[Movie Rec] Wrap up Report

Movie Recommendation Team Wrap-Up Report (1조)

1. 개요

1.1. 프로젝트 주제

Movie Recommendation

- 사용자의 영화 시청 이력 데이터를 바탕으로, 사용자가 시청할 영화 예측
- 사용자별 시청 이력에서 마지막 n개, 그 이전 m개의 데이터 추출하여 테스트 데이터로 사용
- 전체 사용자에 대해 각각 10개의 영화 예측
- Recall@10으로 평가
- **input:** 사용자의 implicit 데이터, 아이템(movie)의 메타 데이터
- **output:** 사용자에게 추천하는 아이템을 user, item이 ','로 구분된 파일(csv) 로 제출합니다.

1.2. 개인 목표

이름	개인 목표
곽동호 T5013	강화학습 추천 적용하기
권수훈 T5017	머신러닝 관련된 엔지니어링 활용 + RecSys 강의에 나온 모델 적용
박상우 T5081	성적과 별개로 많은 실험 + 프로젝트 엔지니어링 마스터하기
이민호 T5140	기본 모델들 복습 및 구현 + 효율적인 코드 작성
이한정 T5166	Side information 활용해 보기 + 모델 실행 코드를 빠르게 완성해 보기
이준원 T5237	Top-Down 방식으로 많은 모델과 툴을 적용해 보고 설득력 있는 결과물 만들기

1.3. 프로젝트팀 구성 및 역할

곽동호	DQN 구현 시도
권수훈	EDA, implicit를 활용한 ALS 모델 구현, dockerizing을 활용한 환경 분리
박상우	RecBole 파이프라인 구현 및 모델 실험
이민호	ALS 모델 구현, Sequential 모델 구현 및 실험
이준원	EDA, EASE 모델, GitHub 및 Notion 관리
이한정	EDA, EASE 모델 구현

1.4. 협업

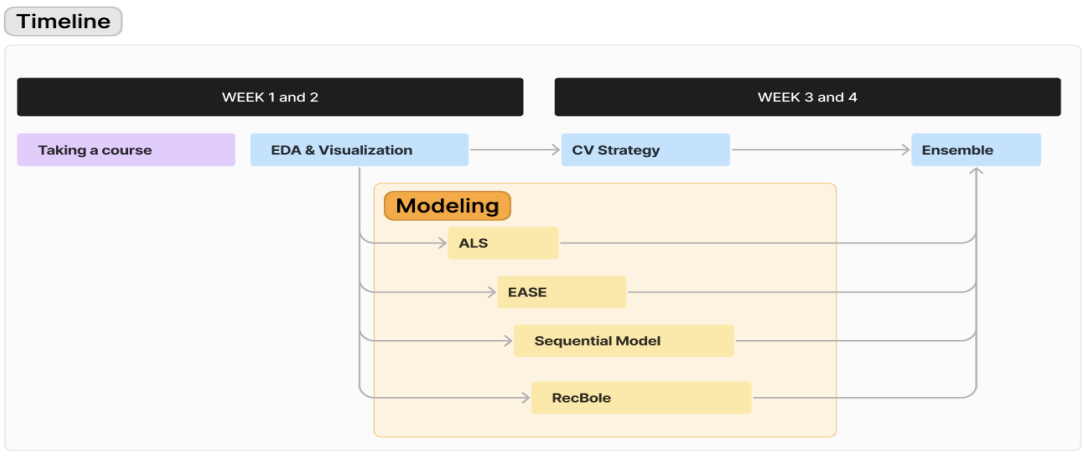
환경 및 툴

개발 환경 (AI Stage Server)	OS : Ubuntu 18.04.5 LTS | GPU : Tesla V100-SXM2-32GB
협업	GitHub, Notion, WandB
의사소통	Zoom, Slack, Offline

프로젝트 진행 기간

2023.05.31 - 2023.06.22

프로젝트 일정



2. 실험 과정

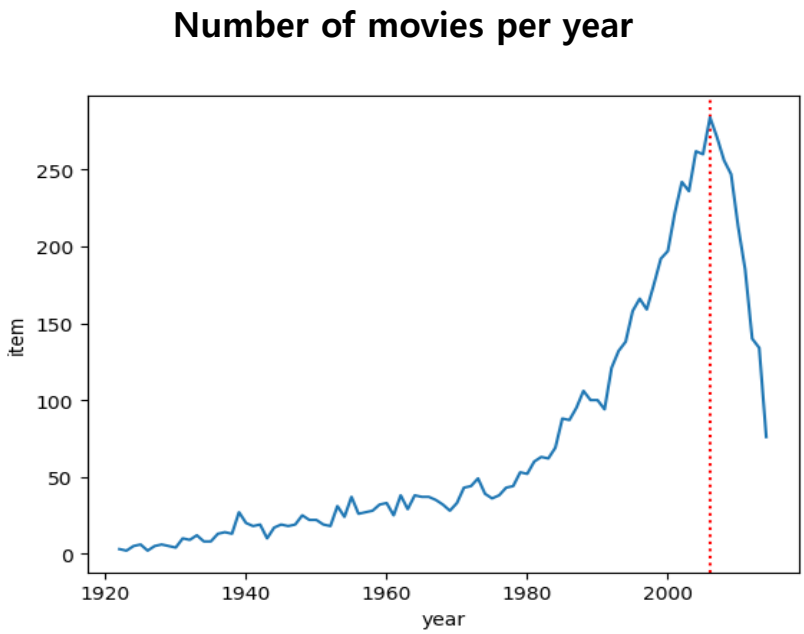
2.1. EDA & Feature Extraction

Train data

- Shape: (5154471, 3) → [user, item, time]
- 사용자 수: 31360
- 아이템 수: 6807
- 희소 비율: 0.975

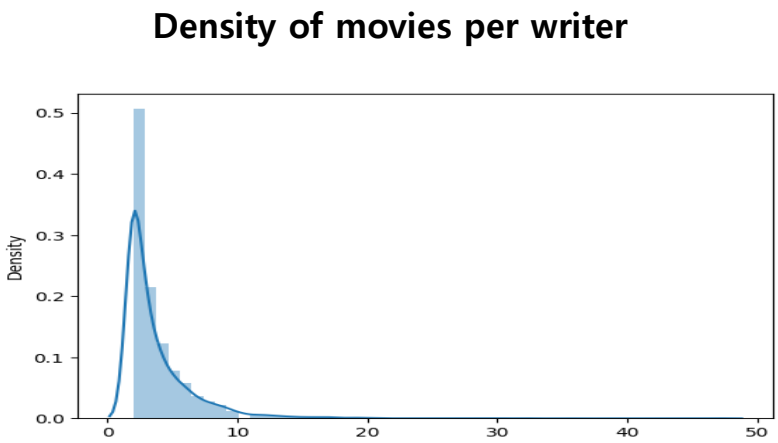
Side information

2.1.1. year



- Shape: (6799, 2) → [item, year]
- 아이템 수: 6799
- 학습 데이터에 연도를 채울 수 없는 아이템 수: **8 → title 뒷부분에 적혀있는 연도로 채울 수 있음**

2.2.2. writer

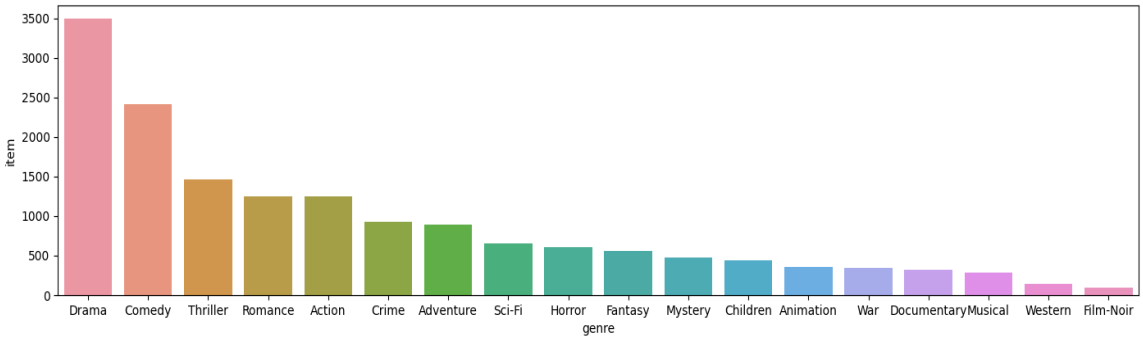


- Shape: (11306, 2) → [item, writer]
- 아이템 수: 5648
- 학습 데이터에 작가를 채울 수 없는 아이템 수: **1159**

2.2.3. title

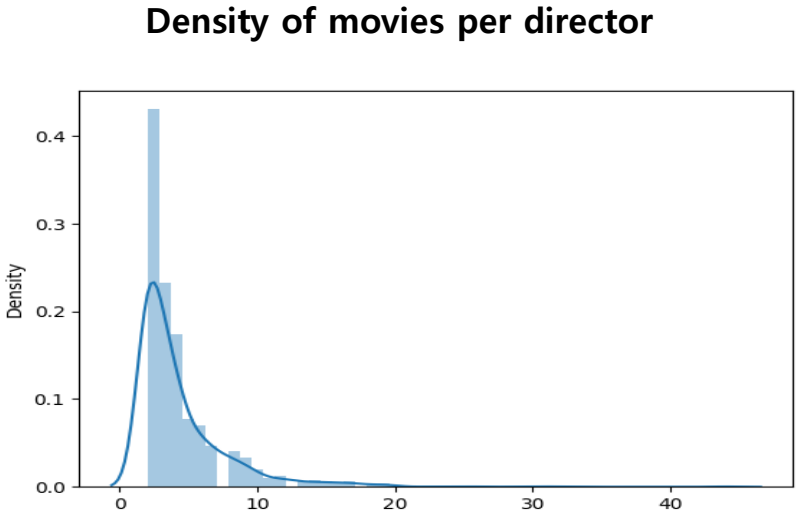
- Shape: (6807, 2) → [item, title]
- 아이템 수: 6807
- 학습 데이터에 제목을 채울 수 없는 아이템 수: 0
- War of the Worlds (2005)라는 같은 이름을 가진 영화가 존재

2.2.4. genre



- Shape: (15933, 2) → [item, genre]
- 아이템 수: 6807
- 학습 데이터에 장르를 채울 수 없는 아이템 수: 0

2.2.5. director



- Director data shape: (5905, 2) → [item, director]
- 아이템 수: 5503
- 학습 데이터에 감독을 채울 수 없는 아이템 수: **1304**

2.2. 모델 선정

- ALS (Alternating Least Squares)
  - 모델 구조가 단순해 학습 시간이 빠름
- EASE
  - Sparse한 데이터와 cold start problem에 강점을 가짐
  - 대회 데이터가 중간에 랜덤으로 샘플링되어 있어 Sparse함
- Sequential
  - 사용자 고유의 특성을 시계열적으로 파악하기 위해 사용
  - BERT4Rec을 기본으로 모델 구조 변형
- RecBole
  - RecBole의 추천 모델 대분류인 General, Context-Aware, Sequence 모델 사용
  - MovieRec 데이터셋에 적합하지 않은 Knowledge-Based Recommender는 제외
  - Inference 시 기존에 추천된 아이템이 다시 추천되는 에러가 있어, 코드를 수정

2.3. 팀별 모델 개발 과정

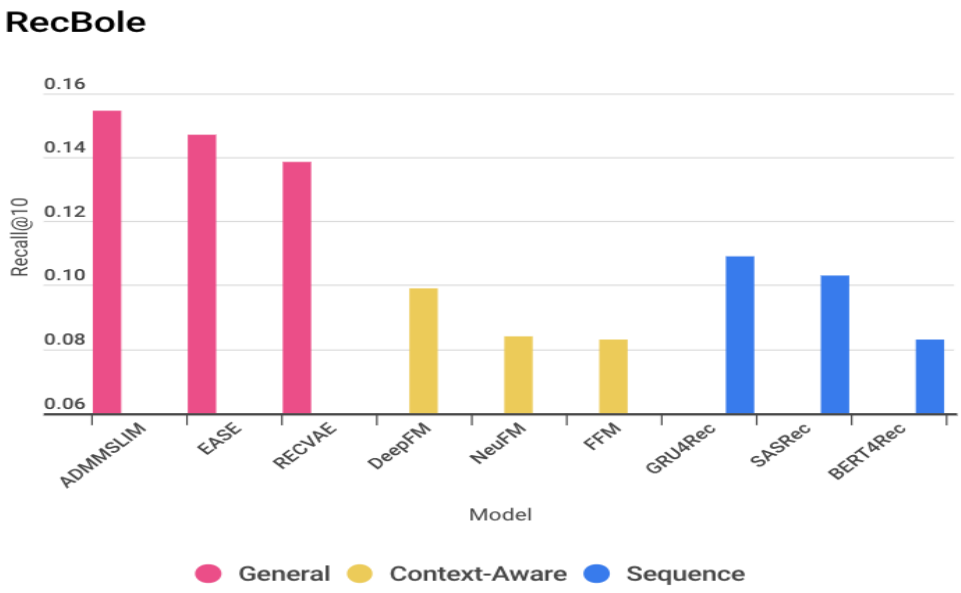
- ALS
    - Implicit 라이브러리 사용해 구현
- EASE
    - Numpy를 사용해 구현
- Sequential
    - PyTorch 사용해 구현
    - Attention 구조에서 look-ahead 마스크를 사용하지 않고, 시퀀스 중간중간 마스킹을 넣어 예측하는 BERT4Rec의 아이디어 사용
    - 사용자 시퀀스 샘플링, 아이템 마스킹, validation 부분에서 BERT4Rec과 다름
    - 사용자와 아이템의 side information을 사용해도 성능이 오르지 않았음
- RecBole
    - RecBole Model 대분류에 맞춰 총 3가지 파이프라인을 제작
    - 스크립트 한 줄로 RecBole에 필요한 Dataset, Yaml 파일 제작 및 모델 학습
    - RecBole 내장 함수인 Run_hyper를 통해 HyperOpt 방식의 하이퍼파라미터 최적화 사용
    - General, Context-Aware 모델은 8:1:1로 성능 검증하였으며, Sequence 모델은 Leave-One out CV를 추가로 사용

2.4. 앙상블 전략

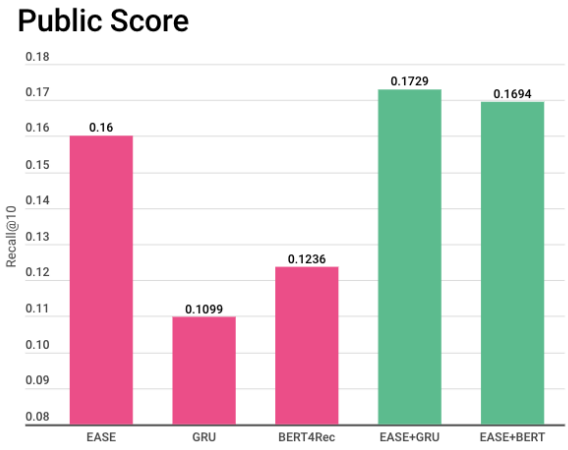
- Soft
    - 모델별로 임의의 N개의 정렬된 추천 아이템 추출
    - Ranking에 모델별 임의 가중치를 통해 합하여 새로운 Ranking Table 생성
    - 새로운 Ranking Table에서 상위 10개 아이템을 최종으로 제출
- Hard
    - 더 성능이 좋은 모델을 기준으로 상위 N개 아이템 선택
    - 다른 모델에서 겹치지 않게 상위 10-N 개의 아이템 선택
    - 두 모델에서 선택된 10개의 아이템 최종 제출
- Hybrid
    - 사용자를 상호작용한 횟수를 기준으로 나눠 다른 모델을 적용
    - EASE 모델은 한 명의 사용자에 대해 implicit data가 적을수록 성능이 높음

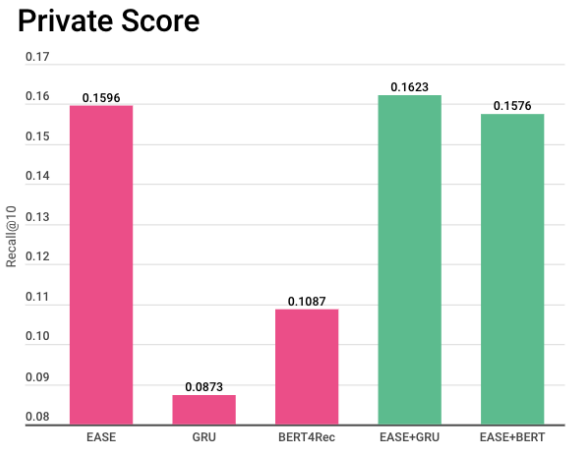
3. 프로젝트 수행 결과

3.1. 모델별 성능



3.2. 최종 모델의 단일 성능과 앙상블 결과





- EASE + GRU (7:3), EASE + BERT (4:6) -> public, private 모두 GRU를 앙상블 한 경우에 성능이 더 좋았다.
- 제출 해 본 모델 중 가장 성능이 좋았던 모델은 EASE + BERT (7:3)으로 Recall@10은 0.1659를 기록했다.

4. 결과 및 분석

- EASE 단일 모델의 성능은 public, private에서 모두 좋은 성능을 보여줌
- 대체로 Public Score에 비해 Private Score가 더 낮음
- BERT4Rec
    - Public Score 기준 0.06대에서 0.12대까지 상승
    - Private Score는 이보다 약간 낮았지만, 전반적으로 valid score와 일치했음
- GRU4Rec을 높은 비율로 한 모델이 Private Score에서 안 좋은 성능을 보여 순위 하락에 일조함
- Seed Ensemble 등 좀 더 Robust 한 모델을 만들어 Private Data에 대응해야 했을 것 같다고 생각함