



---

# Movie Recommendation

부스트캠프 AI Tech 7기

---

## 목차 table of contents

1 프로젝트 목표 및 협업 과정

2 대회 소개 및 EDA

3 모델 선정

4 최종 결과

Part 1

# 프로젝트 목표 및 협업 과정



Inductive bias를 고려한 모델 선정

다양한 side information 취합

읽었던 논문들 구현 및 적용

## 11-25 회의록



소유자: 이효준 ...

조금 전에 마지막 업데이트 • 이 페이지를 본 사용자의 수 보기



날짜

2024년 11월 25일



목표



토론 주제

| 항목            | 비고   |
|---------------|--|
| 모델의 예측 아이템 분석 | <ul style="list-style-type: none"> <li>6800개 아이템 존재. Top 300(평이 많은 수 하거나 hacker score) 인기 있는 아이템으로 보고 이 아이템들이 추천되는 비율, 원본 트레인 데이터랑 비교하고 모델들끼리 비교해보기 <a href="#">X10-88: 인기도 기반 분석</a> 완료</li> <li>cold start, interaction이 상대적으로 적은 유저 군에 대해서 정답 확률이 어떤지 확인 <a href="#">X10-89: 유저 interaction 그룹 분석</a> 완료</li> <li>interaction이 상대적으로 많은 유저 군에 대해서 정답 확률이 어떤지 확인</li> <li>성능 제일 좋은 모델들과 공통 정답이 몇 개인지 확인 (중복이 적을 수록 이상불 성능이 올라갈 확률이 높다)</li> <li>추천된 영화의 개봉 연도별 히스토그램 기반 분석</li> <li>모델별 추천 아이템의 장르, 감독, 작가 top10 분석</li> </ul> |

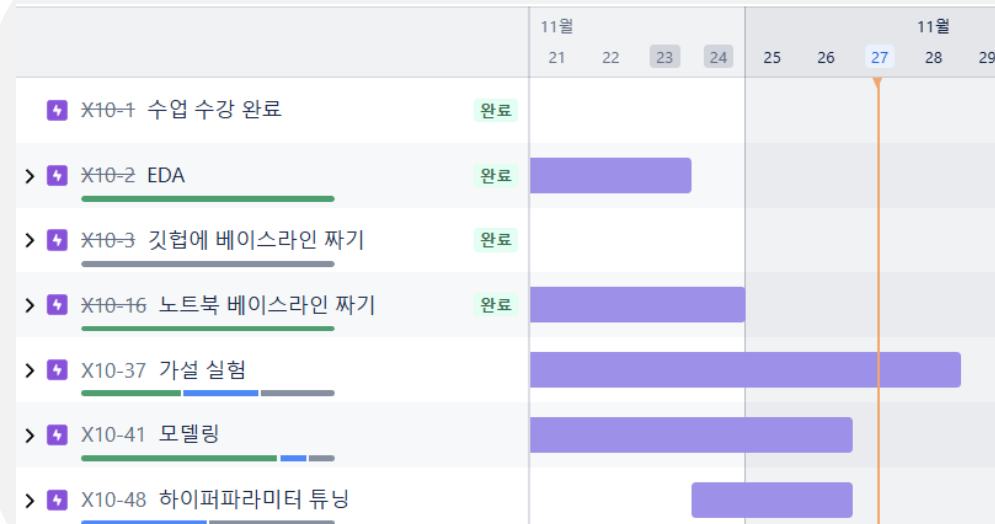
회의록 작성

|                |  |
|----------------|--|
| 실험 결과          | 분할 결과                                  |
| 실험 소유자         | @Jeongmoo Kwag                         |
| Jira 티켓        | 39                                     |
| 상태             | 완료                                     |
| 이 페이지에서 다루는 가설 | <a href="#">Train/Validation split</a> |

|        | Validation Recall@10  |
|--------|---|
| 설명     | Multi-VAE의 recall@10, ndcg metric 추적을 하였다   |
| 장점과 단점 | + 점진적으로 성능이 올라감을 확인할 수 있다   |
| 평가 결과  | <div>Epoch 1/20<br/>Training Loss: 1270.9060<br/>Evaluation Loss: 57.6237<br/>Recall@10: 0.0724, NDCG@10: 0.0545<br/>Validation Loss: 57.6237<br/>Recall@10: 0.0724, NDCG@10: 0.0545</div> <div>Epoch 2/20<br/>Training Loss: 1199.4623<br/>Evaluation Loss: 89.8114<br/>Recall@10: 0.0870, NDCG@10: 0.0650<br/>Validation Loss: 89.8114<br/>Recall@10: 0.0870, NDCG@10: 0.0650</div> <div>Epoch 19/20<br/>Training Loss: 1112.2895<br/>Evaluation Loss: 93.8965<br/>Recall@10: 0.1300, NDCG@10: 0.0988<br/>Validation Loss: 93.8965<br/>Recall@10: 0.1300, NDCG@10: 0.0988</div> <div>Epoch 20/20<br/>Training Loss: 1110.7793<br/>Evaluation Loss: 93.9051<br/>Recall@10: 0.1296, NDCG@10: 0.0979<br/>Validation Loss: 93.9051<br/>Recall@10: 0.1296, NDCG@10: 0.0979</div> <p>20 번의 epoch을 돌린결과 recall@10이 점진적으로 내려감을 확인했고</p> <p>0.1338<br/>-</p> <p>대회 제출 후 0.1338이 나와 최종적으로 신뢰성 있는 train/val split 방법임을 확인했다.</p> |

실험 기록

| 유형 ↑ ▾ | # 키    | 요약           | 상태   |
|--------|--------|--------------|------|
| ⚡      | X10-1  | 수업 수강 완료     | 완료   |
| > ⚡    | X10-2  | EDA          | 완료   |
| > ⚡    | X10-3  | 깃헙에 베이스라인 짜기 | 완료   |
| > ⚡    | X10-16 | 노트북 베이스라인 짜기 | 완료   |
| > ⚡    | X10-37 | 가설 실험        | 진행 중 |
| > ⚡    | X10-41 | 모델링          | 진행 중 |



## Jira를 활용한 이슈 관리 및 타임라인 설정

Jira를 통해 업무를 선정 및 배분하고 타임라인을 설정했다.

Make Something

Make Something

Make Something

Make Something

Make Something

Part 2

# 대회소개 및 EDA

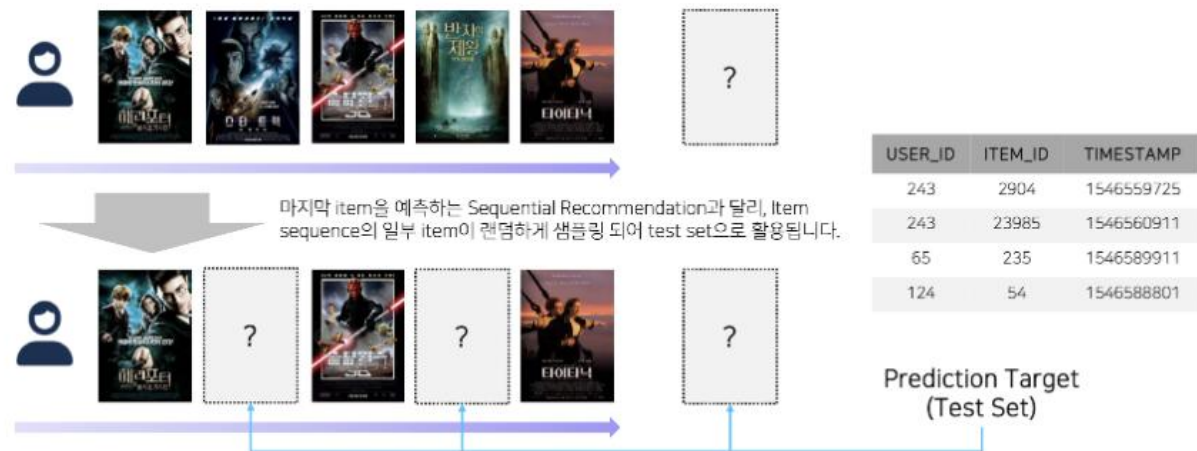
Make Something

Make Something



# MovieLens 데이터

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





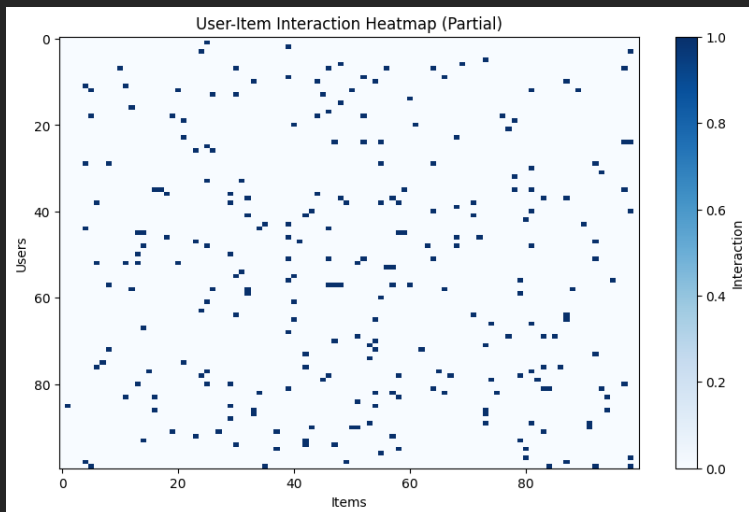
| RATINGS | UNIQUE  |
|---------|---------|
| USER    | 31360   |
| ITEM    | 6807    |
| TIME    | 5074973 |

평점 데이터

| SIDE-INFO | ITEM | UNIQUE |
|-----------|------|--------|
| TITLE     | 6807 | 6806   |
| GENRE     | 6807 | 18     |
| DIRECTOR  | 5503 | 1340   |
| WRITER    | 5648 | 2989   |
| YEAR      | 2989 | 93     |

Side Information 데이터

|                   | ML-20M      | ML-1M       | 대회 Data     |
|-------------------|-------------|-------------|-------------|
| # of users        | 136,677     | 6400        | 31360       |
| # of items        | 20,108      | 3900        | 6807        |
| # of interactions | 20M         | 1M          | 5.1M        |
| % of interactions | 0.36%       | 4.27%       | 2.65%       |
| Date range        | 1995 - 2015 | 2000 - 2003 | 2005 - 2015 |

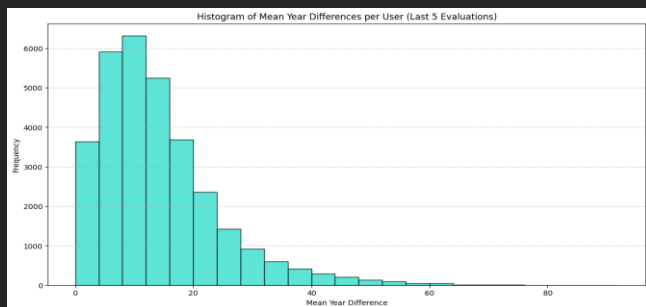
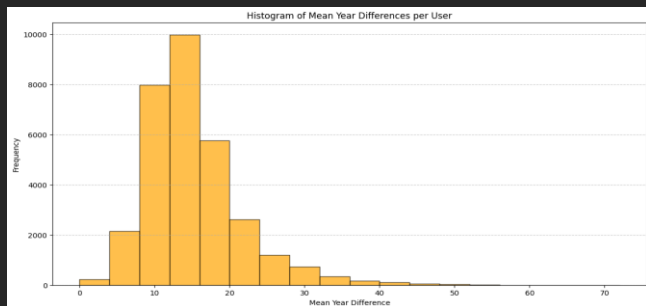


The Sparsity of the Dataset : 97.35%

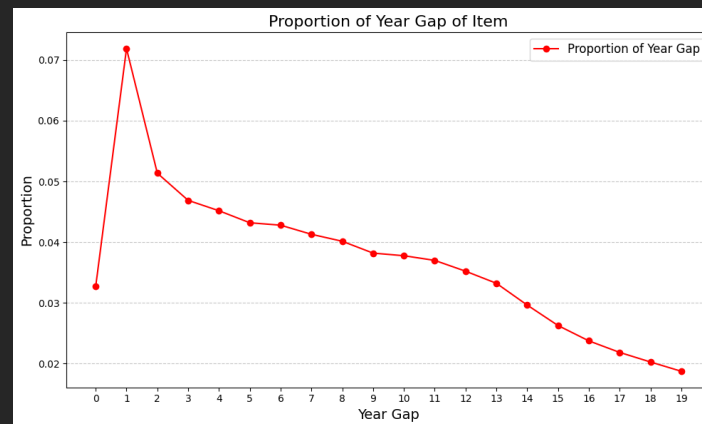
평점 데이터 중에서는 Dense한 Data

|          | Unique | Isnull | Item unique |
|----------|--------|--------|-------------|
| Director | 1399   | 301445 | 1304        |
| Writer   | 3838   | 312692 | 1159        |
| Year     | 93     | 1832   | 8           |

감독, 작가, 개봉연도의 고유값과 결측치  
(통합데이터 행 기준)



평가연도와 개봉연도의 차 그래프  
(전체 데이터 기준, 유저별 마지막 5개 데이터 기준)



개봉 N년 후 평가된 영화 비율

## Negative Sampling

- FM 계열 모델에서는 Negative Sampling이 필수적
- Negative Sampling이 성능에 유의한 영향을 미치지만 기법 선정에 어려움이 있다.

FM

## 감독, 작가 피처 처리 방안

| 감독   | 작가   |
|------|------|
| 1340 | 2989 |

- 감독, 작가의 경우 고유값, 다중값이 많아 처리를 하기에 어려움이 있다.

LightGBM

DeepFM

## 콜드 스타트

문제

- 추천시스템 데이터에서는 Cold Start 문제가 발생

대회 Data

- Cold Start에 해당하는 유저가 제외된 데이터

|       |        |
|-------|--------|
| Count | 31360  |
| Mean  | 164.36 |
| Min   | 16     |
| 25%   | 71     |

## 평가 시점

문제

- Timestamp를 사용하지 않을 시 leakage 발생 우려
- Static : 마스킹된 아이템의 시점 불명확
- Sequential : 마지막 리뷰 시점 불명확



대처 방안

- 추후 실험을 통한 확인 필요

단일 모델 검증 : Random + Sequence

**Static : Sequential 비율 기준 분할**

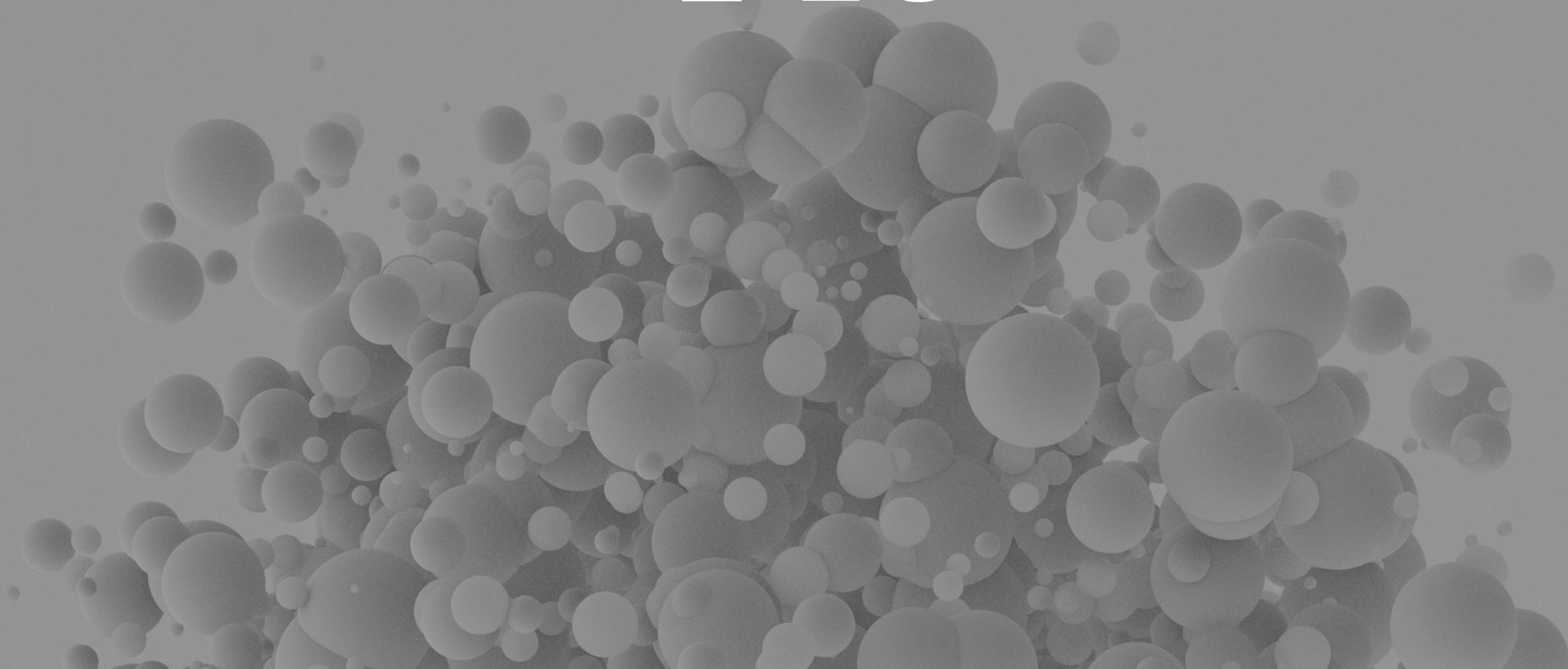
**Static : Sequential 비율 기준 분할**

모델의 특성을 고려한 검증  
Static : Random  
Sequential : Leave-One-Out



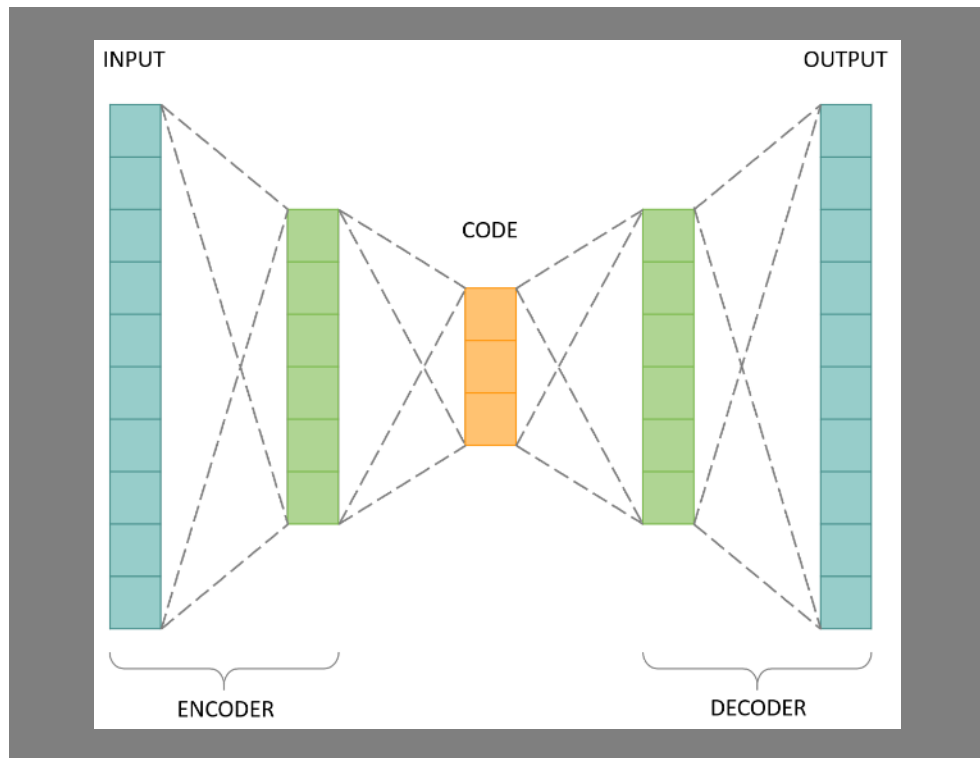
Part 3

# 모델 선정



### Part 3

## Static 모델 선정 이유



AutoEncoder 계열 모델

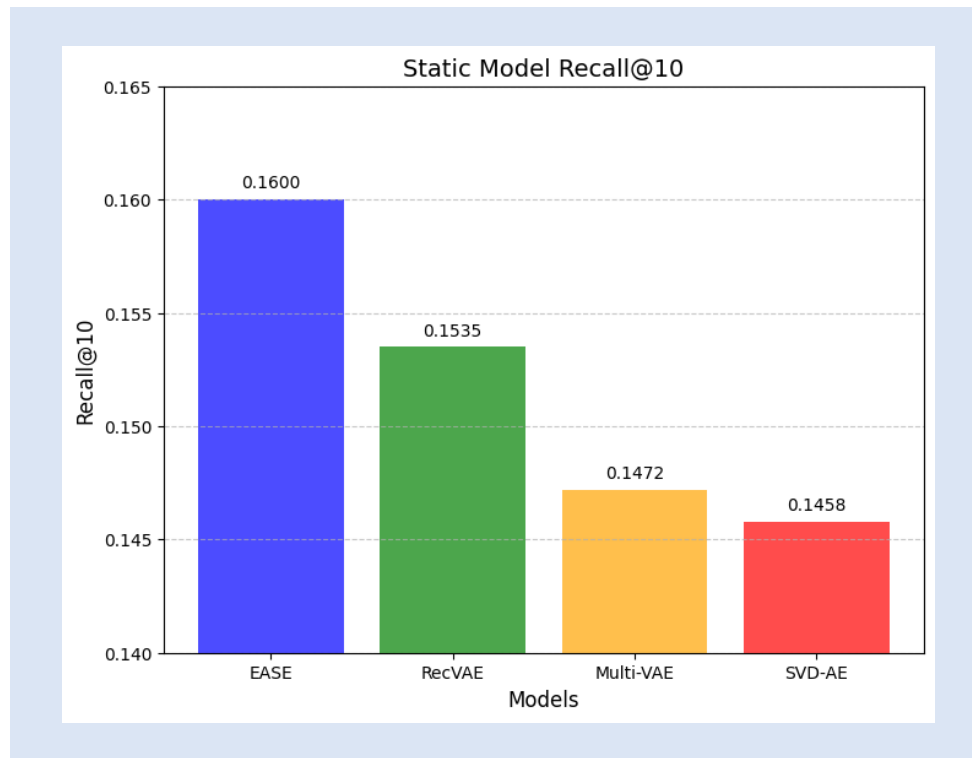
Multi-VAE

CVAE

EASE

SVD-AE

RecVAE



Static Model별 Recall@10

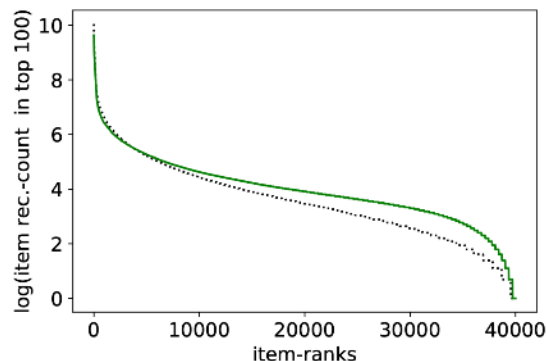


Figure 3:  $EASE^R$  (green) recommends long-tail items more often in the top-100, compared to  $MULT-VAE^{PR}$  (dotted), on MSD data.

### (c) MSD

|                 |       |       |       |
|-----------------|-------|-------|-------|
| popularity      | 0.043 | 0.068 | 0.058 |
| $EASE^R$        | 0.333 | 0.428 | 0.389 |
| $EASE^R \geq 0$ | 0.324 | 0.418 | 0.379 |

results reproduced from [13]:

|                 |                            |       |       |
|-----------------|----------------------------|-------|-------|
| SLIM            | — did not finish in [13] — |       |       |
| WMF             | 0.211                      | 0.312 | 0.257 |
| CDAE            | 0.188                      | 0.283 | 0.237 |
| $MULT-VAE^{PR}$ | 0.266                      | 0.364 | 0.316 |
| $MULT-DAE$      | 0.266                      | 0.363 | 0.313 |

### Multi-VAE, EASE 논문 인용해 예상되는 결과

"The notable improvement of EASE over the other models on the MSD data suggests that it is able to better recommend personally relevant items."

"EASE 모델이 MSD 데이터에서 다른 모델들보다 눈에 띄게 향상된 성능을 보인 것은, 이 모델이 개인에게 더 적합한 아이템을 추천하는 능력이 우수함을 시사합니다."

## 인기도 분석 (Top 300)

|       | EASE   | SVD-AE | Multi-VAE | RecVAE |
|-------|--------|--------|-----------|--------|
| 추천 비율 | 0.8520 | 0.8422 | 0.7974    | 0.7772 |

EASE가 예상과 달리 인기도 기반 추천이 가장 많음

## 아이템 다양성 분석

| Model     | Coverage |
|-----------|----------|
| RecVAE    | 0.4197   |
| Multi-VAE | 0.4162   |
| SVD-AE    | 0.3489   |
| EASE      | 0.2724   |

EASE가 예상과 달리 다양성이 가장 낮음

## Sequence Length

- 데이터셋의 Sequence length가 평균 164, 최대값 2912로 길다.
- RNN 기반인 GRU4Rec보다는 SASRec, BERT4Rec 고려했다.

SASRec

BERT4Rec

|              |        |
|--------------|--------|
| <b>Count</b> | 31360  |
| <b>Mean</b>  | 164.36 |
| <b>Std</b>   | 150    |
| <b>Min</b>   | 16     |
| <b>25%</b>   | 71     |
| <b>50%</b>   | 114    |
| <b>75%</b>   | 200    |
| <b>Max</b>   | 2912   |

Sequence length 통계량

## Sequential Recommendation의 Loss 를 다룬 논문

「 Improving sequential recommendation models with an enhanced loss function'', Li et al., 2023 」

「 Training Dross into gold loss: is BERT4Rec really better than SASRec?'', Klenitskiy et al., 2023 」

## Sequential Recommendation의 Loss 를 다룬 논문

〔 기존 SASRec의 Loss: Positive Sample 한 개당 Negative Sample 한 개를 통해 BCE 계산 〕

〔 BERT4Rec의 Loss: 마스킹 된 아이템에 대한 Cross Entropy 계산 〕



## SASRec은 시퀀스의 모든 Timestamp에서 학습 (Loss 계산)

A User Sequence Example:

|    |    |    |   |    |    |
|----|----|----|---|----|----|
| 25 | 17 | 11 | 9 | 36 | 22 |
| 1  | 2  | 3  | 4 | 5  | 6  |

|       |
|-------|
| Item  |
| index |

Item: User interaction item  
Index: User sequence index

Input:

|    |    |    |   |    |
|----|----|----|---|----|
| 25 | 17 | 11 | 9 | 36 |
| 1  | 2  | 3  | 4 | 5  |

Target:

|    |
|----|
| 22 |
| 6  |

Output:

|                |
|----------------|
| R <sub>5</sub> |
| 5              |

(a)

Input:

|    |    |    |   |    |
|----|----|----|---|----|
| 25 | 17 | 11 | 9 | 36 |
| 1  | 2  | 3  | 4 | 5  |

Target:

|    |    |   |    |    |
|----|----|---|----|----|
| 17 | 11 | 9 | 36 | 22 |
| 2  | 3  | 4 | 5  | 6  |

Output:

|                |                |                |                |                |
|----------------|----------------|----------------|----------------|----------------|
| R <sub>1</sub> | R <sub>2</sub> | R <sub>3</sub> | R <sub>4</sub> | R <sub>5</sub> |
| 1              | 2              | 3              | 4              | 5              |

(b)

SASRec 학습 방식

Input:

|    |      |    |   |      |    |
|----|------|----|---|------|----|
| 25 | Mask | 11 | 9 | Mask | 22 |
| 1  | 2    | 3  | 4 | 5    | 6  |

Target:

|    |    |
|----|----|
| 17 | 36 |
| 2  | 5  |

Output:

|                |                |
|----------------|----------------|
| R <sub>2</sub> | R <sub>5</sub> |
| 2              | 5              |

(c)

BERT4Rec 학습 방식

**"The BERT4Rec training objective is only weakly related to the final goal of sequential recommendations. For SASRec, tasks for the stages of training and prediction are exactly the same."**

**"BERT4Rec의 학습 목표는 순차적 추천의 최종 목표와 연관성이 낮습니다.  
반면 SASRec의 경우, 학습 단계와 예측 단계의 프로세스가 정확히 동일합니다."**

### ML Dataset Benchmark (Leave one out)

| Dataset | Metrics | SASRec | BERT4Rec <sub>hf</sub> | BERT4Rec-VAE | <b>SASRec</b>                |
|---------|---------|--------|------------------------|--------------|------------------------------|
| ML-1M   | HIT@10  | 0.2089 | 0.2594                 | 0.2512       | <b><u>0.2849(36.38%)</u></b> |
|         | NDCG@10 | 0.1119 | 0.1409                 | 0.1387       | <b><u>0.1642(46.74%)</u></b> |
| ML-20M  | HIT@10  | 0.1437 | <u>0.2378</u>          | 0.1558       | <b><u>0.2176(51.43%)</u></b> |
|         | NDCG@10 | 0.0716 | <u>0.1310</u>          | 0.0892       | <b><u>0.1214(69.55%)</u></b> |

Base



Max Len 조정

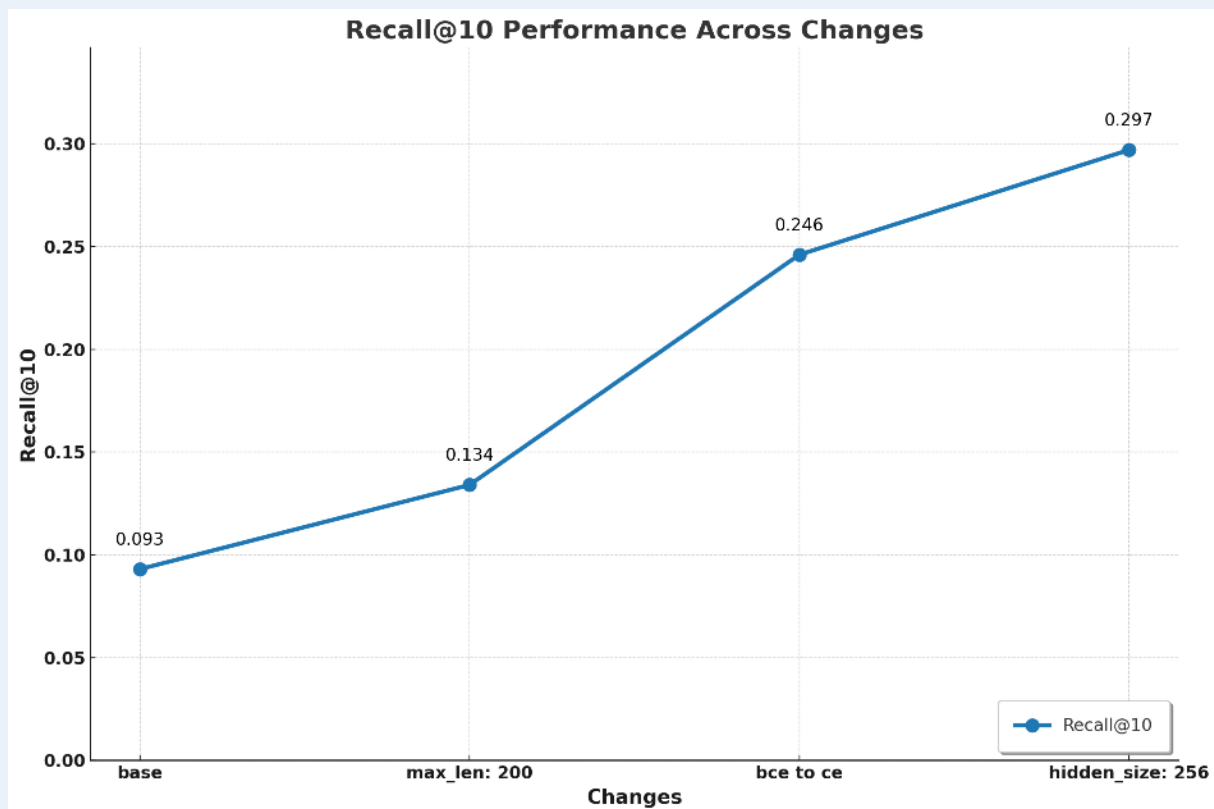


Loss 조정



Hidden Size 조정

## 점수변화 과정 그래프



**Static  
Model**

**+**

**Sequential  
Model**

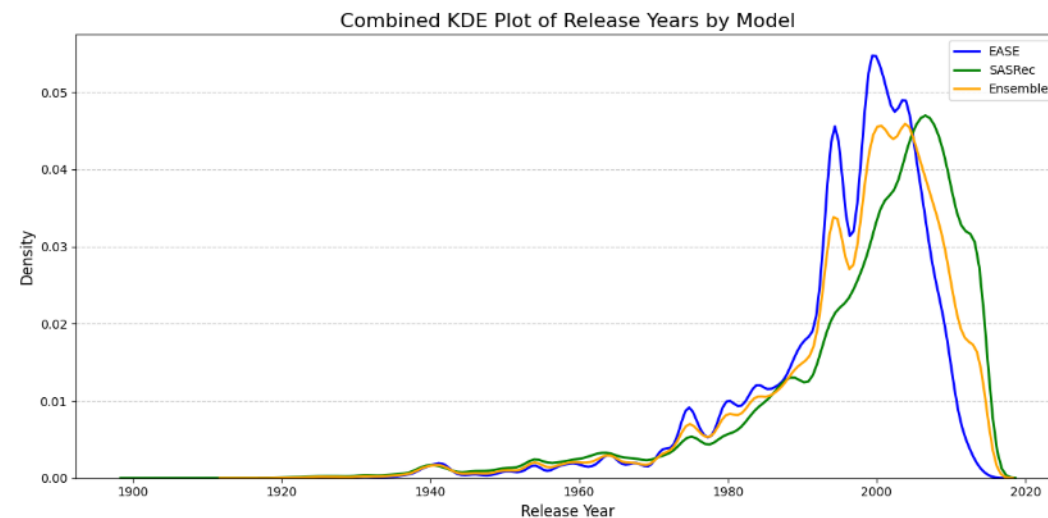
**=**

**Ensemble  
Model**

## EASE & SASRec 공통 추천

|                    |       |
|--------------------|-------|
| 유저별<br>평균 공통 추천 개수 | 0.55  |
| 전체 공통 추천 개수        | 17356 |
| 전체 공통 추천 비율        | 5.53% |

## 개봉 연도



각 모델이 의도한대로 서로 다른 측면을 효과적으로 포착

## 인기도 분석 (Top 300)

|       | EASE   | SASRec | Ensemble |
|-------|--------|--------|----------|
| 추천 비율 | 0.8520 | 0.3748 | 0.6398   |

## 아이템 다양성 분석

| Model    | Coverage |
|----------|----------|
| EASE     | 0.2724   |
| SASRec   | 0.9379   |
| Ensemble | 0.8626   |

앙상블 모델이 두 모델의 추천결과를 적절히 반영





Part 4

**최종 결과**

## Static Model

AutoEncoder 계열  
인기도 기반 추천

+

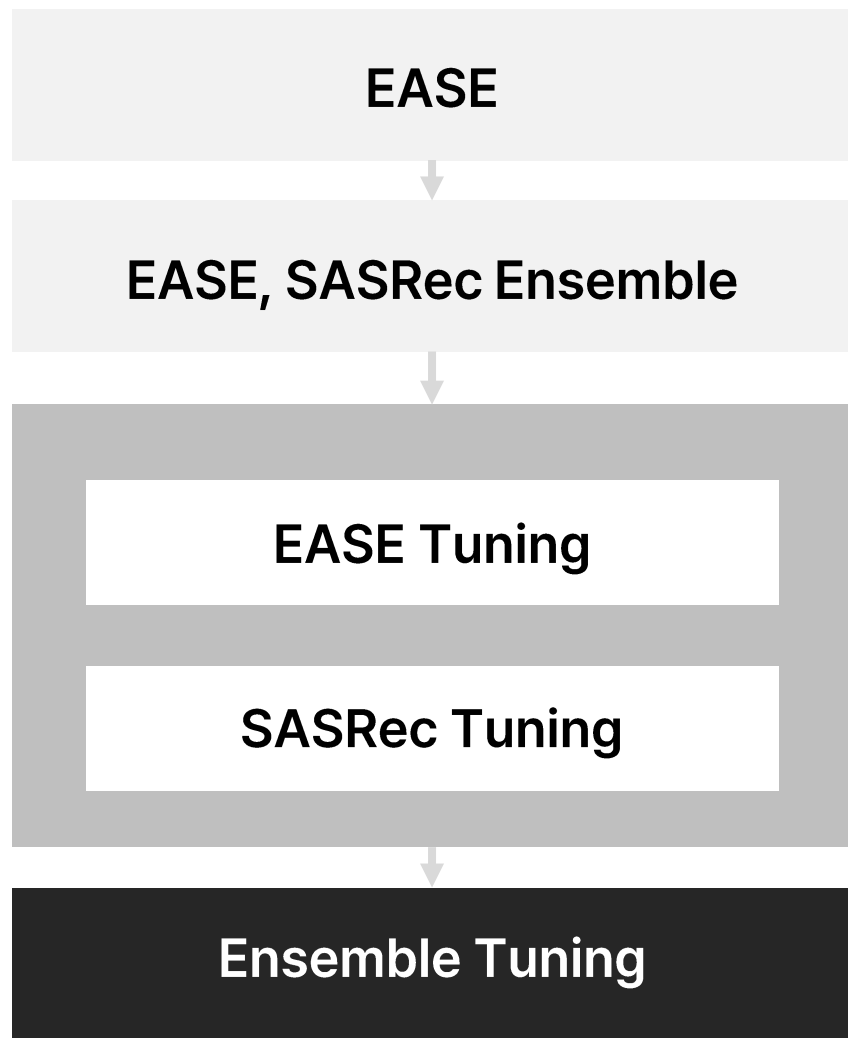
## Sequential Model

SASRec  
평점 시퀀스 기반 추천

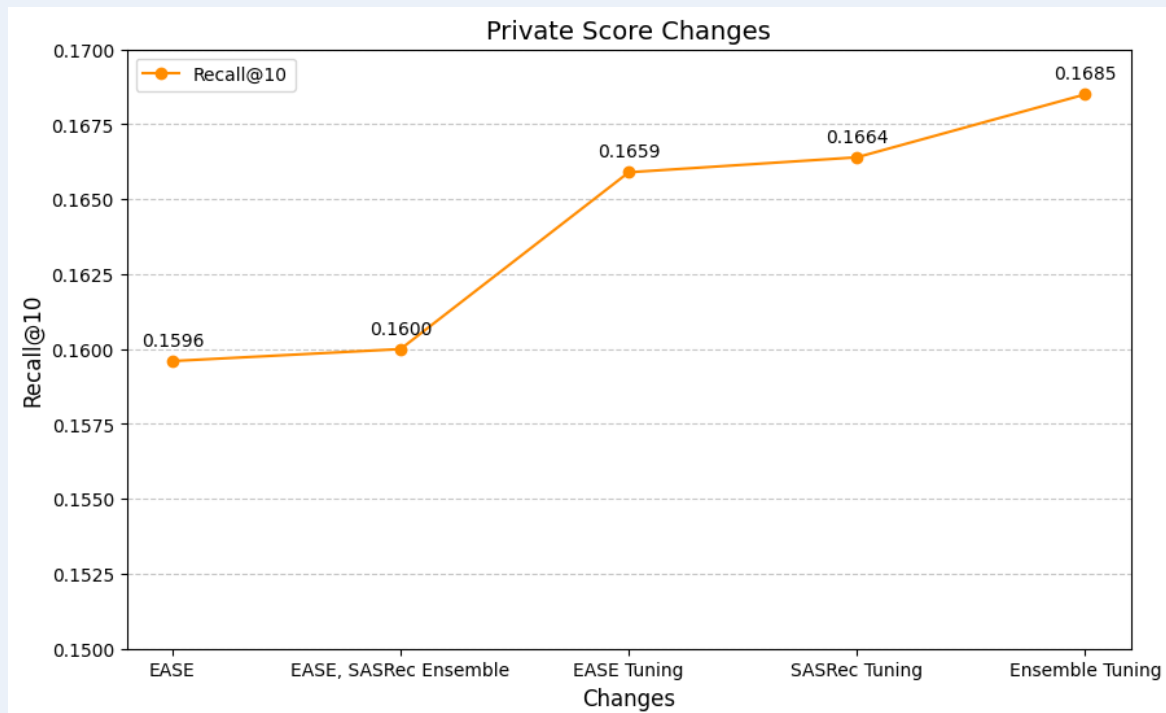
=

## Ensemble Model

최종 모델  
인기도 기반 + 평점 시퀀스



## 점수변화 과정 그래프



## 대회 vs 현업

이번 대회는 추천보다는 예측에 가까운 환경이었기에 해당 모델들의 앙상블이 좋은 성적을 낼 수 있었다. 현업에서도 위의 모델들이 좋은 성과를 낼 수 있을지에 대한 고민이 필요하다.

마스킹한 아이템을 복원하는 Bert4Rec이나 Contrastive learning 을 통한 User representation 을 학습하는 CL4SRec을 대회에 맞게 sequential한 아이템이 아니라 static한 아이템을 예측하게 만들어 볼 수 있지 않을까?

시퀀스 모델에서 timestamp 정보를 활용하려면 유저의 평점 시간 간격을 활용해서 테스트 시점의 timestamp 예측해서 사용할 수 있지 않을까?

Side-Information을 활용할만한 다른 모델은 없을까?

## 잘한 점

논문 및 관련 자료 리서치 후 데이터셋에 맞게 적용  
활발한 회의 진행으로 원활한 의견 교류

## 아쉬운 점

실험으로 실현되지 못한 가설들이 많았다.  
Side Information 활용 부재  
다양한 모델의 시도는 좋았으나 깊이가 부족했다.

“엄마는 항상 인생을 초콜릿 상자와 같은 거라고 말씀하셨습니다.  
그걸 집기 전에는 알 수 없는 거라고.”

포레스트 검프



End

---

# Q & A

