Wrap-up Report

주차	Level1 Project
날짜	@2023년 4월 21일
모더레이터	

Part1. 팀 프로젝트 Wrap Up

1-1. 프로젝트 개요

프로젝트 주제

- 사용자의 책 평점 데이터를 바탕으로 사용자가 어떤 책을 더 선호할지 예측하는 태스크입니다.
- 해당 경진대회는 이러한 소비자들의 책 구매 결정에 대한 도움을 주기 위한 개인화된 상품 추천 대회입니다.
- 리더보드는 평점 예측에서 자주 사용되는 지표 중 하나인 RMSE (Root Mean Square Error)를 사용합니다.

활용 장비 및 재료

- ai stage server : V100 GPU x 4
- python==3.8.5
- torch==1.7.1
- CUDA==11.0

프로젝트 구조 및 사용 데이터셋의 구조도

level1_bookratingprediction-recsys-13
├ ensemble.py
├ main.py
├ src
| ├ preprocess
| ├ ensembles
| ├ models
| └ train
└ submit

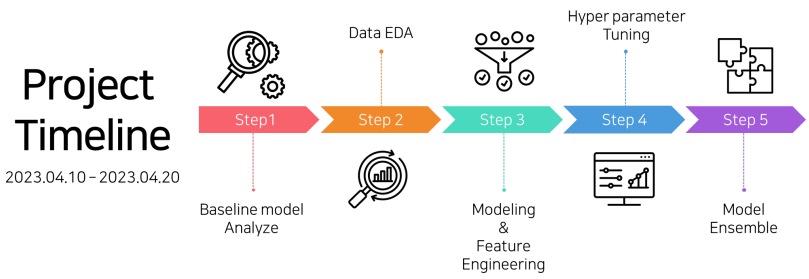
data
├ users.csv
├ books.csv
├ sample_submission.csv
├ test_ratings.csv
└ train_ratings.csv

1-2 프로젝트 팀 구성 및 역할

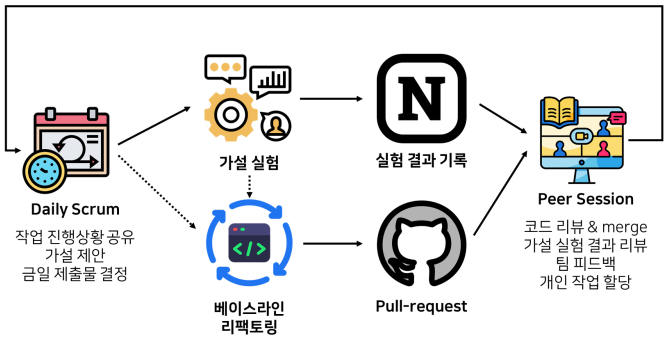
- 곽희원_T5015
  - 프로젝트 개발(`EDA`, `Hyper Parameter Tuning`, `Stratified K-Fold`, `Modeling`, `Ensemble`)
- 임도현_T5170
  - 프로젝트 개발(`EDA`, `Data Preprocessing`, `Feature Engineering`, `Modeling`)
- 임지수_T5176
  - 프로젝트 개발(`EDA`, `Data Preprocessing`, `Ensemble`)
- 조현석_T5205
  - 프로젝트 개발(`EDA`, `Hyper Parameter Tuning`, `Manage Data Pipeline`)

1-3 프로젝트 수행 절차 및 방법

- 수행 절차



- 수행 방법



1-4 프로젝트 수행 결과

- 탐색적 분석 및 전처리 (학습데이터 소개)
  - 학습 데이터는 306,795건의 평점 데이터(`train_rating.csv`)이며, 149,570건의 책 정보(`books.csv`) 및 68,092명의 고객 정보(`users.csv`) 또한 주어집니다.
  - 각 데이터가 구성하고 있는 변수는 다음과 같습니다.
    - 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

      - `user_id` : 유저를 식별하는 아이디입니다.
      - `location` : 유저의 위치 정보입니다. 도시(city), 주(state), 나라(country)의 정보가 모두 포함되어 있습니다.
      - `age` : 각 유저의 나이입니다.
    - books.csv

isbn	book_ title	book_ author	year_of_ publication	publisher	language	category	summary	img_url	img_
0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	en	['Actresses']	In a small ...	http://....	....jp
0060973129	Decision in Normandy	Carlo D'Este	1991.0	HarperPerennial	en	['1940-1949']	Here, for the ...	http://....	....jp

    - `isbn` : 국제표준도서번호. 도서·자료 정리를 위해 만들어진 국제적인 기호로, 책을 구분하는 고유한 아이디입니다.
    - `book_title` : 책의 제목입니다.
    - `book_author` : 책을 집필한 작가입니다.
    - `year_of_publication` : 책을 발행한 연도입니다.
    - `publisher` : 책을 발행한 출판사입니다.
    - `img_url` : 책 표지 이미지에 접속할 수 있는 url입니다.
    - `language` : 출판 언어입니다.
    - `category` : 책에 대한 카테고리입니다.
    - `summary` : 책에 대한 요약 설명입니다.
    - `img_path` : 책 표지 이미지가 들어 있는 파일 경로입니다. data 디렉토리 경로하단에서 각 책에 해당하는 이미지를 확인할 수 있습니다.
    - train_ratings.csv

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

      - `user_id` : 유저를 식별하는 고유 아이디입니다.
      - `isbn` : 책을 식별하는 고유 아이디입니다.
      - `rating` : 해당 유저가 해당 책에 대해 매긴 평점으로 1점부터 10점을 부여합니다.
  - users
    - `location` 변수에 존재하는 특수문자들을 정규표현식을 활용하여 제거하였습니다.
    - `location` 변수는 순서대로 도시(city), 주(state), 국가(country) 로 이루어져 있습니다. 이를 ',' 기준으로 분리하여 `location_city`, `location_state`, `location_country` 변수를 생성하였습니다.

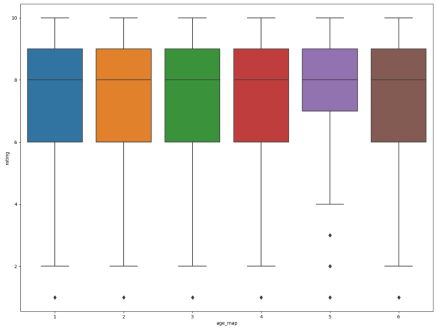
- `location_city` 의 값은 존재하지만 `location_country` 는 결측인 경우 해당 도시에서 가장 많이 관측되는 국가로 `location_country` 의 결측치를 채웠습니다.
  - ex)

user_id	location_city	location_country
116866	ottawa	NaN
115097	seattle	NaN
245827	albuquerque	NaN
226745	humble	NaN
38718	aloha	NaN

- `location_country` 는 존재하지만 `location_city` 는 결측인 경우 해당 국가 도시의 최빈값으로 `location_city` 의 결측치를 채웠습니다.
  - ex)

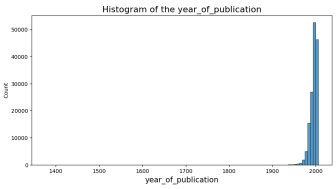
user_id	location_city	location_country
1008	NaN	usa
1065	NaN	united kingdom
1200	NaN	usa
1590	NaN	usa
1819	NaN	united kingdom

- `age` 변수를 10살 단위로 맵핑하여 `age_map` 변수를 생성하였습니다.
  - `age_map` 변수에서 50대 유저들은 평점(rating) 분포가 다른 유저들과 상이한 모습을 보였기 때문에 더미변수로 추출하여 `age_map_5` 변수를 생성하였습니다.
    - ex)



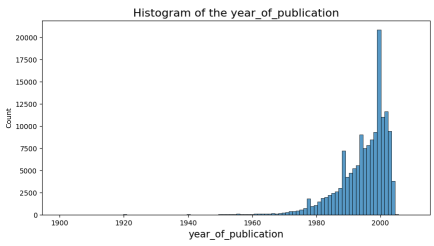
- books
  - `publisher` 변수의 유명 출판사들 중 같은 출판사이지만 표기 오류나 표기 방식에 의한 차이로 다른 출판사로 분류된 경우가 있었으므로, 해당 출판사를 단어로 포함하는 경우 모두 같은 출판사로 대체하였습니다.
  - `category` 변수에 불필요한 대괄호를 제거하였으며 조금 더 편리한 핸들링을 위해 소문자로 변환하였습니다.
  - `category` 변수에 표기 오류 등으로 인해 고유값이 4105 개로 불필요하게 많았습니다. 때문에 가장 널리 사용되는 43개의 상위 카테고리를 지정하여 해당 단어를 포함하면 상위 카테고리로 지정하였습니다. 또한 10개 이하의 샘플을 가진 `category` 를 others 라는 값으로 대체하여 총 4105 개의 고유값을 가진 `category` 변수를 통해 209 개의 고유값을 가진 `category_high` 변수를 생성하였습니다.
  - 출판년도가 1300 년대에 존재하는 등 `year_of_publication` 변수에 오류가 있음을 확인하였습니다. 직접 해당 책을 검색하여 출판 년도를 정상적으로 바꿔주었습니다.
    - `year_of_publication` 변수의 분포) 변환 전

	isbn	year_of_publication
104259	9643112136	1378
121860	964442011X	1376
129205	0781228956	1806



    - `year_of_publication` 변수의 분포) 변환 후

	isbn	year_of_publication
104259	9643112136	2010
121860	964442011X	1997
129205	0781228956	1806

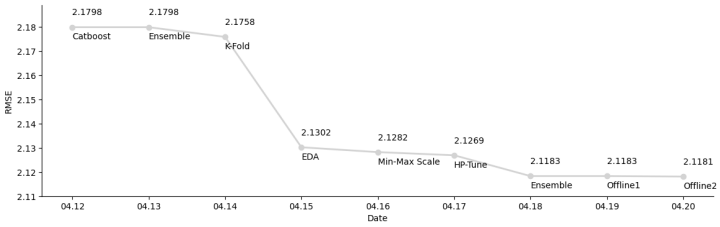


  - `year_of_publication` 변수의 값을 10년 단위로 맵핑한 `year_of_publication_map` 변수를 생성하였습니다.

- 변수 선택(feature selection)
  - 피처 엔지니어링 단계에서 다양한 변수들을 생성하였기 때문에, 순열 중요도(permutation importance) 기반 변수 별 중요도를 계산하여 전진 선택(forward-selection) 방식으로 모델에 유의미한 영향을 주는 변수들을 선택하였습니다.
- 모델 선정 및 분석
  - 저희는 CatBoost 모델을 선정하였고, `hyperopt`를 활용하여 하이퍼파라미터를 튜닝하였습니다. 모델 학습시에는 GPU를 이용하여 병렬 처리를 하였으며, 데이터셋이 매우 큰 경우에도 빠른 학습이 가능하였습니다.

Aa 모델 이름	# RMSE
FM	2.4605
CNN_FM	2.3452
DCN	2.3666
DeepCoNN	2.3695
FFM	2.4636
NCF	2.3705
WDN	2.4733
Catboost	2.1269
LGBM	2.2836

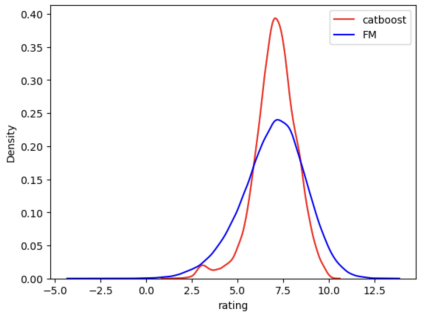
- 모델 평가 및 개선
  - 모델 평가 지표로는 RMSE(Root Mean Squared Error)를 사용하였습니다.
  - 초기 모델의 RMSE는 2.4733 였으며, 하이퍼파라미터 조정과 모델 학습 시간 증가 등의 개선 작업을 통해 최종적으로 RMSE를 2.1181 수준으로 줄일 수 있었습니다.
- 시연 결과
  - 모델 성능
    - CatBoost 모델을 사용하여 예측하여, 테스트 데이터셋에 대한 예측을 진행하였습니다.예측 결과 RMSE는 2.1181로, 대회의 6위 성적을 기록하였습니다.



  - 앙상블 성능

models	RMSE
CNN_FM Catboost DCN DeepCoNN FFM FM Lgbm	2.1181
Catboost FFM FM	2.1190
CNN_FM Catboost DCN DeepCoNN FFM FM Lgbm NCF	2.1200
CNN_FM Catboost DCN DeepCoNN Lgbm	2.1228

- 데이터 후처리
  - 앙상블 작업을 위해 모델의 예측값의 분포를 확인하던 중, 평점의 정상 범위(1~10)를 넘어서는 비정상적인 값들을 발견했습니다. 따라서 해당 값들을 정상 범위내로 수정해주는 알고리즘을 구축하고 예측값의 분포가 상이한 모델들을 앙상블하여 일반화 성능을 높혔으며 결과적으로 성능 향상을 이끌어냈습니다.
    - ex)



1-5 자체 평가 의견

- 잘한 점들
  - 모델링 및 실험
    - 첫째 날 대회의 Tesk에 적용 가능한 모든 베이스라인 모델의 성능을 실험 했고 그 결과, `GBDT 모델`이 Task에 유효하다는 것을 파악하고 빠르게 베이스라인에 적용할 수 있었다.

        - 가능한 모든 제출 기회(`일 10회`)를 소진하여 많은 결과를 얻었다.
        - 어떤 결정을 할 때 모델의 성능을 보고 결과론적으로 결정한 것이 아닌 `가설 설정`과 `실험 결과`에 입각해서 결정하였다.
            - 예를 들어, rating의 분포가 Imbalance한 것을 파악해서 naive K-Fold에서 `Stratified K-Fold` 적용해 개선하였다.
    - 팀 플레이
        - 각자 구상한 가설 구현 및 검증에 제한을 두지 않고 존중하여 제출 기회를 부여하였다.
        - 제출 직전 결과의 분포를 분석하는 과정에서 의견 교환과 피드백을 진행했다.
        - 팀원 모두가 하나의 역할에 몰두한 것이 아닌 전처리, EDA, feature engineering 등 다양한 역할을 번갈아 수행했다.
- 시도 했으나 잘 되지 않았던 것들
    - 피처 엔지니어링
        - rating 에 대하여 분포가 상이했던 범주형 변수들을 `원-핫 인코딩`하여 feature로 활용했다
            - ex) `age_map` 변수에서 `age_map_5` 더미 변수 추출
        - 평점(rating)에 대한 user_id별, category별 통계치(`mean`, `median`, `var`, `std`)를 feature로 활용했다
            - 결과적으로 약 40개 가량의 feature를 만들어 활용했지만 `over-fitting` 문제가 발생했고, 해결하지 못했다.
    - 전처리
        - Category별 books 데이터의 분포가 매우 상이했기 때문에 대표 n개의 카테고리로 `high-categorize`를 시도했지만, 기대한만큼의 성능 개선은 이끌어내지 못했다.
        - books 데이터에 Category와 Language 컬럼의 결측치를 `Logistic Regression` 등의 모델로 보정을 시도했지만, 마찬가지로 기대한 만큼의 성능 개선은 이끌어내지 못했다.
    - label 로그 변환
        - rating이 특정 값(6~9 점)에 치중한 분포를 보았기 때문에 skewness 감소를 위해 label에 `로그 변환`을 적용한 모델과 변환하지 않은 모델을 앙상블 했지만 오히려 약간의 성능 저하를 보였고, 이유를 결론 짓지 못했다.
    - AutoML (PyCaret)
        - PyCaret을 활용해 `AutoML` 모델을 활용하려 시도했지만 시간 내에 전처리 파이프라인에 도입이 어려웠다.
        - 작업 단위로 코드를 수정할 때 확장성을 고려하지 못한 것이 가장 큰 원인이라고 생각하며, 프로젝트에서 객체지향적인 코드 구조를 늘 유지하는 것이 중요하다는 것을 느꼈다.
- 아쉬웠던 점들
    - 대회 후반부의 집중력이 현저하게 떨어진 점
        - 리더보드가 열리기 전부터 대회 준비를 열심히 했지만, 후반부 체력과 뒷심이 달렸다.
        - 때문이 기존 논리 기반으로 의사결정을 하던 작업들을 뒤로 갈수록 확실하게 매듭짓지 못하고 제출하기 급급했다.
    - 프로젝트 타임라인 관리가 미흡했다.
        - 시작부터 타임라인 관리를 하지 않았고, 전처리 및 feature engineering 단계에서 하고 싶은 것들을 다 하다보니 실제로 우선적으로 해야하는 것, 나중에 해야할 것의 구분이 잘 안되어 전반적으로 프로젝트 진행이 비효율적이었다.
- 프로젝트를 통해 배운 점 또는 시사점
    - 프로젝트애서 각자 경험을 위해 실험 및 결과를 도출하더라도 팀 단위의 큰 그림, 전반적인 계획 수립이 꼭 필요할 것 같다.
        - 프로젝트 초기 단계부터 프로젝트의 관리를 담당할 PM 을 정하는 것이 좋을 것 같다.
    - 프로젝트에 대한 꾸준한 기록과 소통은 중요하고, 어려운 것이다,
        - 팀의 룰(1일 10제출) 영향으로 제출을 너무 많이 하다보니 오히려 기록과 소통이 잘 안되었다.
        - 특정 시간을 정해두고, 확실한 convention을 바탕으로 기록에 조금의 강제성을 부여하는 방향으로 룰을 수립하면 좋을 것 같았다.

Part2. 개인 프로젝트 Wrap up

곽희원_T5015

내 학습 목표

- 진심으로 참여한 대회는 오랜만이지만 **1등**을 하기!
- 최대한 조원의 의견을 듣고 이해하며 존중하기!
- 하고 싶은 시도를 다 하여 후회 없이 대회를 마무리 하기!

모델을 개선을 위한 나의 시도

- 새로운 시도
    - `K-Fold` 적용 → 평점을 예측하기 위해 회귀 모델을 사용하여 예측하지만 불균형 데이터 이므로 `Stratified K-Fold`를 구현
    - `hyperopt`를 이용한 하이퍼 파라미터 튜닝을 통하여 성능 향상
    - 다양한 모델들을 `앙상블`하여 성능 향상
- **Modeling**
    - GBDT 모델들(xgboost, lgbm, catboost), Tabnet을 구현하여 기존의 모델들보다 성능 향상
- 아쉬운 시도
    - `OPTUNA`를 이용한 하이퍼 파라미터 튜닝을 수행하고 싶었지만 모듈 에러가 발생하였으며 이를 해결 못해 아쉽습니다.
    - 과적합을 만드는 새로운 변수를 찾았지만 어떻게 수정하여 적용해보는 시도를 못 해봐서 아쉽습니다.
    - `Tabnet` 모델을 구현했지만 사용법을 익숙하지 않아서 다루지 못했습니다.

- 너무 많은 것을 시도하려고 하여 각각의 경우에 집중을 못 하여 결과를 못 얻은 경우도 있었습니다.

느낀 점과 고칠 점

- `스케줄링`의 중요성
  - 작업의 시간을 고려한 효율적인 스케줄링
  - 하루 단위 스케줄링보다 프로젝트 전체 기간의 스케줄링
- `커뮤니케이션`의 중요성
  - 나의 생각이 팀원들에게 잘 전달하는 스킬
  - 팀원들이 의견을 경청하며 듣는 스킬
- `준비`의 중요성
  - 아쉬운 시도들을 다음 프로젝트에서는 성공한 시도로 적용할 실력

임도현_T5170

학습 목표

- 최대한 다양한 방법들을 시도해보기
  - 그 동안 부스트캠프 교육과정을 통해 배운 내용들을 최대한으로 적용해보고 싶었기 때문에 대회를 진행하며 데이터 전처리, 피처 엔지니어링, 모델링 등 머신러닝 프로젝트에서 성능 향상을 위한 모든 단계들에서 최대한 다양한 방법들을 시도해보고자 했습니다. 실제로 부스트캠프 교육과정에서 배운 내용을 토대로 수 많은 가설 검증 단계를 거치며 모델 성능 향상을 이끌어 냈습니다.
- 프로젝트 팀원들과의 협업에 익숙해지기
  - 이전에 진행해왔던 ML 경진대회들에서는 주로 본인 혼자 모든 단계를 맡았습니다. 때문에 이번 대회를 통해 팀원들과 어떻게 협업을 해야 혼자 진행하는 것 보다 효율적으로 프로젝트를 진행할 수 있는지를 배우고자 했습니다. 실제로 Git 을 활용한 협업 및 노션을 이용하여 작업 중요도를 파악하여 팀원들 간의 일정 조율 등에 효율적인 방법들을 많이 배웠습니다.

모델 개선 시도

- 변수 중요도(feature importance) 기반 변수 선택 프로세스 구축 및 피처 엔지니어링(feature engineering)
  - 타겟 변수의 예측에 좋은 영향을 갖는 변수를 찾아내기 위해 범주형 변수들의 값들에 대한 타겟 변수 '평점' 의 분포를 boxplot 을 활용하여 관찰한 뒤, 유의미한 차이를 보이는 값에 대한 더미 변수를 생성하여 여러가지 변수를 생성하였습니다.
  - 베이스라인 CatBoost 모델을 학습 한 뒤, 순열 중요도(permutation importance) 를 계산하여 전진 선택법(forward selection) 방식의 변수 선택 프로세스를 구축했습니다. 앞선 피처 엔지니어링 단계에서 다양한 변수들을 생성하여 그 중 효과적인 변수들을 선택하여 차원을 축소하는 것이 중요하다고 생각했기 때문입니다.
- 데이터 전 처리 및 후 처리
  - 탐색적 데이터 분석(EDA) 과정에서 '카테고리' 변수에 출판년도가 입력되어 있는 등 각 변수들에 잘못된 값들이 기입되어 있는 경우를 발견했습니다. 해당 값들을 EDA 과정에서 발견한 데이터 간의 관계를 통해 전처리 하였습니다.
  - 나이(age) 변수의 결측치를 도시 별 사용자 나이의 중앙값으로 채워 92662 개의 결측치를 15102 개의 결측치로 줄였으며, 남은 결측치는 전체 사용자의 중앙값으로 대체하였습니다.
  - 고의적으로 또는 매우 부정적으로 평가를 하는, 즉 여러 평가를 한 유저 들 중 모든 책들에 1 점의 평점을 매긴 유저들을 파악하여 Test 데이터에서 해당 유저들의 평점을 1 점으로 변환하는 규칙 기반 모델을 설계했습니다. 이후 가장 성능이 좋게 나온 모델과 산술 평균 가중치를 이용하여 앙상블 하여 편향을 줄이는 방향을 선택하였습니다.
  - 실제 평점(1~10점) 의 범위를 벗어나는 모델의 예측값들이 존재했기 때문에 예측 이후 해당 값들을 정상적인 범위 내의 값으로 변환하였습니다.
- 위의 시도들이 모두 모델 성능향상을 이끌어냈습니다.

한계점 및 아쉬웠던 점

- 부실한 실험 관리
  - 하루 최대 제출을 목표로 하였기 때문에 많은 가설을 세우고 검증해내는 단계들을 하루에도 수 십번씩 진행하다보니 실험 관리 부분이 조금 어수선한 느낌을 받았습니다. 다음 프로젝트에서는 트렐로(Trello), Weight & Biases (wandb) 등 전용 툴을 사용하여 체계적인 실험 관리 및 일정 관리를 시도해보려 합니다.
- 단조적인 모델링
  - 범주형 변수들이 대부분을 차지하였고 CatBoost 모델이 다른 모델들에 비해 압도적으로 좋은 성능을 보였기 때문에 높은 성능에 집착하여 다른 모델들을 많이 시도해보지 못한점이 아쉬웠습니다. 때문에 다음 프로젝트에서는 성능에만 집착하지말고 본인의 경험을 위한 여러가지 딥러닝 기반 모델링을 시도해보고 싶습니다.

임지수_T5176

이번 대회의 목표

- 이론으로 학습했던 추천시스템의 모델들을 몸소 체감하기
  - 추천시스템을 공부하며 context 기반 모델, 딥러닝 기반 모델, 비정형 데이터를 활용한 multi-modal 모델, boosting 계열의 모델까지 여러 모델 및 기법들을 학습했습니다. 하지만 각 모델들이 어떠한 상황이나 Task에서 더 유효하게 활용될 수 있는지 감을 잡기는 어려웠는데, 이번 대회를 통해서 **모델들을 최소 한 번 이상 학습 시켜보는 방법으로 모델들을 체감**해보고자 했습니다.
  - catboost, lgbm 등 개인 학습에서 주요하게 다루지는 않았던 **부스팅 계열의 모델들이 이번 대회의 Task에서 굉장히 좋은 성능을 보임**을 확인했고 핵심적으로는 **딥러닝 기반의 모델들과 multi-modal 모델들은 이번 Task에서 단독으로 활용하게 되면 좋은 성능을 보이지 못한다는 것**을 확인했습니다.
- 대회 다경험 팀원들을 팔로우하며 노하우 배우기
  - 대회 경험이 전무했기 때문에 대회 경험이 많은 팀원들을 팔로우하며 어깨너머로 노하우를 배우고자 했습니다. 이 과정에서 **현실적으로 실험을 설계하는 방법**과 보다 **효율적으로 Feature Engineering**을 할 수 있도록 Feature Selection을 하는 것 등 많은 팁을 배울 수 있었습니다.

모델을 개선하기 위한 시도

- ML기반 분류 모델을 활용한 결측 데이터 보정

- books 데이터에 'category', 'language' 등 핵심적인 정보를 담고있다고 생각한 컬럼에 결측값이 많았습니다. 유의미하게 결측 값을 채우기 위해 결측이 없는 책의 제목(title)과 저자(author)를 Tf-Idf Vectorizer하고, logistic regression, random forest 등의 **ML 모델을 활용하여 결측 값을 다른 값 중 하나로 분류**했습니다. 약간의 성능 향상(**RMSE -0.001**)을 이끌어냈습니다.

- 다른 계열의 모델 ensemble
  - 부스팅 계열 모델의 성능이 이번 Task에서 단독으로 활용했을 때 가장 좋은 성능을 내고 있음을 확인했기 때문에 이를 베이스 모델로 두고, 다른 feature를 잡아낼 수 있는 다른 계열의 모델들을 ensemble하는 전략을 활용했습니다. 특히, feature engineering을 통해 feature의 수를 늘린 상황에서 각 feature들의 상호작용을 고려할 수 있는 **FM과 FFM 모델이 catboost와 weight ensemble(8:2)**했을 때 좋은 결과(**RMSE -0.01**)를 보였습니다. 그리고 **딥러닝 기반 모델과 multi modal 모델**을 함께 **weight ensemble(8:1:1)**했을 때에도 유의미한 성능 향상(**RMSE -0.02**)을 보였습니다.

한계점과 아쉬웠던 점

- 베이스라인 코드 구조에 의존한 실험 설계
  - 정말 잘 구성된 베이스라인이 주어졌기 때문에 필요한 부분만 덧대는 방식으로 편하게 실험할 수 있었지만, 전반적인 프로젝트의 구조를 이해하지 못한 상태에서의 **확장성을 고려하지 않은 코드 수정이 대회가 진행될 수록 추가적인 실험을 어렵게 만들었습니다.** 다음 프로젝트에서는 이처럼 베이스라인 코드 구조에 의존하지 않고 전체 구조를 이해한 다음 객체지향적으로 코드 수정을 할 수 있도록 깊게 생각하고 수정해야겠습니다.
- 개인적인 프로젝트의 타임라인을 정하지 않았던 점
  - 개인적으로 프로젝트의 타임라인을 정하지 않고 작업단위로 실험을 반복하다 보니 대회 후반부로 갈수록 시간의 제약을 받았고 효율적인 프로젝트 진행의 어려움을 느꼈습니다. 앞으로의 프로젝트에서는 대회 초반 타임라인을 확실하게 정해두고 효율적으로 프로젝트를 진행해야겠습니다.

조현석_T5205

- 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?
  - 우리 팀과 나의 학습목표는 모델의 정확도를 최대한 높이는 것이었습니다. 개인적으로는 모델 개발 및 평가를 위해 Python, pandas, numpy, scikit-learn, PyTorch와 같은 도구와 라이브러리를 활용하는 방법을 배우는 것이 목표였습니다.
- 나는 어떤 방식으로 모델을 개선했는가?
  - 모델 개발을 위해 사용한 지식과 기술에는 Gradient Boosting, Random Forest, XGBoost, LightGBM, Catboost와 이들의 앙상블 방법이 포함됩니다. 이러한 앙상블 방법을 사용하여 모델의 정확도를 개선하였습니다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?
  - 데이터 전처리와 피쳐 엔지니어링이 모델의 성능에 큰 영향을 미친다는 것을 깨달았습니다.
- 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
  - 저는 이번 대회에서 다양한 시도를 해보면서 성장하는 기회를 가졌습니다. 이 중에서도 가장 큰 변화는 모델의 튜닝 방식이었습니다. 이전까지는 주로 Grid Search와 Random Search를 사용하여 하이퍼파라미터를 튜닝했었지만, 이번 대회에서는 Bayesian Optimization을 처음으로 시도해보았습니다. Bayesian Optimization은 이전 실험 결과를 고려하여 하이퍼파라미터를 조정해나가는 방식으로, 기존의 방법에 비해 높은 성능을 보였습니다. 이를 통해 모델 성능을 높일 수 있는 다양한 하이퍼파라미터를 발견하고 조정할 수 있었습니다. Bayesian Optimization은 기존의 방법보다 수행시간이 오래 걸리는 단점이 있었습니다. 또한, 복잡한 모델의 경우 하이퍼파라미터 공간이 매우 크기 때문에 최적의 하이퍼파라미터를 찾는 것이 어려웠습니다. 이러한 한계점을 극복하려면 모델과 데이터에 맞는 최적의 하이퍼파라미터 튜닝 방법을 선택해야 할 것입니다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
  - 이번 대회에서 아쉬웠던 점은 모델의 성능 향상을 위해 앙상블의 weight를 자동으로 최적화하지 못했다는 점입니다. 앙상블은 여러 모델의 예측 결과를 결합하여 더 높은 성능을 얻을 수 있는 방법으로, 대부분의 우승자들이 사용하는 기법 중 하나입니다. 다음 대회에서는 이러한 앙상블의 weight를 자동으로 테스트하여 모델 성능을 높일 계획입니다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?
  - 이번 대회에서의 한계점과 교훈을 바탕으로 다음 프로젝트에서는 다양한 모델과 튜닝 방법, 앙상블 기법 등을 시도해보고, 최적의 모델을 찾는 것이 목표입니다. 또한, 데이터 전처리와 피쳐 엔지니어링에도 더 많은 시간을 투자하여 모델 성능을 높일 계획입니다. 이러한 시도를 통해 더 나은 모델을 만들어 실무에서 높은 성과를 이루어내고자 합니다.
  - 하이퍼파라미터 튜닝은 병렬 처리를 통해 빠르게 여러번 돌리려고 합니다.
    저번 대회에서의 한계와 교훈으로, 모델의 하이퍼파라미터를 튜닝하는 작업이 매우 시간이 오래 걸리는 것을 깨달았습니다. 이를 개선하기 위해 다음 대회에서는 병렬 처리를 활용하여 더욱 효율적으로 하이퍼파라미터를 탐색하고자 합니다. 더욱 다양한 하이퍼파라미터 조합을 시도하며, 이를 병렬 처리를 통해 빠르게 탐색하고자 합니다.
  - 또한, 다음 대회에서는 모델의 구조 자체를 수정하는 작업도 시도해볼 예정입니다. 이번 대회에서는 전처리 과정과 하이퍼파라미터 조합을 수정하는 방식으로 모델 성능을 개선했습니다. 그러나 다음 대회에서는 모델 자체를 수정하여 성능을 향상시키고자 합니다. 이를 위해 다양한 모델 아키텍처를 탐색하며, 이를 병렬 처리를 통해 빠르게 탐색하는 방식을 활용할 예정입니다.