Book Rating Prediction

Recsys-10조(물어봐조)
곽정무, 박준하, 박태지, 신경호, 이효준

1. 프로젝트 개요

1-1. 개요

해당 프로젝트는 책과 관련된 정보와 소비자의 정보, 그리고 소비자가 실제로 부여한 평점, 총 3가지의 데이터 셋을 활용하여 각 사용자가 주어진 책에 대해 얼마나 평점을 부여할지에 대해 예측하는 것이 목표이다.

2. 프로젝트 팀 구성 및 역할

공통	EDA 및 랩업 리포트 작성
곽정무	PM, Text, Context 모델 실험 및 피쳐 엔지니어링
박준하	LightGCN 모델링 및 하이퍼파라미터 튜닝
박태지	모델 실험 및 하이퍼파라미터 튜닝
신경호	앙상블 검증 프로세스 세팅 및 모델 실험
이효준	Confluence 템플릿 구축, 회의록 작성, 피처 엔지니어링, 랩업 리포트 관리

3. 프로젝트 수행 절차 및 결과

3-1. Timeline

협업은 Jira 와 Confluence 중심으로 진행했다. Confluence에 EDA와 가설을 공유하고 이를 중심으로 피처 엔지니어링과 모델링을 진행하였으며 매일 2회, Jira를 통해 서로의 진행상황과 계획을 알 수 있도록 하였다. 1주차에는 수업이 상당부분을 차지했고 EDA 기본적인 부분을 수행했으며 2주차에 심화 EDA 및 피처 엔지니어링, 하이퍼파라미터 튜닝, 그리고 앙상블을 진행한 후 대회를 마쳤다.

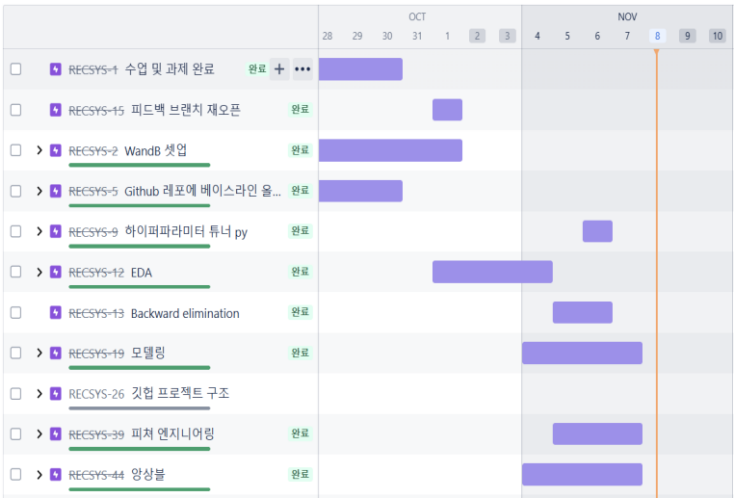

그림 1. Jira Timeline

3-2. EDA

이번 대회에서는 약 15만개의 책과 7만여명의 사용자데이터와 거기서 파생된 30만건의 평점데이터가 주어졌다. 이를 기반으로 각 데이터에 대한 EDA를 진행하였으며 진행과정에서 추천시스템 데이터의 가장 큰 특징인 sparsity를 확인할 수 있었고 또한 여러 피처에서 크게는 20%정도의 많은 결측치를 보유한 것 역시 확인할 수 있었다.

추가적으로 각 컬럼의 빈도별, 평점 빈도별, 평점 평균별 편향을 확인할 수 있었고 또한 각 컬럼의 값들이 매우 다양한 고유 값과 빈도를 가진다는 것 역시 확인할 수 있었다.

표 1. 컬럼별 고유 값

	unique
user_id	59803
isbn	129777
rating	10
age	91
country	208
state	1405
city	11189
book_title	117729
book_author	54715
year_of_publication	92
publisher	10408
img_url	129777
language	24
category	3872
summary	70061
img_path	129777

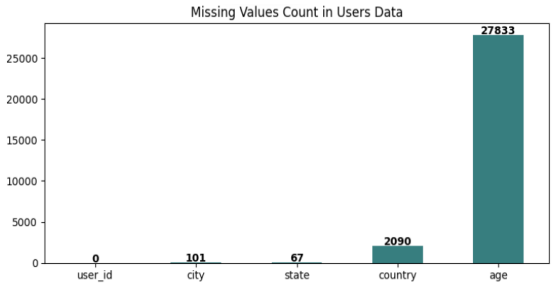

그림 2. 사용자 데이터 결측치

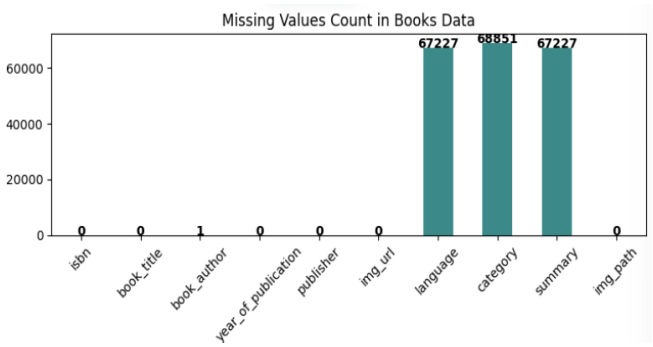

그림 3. 도서 데이터 결측치

3-3. 피처 엔지니어링

이번 대회 데이터셋의 경우 sparse하다는 특징이 있었고 새로운 유저와 아이템의 유무는 피처 엔지니어링이 중요할 것이라고 예상됐다. EDA를 통해 얻은 인사이트를 기반으로 FM 기반 모델을 중심으로 사용하고자 하였고 baseline이 라벨 인코딩된 변수를 input으로 사용했기에 어떻게 범주를 나눌 것인지가 주요 과제였다.

우선 중요도가 낮은 주와 도시에 대한 피처를 제거하고 피처 엔지니어링을 추가로 진행했다. 가장 성능 향상을 이끈 피처는 사용자 나이의 유무에 대한 이진 변수로 EDA 상 평점 차이가 가장 심한 요소에 속했다. 그 외에도 리뷰가 많은 책인지의 여부와 사용자의 리뷰 빈도를 활용한 피처를 추가했다. 이 과정에서 분할점을 선정하는 데에 어려움이 있었으나 성능 향상을 이끌어낼 수 있었다. 한 작가, 장르, 출판사의 경우 고유 값의 수가 많고 각 고유 값의 수가 적은 경우가 많았기에 각 피처별로 임계값을 정해 그 미만을 others로 할당했다.

3-4. 모델 실험 및 결과

먼저, 실험 환경으로는 Ubuntu 20.04 상의 V100 GPU를 사용하였으며, 성능 검증을 위해 train set과 validation set을 각각 0.8, 0.2만큼 두었다. 모델은 sparse한 데이터에 강점을 보이고 context-aware한 FFM(Field-aware Factorization Machines)을 사용했다. 실험적으로도 FFM 모델이 다른 baseline 모델들에 비해 validation set에서 우수한 성능을 보였다. 여기에 더 나아가 baseline FFM 모델에 feature engineering을 통해 얻은 피처들을 추가하고 hyperparameter tuning을 진행한 결과, 아래 그래프와 같은 결과를 얻었다. Baseline 모델과 feature가 추가된 모델은 3 epoch에서 최소 loss를 보였으며, 이후 epoch에서는 loss가 급격히 상승하였다. 반면 hyperparameter tuning이 완료된 모델에서는 epoch가 진행될수록 loss가 감소하고, 점차 수렴함을 알 수 있었다.

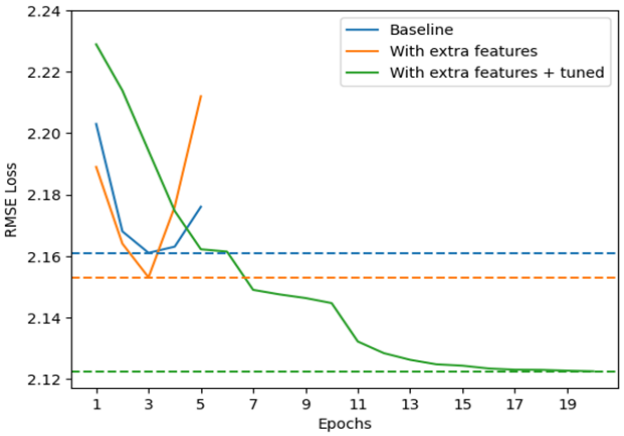

그림 4. 실험 결과

표 2. FFM 모델 최종 성능 비교

	Best RMSE loss	Loss 감소
Baseline FFM	2.161	-
FFM with extra features	2.153	baseline 대비 -0.37%
FFM with extra features and tuning	2.122	baseline 대비 -1.83%

결론적으로, feature 추가와 hyperparameter tuning을 통해 -1.83%의 loss 감소를 이뤄냈으며, test dataset에서도 2.1316이란 우수한 성능을 기록하였다.

4. 한계

이번 대회에서 추천 모델의 성능을 향상을 위해 다양한 파생변수를 생성했다. 사용자와 도서 간의 관계를 세분화하고, 개인화된 추천을 강화하는 특성을 추가함으로써, 보다 정교한 추천 모델을 구축하고자 했다. 피처의 수가 증가함에 따라 모델의 효과적인 학습을 위해 하이퍼파라미터 최적화가 필요했으나 시간의 제약으로 인해 하이퍼파라미터 튜닝을 충분히 진행하지 못했다.

특히, DeepFM 의 mlp_dims(각 층의 차원을 결정), FFM 의 embed_dim(sparse feature 임베딩 차원을 설정) 등 피처 간 상호작용을 효과적으로 학습시키는 하이퍼파라미터와 학습률과 같이 다수의 피처에 대해 정밀한 최적화를 진행하는 하이퍼파라미터들을 충분히 실험하여 최적화하지 못한 점이 모델 성능 개선에 한계로 작용하였다.