지식**: 소프트웨어 전공 팀원분이 이미지 임배딩 벡터를 빠르게 로드하는 기능을 구현하시고 해당 내용을 공유해주셨는데 당시에는 거의 이해할 수 없었음. 컴퓨터 전공 팀원과의 원활한 의사소통을 위한 CS 기초 지식을 따로 공부해야겠다는 생각이 들었음

5. 다음 프로젝트에서 시도해볼 것

- **병렬처리**: 프로젝트를 진행하면서 병렬처리를 통해 데이터 처리와 학습에 소요되는 시간을 단축시킬 수 있는 여지가 많음을 느낌. 컴퓨터 구조, 운영체제, 병렬처리 라이브러리 공부를 통해 다음 프로젝트에 병렬처리를 적용해 볼 계획임
- **더 좋은 협업문화**: 실험결과 공유 부분에서 노션은 효과적이지 않다는 생각이 들었고 매일 쏟아져 나오는 실험결과 파일들을 저장하고 관리하는 부분에서도 어려움을 느낌. case study 를 통해 더 좋은 툴이나 방법을 찾아서 다음 프로젝트 때 도입해 볼 계획임

2-4. 이준영_T5158

1. 내 학습목표를 달성하기 위해 한 노력

- **대회 프로세스 적응 & 이해**: EDA → 모델링 → 앙상블 순의 프로세스를 시도해보며 진행 과정에 대해 이해하려고 했다.
- **협업 환경에 적응하기**: Github, Wandb같은 협업 도구를 사용했다.

2. 내가 모델을 개선하기 위해 한 노력

- **모델 튜닝**: wandb sweep을 이용해 다양한 모델을 튜닝 했다.
- **모델링**: CNN과 Deep Conn을 합치는 등의 시도를 해보았고, 모델링 코드에 대한 이해도를 높일 수 있었다.

3. 내가 한 행동의 결과로 달성한 것 및 얻은 깨달음

- **불완전한 실험환경**: 실험환경을 어떻게 구축해야 하는지 잘 모르기 때문일 수 있지만, 실험 환경 구축 후 일부 기능은 오프라인으로 공유하거나 시각화 해야 하는 문제가 있었다. 때문에 실험환경에 필요한 것이 뭔지 더 이해가 필요하고 새로운 기능&툴을 찾아보면 좋을 것 같다.
- **라이브러리 활용**: baseline 코드에서 우리가 원하는 모든 기능을 담을 수는 없어 다소 불편한 점이 있었다. 때문에 모듈화를 조금씩 시도했는데, 보다 나은 모듈화를 위해선 추상화된 라이브러리 (pytorch-lightening, scikit-learn)를 사용하는 것이 좋을 것 같다.

4. 다음 프로젝트에서 시도해볼 것

- **협업 관리 툴 (ex: JIRA)**: 노션도 좋지만, 기록 정리 이외에 협업 툴로 제대로 활용하지 못하는것 같다. 따라서 더 효율적인 협업 툴이 필요하다. 경우의 따라선 협업 방식을 고민해 봐야 할 수 있다.
- **접근성 좋은 시각화**: wandb등을 이용해 실험의 결과를 더 쉽게 볼 수 있게 만들 필요가 있다. 예컨데, 모델 학습 후 자동으로 예측 결과의 분포를 wandb상에서 histogram으로 보여줄 수 있다.
- **더 빠른 모델 학습**: 우리의 머신은 GPU를 모두 활용하지 못한다. 이는 학습 데이터 준비, GPU 사용량 제한 등의 이유가 있는 것 같다. 병렬로 모델을 학습해 더 빠른 모델 학습 프로세스를 추구할 것이다.
- **테스트 프로세스**: 이번 프로젝트에선 자동화 테스트를 전혀 하지 않았다. 때문에 직접 사람이 모델을 돌려가며 테스트를 해야 했고, 오류가 있는 커밋을 머지하는 일이 잦았다. 리뷰를 적극적으로 해도 이런 문제는 잡기 어려울 수 있다. 때문에 다음엔 Github Action이나 Jenkins를 도입하는 게 좋겠다

2-5. 주혜인_T5208

1. 내 학습목표를 달성하기 위해 한 노력

- 회의 내용 노트에 열심히 적기 : 당장에 내가 다 내용을 흡수하지 못하더라도 기억하기 위해서 회의내용을 따로 열심히 기록해뒀다.
- 머신러닝 모델 도입 : 특히, 범주형 변수가 많은 상황에 효과적이라고 알려진 모델을 활용했다.
- Optuna, gridsearch cv : 하이퍼파라미터 탐색에 소요되는 시간을 줄이기 위해 시도했다.

2. 내가 모델을 개선하기 위해 한 노력

- Optuna 도입
- early stopping 구현

3. 내가 한 행동의 결과로 달성한 것 및 얻은 깨달음

- automl은 참고로 사용해야겠다. 마스터클래스에서 local minimum에 빠질 수 있다고 해주셨기 때문이다.
- 결국 모델 튜닝을 잘하기 위해선 실험의 결과를 잘 기록하고 분석에 많은 시간을 들이는게 좋은 것 같다. 이 부분이 나는 부족했는데 팀원들이 잘 해줘서 배울 수 있었다.

4. 내가 새롭게 시도한 변화와 효과

- 주어진 평점은 사실 이산형이고 우리 모델이 예측하는 평점은 연속형이다. 따라서 예측값을 반올림하면 어떻게 될지 궁금해서 해봤는데 rmse 가 낮아졌다. 이건 모델 성능이 아주 우수해야 효과가 있었을 것 같다. 또, 굳이 그럴 필요가 없는 것 같기도 하다.

5. 마주한 한계와 아쉬웠던 점

- **Git 사용**: 우리 팀에서 git 을 가장 낯설어한 팀원이 나인 것 같아서 초반에 조금 힘들었다. 내 개인 저장소에만 영향을 미치는게 아니다보니 나의 실수로 잘못될까봐 걱정이 많았다. 그래도 지금 많이 물어서 익숙해지지 않으면 나중에 더 어려울 것이라는 생각이 든 이후로는 두려워하지 않고 적극적으로 PR 도 날리고 이슈도 올려보고 했던 것 같다.
- **강의**: 프로젝트를 하면서 하루에 몇시간 투자해 강의를 듣는게 어려운 일이 아니었는데 강의를 좀 미뤄둔게 후회된다.
- **실험 결과 관리**: 실험을 많이 돌렸지만 결과를 제대로 관리하진 않았던 것 같다. 지속적으로 추적하면서 개선해나갔어야 했는데 실험을 많이 하는 것에만 좀 매몰되었던 것 같다.
- **딥러닝 모델 시도**: 어쩌다보니 팀 내에서 catboost 와 lgbm 을 중점적으로 맡게되었다. 다음에는 비교적 경험이 부족한 딥러닝 모델을 시도하고 싶다.

6. 다음 프로젝트에서 시도해볼 것

- 내 적극적으로 생각 공유하기 : 우선, 사실 선언했다가 실패하는게 걱정이 되어서 머릿속에 떠오른 아이디어를 선뜻 말을 미리 못한게 아쉽다. 다음 프로젝트에서는 조금 더 도전적으로 목소리를 내고 싶다. 그렇게 팀원들과 먼저 얘기하다보면 더 쉽게 해낼 수 있는 방법을 찾을 수도 있고 의견을 나누는 과정에서 시야가 넓어질 수 있는건데 두려움이 더 앞섰던 것 같다.
- 딥러닝 모델 구조 개선 : 레이어의 개수나 모델 구조를 직접적으로 개조해보는 시도를 하고 싶다.!
- 사실 프로젝트 기간에 그때그때 많은걸 느끼고 또 결심도 많이 했는데 막상 프로젝트가 끝나니까 잘 기억나지 않는 부분이 많다. 그래서 다음 프로젝트에서는 한줄이라도 매일 회고를 작성해두려고 한다.

3. 부록 (Appendix)

3-1. books.csv 의 컬럼별 EDA 결과

컬럼명	설명	특징 및 분석 결과
isbn	국제표준도서번호	- isbn 형식과 다른 값이 들어가 있는 경우가 있음
book_title	책의 제목	- 동일한 책이 다른 언어나 출판사를 통해 등장하는 경우도 있음
book_author	작가의 이름	- 결측값 없음
year_of_publication	책 발행 연도	- 대부분이 1900 년대 중반 이후이며, 종종 1400 년대의 이상치가 존재
publisher	책 발행 출판사	- 출판사의 이름이 잘못 표기된 경우가 있음 - 한 권만 출판한 출판사가 대부분 (6295/11571)
img_url	책 표지 이미지 접속 url	- 파일 경로와 동일
img_path	책 표지 이미지 파일 경로	- 이미지 접속 url 과 동일
language	출판 언어	- 결측치가 44% - 대부분의 language value 가 영어로 몰려있다.
category	책 카테고리	-결측치가 46%이며, 카테고리가 결측치 일때 language 와 summary 도 대부분 결측치 - 대부분의 category value 가 fiction 에 몰려있음
summary	책 내용 요약 설명	- 결측치가 45%

3-2. users.csv 의 컬럼별 EDA 결과

컬럼명	설명	특징 및 분석 결과
user_id	유저 식별 아이디	- 결측값이 발견되지 않음
location	유저의 위치정보	- 유저의 위치정보가 일반적으로 반점(,)으로 구분되어 표기됨 (city, state, country) - 하지만 많은 데이터가 제대로 표기가 되어있지 않음
age	유저의 나이	- 결측치가 40%이며 이상치가 존재 - 20 대 초반 ~ 30 대 중반까지의 사용자가 많음

3-3. train_ratings.csv 의 컬럼별 EDA 결과

컬럼명	설명
user_id	유저 식별 아이디
isbn	국제표준도서번호이자 책을 식별하는 고유 아이디
rating	해당 유저가 해당 책에 대해 매긴 평점 (1 점~10 점)

3-4. 사용한 모델 개요

Model	특징	구현 방법
FFM	user 와 book 의 서로 다른 feature 들의 interaction 을 학습하기 위한 모델 이미지나 텍스트를 제외한 book, user context dataset 을 이용해 학습	baseline 제공
DCN	user 와 book feature 들의 상호관계를 학습하기 위한 "Cross network"과 비선형 함수로 매우 복잡한 상호작용을 학습하기 위한 "Deep network(DNN)"으로 구성된 모델	baseline 제공
NCF	user, book 의 latent vector 들을 MLP 모델에 넣어 학습하는 user 와 book 의 상호작용을 비선형 함수로 학습하는 모델 초기에 튜닝없이 baseline 모델 중 꽤 성능이 잘 나왔던 모델	baseline 제공
DeepCoNN	book 의 summary 를 BERT 모델을 이용해 전처리 한 뒤, user, book vector 와 결합해 FM 을 적용한 모델 Cross validation 같은 기법이 없이도 일반화가 잘 됨	baseline 제공
3layer CNN context FM	CNN layer 로 이미지 vector 를 학습한 뒤 user, book vector 와 결합해 FM 으로 학습하는 모델 CNN 의 layer 수를 2 에서 3 으로 높임으로써 더 낮은 loss 에 더 적은 수의 epoch 으로 빠르게 도달 baseline 기준 단일 모델로 가장 성능이 잘 나와서 user, book vector 에 context dataset 까지 추가	baseline 참고해서 직접 구현
FFDCN	FFM 과 DCN 을 병합한 하이브리드 모델 FFM 의 출력과 DCN 의 출력을 concat 한 후, linear layer 의 출력으로 평점을 예측	baseline 참고해서 직접 구현
DeepCoNN_CNN	DeepCoNN 모델의 FM layer 에 이미지 vector 를 추가한 모델 DeepCoNN 보다 약간 성능이 좋아짐	baseline 참고해서 직접 구현
Catboost	트리 기반 모델로 범주형 dataset 에서 예측 성능이 우수한 모델 앙상블 효과가 가장 뛰어난 모델이며 단독 성능도 나쁘지 않았던 모델. Language 를 제외한 모든 feature 를 사용했을 때 가장 좋은 성능을 보임.	Scikit-learn 라이브러리를 활용하여 직접 구현
LGBM	트리 기반 모델로 Leaf-wise 방식을 사용해 학습속도가 빠르다는 장점을 갖는 모델. 단독 성능은 좋지 않았으나 다른 모델과 앙상블한 경우 좋은 결과를 보임. Language 를 제외한 모든 feature 를 사용했을 때 가장 좋은 성능을 보임.	Scikit-learn 라이브러리를 활용하여 직접 구현
Naïve Bayesian	범주형 데이터에서 성능이 좋다고 알려진 베이즈 정리를 적용한 확률적 분류기법 성능이 좋지 않아(2.603) 활용하지 않음	scikit-learn 라이브러리를 활용하여 직접 구현

3-5. Age 필드의 결측치 방법 별 설명

방법	설명
Normal Random	- 정규분포로 채우는 방식 - 이후 모델을 학습할 때 많이 사용
Total Zero	- 결측치를 0 으로 채우는 방식 - 이후 모델을 학습할 때 주로 적용
Total Average	- baseline 에서 처리하는 방식 - 성능은 나쁘지 않지만 분포가 고르지 않음
Country Average	- 큰 차이가 없음. 오히려 성능이 안좋아졌음 - 국가별 분포가 전체 분포와 크게 다르지 않아 그런 것 같음
Uniform Random	- 분포는 보다 고르게 나오지만 성능이 좋지 않음
KNN	- 인접 k 개의 이웃의 평균값으로 대체 - 전체 평균과 비슷한 결과가 나옴
Stratified	- 기존 분포와 같은 분포로 채움