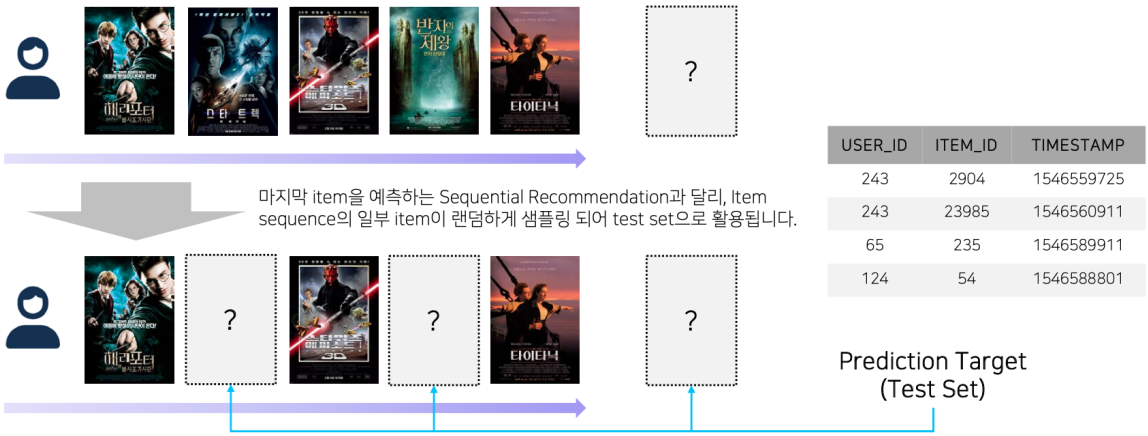
Wrap Up Report

1. 프로젝트 Wrap Up

1-1. 프로젝트 개요

- 프로젝트 주제: **Movie Recommendation**

  사용자의 영화 시청 이력 데이터를 바탕으로 사용자가 다음에 시청할 영화 및 좋아할 영화를 예측하는 문제



USER_ID	ITEM_ID	TIMESTAMP
243	2904	1546559725
243	23985	1546560911
65	235	1546589911
124	54	1546588801

- 평가지표: (normalized) Recall@10
- 활용 장비 및 재료
  - Tool: Notion, Slack
  - Library/Framework: RecBole, PyTorch
  - Optimization: HyperOpt, Ray, WandB

1-2. 프로젝트 팀 구성 및 역할

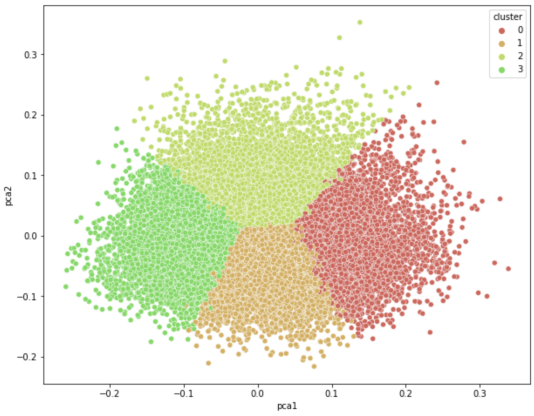
- RecSys 13조
- 구성원: 하태찬(PM), 구진범, 유영훈, 이환주, 임도연
- EDA, 데이터 전처리, 모델 설계 및 실험, 모델 튜닝

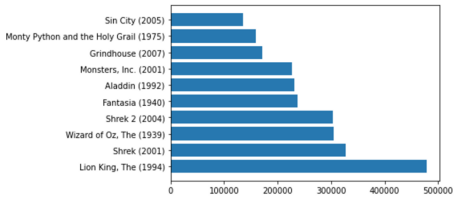
1-3. 프로젝트 수행 절차 및 방법

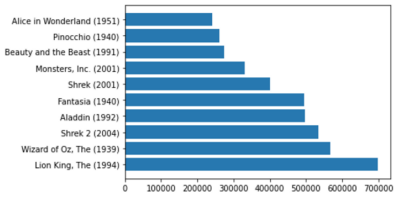
- EDA
  - 유저의 시청기록 분석
  - 시청한 장르에 따라 cluster로 나눠서 분석

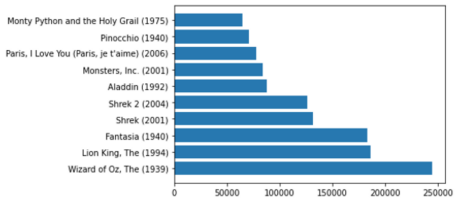
    각 user가 시청한 item에 대한 genre 정보를 multi-hot encoding으로 표현한 후 모든 item의 vector 값을 합산한 후 정규화를 진행했다. 그렇게 나온 각 user가 시청한 모든 item에 대한 genre 정보를 이용해 총 4

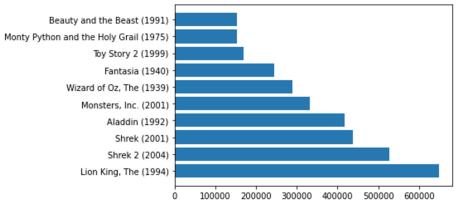
개의 그룹으로 분류했다.











- 이상치 제거

	item	title
349	34048	War of the Worlds (2005)
1926	64997	War of the Worlds (2005)

타이틀은 같은데 item id가 다른 데이터 하나가 존재했다.

직접 확인해보니 해당 연도에 동명의 이름을 가진 영화는 하나 밖에 존재하지 않았기 때문에 둘 중 하나가 잘못된 데이터라고 판단했다. 둘 중 삭제할 데이터를 선택하기 위해 interaction 데이터를 확인해보니 id 34048에 해당하는 데이터가 다른 데이터보다 16배 많아 전체 데이터에서 id 64997를 34048로 대체하였다.

성능 개선 실험

기준 모델: FM

Data split: [0.4, 0.1, 0.1]

Metric: Recall@10

이상치 제거 전	이상치 제거 후
0.082	0.0823

- 결측치 제거

```
years 전체 데이터: 6798 결측치: 8
directors 전체 데이터: 5503 결측치: 1303
writers 전체 데이터: 5647 결측치: 1159
```

year, director, writer 데이터는 각각 8, 1303, 1159개의 결측치가 존재했다.

year 결측치는 title 데이터에 표기되어 있는 연도를 가져와 채웠지만 director와 writer의 결측치는 채울 수가 없어서 그대로 두었다.

성능 개선 실험

결측치 제거 전	결측치 제거 후
0.082	0.081

- Feature 추가/제거 실험
  - 기준 모델: FM
  - Data split: [0.98, 0.01, 0.01]
  - Feature set: year, title, director, writer, genre
  - Epoch 20

추가한 Features	Recall@10	MRR@10	NDCG@10	Hit@10	Precision@10
안 넣은 것	0.1395	0.1269	0.0965	0.2876	0.0358
year	0.1377	0.1232	0.0938	0.2857	0.0348
title	0.1351	0.1202	0.092	0.2806	0.0343
director	0.1451	0.1246	0.0975	0.2889	0.0354
writer	0.1441	0.1230	0.0959	0.2867	0.0351
genre	0.1369	0.1208	0.0925	0.2836	0.0348
ratings	0.1464	0.0819	0.0862	0.194	0.0208
popularity	0.1457	0.0808	0.0855	0.1928	0.0207
year, title, director, writer, genre	0.1365	0.1233	0.093	0.2829	0.0346

- 모델
  - General model: EASE, ADMMSLIM, RecVAE, NCEPLRec

    - Context-aware model: xDeepFM, FM, FFM
    - Sequential model: S3Rec, SASRec
- 하이퍼 파라미터 튜닝
    - HyperOpt, Ray와 WandB 등을 활용
- Ensemble
    - Model 마다 서로다른 weight를 부여

```
graph TD
  ENSEMBLE[ENSEMBLE 0.1609]
  EASE     --> |2|ENSEMBLE
  NCEPLRec --> |1|ENSEMBLE
  RecVAE   --> |2|ENSEMBLE
  S3Rec    --> |1|ENSEMBLE
```

1-4. 프로젝트 수행 결과

- EASE(2) + NCEPLRec(1) + RecVAE(2) + S3Rec(1) → 0.1609 (최종 제출)
- EASE → 0.1601 (최종 제출)
- 최종결과: 11등

순위	팀 이름	팀 멤버	Recall@10	제출 횟수	최종 제출
11 (1 ▼)	RecSys_13조		0.1609	80	2d
1	RecSys_08조		0.1742	63	2d
2	RecSys_04조		0.1699	97	2d
3	RecSys_02조		0.1688	55	13d
4	RecSys_10조		0.1651	59	17h
5	RecSys_09조		0.1630	91	16h
6	RecSys_12조		0.1626	62	17h
7	RecSys_07조		0.1623	82	5d
8	RecSys_01조		0.1622	33	1d
9	RecSys_03조		0.1619	78	19h
10	RecSys_06조		0.1617	56	2d
11	RecSys_13조		0.1609	80	2d
12	RecSys_11조		0.1593	43	12d
13	RecSys_05조		0.1418	13	16h

1-5. 자체 평가 의견

- 잘한 점
  - 코어 타임 동안 줌을 켜놓고 모르는 것이 있으면 바로 물어보고 공유했다.
  - 슬랙을 적극적으로 사용해서 코드와 자료, 진행상황 등을 공유했다.
  - 클러스터를 만들고 각 클러스터 별로 모델을 적용하는 등 다양한 아이디어를 시도해봤다.
- 아쉬운 점
  - feature engineering 열심히 했지만 의미있는 feature를 찾아내지 못했다.
  - side-information을 잘 적용하지 못했다.
  - 단순 hard-voting 이외의 효과적인 ensemble 기법을 적용하지 못했다.
  - GitHub를 잘 활용하지 못했다. 코드를 짰으면 깃헙을 통해서 올려줬으면 좋겠다. 코드 공유나 디버깅이 더 쉬울 것 같다.
  - requirements.txt를 활용해서 라이브러리 버전도 통일했으면 좋겠다.
  - 초반에 서로 공유하거나 질문을 하지 못했다.
  - Docker를 활용했으면 서로의 문제를 더 빨리 해결할 수 있었을 것 같다.
- 프로젝트를 통해 배운 점
  - 줌과 슬랙을 잘 활용하면 협업을 잘 할 수 있다는 것을 배웠다.
  - 파이프라인을 빠르게 구축해야 한다.

2. 개인 회고

구진범

모델 개선

- 새로 적용 및 구현한 모델

  이번 task에서는 sequential한 예측과 static한 예측이 모두 필요했다. 베이스라인에서 sequential 모델인 S3Rec 모델이 제공되어 static한 모델에 집중하여 진행하였다.

  - Multi-VAE

    Multi-VAE 모델의 경우 multinomial distribution을 사용하여 분포합을 1로, 이미 시청한 아이템에는 더 큰 확률 질량을 부여하기 때문에 implicit data에 적합하다고 판단하여 구현하여 모델에 적용하였다.

    Recall@10 : 0.1126/0.1155 (public/private)

  - RecVAE

    Multi-VAE 모델이 고차원의 희소행렬을 input / output으로 사용하면서 다른 부분의 매개 변수를 손상 시킬 수 있는 불안정함을 개선하였기 때문에 성능 향상을 기대하고 구현하여 모델에 적용하였다.

    Recall@10 : 0.1480/0.1506 (public/private)

- Clustering

    - 모델에서 유저의 임베딩을 사용하여 유저를 그룹화 하고 각각의 그룹 별로 다르게 추천하는 방식을 사용하였다.
- Ensemble
    - 단순 hard-voting을 통하여 ensemble을 진행하는 것이 아닌 score을 정규화하여 합산하는 방식과 순위별 점수를 차등 분배하여 ensemble하는 방식을 사용하여

대회를 통해 얻은 것

- 이번 프로젝트의 베이스라인으로 SOTA 모델인 S3Rec이 제공되었는데 데이터와 task의 특성상 좋지 않은 성능을 보여주었다. 또한 유저와 아이템의 상호작용 만을 사용하는 EASE모델이 단일 모델 중 가장 높은 성능을 보여주었다. 이를 통해 다시 한번 데이터와 task에 따른 모델 선정이 추천시스템에서 큰 비중을 차지함을 느낄 수 있었다.
- 팀 내에서 협업하는 분위기를 성공적으로 정착한 것이 이번 프로젝트의 가장 큰 성과라고 생각한다.

마주한 한계와 아쉬웠던 점

- Clustering을 통하여 조금 더 세분화된 추천을 할 수 있을 것이라고 생각했는데 오히려 데이터의 수가 더 적어짐으로 인하여 성능이 떨어지는 모습을 확인할 수 있었다.
- 모델 실험을 계획적으로 진행하지 않아 파라미터 튜닝을 급하게 진행하는 등 많은 문제가 발생하였다. 다양한 모델을 적용하고 개선하는 것이 오히려 충분한 실험을 하지 못하게 되어 의미가 없어지게 되었다. 다음 프로젝트에서는 데이터의 특성에 따른 모델을 선정을 한 뒤 계획적인 실험을 진행해야 더 좋은 성과를 얻을 것이라 생각한다.
- 프로젝트 막바지가 되어서야 팀 내에서 협업하는 분위기가 형성된 것이 아쉬웠다.

유영훈

내가 시도한 것

1. 가설에 기반한 Rule-based 접근
    - Movie Lens dataset은 영화리뷰 사이트의 리뷰를 바탕으로 한 데이터이다.
    - 따라서 한 영화에 대한 기록이 2개 이상있을 수 없다. → 중복된 아이템 추천 X
    - 사이트에 활동기간에 따라 유행하는 영화가 있었을 것이다. → 활동기간에 따른 인기아이템 추천
    - Public에서는 감소했지만 Private까지 확인한 결과 옳은 접근이었다.
        - Public: 0.0673 → 0.0668
        - Private: 0.0671 → 0.0684
2. RecBole을 활용한 Sequential model
    - 미래의 아이템을 예측하는 부분이 있기때문에 Sequential model을 사용하였다.
    - 여러 모델들을 사용해보았지만 CV에서는 좋게나왔지만 LB에서는 좋은 성적이 나오지 않았다.
        - 결과: GRU4Rec 기준 CV: 0.1518 → Public: 0.0672 → Private: 0.0539
    - 이후 앙상블에서 활용하기위해 WandB Sweep을 통한 최적화를 시도했다.

- 내가 생각한 LB에서 안좋게 나온 이유
    - 중간에 있는 아이템도 예측해야 하기 때문에 Sequential model이 task에 맞지 않았다.
    - 대부분의 시청기록들이 신작보다 오래된 영화에 집중되어있어서 아이템의 출시시기도 크게 중요하지 않았다.

3. 앙상블과 결과 후처리
    - EASE 모델과 S3Rec 에서 5개씩 추천을 받아서 합쳤다.
        - 결과: 0.1042 (S3Rec) + 0.1562 (EASE) → 0.1518
    - 팀원의 성능이 잘나온 EASE 모델에서 20개를 뽑은 후 인기가 많은 10개의 아이템을 제출했다.
        - 결과 Public 기준: 0.1600 → 0.1037

소감 / 후기 / 느낀 점

- Sweep의 범위를 너무 넓게 줘서 제한시간내에 효율적인 파라미터를 찾지못했다.
- 빠르게 파이프라인을 구현하는게 중요한 것 같다.
- 이번 프로젝트에는 팀원들과 Git과 GitHub의 다양한 기능을 활용해볼 수 있어서 좋았다.
- 다음에는 Docker를 활용해보고 싶다.
- 유저와 아이템의 상호작용 데이터만을 활용한 모델들이 대체적으로 성능이 잘나왔다. 최소한 추천분야에서는 간단한 모델들이 더 성능도 내기 쉬운 것 같다.

이환주

내가 시도한 것

- user별, item별 특성을 확인하여 feature engineering 해볼 만한 것들에 대해 고민하였다.
- user들이 적게 시청한 영화를 제거하기 위해 ratings(시청 횟수가 55 이하인 것은 0, 나머지는 1)라는 feature를 만들어 넣어주었다.
- title을 SBERT로 embedding한 후에 xDeepFM으로 학습하였다.

마주한 한계와 도전 숙제

- 데이터파트에서 가설을 세우는 것이 어려웠다. 그리고 가설에 대한 검증을 통해 의미있는 feature를 뽑아내는 것도 어려웠던 것 같다.

  → 이번 대회를 통해 어떻게 해야할지 감이 좀 생긴 것 같다. 시간이 부족해서 다양한 시도를 해보지 못한 것이 아쉽고, 최종 프로젝트 때에는 더 많은 가설을 세우고 검증함으로써 의미 있는 데이터를 뽑아내고 싶다.

- RecBole을 처음 사용해봤는데 전체적인 이해 없이 바로 코드를 수정했더니 처음에 model이나 feature를 변경할 때마다 에러도 많이 나고 많이 헤맸던 것 같다.

  → 다음부터는 전체적인 부분을 먼저 이해하고 수정할 부분을 파악한 후에 코드를 수정해야겠다.

아쉬웠던 점

- 대회 초반에 번아웃과 감기로 인한 컨디션 난조로 시간을 많이 흘려 보냈는데 대회 후반엔 해보고 싶은게 많아서 일주일만 더 있었으면 할 정도로 시간이 부족한게 아쉬웠다.
- GitHub이나 Docker를 잘 활용하지 못한 것이 아쉽다.

다음에 도전할 것

- GitHub과 Docker를 잘 활용해보고 싶다.
- 다음 대회에서는 파이프라인을 빠르게 구축함으로써 대회를 빠르게 진행하여 더욱 다양한 시도를 해보고 싶다.

임도연

내가 시도한 기술적인 도전

- user based collaborative filtering 구현
  - user와 item 시청 interaction 데이터만을 이용해서 가장 유사한 user를 뽑아 시청한 item 기준으로 추천
- recbole 패키지의 context-aware recommendation model을 맡아 진행
  - 기존 interaction 데이터뿐만 아니라 다른 item에 대한 genre, director, writer 정보를 활용해 feature를 추가했을 때 성능 개선을 확인, 가장 잘 나온 xDeepFM 모델의 튜닝을 진행
- genre에 따라 user를 clustering함
  - 5개의 cluster가 가장 잘 나누어진다고 판단하여 clustering 후 각 cluster 집단에 다른 feature 값을 추가한 xDeepFM 모델 적용

마주한 한계와 도전 숙제

- 데이터를 조금 더 자세히 보아 좋은 피처를 만들어보고 싶음 -> 어떤 피처를 만들어보면 좋을지 생각이 잘 떠오르지 않음

느낀점

- 마지막 대회인만큼 팀원들과 다양한 방법들을 시도해봄
- 서로 깃허브도 협업을 해가며 계속 진행상황을 공유하면서 활발히 소통을 하는 협업 방법에 대해서도 배울 수 있었음

하태찬

내가 시도한 기술적인 도전

- 테스트 셋이 데이터 셋에서 중간중간 랜덤 샘플링해서 구성되었기 때문에 베이스라인으로 주어진 S3Rec 모델과 같은 Sequential 모델을 의미가 없을 것이라고 판단했다. 그래서 유저의 전반적인 영화 소비 성향을 어떤 식으로든 잘 표현하는 방법을 찾고자 했고, Item2Vec을 사용해서 각 유저가 소비한 영화 각각을 임베딩 한 후 이를 평균내어 각 유저마다 벡터 하나씩을 구해놓고, 이미 소비하지 않은 영화와 유사도를 구해 top 10을 추천하는 방식을 진행했다. 결과적으로 Recall@10 기준으로 0.09 정도 나왔다.

학습과정에서의 교훈

- 깃헙이나 슬랙을 적극적으로 활용해서 협업을 잘 해야 팀 전체적으로 성과가 잘 나올 수 있다고 깨달았다.

마주한 한계와 도전 숙제

- Item2Vec을 조금 더 발전시키지 못한 것이 아쉽다. 임베딩 잘 되었는지 확인하는 방법과 전체 평균 말고 window를 잡아서 그만큼만 평균하는 것, 모델이 못하는 게 뭔지 등을 확인해봐야겠다.

좋았던 점

- 늦게나마 팀원들과 매일 진행 상황과 성과에 대해서 공유함으로써 다른 사람의 작업을 반복하지 않을 수 있었다.