Wrap-Up Report, RecSys09

좋아요구독추천AI Team

Team 소개

👍 **좋**아요**댓**글**구**독**알**림~

> BoostCamp AI Tech 4기 추천 시스템 9조, 좋아요… 팀!
>
> **T4056_김찬호**
> **T4096_배성수**
> **T4171_이지훈**
> **T4196_정소빈**
> **T4210_조원삼**

목차

우리팀의 목표

“협업을 통해 대회를 진행하자 ! ”

- #협업
- #성장
- #소통
- #문제 해결
- #추천 마스터

1-1. 프로젝트 개요

▼ 프로젝트 주제

Movie Recommendation

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

▼ 데이터 개요

- **train_ratings.csv (**5,154,471 행)

  주 학습 데이터, user*id,* *item*id, timestamp(초)로 구성

- **titles.tsv** (6,807행)

  영화 제목

- **years.tsv** (6,799행)

  영화 개봉년도

- **directors.tsv (**5,905 행)

  영화별 감독

- **genres.tsv (**15,934 행)

  영화 장르 (한 영화에 여러 장르가 포함될 수 있음)

- **writers.tsv (**11,307행)

  영화 작가

▼ 프로젝트 구조도

▼ 활용 장비 및 재료

개발환경 (AI Stage Server)

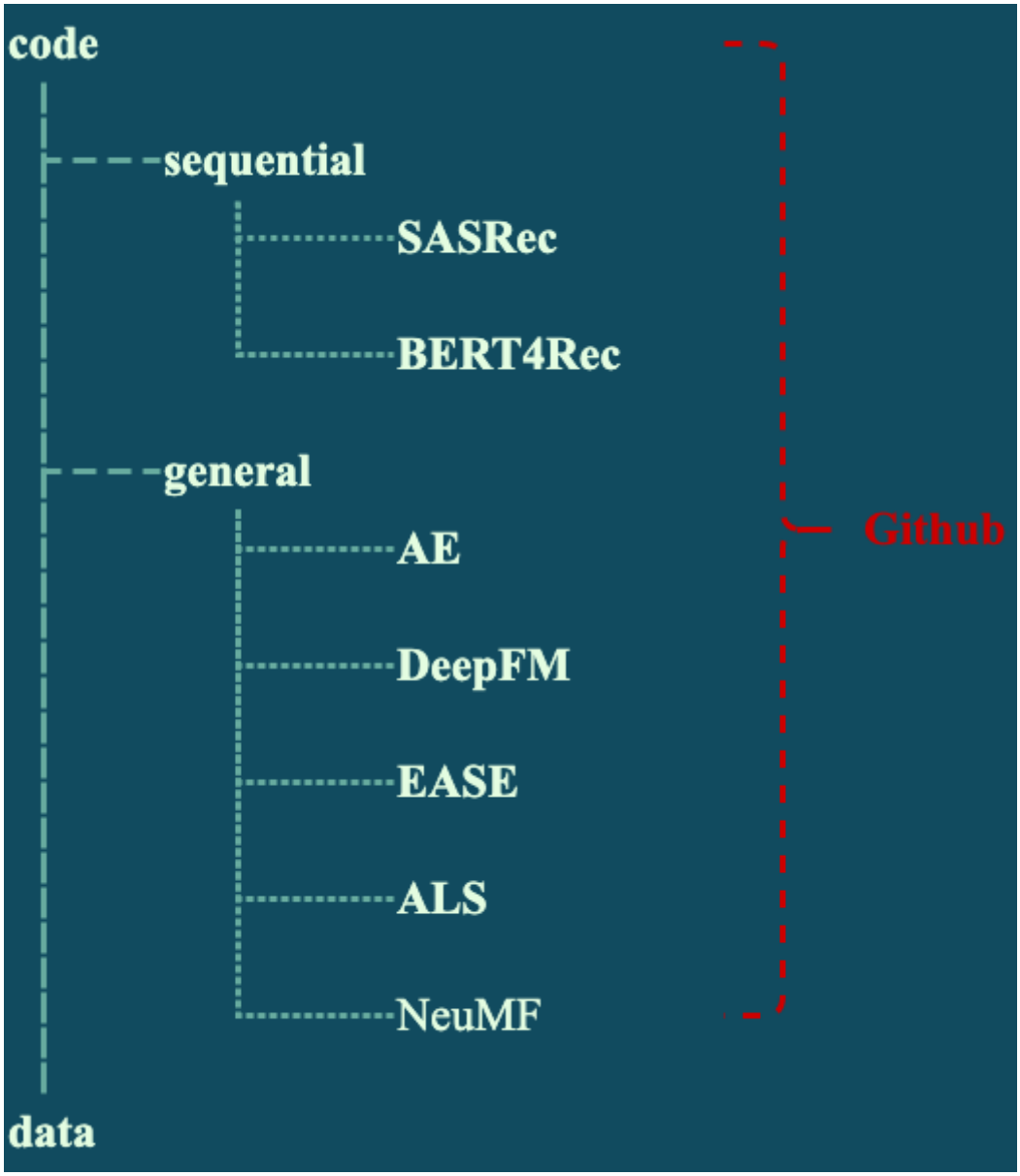
- OS: Ubuntu 18.04.5 LTS
- GPU: Tesla V100-SXM2-32GB

협업

- Github
- Slack
- Zoom

Tools

- Python
- Pytorch
- Weights & Biases + Sweep
- Optuna



1-2. 프로젝트 팀 구성 및 역할

- **T4056_김찬호:**
  - 노션 프로젝트 페이지 정리
  - 베이스라인 인퍼런스 구현 및 코드 정리
  - EASE 모델 구현
  - Voting 앙상블, Top-k 앙상블
- **T4096_배성수:**
  - 모델 탐색 방향성 설정

    - 데이터 전처리, Autoencoder 계열, ALS 베이스라인 개발
    - Mult-DAE, Mult-VAE, Bert4Rec, ALS 모델 실험
    - Voting 앙상블
- **T4171_이지훈:**
    - 모델 탐색
    - Multi-DAE, Multi-VAE, NMF 모델 실험
    - Nue-MF Pytorch Project 개발
- **T4196_정소빈:**
    - 모델 탐색
    - Sequential 계열 모델 실험
- **T4210_조원삼:**
    - EDA
    - DeepFM
    - Bert4rec

1-3. 프로젝트 수행 절차 및 방법

▼ EDA

Rating

시청기록에 대한 데이터를 EDA함으로 전반적인 데이터의 구조를 살펴보고자 함. User의 시청기록과 Item의 시청기록 분포는 롱테일의 모습을 띄고 있기에 극단에 존재하는 경우의 Bias가 크게 생길 것을 확인 가능.

시계열적 요소 역시 존재해, 여름과 겨울에 시청기록이 늘어났으며 전반적인 시청기록은 우하향하는 모습을 보임. 이를 통해 Item의 개봉일자와 시청기록에 대한 상관관계를 파악할 수 있었음.

User별로 Item 시청간격이 비정상적으로 짧다는 것을 확인해, 직전 시청 Item과 간격이 일정 시간 이상인 경우만 전처리해 다시 확인. 기존 데이터셋의 약 10%로 데이터 감소.
이때 전반적인 분포는 유사하나, 시청기록의 분포가 덜 Sparse해진 경향을 보임으로, 해당 데이터를 마치 이상치를 제거한 데이터처럼 사용 가능하다 판단.

Genre

한 Item에 대해 여러 Genre row가 있는 상황이라 하나의 row로 만들어주는 전처리 수행. 가장 많은 Item이 두 개의 Genre를 갖고 있고, 한개 혹은 세개의 Genre를 갖는 Item은 그 다음으로 많이 나타남.. 전반적으로 모든 Item에서 드라마, 코미디, 로맨스 Genre가 제일 많이 나타났고, 다른 장르와 섞이기 힘든 다큐멘터리 혹은 호러와 같은 Genre는 단일 Genre Item에서 많이 나타남.

그러나 시청간격을 고려한 10%데이터에서는 로맨스의 비중이 줄어들고, 액션이나 어드벤쳐, 범죄의 Genre 비중이 늘어남을 알 수 있음.

Title, Writer, Director

큰 EDA 없이 진행. 다만 Writer나 Director는 하나의 Item에 평균적으로 각각 3.8명, 2명씩 배치되어있는 경우. 실제 User가 Item을 선택하는데에 이 요소는 큰 영향이 없을것이라는 정성적인 판단하에 배제.

▼ 모델 탐색

모델 탐색 방향설정

본 대회의 test dataset은 사용자의 아이템 sequence에서 일부 아이템이 누락된 실제와 비슷한 상황을 가정하여, 중간에 누락된 아이템과 다음에 시청할 아이템 각각 5개로 유저별로 총 10개씩 구성된다. 따라서 누락된 아이템 예측에 강한 모델과 다음에 시청할 아이템 예측에 강한 모델을 탐색해 앙상블하는 방향으로 탐색을 진행했다.

Valid Dataset 구성

Valid dataset은 누락된 아이템 예측인지, 다음에 시청할 아이템 예측인지에 따라 다르게 구성된다. 누락된 아이템 예측의 경우 랜덤하게 유저들을 골라 유저 sequence 중 일부분을 랜덤하게 누락시켜 입력으로 사용하고, 누락된 데이터를 라벨로 사용한다. 다음에 시청할 아이템 예측은 각 유저 별 마지막 아이템을 valid dataset으로 사용한다.

Context-Aware Model

- **DeepFM**
    - **모델을 사용한 이유**

Item Feature가 많은 데이터셋인 만큼 Context 기반의 모델인 Deepfm이 효과적일 것이라 판단. 또한 단순히 Linear한 Fm보다 DNN layer가 추가되었기에 보다 더 효과적인 학습이 가능할 것이라 판단.

- **가설1**: Negative Sampling을 단순 Random으로 잡는 것이 아닌, User별로
  시청기록이 제일 적은 장르의 Item을 Sampling. 이를 통해, User가 더욱 선호하는 Item, 특히 Sequence 중간의 Item 예측에 효과적일 것이라 판단.
  - 결과: 기존의 Random Sampling을 사용하는 것과 큰 차이가 없음.
  - 이유: Random Sampling보다 특정 Gerne나 Item에 Bias가 크게 잡힐 가능성이 높기때문에 오히려 학습에 부정적인 영향
- **가설2**: 시청간격을 또한 FM Layer의 Feature로 사용. 이를 통해 시청간격이 과하게 낮은 경우의 기록에 대한 Bias를 줄이고자 함.
  - 결과: 기존의 결과값보다는 개선되었으나 절대적인 성능의 개선은 없음.
  - 이유: 실제 시청에 대해 유의미하지 않다고 생각했던 시청간격 이상치 데이터들이 실제로, 1초라도 시청을 했다는 것 자체가 유의했다고 판단.

MF Model

- **Alternating Least Squares**
  - **모델을 사용한 이유**

    EASE와 한개의 Linear layer 모델이 좋은 성능을 보이는 것을 확인하고 딥러닝 모델보다 단순한 MF 계열의 협업필터링 모델을 사용해 보기로 함. 또한 ALS는 Sparse한 데이터에 Robust한 모습을 보여주기 때문에 누락된 데이터 예측에 좋은 성능을 기대할 수 있음.
  - **가설:** 검증용 데이터를 제외하기 보다는 모든 데이터를 사용해 학습을 진행.
    - **결과:** Factors를 60으로 설정했을 때 Recall@10: 0.1395로 괜찮은 성능을 얻을 수 있었음.
    - 이유: Matrix Factorization은 비지도 학습이므로 validation이 불필요하다 판단. Factors를 valid data set

을 만들어 튜닝한 뒤 해당 factors 값을 모든 데이터를 사용해 학습할 때 사용함.

- **Nue-MF**
  - **모델을 사용한 이유**

    유저와 아이템 간의 상호작용을 이용하는 Multi-AE 와 ALS의 성능이 좋아서 유사한 모델 작동 원리를 이용하는 MF계열 모델 실험을 진행함. Negative Sampling과 다른 Feature를 활용하여 Input을 구성하고, 모델의 결과값을 점수가 높은 순서로 랭킹매김
  - **가설**: Genre Feature를 MLP Layer에 Input하고, 시청한 데이터 제외하고 Inference 진행
    - **결과**: recall_10k의 결과는 약 0.31, NDCG 약 0.4의 준수한 성능
    - **이유**: 늦게 가설을 세우고 진행을 했었기에 다양한 실험을 진행하지 못했음. 논문에서는 8개의 Feature를 사용하면 좋다고 하였지만, 총 3개의 Feature만 활용하였고, 모델 고도화 역시 진행하지 못한 부분이 아쉬움.

Autoencoder Model

- **Mult-DAE**
  - **모델을 사용한 이유**

    대회의 중요한 **Task** 중 하나인 중간에 빈 값들을 예측하는 것이 **모델의 작동 원리와 유사**하다고 판단했기 때문에 사용함. **특징**은 Dropout(Noise)을 큰 값인 **0.5**로 설정함으로써 실제로 비어있는 값과 함께 모델을 실행하고, Deterministic hidden neuron통해서 원본 데이터를 다시 복원하는 방식
  - **가설1**: Mult-DAE에 **Side-Information**을 추가하면 더 좋은 성능을 내는 논문을 참고하여 **Genre Feature**를 추가함.
    - **결과:** 성능적인 부분에서 피쳐를 추가하기 전보다 성능이 **약 5% 하락**
    - **이유1:** 피쳐를 추가한 뒤, 추가적인 하이퍼 파라미터를 수정하지 않았고, 그에 맞게 모델을 최적화 시켜주지 않았기 때문이라 판단
    - **이유2**: 장르의 범주가 18개여서 임베딩화하여 학습을 시키고 싶었으나, 모델의 특성상 임베딩을 진행한다면 차원이 맞지 않았기 때문에

모델에 큰 영향을 주진 못했을 것이라 판단

- **가설2**: Mult-DAE의 성능이 좋은 이유는 누락된 데이터를 예측하는 데 좋을 뿐만이 아니라 다음 아이템 예측에도 좋기 때문.
  - **결과 :** 다음 아이템 10개를 valid dataset을 사용해 성능을 측정해본 결과 다음 아이템 예측에도 recall@10에서 약 0.06의 성능을 보임.
  - **이유1 :** Dropout이 랜덤하게 작용하기 때문에 학습 중 확률적으로 다음 아이템을 예측하는 구성으로 학습했기 때문이라 판단.
  - **이유2 :** 시리즈 영화를 제외하면 볼 영화를 선택하는 데 있어 정해진 순서가 없기 때문이라 판단.
  - **결론 :** Top-K가 커질 수록 recall값이 커지는 것으로 보아 모델 출력에서 누락된 아이템은 높은 순위에 분포해 있고, 다음 아이템은 상대적으로 낮은 순위에 분포해 있음. 따라서 앙상블을 할 때 Top-K의 K 값을 10보다 큰 순위의 출력을 사용해 적용해 보는 것을 고려할 수 있음.
- **가설3:** 좋은 성능을 내는 EASE 모델을 참고해, 한개의 Linear Layer를 가지고 Mean Squared Error를 사용해 학습했을 때 Mult-DAE보다 좋은 성능을 보여줬음. 따라서 MSELoss를 사용하는 MSE-DAE 실험은 진행.
  - **결과 :** 기존의 Mult-DAE의 성능에서 Recall@10: 0.1462 → 0.1534 로 유의미한 성능 향상을 확인.
  - **이유 :** 본 대회의 데이터는 Multinomial Likelihood보다 Gaussian Likelihood가 학습에 알맞다는 것을 알 수 있음.

- **Mult-VAE**
  - **모델을 사용한 이유**

    Multi-dae와 유사하게 대회의 중요한 **Task** 중 하나인 중간에 빈 값들을 예측하는 것이 **모델의 작동 원리와 유사**하다고 판단했기 때문에 사용함. **특징**은 bottleneck(z)이 latent한 확률 변수이고, **z의 평균과 분산, 그리고 복원된 output값**이 loss를 통해 모델의 작동하는 방식
  - 가설 : Mult-VAE가 Mult-DAE보다 좋은 성능을 낼 것이다.
    - **결과:** Mult-DAE의 성능이 Mult-VAE보다 조금더 좋았고, 하이퍼파라미터 튜닝 후에는 Mult-DAE의 성능이 0.01 더 높았음.

        - **이유:** 논문에 의하면 유저의 활동량이 높을 수록 Mult-VAE의 강한 사전가정이 성능에 안좋은 영향을 끼치고, Mult-DAE의 성능은 좋아진다고 한다. 본 대회의 학습데이터는 최소 16편 그리고 평균 165편의 영화를 본 유저들로 이루어져 있어 논문의 ML-20M실험 환경보다 유저활동량이 높아 Mult-DAE의 성능이 더 높다고 생각됨.

- **EASE**
    - **모델을 사용한 이유**

      베이스라인에 있던 모델들 중 AE 기반의 모델들의 성능이 가장 좋아 AE 기반의 모델들을 찾아보던 중, 모델구조가 무척 가벼운 EASE 모델이 Movielens 20M 데이터에서 좋은 성능을 보여준다는 정보를 보고 사용하게 됨. EASE 모델이 sparse한 데이터와 cold start problem 문제에 강한 특징을 가지므로 유저의 시퀀셜 데이터의 중간 부분이 누락되어 있는 본 프로젝트 문제에서 좋은 성능을 기대할 수 있음.

    - **가설 :** 유일한 파라미터인 정규화 파라미터 lambda의 기본값이 낮아(0.5), 보다 큰 값으로 정규화를 진행하여 성능 향상 기대.
        - **결과 :** lambda: 0.5 → 605로 조정하여 recall@10: 0.1422 → 0.1600이라는 단일 모델 최고 성능을 얻음.
        - **이유 :** 본 프로젝트의 데이터가 sparse하고, 유저간 데이터 갯수의 편차가 커서 과적합 문제가 일어나기 쉽기 때문에 정규화 파라미터를 그게 조정한 것이 유의미한 효과가 있었던 것으로 생각됨.

Sequential Model

- **Bert4Rec**
    - **모델을 사용한 이유**

      Self-attention mechanism구조와 bidirection 구조를 사용하는 모델로 user history의 long-term, short-term dynamics를 모두 반영해 다음 아이템 예측에 유리하고, 또한 랜덤하게 시퀀스의 일부분을 마스킹 해 마스킹된 아이템을 예측하는 cloze task가 대회의 방향과 비슷해 다음 아이템 예측에서 좋은 성능을 기대할 수 있음.

    - **가설1 :** Sliding Window를 사용한 Data Augmentation으로 성능 향상 기대
        - **결과 :** public과 private 성능 모두 하락함. 특히 private 성능은 0.01 하락.

        - **이유 :** 유저의 평균 영화시청 수인 165편인데 반해 1000편 이상으로 많이 본 유저들이 존재함. 따라서 영화를 많이본 유저의 데이터가 많아져 해당 유저들에게 편향되게 학습된다고 판단.
    - **가설2 :** 장르와 년도를 추가하고, 다음 1개의 아이템만이 아닌 5개의 아이템을 라벨로써 학습.
        - **결과 :** public성능은 하락했으나, private성능은 0.0734 → 0.0845로 상승함.
        - **이유 :** 다음 아이템 1개만을 라벨로 뒀을 때는 public과 private 둘다 하락하는 모습을 볼 수 있었음. 따라서 다음 5개의 아이템을 학습함으로써 보다 robust한 모델을 만들 수 있었다고 생각함.
    - **가설3 :** Cloze task와 mask token을 없애고 기존 Bert와 비슷하게 학습한 실험.
        - 결과 : public과 private 모두 Bert4Rec 베이스라인보다 좋았고, public 성능이 모든 실험 중 가장 좋은 성능을 보여줬음. 하지만 private 성능은 **가설2** 보다 떨어짐.
        - **이유 :** Cloze task가 본 대회의 목적에 부적합한지에 대한 실험 데이터는 부족하다. 논문에서 언급했듯이 Cloze task가 inference 때 활용되기 힘든 점을 고치기 위해 mask token을 사용했다. 하지만 중간에 누락된 데이터가 있어 유저의 sequence가 끊겨 있고, 정확하게 어느 시점인지 알 수 없어 Cloze task와 mask token이 우리의 task에 좋지 않다고 판단.
    - **가설4** : 기존 방식은 라벨을 0으로 두어 학습을 안시키는 부분을 확률값을 지정해서 random negative sampling을 진행했고 생각보다 많은 예측값이 0으로 라벨링 되었음. 이때 확률값이 cloze task를 위한 마스킹 확률에 종속적이여서 이 확률값을 변경하면 함께 영향을 받을거라 생각해 random negative sampling을 위한 확률값을 새로 만들어 설정하는 방법 대신에 인기도 기반의 negative sampling을 적용한다면 기존 방식보다 학습에 사용되는 데이터도 많아지고 인기도 기반의 아이템 위주로 학습하기 때문에 성능도 올라갈거라 판단.
        - 결과 : public score기준으로 0.0775 → 0.0760으로 성능이 하락하였음.
        - 이유 : 첫번째로는 max_len에 의해 학습에 사용될 시퀀스가 결정되고, 비교해본 결과 인기도 기반일때와 랜덤일때의 시퀀스 값이 크게 달라지지 않았음. 이미 유저들은 어느정도 인기가 있는 영화들을

시청했기 때문에 인기도 기반의 negative sampling이 크게 효과가 있지 않았다고 생각함.

두번째로는 인기도 기반의 negative sampling을 할 경우에 이미 정답일 확률이 높은것들로 학습하기 때문에 robust한 모델로 학습이 되지 않음. 이런 경우에는 라벨이 인기가 없는 영화일 경우에 예측을 못 할 확률이 높아 오히려 성능이 떨어졌다고 생각함.

- **SASRec**
  - **모델을 사용한 이유**

    Self-attention mechanism구조를 사용하는 모델로 user history의 long-term, short-term dynamics을 모두 반영할 수 있어 중간에 빠진 아이템 예측과, 앞으로의 아이템 시퀀스 예측모두 가능할거라 판단.

  - **가설1**: 마지막 아이템 시퀀스 뿐만이 아니라 중간에 빠진 아이템들도 예측해야 하는 task이기 때문에, 원래 시퀀스를 순서대로 학습하면 마지막 아이템 시퀀스 예측에 유리해지고 랜덤하게 순서를 섞어 또 학습하면 중간에 빠진 아이템 예측에 유리해 지고, robust한 모델이 될것임.
    - 결과 : leave-one-out validation이용했으며 CV값에서 큰 성능향상은 없었음. (0.72→0.73)
    - 이유 : 중간에 빠진 아이템 예측도 short-term dynamic한 특징이 있기 때문에 생각보다 유저의 action 시퀀스가 중요하다고 생각함.
  - **가설2**: 기존의 방식은 시퀀스의 아이템의 임베딩으로부터 implicit한 유저 임베딩을 추론하는데 개인화된 추천을 제공하기위해 유저의 선호를 나타내는 explicit한 유저 임베딩 매트릭스를 사용하면 유저의 선호도를 더 잘 표현할것임.
    - 결과 : 논문대로 explicit한 유저 임베딩 메트릭스를 마지막 레이어에 추가했지만 큰 성능향상은 없었음.
    - 이유 : 기존방식인 implicit한 유저 임베딩도 이미 모델이 유저의 모든 action을 고려한 방식이기 때문에 explicit한 유저 임베딩과 크게 차이가 없었다고 생각함.

▼ 모델 고도화

하이퍼 파라미터 튜닝

- 논문 참고 : 여러 모델들의 빠른 하이퍼 파라미터 튜닝을 위해 논문에서 말하는 하이퍼파라미터의 경향성이나 실험환경에서 사용한 값을 가져와 테스트하는 방식으로 진행함. AE계열의 Mult-VAE, Mult-DAE와 EASE모델들은 준수한 성능 향상을 보여주는 것을 확인함.
- WandB : 논문에서 가져온 하이퍼 파라미터 값에서 더 높이거나 줄임에 따른 성능을 시각화 하는데 사용함.

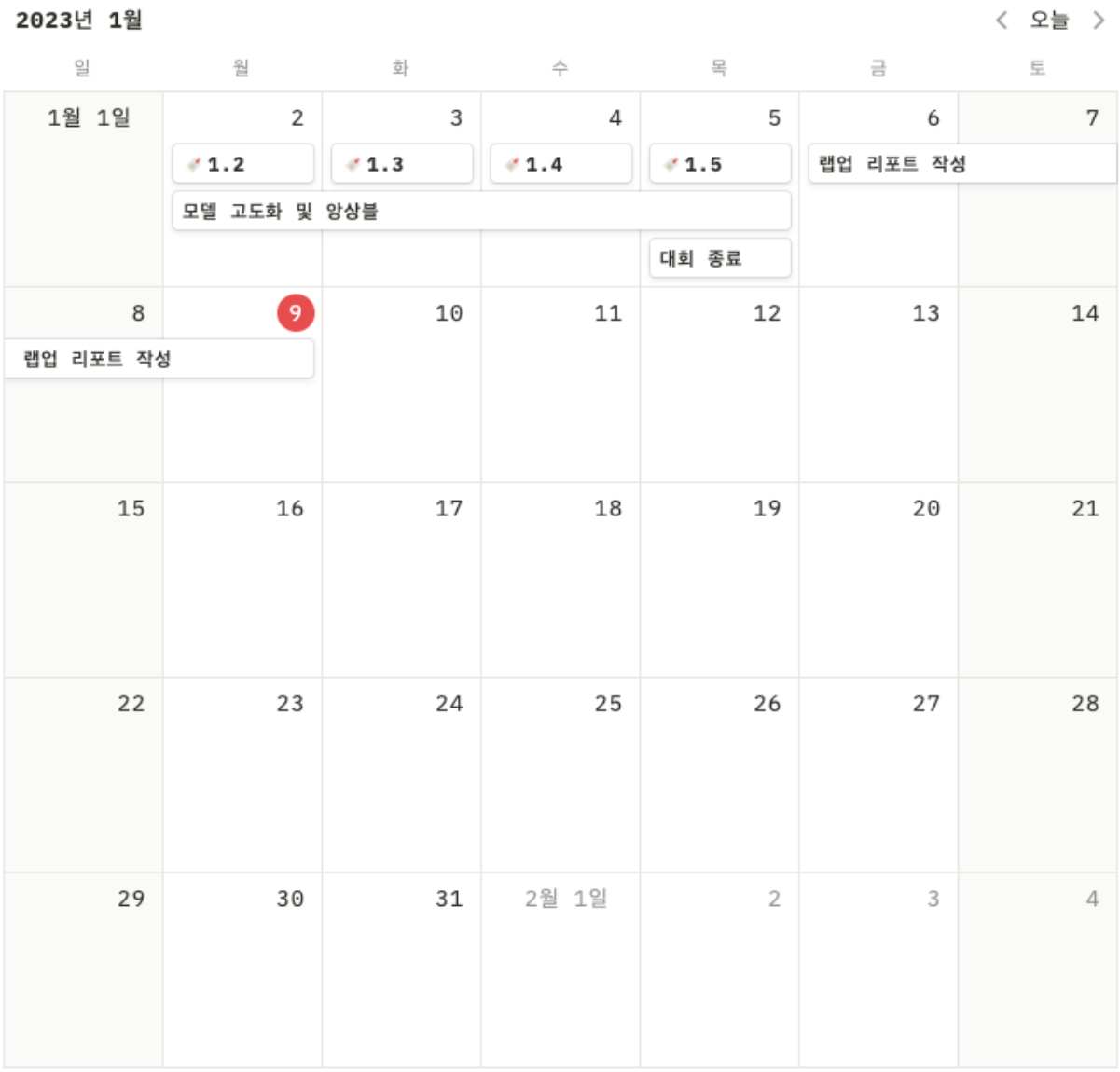
Ensemble

- **모델 기반 앙상블**
  - 앙상블 기법으로 각 모델들의 top10 결과를 기반으로 hard voting 방법 사용.
  - 시퀀셜, AE기반 모델들 등 서로 다른 계열의 모델들이 앙상블로 조합했을 때, 시너지가 낼 수 있을 것으로 생각하고 실험을 진행하여 최적의 모델 조합을 찾음.
    (ease, mse-dae, multi-dae, bert4rec, sasrec)
  - 모델 간 가중치는 각 모델들의 public test 성능을 사용함.
- **top-k 스코프 범위 확장**
  - 각 모델의 top10 범위 밖에도 정답이 존재할 가능성을 고려하여 top-k 스코프 범위 확장
  - k값을 20까지 확장하여 실험을 진행한 결과 public test 성능 기준으로 k=15~20일 때 앙상블 성능이 0.1646으로 가장 좋았음.
  - private test 성능을 확인해본 결과 k=20일 때 더 좋은 성능을 보여주는 것으로 확인되어 K값의 스코프를 확장시키는 것이 일반화 성능을 향상시키는 것으로 생각됨.

1-4. 프로젝트 로드맵

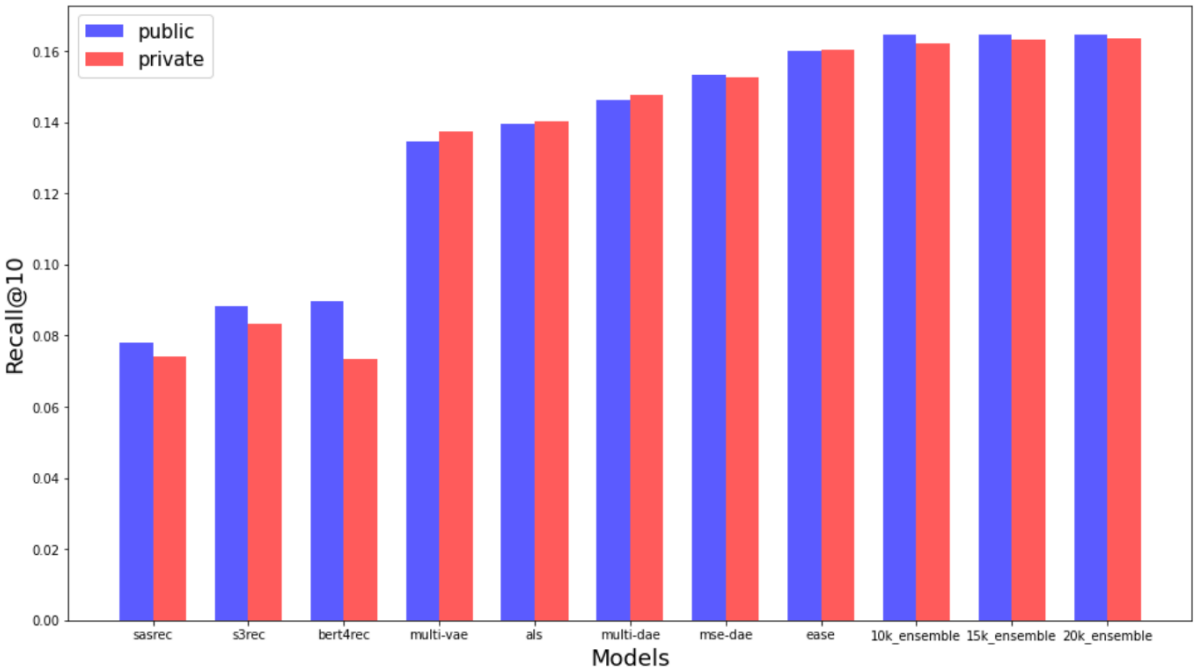
2022년 12월 오늘

일	월	화	수	목	금	토
27	28	29	30	12월 1일	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
	강의 수강 및 베이스라인 이해					12.17
	12.12	12.13	베이스라인 및 미션 모델 수정			
	대회 공개		12.14	12.15	12.16	
			리더보드 오픈			
18	19	20	21	22	23	24
12.18	12.19	12.20	12.21	12.22	12.23	12.24
	모델 탐색 및 EDA					
25	26	27	28	29	30	31
12.25	12.26	12.27	부캠 겨울방학			
	모델 탐색 및 모델 고도화 ⇒ 목표치 달성하기(0.17)					



1-5. 프로젝트 수행 결과

5️⃣ **Recall@10: `(public) 0.1646 ⇒ (private) 0.1634` | 최종 5위**



1-6. 자체 평가 의견

잘한 점

- 다양한 모델을 탐색하고 사용함.
- 이전 대회보다 협업적인 측면(Git Convention설정 및 Github활용)에서 큰 성장을 할 수 있었음.
- 모델 구조만 아니라 학습 중간에 사용되는 테크닉에 대해 개선.
- 논문을 바탕으로 모델에 대한 이해를 높이고 실험 방향을 정함.

시도했으나 잘되지 않았던 점

- 시퀀셜 데이터를 수정하여 시퀀셜 모델의 성능 향상을 원하는 만큼 달성하지 못함.
- Multi-dae 모델에 대해서 여러 피쳐를 추가하면서 실험을 진행하지 못함.
- Deepfm 관련한 실험에서 성능 달성이 제대로 이뤄지지 않음

아쉬운 점

- RecBole과 같은 라이브러리를 활용하여 다양한 모델을 적은 리소스

프로젝트를 통해 배운점

- 성능 지표로 경쟁하는 competition에서의 앙상블의 중요성.

로 실험하는 베이스라인을 구성하지 못함.

- Wandb를 활용하여 다양한 실험을 진행하지 못함.
- 추가 새로운 모델을 많이 탐색해서 새로 구상하지 못함
- 데이터에 기반한 가정과 실험을 제대로 진행하지 못함

- 깃&깃헙을 통한 협업의 중요성.
- 모델에 대한 이해와 가설 설정의 중요성.
- 모델 전반적인 구조만 아니라 중간 중간 테크닉의 변화도 필요함을 암.
- 논문을 통한 모델 이해의 중요성.