Book Rating Prediction Wrap-up Report

RecSys-4조(점메추)

김시윤, 박승아, 이재권, 이현주, 장재원, 홍훈

1 프로젝트 Wrap Up

1.1 프로젝트 요약

본 프로젝트는 사용자와 아이템의 상호작용 정보와 메타 데이터를 활용하여 소비자의 책 평점 예측을 위한 효과적인 모델을 제작하는 것에 목적이 있다. 다양한 유형의 데이터를 활용하기 위해 FFM, DCN, DeepCoNN, ROP-CNN, CNN_FM, Catboost, XGBoost과 같은 여러 모델들을 사용하였다. 이 모델들의 장점을 취합하기 위해 이를 앙상블하여 최종적인 결과를 도출하였다. 본 프로젝트의 결과 catboost : DCN : others를 7:2:1의 비율로 앙상블한 모델의 Test RMSE가 2.14로 가장 성능이 높았으며 최종적으로 이를 제출하였다.

1.2 프로젝트 개요 및 환경

1.2.1 개요

뉴스기사나 짧은 러닝 타임의 동영상처럼 간결하게 콘텐츠를 즐길 수 있는 '숏폼 콘텐츠'에 비해 소비자들이 부담 없이 쉽게 선택할 수 있지만, 책은 완독을 위해 보다 긴 물리적인 시간이 필요하다. 또한 제목, 저자, 표지, 카테고리 등 한정된 정보로 내용을 유추하고 구매를 결정해야 하므로 선택에 더욱 신중을 가하게 된다.

우리는 소비자들의 책 구매 결정에 대한 도움을 주고자 책에 대한 메타 데이터(books), 고객에 대한 메타 데이터(users), 고객이 책에 남긴 평점 데이터(ratings)를 활용해, 1과 10 사이 평점을 예측하는 모델을 구축하고자 한다.

1.2.2 환경 -> 개발환경

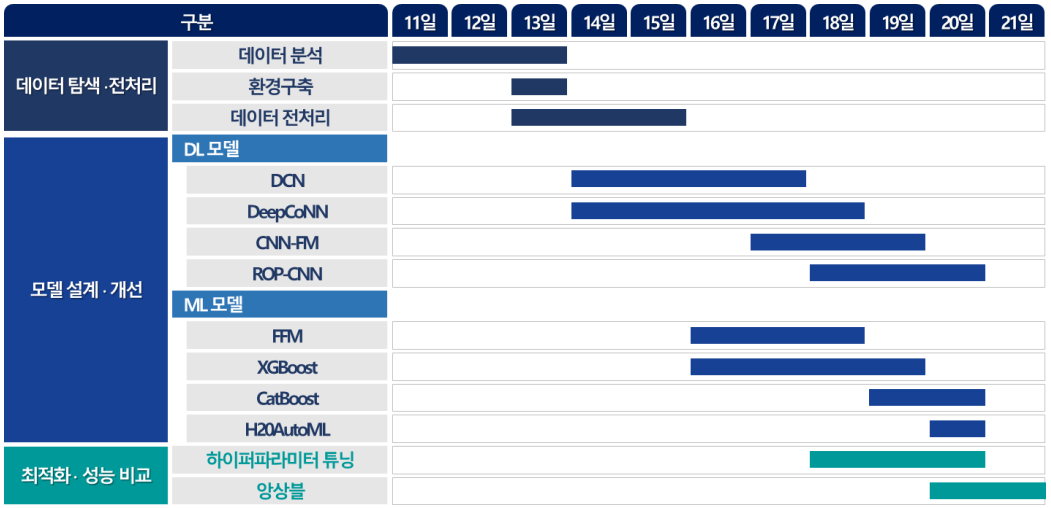
```
conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia pip install -r requirement.txt
```

conda 를 이용하여 가상환경 안에 공통적으로 사용하는 라이브러리를 설치하였다. 또한 나머지 라이브러리는 code 폴더 안에 사용하는 requirement.txt 파일을 이용하여 개발 환경을 구성하였다.

1.3 프로젝트 팀 구성 및 역할

전체	모델 실험, Wrap-up report 작성
김시윤	데이터 전처리, FFM 모델링, Catboost 모델 최적화
박승아	데이터 전처리, DCN 아키텍처 변경 및 실험, 하이퍼파라미터 튜닝
이재권	데이터 전처리, DeepCoNN 모델링, Catboost 모델링
이현주	데이터 전처리, CNN FM 아키텍쳐 및 AutoML 변경, WandB 적용
장재원	데이터 전처리, DeepCoNN 아키텍처 변경, ROP-CNN 모델 구현
홍훈	XGBoost 모델 구동, 앙상블 전략 수립, 하이퍼파라미터 튜닝

1.4 프로젝트 수행 절차 및 방법



1.4.1 프로젝트 개발환경 구축(Server, WandB, Github 등)

1.4.2 EDA를 통한 데이터 구조 파악 및 결측치 전처리 방안 논의

1.4.3 베이스라인 기반 추천시스템 모델링 – FFM, DCN, DeepCoNN, CNN-FM

1.4.4 논문 구현 - ROP-CNN

1.4.5 ML 기반 회귀 모델 구축 - CatBoost, XGBoost, H2OAutoML

1.4.6 Hyper Parameters Tuning

1.4.7 Weighted Ensemble을 통한 최종 결과물 제작

1.5 프로젝트 수행 결과

1.5.1 Data EDA 및 공통 전처리

Data EDA

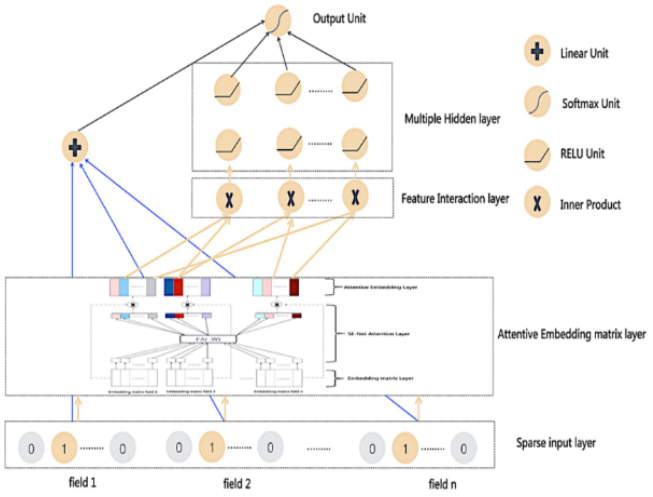
- 프로젝트에서 사용한 데이터는 kaggle의 Book-Crossing: User review ratings의 유저 프로필(Users), 책 프로필(Books), 유저-책 평점(Ratings)으로 이루어진 데이터 셋으로, 원본에는 이상치가 많았으나 대회용 데이터는 결측치만이 존재하였다.
- Users data
  - 유저 68092 명의 정보
  - ‘user_id’, ‘age’, ‘location’ 3개의 컬럼으로 구성
  - ‘age’에서 27833 건(약 41%)의 결측치 존재
  - ‘location’을 city, state, country로 나누어 결측치를 확인해본 결과, 각각 122, 3254, 2124 건의 결측치 존재
- Books data
  - 책 149570권의 정보
  - 'isbn', 'book_title', 'book_author', 'year_of_publication', 'publisher', 'img_url', 'language', 'category', 'summary', 'img_path' 총 7개의 컬럼으로 구성
  - 이 중 ‘language’, ‘category’, ‘summary’에 약 46%의 결측치가 존재. ‘category’의 경우, 같은 책 제목에 대해서 값이 있는 데이터와 누락된 데이터 존재
  - 또한, 책 표지 데이터 중 이미지 크기가 1x1 인 결측치 41802건 존재

공통 전처리

- Users - location
  - location 컬럼에 city, state, country 정보가 모두 담겨 있어 이를 ‘,’를 기준으로 분리해주었다.
  - city 값은 존재하지만 state와 country가 없는 경우, city 값을 기준으로 가장 많이 존재하는 location의 값으로 이를 대치하였다.
  - 이후에도 결측치로 남아있는 값은 ‘etc’ 처리를 해주었다.
- Books - language
  - 출판사가 모든 국가의 언어를 고르게 내는 경우는 많이 없을 것이라 생각해 ‘language가 null인 책의 출판사가 어느 언어로 책을 가장 많이 펴냈는가?’를 기준으로 채워넣었다.
  - 이후에도 채워지지 못한 약 160 개의 결측치는 ‘etc’로 처리하였다.
- Books - publisher
  - ISBN의 앞 4자리가 국가와 출판사를 나타낸다는 정보를 활용하여 같은 출판사로 추청되지만 이름이 일부 다른 경우, 대치하여 출판사 항목 수를 줄였다.
- Books - year_of_publication
  - 1300년대부터 2006년까지의 책으로 구성되어 있으나 대게 1990년대의 책이 주로 분포하였다.

○ 따라서, 1991년 이전에 발간된 책은 '1990', 1991~1995년에 발간된 책은 '1995', 1996~2000년에 발간된 책은 '2000', 2001년 이후에 발간된 책은 '2005'로 범주화 하였다.

1.5.2 FFM



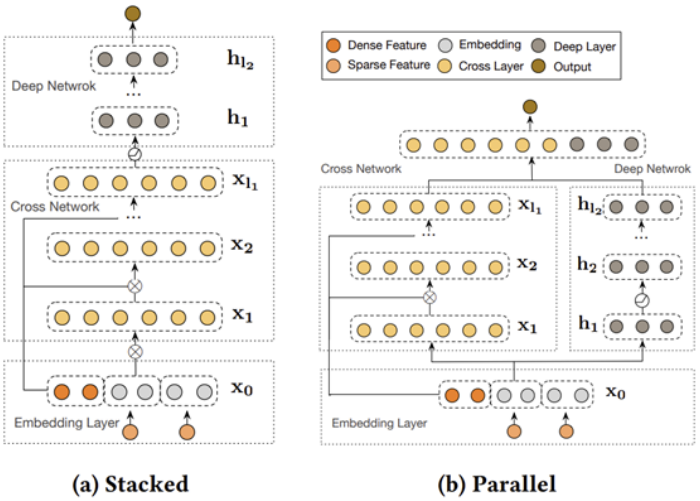
본 데이터는 유저와 책의 레이블을 받아 레이팅을 예측하는 데이터 구조임을 파악했다. 따라서 유저의 age, country 등의 variable을 카테고리화 하여 상호작용하는 필드로 사용할 수 있다고 생각하였다. 또한 많은 column들이 존재한다고 생각하였기에 각 column을 카테고리를 가진 필드로 구성하면 각 요소의 단순 상호작용보다 더 나은 Book Rating Score 예측을 보일 것이라는 가설을 바탕으로 모델링을 진행하였다.

먼저 Users는 age, location을 주로 처리하였다. 먼저 연령은 10살 이하 = 0 , 10~20살 = 1,..., 결측치 = 9로 categorize를 진행하였다. Books에서는 category에 대한 데이터 전처리를 진행하였다. 카테고리를 범주화 하기 위해 일반적으로 사용되는 책에 대한 카테고리 50여개를 임의로 만들어 해당하는 부분을 치환하는 category_high 칼럼을 생성하였으며 결측치는 fiction으로 처리하였다.이후 label encoding을 통해 수치화한 category_high_encode를 최종적으로 사용했다.

전처리 결과를 바탕으로 모델링을 진행한 후 HyperParameter Tuning을 실시하였다. 최종적으로는 embed_dim = 32, batch_size = 1024, epochs = 30을 사용하였다. 그 결과, baseline의 점수인 2.5047에서 2.4685(private 2.454)로 valid_loss를 감소시켰다.

1.5.3 DCN

해당 프로젝트에서 사용한 'Deep Cross Network-Version 2(이하 DCN)'는 DNN(Deep Neural Network)과 Cross Network를 결합해 여러 feature들 간의 복잡한 상호작용을 모델링하였다. feature들 간의 명시적 상호작용을 계산하는 Cross Network와 비선형 feature를 모델링하는 DNN을 결합하는 방식에 따라 Stacked 아키텍처와 Parallel 아키텍처로 나뉘게 된다.



(a) Stacked (b) Parallel

Stacked 구조는 Cross Network와 DNN이 순차적으로 결합해 Cross Network의 출력 값이 다음 네트워크의 입력 값으로 사용된다. 따라서 feature interaction을 보다 명확하게 구분할 수 있어 더 쉽게 해석할 수 있다. 반면, Parallel 구조는 Cross Network와 DNN을 병렬로 결합하여 출력을 연결하거나 합산하므로 계산 및 훈련 과정에서 더 효율적이다.

DCN의 경우, 논문에서 밝힌 바와 같이, 더 좋은 성능을 보이는 아키텍처가 데이터 셋에 따라 다르다. Stacked 구조는 0과 1로 이루어진 클릭 로그 데이터(Criteo)에, Parallel 구조는 1점부터 5점까지의 rating으로 이루어진 MovieLens에 강점을 보인다. 베이스라인으로 주어진 DCN은 Stacked 구조가

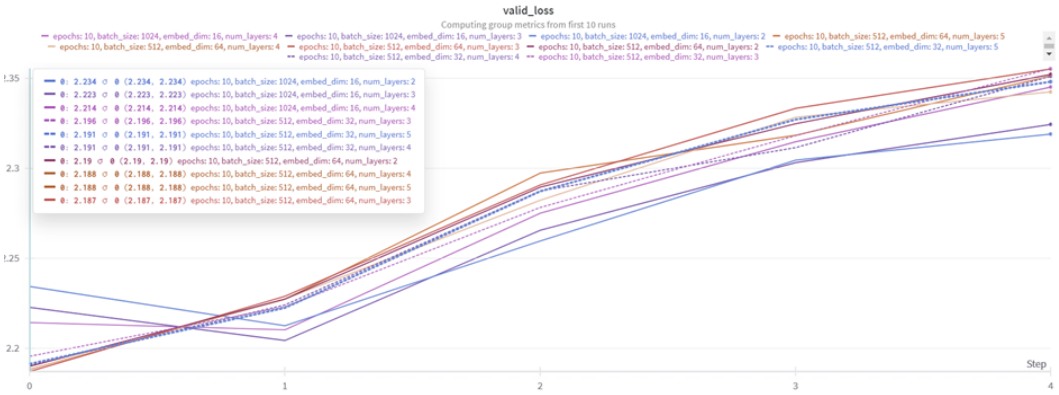
사용되었는데, 이번 프로젝트의 데이터셋이 MovieLens와 유사하므로 'Parallel 구조에서 더 나은 Book Rating Prediction 성능을 보일 것이다.'라는 가설을 세우고 이를 구현하였다.

사용된 데이터는 user_id, isbn, ratings이며 편의를 위해 indexing 처리하였다. 다음은 기본 parameter 상태에서 DCN-Stacked와 DCN-Parallel의 대회 Public Score(RMSE) 값을 나타낸 표이다. 가설과 같이, Parallel 구조에서 향상된 성능을 보임을 알 수 있었다. 모델의 활성화 함수를 기존의 ReLU에서 LeakyReLU, Sigmoid 등으로 바꾸어 보았으나 ReLU가 가장 좋은 성능을 보였다.

DCN-Stacked	DCN-Parallel
2.3391	2.2125

이후, Hyper Parameter Tuning을 진행하였다. Grid Search 방식을 사용하였으며 대상 파라미터와 후보군은 batch_size={512, 1024, 2048}, embed_dim={16, 32, 64}, num_layers={2, 3, 4, 5} 였다. epochs 또한, Hyper Parameter Tuning 대상으로 고려할 수 있었으나 대부분의 모델에서 epochs가 3 이상이 되는 경우, 과적합이 발생하는 양상을 보여 기본 세팅인 10으로 진행하였다.

아래는 Grid Search의 결과이며, batch_size={512}, embed_dim={64}, num_layers={3} 일 때 valid_loss(RMSE)에서 가장 좋은 결과가 나와 이를 모델에 적용하였다.



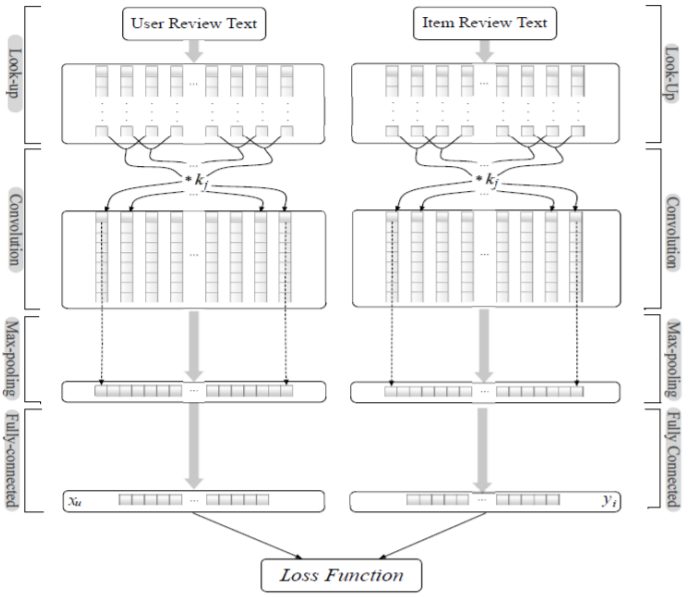
이후, mlp의 차원, learning rate, seed를 추가로 Grid Search 하였으나 mlp의 차원, learning rate는 기본 설정인 (16, 16), 0.001에서 가장 나은 성능을 보였으며 seed에 변화를 주자 성능이 약간 향상되는 모습을 보였다. 이를 통해 최종적으로 valid_loss 2.175인 모델을 구현하였다.

1.5.4 DeepCoNN

텍스트 데이터로 책 요약이 주어졌고, 이를 활용하여 책 추천에 다양성을 부여하고자 하였다. 책 요약의 45%는 결측치였기 때문에 결측치가 존재하지 않는 책 제목으로 책 요약의 결측치를 대체하여 사용하였다. 또한, BERT 기반의 언어 모델에서는 특수문자나 불용어를 제거하는 전처리 방법이 정보 손실을

일으킬 수 있으므로 전처리에서 제외하였고, html 기호 처리, 공백 제거, 소문자로 변환만을 텍스트 전처리로 활용하였다.

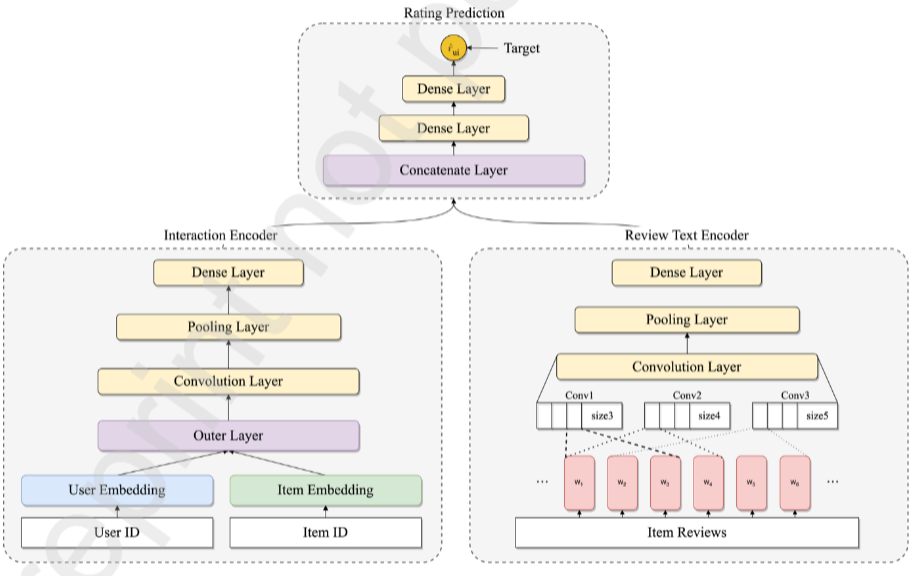
DeepCoNN은 사용자와 아이템의 리뷰 텍스트 정보를 임베딩하고, CNN 구조로 학습하여 사용자와 아이템의 상호작용을 예측하는 모델이다. 이 프로젝트에서는 리뷰 대신 책의 요약을 아이템 임베딩에 활용하고, 사용자가 평점을 매긴 책의 요약을 사용자 임베딩에 활용하였다.



사용자와 아이템에 대한 임베딩 차원이 특징을 표현하기에 충분하지 않다고 판단하여 32차원에서 64차원으로 변경하였다. 또한, 더 효과적인 텍스트 임베딩을 추출하기 위해 사전학습 모델로 DeBERTa-v3를 활용하였다. 나머지 부분에서는 DeepCoNN의 기본 구조를 사용하였다. 결과적으로 valid loss 2.331을 기록하였다.

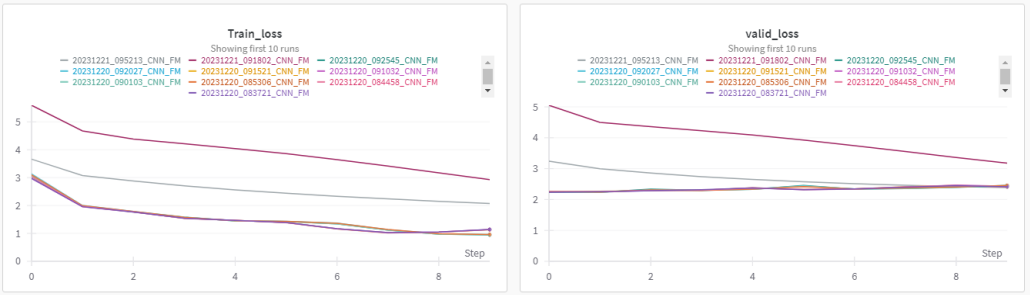
1.5.5 ROP-CNN

DeepCoNN은 사용자 리뷰 데이터를 활용하는 모델이다. 위 논문에서 말하는 리뷰 데이터란 어떠한 User가 어떤 Item에 리뷰를 남겼는지를 포함하고 있는 데이터이다. 그러나 우리에게 주어진 데이터에서는 Item에 대한 User의 Rating 데이터와 Item의 Summary 데이터이다. 이 데이터만으로는 User와 Item의 상호관계를 알기 힘들다고 판단하였다. 그래서 우리는 User와 Item의 상호작용을 고려하기 위해서 ROP-CNN을 활용하였다[논문1]. ROP-CNN은 User와 Item간의 복잡한 상호작용을 설명 못하는 DeepCoNN의 한계를 해결하기위해서 제안된 외적(Outer Product)기반 하이브리드 추천 시스템이다.



Rop-CNN의 구조는 Review Text Encoder, Interation Encoder, Rating Prediction으로 나누어진다. Review Text Encoder에서는 DeepCoNN과 같은 Item Reviews 데이터를 사용하였고, Interaction Encoder에서는 Item에 대한 User의 Rating 데이터를 사용하였다. 기본적인 구조는 논문과 동일하게 구현하였으며, 우리 문제상황에 맞게 User, Item의 Embedding 및 Layer(정규화, 활성화함수)를 변경하였다.

1.5.6 CNN-FM



CNN FM 은 유저/상품 벡터와 이미지 벡터를 결합하여 FM 로 학습하는 모델이다. 기존 베이스라인에서는 32x32 크기의 이미지 데이터를 사용하였는데 이미지 특징을 추출할 수 없는, 1x1 의 이미지를 제외한 이미지 데이터들의 통계를 내어봤을 때 이미지의 평균적인 크기는 가로 50.09, 세로 74.75 으로 기존의 방법이 이미지의 특징 정보를 충분히 추출할 수 없다고 판단하여 입력 이미지의 크기를 64x64로 변형하였다. 또한 단순 Resize 가 아닌 RandomResizedCrop 으로 transforms 을 적용하고, CosineAnnealingWarmRestarts 스케줄러를 사용하여 진행하였다.

기존 베이스라인 코드에서 CNN Layer 를 정의할 때는 Conv2D, ReLU, MaxPool2d, Conv2d, ReLU, MaxPool2d 순서로 기초적인 이미지 특징을 추출했었다. 실험을 통해 레이어를 하나 더 쌓는 것이 성능이 높았기에

이미지로부터 좀 더 복잡한 특징을 추출할 수 있다고 판단하였고, 변경된 구조에서는 Conv2D, ReLu, MaxPool2d, ReLu, MaxPool2d, Conv2d, ReLU, MaxPool2d 순서로 CNN Layer 를 정의하여 사용하였다. 이를 통해 CNN FM 은 베이스라인 Train Loss (1.142), Valid Loss (2.433) 에서 최종 성능 Train Loss (1.386), Valid Loss (2.367) 를 기록하였다.

1.5.7 Catboost

본 프로젝트는 카테고리형 데이터를 바탕으로 rating을 예측하는 과제이기 때문에 카테고리형 자료를 분석하는데 뛰어난 효과를 보이는 Catboost 모델을 사용하고자 하였다. 모델링을 진행하고자 다음과 같은 절차에 따라 진행하였다. 먼저 EDA 결과를 바탕으로 preprocessing의 결과를 적용하였다. 추가적으로 공통 EDA에서 사용된 publication_year처리를 바탕으로 사용하였으며, Author의 결측치는 others로 대체하였다. Catboost는 HyperParameter Tuning에 초점을 두어 진행하였으며 K=5인 KFold를 진행하였다. 데이터 간의 동질성을 유지하기 위해 StratifiedKFold를 사용하였다. 튜닝의 결과 최종적인 파라미터는 다음과 같이 설정하였다. 결과는 2.1565(private 2.133)를 기록하였다.

```
params = {
    "random_state": 42,
    "objective": "RMSE",
    "learning_rate": 0.01,
    "bagging_temperature": 0.1,
    "n_estimators": 5000,
    "max_depth": 10,
    "random_strength": 10,
    "l2_leaf_reg": 1e-7,
    "min_child_samples": 20,
    "max_bin": 300,
    "od_type": "Iter",
    "task_type": "GPU",
    "devices": "0",
    "cat_features": list(train_ratings.drop(['rating'], axis=1).columns)
}
```

1.5.8 XGBoost

프로젝트의 문제가 Book Rating을 예측하는 회귀문제였기 때문에 회귀문제를 해결하는데 뛰어난 모델인 XGboost 모델을 이용하기로 결정하였다. 하이퍼 파라미터는 다음과 같이 설정하였으며 5-fold validation을 이용하였다. 결과는 Valid loss. average 2.224를 기록하였다.

```
# GENERAL PARM
# gbtree(기본값), gblinear, dart
param = {'booster' : 'gbtree'}
#param = {'booster' : 'dart'}

# BOOSTER PARM
# eta = lr 0.3
param['eta'] = 0.3
# weak learn 반복수 10
param['num_boost_around'] = 10
# 트리 최대 깊이 6
param['max_depth'] = 9
# L2 Reg 적용 값 1
#param['lambda'] = 1
# L1 Reg 적용 값 0
#param['alpha'] = 0

# TRAIN PARM
# reg:squarederror(기본값), binary:logistic, multi:softmax, mult
param['objective'] = 'reg:squarederror'

# EVAL metric
# rmse, mae, logloss, error, merror, mlogloss, auc
param['eval_metric'] = 'rmse'
```

1.5.9 H2OAutoML

다양한 머신러닝 방법론들을 적용하기 위해 AutoML 을 사용하였다. RMSE 뿐만 아니라 MSE, MAE, RMSLE 등의 평가지표를 활용하여 결과를 볼 수 있었고, 여러 모델들 중 RMSE 의 성능이 가장 결과가 좋았던 앙상블 모델을 사용하여 이후 catboost 를 진행하였다.

model_id	rmse	mse	mae	rmsle	mean_residual_deviance
StackedEnsemble_AllModels_1_AutoML_2_20231221_83837	2.26559	5.13288	1.72701	0.402981	5.13288
StackedEnsemble_BestOfFamily_1_AutoML_2_20231221_83837	2.26844	5.1458	1.73035	0.403501	5.1458
XRT_1_AutoML_2_20231221_83837	2.28673	5.22915	1.75791	0.405893	5.22915
DRF_1_AutoML_2_20231221_83837	2.28925	5.24067	1.75523	0.405928	5.24067
GBM_4_AutoML_2_20231221_83837	2.29868	5.28392	1.76298	0.408426	5.28392
GBM_1_AutoML_2_20231221_83837	2.30525	5.31419	1.76927	0.409193	5.31419
GBM_3_AutoML_2_20231221_83837	2.3166	5.36664	1.77924	0.410953	5.36664
XGBoost_2_AutoML_2_20231221_83837	2.32075	5.3859	1.77294	0.408433	5.3859
XGBoost_3_AutoML_2_20231221_83837	2.32284	5.3956	1.78284	0.410894	5.3956
GBM_2_AutoML_2_20231221_83837	2.33185	5.43751	1.79306	0.413061	5.43751

[10 rows x 6 columns]

1.5.10 Model Ensemble 및 Competition 결과

	DCN	DeepCoNN	ROP-CNN	CNN-FM	FFM	XGBoost	CatBoost	H2OAutoML
Valid Loss	2.175	2.331	2.336	2.283	2.468	2.224	2.149	2.265

	Ensemble1	Ensemble2
Test Loss	2.133	2.134

Catboost, DCN-par ensemble (weight: 0.7, 0.3)

전체적으로 성능이 뛰어났던 Catboost를 Hyperparameter 튜닝을 통해 최적화하고 이를 보완해줄 수 있는 DCN-par 모델을 같이 이용하여 ensemble을 진행하였다. Test RMSE 결과는 public 2.14, private 2.133으로 가장 좋은 성능을 보였다.

Catboost, DCN-par, Others(XGboost, H2OAutoML, CNN-FM, DeepCoNN, ROP_CNN) (weight: 0.7, 0.2, 0.1)

가능한 여러가지 모델의 결과를 ensemble하여 모델의 장단점을 보완하려는 시도를 하였다. 각각 모델은 Hyperparameter 튜닝을 통해 최적화된 모델이며, 머신러닝 기반 모델 부터 이미지, 텍스트를 사용하는 모델들을 모두 취합하였다. Test RMSE 결과는 public 2.1408, private 2.1342로 나타났다.

1.6 자체 평가 의견

잘했던 점
● 다양한 모델을 학습하고 구현하려 노력하였다. ● 가설을 설정하여, 해당 가설을 토대로 작업을 구성하였다.
시도했으나 잘 되지 않았던 것들
● DeepFM, GNN 등의 모델을 구현해보려 하였으나 시간 상의 이유로 구현하지 못하였다. ● 여러가지 가설에 기반하여 모델을 구성하고 예측을 시도했으나, 결론적으로 좋은 결과가 나오지 않았다.
아쉬웠던 점들
● 먼저 다같이 EDA를 진행하지 못하였다. ● 논문과 같은 다양한 feature를 사용한 DCN을 시도해보지 못하였다. ● 좋지 않은 결과가 나왔을 때, 해당 문제를 해결하기 위한 적절한 해결방안을 찾지 못하였다.
프로젝트를 통해 배운 점 또는 시사점
● WandB를 통해 실험을 기록하는 방법을 학습하였다. ● 협업을 통해 작업하는 시퀀스를 경험할 수 있었다.

2 개인 회고

김시윤

내가 한 행동의 결과로 어떤 지점을 달성하고**,** 어떠한 깨달음을 얻었는가**?**

: FFM의 경우 기존의 baseline이 2.50였으나, 이와 같이 전처리를 진행하고, 추가적인 필드를 생성하였더니 2.43으로 감소하였다. 이와같이 진행한 후 느낀점은 전처리를

완벽하게 하진 못했지만, 필요한 부분을 캐치하고, 그 부분을 고도화시켜 파생변수를 만드는 것이 중요하다고 생각하게 되었다는 점이다. 추가적으로, 이러한 전처리를 위해 필드의 수, Big O와 같은 메모리 관리도 중요하다는 것을 알게 되었다.

전과 비교해서**,** 내가 새롭게 시도한 변화는 무엇이고**,** 어떤 효과가 있었는가**?**

: 어느 모델 하나만 생각해서 그것만 해보는 것이 아닌, 다양한 모델의 구조를 확인하고, 그 구조에는 어떠한 데이터가 필요한지 직접 살펴보며 생각해보았다. 이를 통해 다양한 task에서 데이터를 어떻게 다루는지 알게 되었으며, 각각의 전처리 방식이 다름을 알게되었다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

: 직접 서버를 세팅하고, 다른 환경을 설정하는 경험이 처음이었기 때문에 시간을 많이 잡아먹어 예상보다 더 늦게 작업을 진행하게 되었다. 충분한 시간이 있었으면 더 많은 시도를 하고, 전처리에 대해서도 만족할만큼 진행할 수 있었을 것이라고 사료된다. 또한 팀플레이로 진행하는 것이 처음이라 기초적인 부분을 까먹고 모델을 구현하는 것에만 몰두했던 것 같다. 천천히 큰 그림을 그리고 거기서부터 시작하였으면 더 좋은 결과를 얻을 수 있었을 것이라 생각된다. 추가적으로 한 두개의 모델을 그저 사용하기만 했다고 생각한다. 그 결과, 하이퍼 파라미터를 고려하여 적절한 튜닝을 진행하지 못하였고, 각 모델을 사용하면서 다른 모델과의 관계를 파악하는 등 이론적인 지식을 활용하지 못하여 이를 활용할 수 있는 다양한 방법을 생각해내지 못했다는 점이 아쉽다고 생각한다.

한계**/**교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까**?**

: 이번 프로젝트에서는 한두개의 모델만 확인하고 고도화 작업을 진행했던 것 같다. 더 많은 모델들간의 상호작용을 파악하고 각 모델이 의미하는 바를 확인한 후, 이를 결합하여 각각의 장점을 살리고, 단점은 상쇄시킬 수 있는 효율적인 모델을 만들어 보고 싶다. 추가적으로 전처리에 있어서도 다른 팀원이 해주겠지하는 마음가짐보다는 내가 모든 다뤄보지 않았던 유형의 데이터도 직접 찾아서 간단하게나마 처리해보는 시도를 하여 다양한 종류의 데이터를 처리할 수 있는 능력을 기를 것이다.

박승아

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가**?**

이번 Book Rating Prediction 프로젝트에서 내 학습목표는 베이스라인 코드를 Hyper Parameter Tuning만 하는 것이 아닌, 그동안 배워왔던 모델의 실제 논문을 읽고 그 논문의 주안점이 무엇인지 파악해 우리의 프로젝트를 위해 데이터와 모델을 어떻게 바꾸어야 할지 고민하는 것이었다. 이를 위해 DCN-V2 논문을 직접 읽어보고 다양한 이들이 작성해 놓은 논문 리뷰를 살펴보며 이해하고 구현하는 연습을 하였다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

객체 지향 프로그래밍, CLI가 익숙하지 않아 베이스라인 코드를 분석, 실행하는 것이 어려웠다. 구조를 파악하는 데 시간을 많이 소요돼 DCN 모델을 더욱 고도화하거나 가설을 명시적으로 설정해 이를 위한 실험을 직접 수행해보지 못하였다. 또한, 베이스라인 코드에 맞게 main.py 파일을 실행하면 바로 분석이 가능한 모델을 만드는

것에만 집중해 도리어 더 잘 알고 있던 ipynb를 이용한 트리 모델을 만들어 보지 못하였다. 그리고 DCN 모델이 다양한 범주형 데이터를 사용할 수 있음을 논문을 통해 알고 있었지만 베이스라인 코드를 어떻게 수정해야 할지 고민하느라 시간을 많이 소요하였다. 마지막으로, VSCode를 활용해 Git을 복제하고 branch를 내어 협업하는 데 익숙하지 않아 GUI를 사용하였다. 팀원들의 코드와 충돌이 일어날까 겁먹고 local에서 주로 작업하였다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

팀 간의 소통을 통해 데이터와 모델 간 상관관계 등을 함께 파악하고 명시적인 가설 설정 과정을 거치며 연구자로서의 모델링 실험을 시도해보고자 한다. 그리고 DCN 논문에서 모델의 성능을 검증할 때 MovieLens 데이터의 7개의 feature를 사용한 점을 바탕으로 이를 다음 프로젝트에서 실제로 구현해보고 싶다. 또한, 베이스라인 코드에 의지하는 것이 아닌, ipynb 형태를 이용하더라도 직접 처음부터 모델을 만드는 경험을 하고 싶다. 더불어 VSCode와 Git을 통해 협업할 때, 적극적으로 충돌에 부딪혀보고 해결해 나가는 경험을 해보고자 한다.

이재권

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

책 추천 프로젝트에서 텍스트 데이터 기반 모델링과 머신러닝 기반 모델링을 맡았다. 사용자와 아이템에 대한 상호작용 데이터와 텍스트 데이터를 활용하여 DeepCoNN을 모델링하였고, 전처리 과정을 수정하고 사전학습 모델을 DeBERTa로 변경하여 모델을 고도화하였다. 추천 시스템에 텍스트 데이터를 활용해 보는 경험을 통해 텍스트 데이터는 성능 향상보다는 추천의 다양성을 향상시키는 데에 활용하는 것이 더 효과적인 것 같다는 생각이 들었다.

딥러닝 기반의 모델들이 효과적이지 못했고, 사용자와 아이템의 상호작용 정보가 희소했기 때문에 아이템의 특징을 잘 반영할 수 있는 머신러닝 기반 모델을 도입했다. 또한, 아이템의 특징 데이터가 대부분 범주형이기 때문에 범주형 데이터 처리에 장점이 있는 Catboost를 선택하였다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

사용자와 아이템의 상호작용 정보가 희소해서 상호작용 기반 모델들의 성능이 좋지 않았다. 또한, 상호작용 정보가 충분하지 않은 상태에서 텍스트나 이미지만으로 책을 추천하는 것은 효과적이지 못했다. 오히려 책의 메타 데이터가 풍부하여 책의 특징들을 잘 반영할 수 있는 Catboost가 책 추천에 더 효과적이었다.

데이터의 이해와 전처리에 시간을 많이 투자하지 못해서 Catboost의 성능을 제대로 끌어내지 못했다. 데이터 전처리를 우선적으로 완료한 후에 모델링을 했어야 하는데 모델링에 너무 집중한 나머지 중요한 부분을 간과했다.

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

모델 이해와 디버깅에 생각보다 많은 시간이 걸려서 데이터의 이해와 전처리에 시간을 투자하지 못했다. 데이터의 이해와 전처리의 중요성을 간과하고, 모델링에 집중한 것이 추천 성능이 좋지 않았던 원인으로 보인다.

다음 프로젝트에서는 데이터를 이해하는 데에 시간을 더 투자하고, 개인별로 전체적인 전처리 과정과 학습 과정을 경험해본 후에 협업을 시작하는 것이 좋을 것 같다.

이현주

내가 한 행동의 결과로 어떤 지점을 달성하고**,** 어떠한 깨달음을 얻었는가**?**

이번 프로젝트에서 이미지 데이터와 CNN FM 을 맡았다. 이미지 데이터에 대한 EDA 를 진행하다보니 1x1 크기의 이미지로부터 피쳐를 추출하여 해당 데이터에 대한 충분한 정보를 추출할 수 없다는 것을 문제로 설정하였기 때문이다. 이를 극복하기 위해 이미지 데이터에 대한 전처리를 진행하고, Transform 방법을 적용하고, 레이어를 다시 쌓아 모델을 구현하고, 여러가지 스케줄러를 사용하며 하이퍼파라미터 튜닝을 하는 과정을 겪는 게 의미 이었다고 생각한다. 대부분의 시간을 CNN FM 에만 집중하였고 해당 모델을 단독으로 사용했을 때는 다른 모델과 비교했을 때 성능이 좋지 못했는데 , 다른 팀원들과 앙상블 하였을 때 성능이 좋아져서 뿌듯함을 느꼈었다.

전과 비교해서**,** 내가 새롭게 시도한 변화는 무엇이고**,** 어떤 효과가 있었는가**?**

팀원들에게 WandB 사용 방법을 공유하거나, Gitmoji 를 도입하자고 하는 등 같이 협업하는 방식을 좀 더 배운 것 같다. 제안을 하고, 코멘트를 달고, 리뷰를 하는 등 좀 더 실시간으로 협업을 할 수 있었던 것 같다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

아무래도 책의 표지 이미지가 평점에 큰 영향을 미치지 않는다는 점이 아쉬웠던 것 같다. 성능보다는 문제를 설정하고, 이를 해결하는 과정을 겪는 게 더 의미있는 일이라 생각해서 중간 중간 의심이 들더라도 계속 뿌리치며 진행했는데, 처음에 충분히 더 생각하고 문제를 설정했으면 성능도 좋지 않았을까란 생각이 들었다.

한계**/**교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까**?**

이미지 데이터에 대한 시각화를 진행하고 싶다. 이번에는 이미지 데이터를 흑백으로 작은 크기로 일부만 봐서 평점에 직결되는지에 대한 여부를 충분히 판단하지 못했다고 생각하는데, 이후에는 원본 데이터에 대해 더 많은 데이터를 시각화를 시도해보고 싶다.

장재원

내가 한 행동의 결과로 어떤 지점을 달성하고**,** 어떠한 깨달음을 얻었는가**?**

나는 최신의 기술을 현 문제상황에 적용 시키는 것을 좋아한다. 이번 프로젝트에서도 텍스트 요약 데이터 활용을 위한 다양한 방법론을 탐색하였다. 그 과정에서 ROP-CNN이 우리의 문제 해결에 적합하다고 판단하였고 구현 및 튜닝에 노력을 하였다. ROP-CNN은 이번달에 등재된 최신 논문이라 참고 자료의 부족이 큰 어려움이었지만, 논문에서 제시된 구조를 바탕으로 구현하고 우리의 문제상황에 맞게 조정하는 과정은 매우 유익하였다. 그러나 논문의 문제상황과 우리의 문제상황이 상이했기에, 원하는 성과를 얻지는 못했다. 이번 경험을 통해, 새로운 기술 적용에 있어서 이론과 실제의 차이를 더욱 명확하게 인지하였다. 이는 앞으로의 프로젝트에서 더욱 효과적인 기술 적용을 위한 중요한 경험이 될 것이다.

전과 비교해서**,** 내가 새롭게 시도한 변화는 무엇이고**,** 어떤 효과가 있었는가**?**

대회에서 이미 검증된 모델을 사용하는 대신, 나는 최신 텍스트 리뷰 기반 모델을 도전해보고 싶었다. 머신러닝 기반의 모델들이 보여주는 높은 성능에 비해, 딥러닝 기반 모델은 그만큼 뛰어나지는 않았다. 하지만, 딥러닝 모델 역시 충분히 경쟁력 있는 성능을 내는 것이 가능하다는 점을 보여주고 싶었다. 이번 경험을 통해, 기존의 방식을 벗어나 새로운 기술에 도전하는 것에 대한 즐거움과 앞으로의 다양한 시도를 위한 동기 부여가 되었다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

가설 설정과 분석 과정에서 아쉬움이 남는다. 나는 DeepCoNN의 한계를 보완하기 위해 ROP-CNN을 구현하는 것에만 초점을 맞추었다. 그러나 우리의 데이터가 사용자 리뷰 텍스트 부분에서 성능을 발휘하지 못하는 원인이 무엇인지, 그리고 이 문제를 해결하기 위해 어떤 방법들을 사용하고, 그 결과가 어떻게 나타날지에 대해 충분히 고민하지 않았다. 다양한 관점에서 가설을 설정하고 보다 전반적으로 알아보지 않은 상태에서 구현한 ROP-CNN은 예상한 성능을 보여주지 못했다. 이번 경험을 통해, 모델을 선택하고 구현하는 것만큼이나 그 기반에 대한 충분한 가설 설정과 분석이 중요하다는 것을 깨달았다. 이는 앞으로의 연구에서 반드시 고려해야 할 부분이다.

한계**/**교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까**?**

이번 경험은 가설 설정과 분석이 얼마나 중요한지를 깨닫게 해주었다. 앞으로는 현상이 발생하는 원인을 철저하게 분석하고 이해하는데 더 많은 시간을 투자할 계획이다. 그리고, 다양한 모델에 대해 미리 가설을 세우고, 그 가설을 체계적으로 검증하는 과정을 거치려 한다. 이러한 과정을 통해 다양한 모델을 문제 해결에 적용할 것이다. 각각의 모델이 가지는 장단점을 이해하고, 문제에 가장 적합한 모델을 선택하고 활용해 나아갈 것이다.

홍훈

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가**?**

프로젝트에서 원하는 목표(Book Rating Prediction)을 확실히 하고 해당 문제를 풀기 위한 최적의 모델(Gradient Boosting 기반 회귀 모델, XGboost)을 선정하였다.

XGboost를 구동한 base 결과 및 커뮤니티 데이터의 속성을 통해 Book Rating 이라는 타겟 데이터의 특성을 파악하고자 노력했다.

데이터 특성들을 잘 예측하기 위한 가설을 설정하였으며, 그에 따른 모델을 구성하였다.

실제 모델링 결과, 모델링 성능이 좋지 않았는데 이에 대한 문제 파악과 대안을 제시하였다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

첫 프로젝트라 모델 구동에 더 집중하였기에 AI 모델에 있어 데이터 전처리의 중요성을 다소 소홀히 한 점이 있다. 모델링 성능에 고품질의 데이터는 반드시 필요하다는 것을 확인하였으며, 프로젝트 후반부에 데이터를 주의 깊게 보면서 생기는 여러가지 인사이트가 있었으나 이를 제대로 활용하지 못한 점이 아쉬웠다.

한계**/**교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가**?**

팀원들과 함께 체계적인 틀 안에서 데이터를 분석해 보고 싶다. 또한, 이번 프로젝트에서는 Github를 이용하여 협업을 수행하였지만 제대로 사용하지 못한 것 같다. 이후 프로젝트에서는 Github의 PR, Issue등을 잘 활용하여 협업을 진행하고 싶다.