

# Wrap-Up Report



## 목차

1. Team Wrap Up Report
  - a. 프로젝트 개요
  - b. 팀 구성 및 역할
  - c. 수행 절차 및 방법
  - d. EDA
  - e. 모델
  - f. 앙상블
  - g. 최종 결과
2. 개인 회고

## 1. Team Wrap Up Report

### 1) 프로젝트 개요

#### 프로젝트 주제: 🎬 Movie Recommendation

이 프로젝트의 목표는 사용자의 영화 시청(평가) 이력 데이터를 기반으로, 사용자가 시청할 영화와 선호하는 영화를 예측하는 것이다.

#### 프로젝트 설명

##### 1. 데이터

- **explicit feedback**인 Movie Lens 데이터를 **implicit feedback** 데이터로 변환
- user-item interaction data와 item 정보에 대한 데이터 존재

##### 2. 문제 정의

- time-ordered sequence에서 일부 item이 누락된 상황을 가정하여 일반적인 **sequential recommendation**보다 복잡한 환경



##### 3. side-information 활용

- item과 관련된 다양한 부가 정보를 함께 사용하는 방향으로 설계 가능

#### 4. 예측 목표

Training Data에 존재하는 전체 유저에 대해서 각각 10개의 아이템 추천

- 특정 시점 이후의 사용자 행동 예측(Sequential)
- 특정 시점 이전 데이터로부터의 사용자 선호 파악(Static)

#### 5. 평가 지표

- **normalized recall@10**

$$Recall@K = \frac{1}{|U|} \sum_{u \in U} \frac{|\{i \in I_u \mid \text{rank}_u(i) \leq K\}|}{\min(K, |I_u|)}$$

### 프로젝트 목표

이번 프로젝트는 다음 두 가지 목표를 가지고 진행되었다.

#### 1. 추천 모델에 대한 학습

- **RecBole**을 활용하여 다양한 추천 모델을 접하고 학습하며, 추천 시스템에 대한 전반적인 이해를 높이는 것을 목표로 삼았다.

#### 2. 모델 구현

- 학습한 모델 중 일부를 선정하여 관련 논문을 읽고 공부하며, 직접 모델을 구현하는 데 도전하였다.

## 2) 프로젝트 팀 구성 및 역할

이름	역할
강현구	Sequential-based models, Ensemble
서동준	EDA, LRML, K-fold Ensemble
양시영	EDA, MLflow, CDAE, Soft Voting
이도걸	ADMMSLIM, Parameter Tuning, General model
이수미	EDA, VAE기반 모델, MultiVAE 구현, Hard Voting
최윤희	EDA, context-aware model, EASE/Multi-EASE 구현

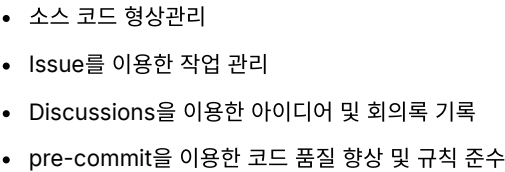
## 3) 프로젝트 수행 절차 및 방법

### 수행 과정

	11월 11일	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
회의	협업 규칙 정하기																	
EDA	EDA																	
Recbole				Recbole 세팅				Recbole 구조 이해 및 모델 탐색										
ML flow								ML flow 모듈화 및 적용										
모델구현 및 실험								KNNCF					Knowledge 모델				MultiVAE	
				Rule-base 실험											CDAE			
										후처리 filtering					LRML			
하이퍼 파라미터 튜닝										Context 모델					EASE			
양상블															하이퍼 파라미터 튜닝			
																	양상블	

### 협업 문화

#### 1. GitHub



**rwflow** 3.18.0
Experiments Models

---

Home >

## ramblunconious-ram-500

**Overview**
Model metrics
System metrics
Artifacts

**Description**

FEWEL-0-1282

**Details**

Created at	2024-11-20 18:03:30
Created by	<a href="#">login</a>
Experiment ID	480154738053491870
Status	<span style="color: green;">✔ finished</span>
Run ID	12d4cd3f0c8b4e7f99641ec7a2e18712
Duration	1.3h
Datasets used	--
Tags	<a href="#">Add</a>
Source	<a href="#">nlp_model.py</a>
Logged models	--
Registered models	--

**Parameters (47)**

Search parameters

Parameter	Value
cfg_grad_norm	None
metrics	"Recall", "MRR", "NDCG", "IR", "Precision"
valid_metric	MRR@10
topk	[10]
enable_sampler	False
sam_dataset	False
metric_validation_place	4
transform	None
strapping_step	10
train_batch_size	2048
loss_type	BCE

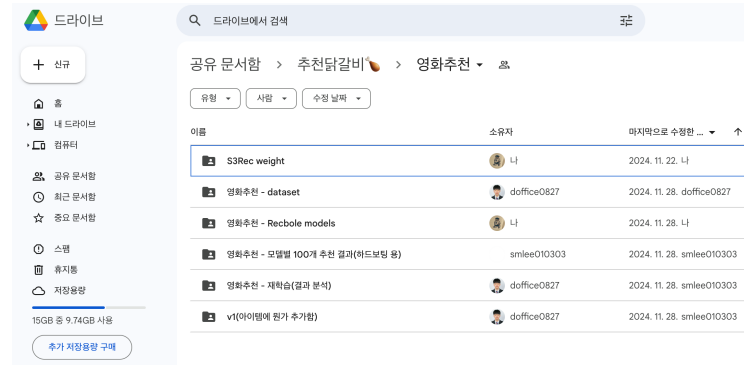
**Metrics (4)**

Search metrics

Metric	Value
valid_score_trigger	1
test_result	0.9531
best_valid_result	0.1445
best_valid_score	0.1445

- 하이퍼파라미터 기록, 모델 성능비교 등의 모델 실험 관리

### 3. Google Drive



- 데이터 및 자료 관리

### 4. Notion

- 실험 결과 및 아이디어 기록 및 공유

### 5. Zoom

- 정기적인 회의를 통한 협업 진행

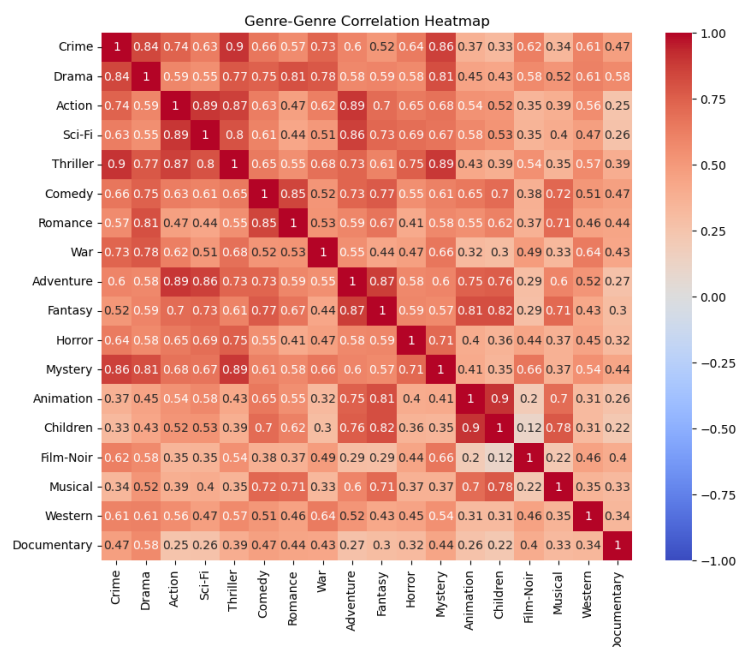
## 4) 탐색적 데이터 분석 (EDA)

### 유저 평가 기간 확인

- 11737(37.4%)명의 사용자가 평가 기간이 1일 이내임
- 평가 기간이 짧은 유저에게 sequential 정보를 기대하기 힘들

### 장르 선호 유사도 확인

- 유저별로 시청한 영화의 장르 통계를 활용하여, 장르별 선호 유사도 확인
- 특정 장르 선호 유저에게 유사 장르 추천, 장르 클러스터링 등에 활용 가능



### 특정 유저(user=11)의 train\_ratings.csv의 year 칼럼 확인

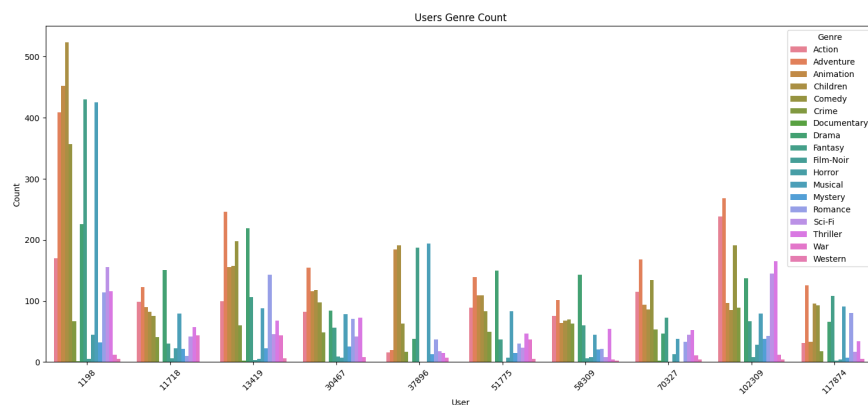
- year값이 2009와 2011 만 있다.
- 아이디어: 유저별로 특정 기간의 영화만 추천해줘야 할 것 같다.

	user	item	time	time_kst	year
0	11	4643	1230782529	2009-01-01 13:02:09	2009
1	11	170	1230782534	2009-01-01 13:02:14	2009
2	11	531	1230782539	2009-01-01 13:02:19	2009
3	11	616	1230782542	2009-01-01 13:02:22	2009
4	11	2140	1230782563	2009-01-01 13:02:43	2009
...	...	...	...	...	...
371	11	48738	1294796106	2011-01-12 10:35:06	2011
372	11	6291	1294796113	2011-01-12 10:35:13	2011
373	11	46578	1294796119	2011-01-12 10:35:19	2011
374	11	7153	1294796132	2011-01-12 10:35:32	2011
375	11	4226	1294796159	2011-01-12 10:35:59	2011

376 rows x 5 columns

### 랜덤한 유저 10명에 대한 장르 선호(preference) 시각화

- 유저별 시청한 영화의 장르 개수를 이용하여 시각화
- 아이디어: 유저별로 특정 장르 몇개에 집중해서 시청하는 경향이 보인다.

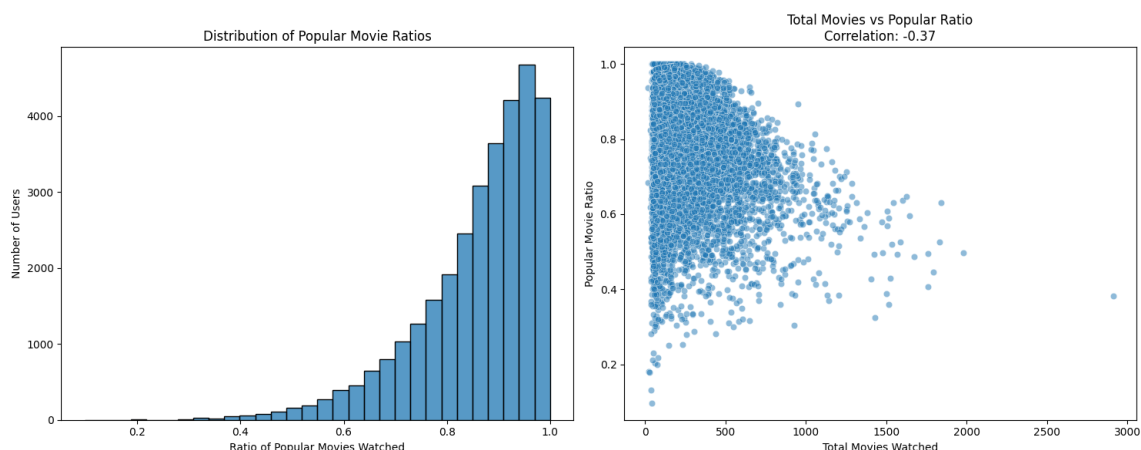


### 더 적은 영화를 시청하는 사용자일수록 인기 영화에 집중하는 경향이 있다

- 통계적 유의성:
  - 영화 시청 수와 인기 영화 선호도 간의 음의 상관관계 존재( $\rho = -0.284$ ,  $p < 0.001$ )
  - 그룹 간 차이가 통계적으로 매우 유의미( $H = 1522.82$ ,  $p < 0.001$ )
- 효과의 크기:
  - Very High 그룹과 Low 그룹 간의 실질적 차이 확인(Cohen's  $d = -0.623$ )
  - 중간~큰 효과 크기로, 실제적으로 의미 있는 차이

- 구체적 수치:
  - 가장 많은 사용자(23,535명)가 속한 Very High 그룹(인기 영화 비율 91%):
    - 평균 136.73편 시청
    - 신뢰구간 [135.5, 137.97]로 매우 안정적
  - 반면 Low 그룹(인기 영화 비율 35%):
    - 평균 322.17편 시청
    - 더 많은 영화를 보며 다양한 영화 탐색

이러한 패턴은 "가끔 영화를 보는 사용자는 검증된 인기 영화를 선호하는 반면, 영화를 자주 보는 사용자는 더 다양한 영화를 탐색한다"는 실제 행동 패턴을 반영한다.



왼쪽: x축-시청한 영화 중 유명한 영화의 비율, y축- 유저 수 / 오른쪽: x축- 영화 시청수 y축- 시청한 영화 중 유명한 영화의 비율

## 5) 모델

RecBole을 통해 데이터와 Task에 적합한 모델을 빠르게 추려내고, 적합한 모델을 직접 구현하는 방식으로 프로젝트를 진행

주어진 상황에 알맞게 Sequential 모델, Static 모델, Context-aware 모델을 모두 다루었다.

- Sequential 모델은 시청할 영화를
- Static 모델은 선호하는 영화를 잘 예측할 거라 생각하였고,
- item 정보를 다루기 위해 context 기반 모델도 사용하였다.

### RecBole 결과

	모델명	결과 (Valid)	결과 (리더보드)
General	ADMMSLIM	0.2244	0.1616
	CDAE	0.1445	0.1378
	DMF	0.1024	-
	EASE	0.19	0.1584
	ENMF	0.1504	0.1148
	FISM	0.08	-
	ItemKNN	0.1378	-
	LightGCN	0.1311	0.1161

	모델명	결과 (Valid)	결과 (리더보드)
	MacridVAE	0.159	-
	MultiDAE	0.1662	-
	MultiVAE	0.1602	0.1245
	NeuMF	0.1421	-
	NGCF	0.1147	0.1147
	RecVAE	0.2017	-
<b>Sequential</b>	BERT4Rec	0.0887	0.0486
	GRU4Rec	0.1750	0.1267
<b>Context-aware</b>	DeepFM	0.1338	-
	EulerNet	0.1444	-
	PNN	0.1442	-
	xDeepFM	0.1386	0.1070

## 결과 분석

- sequential 모델의 성능이 낮았음
  - explicit feedback 정보를 implicit feedback으로 변형한 데이터였고, timestamp의 간격이 매우 짧은 데이터가 많아, sequential 정보량이 적었던 것으로 추측
- EASE와 ADDM SLIM과 같은 item 간 관계를 학습하는 선형 모델의 성능이 높았음
  - 유저() 대비, 아이템()의 개수가 적은 데이터였기 때문에, 아이템간의 관계를 학습하기 용이했을 것으로 추측
  - sparsity가 낮은 데이터의 특성상, 딥러닝 모델보다 단순 선형 모델이 데이터의 단순한 특성을 더 잘 학습함
- Context-aware 모델의 성능이 낮았음.
  - sparse한 데이터에서 FM 기반의 모델들이 비선형 관계를 일반화하기 어려웠을 것으로 추측

## 선정된 모델 공부/구현/발전

RecBole에서 괜찮은 성능을 보였던 모델들을 선정하여 공부,구현 하고 발전 시키려 노력하였다.



### CDAE

오토인코더 대신 디노이징 오토인코더를 사용하여 성능을 개선한 협업 필터링 모델

- 특징
  - 입력 데이터에 노이즈(drop out)를 추가함으로써 모델의 일반화 성능 향상
  - 사용자에게 대한 고유한 잠재벡터를 학습 할 수 있어서 개인화된 추천이 가능함
  - 랭킹을 통해 사용자에게 Top-N 추천을 하는데 사용됨
- 실험 결과
  - drop-out 비율, learning rate 등의 조절에 따라서 성능 및 학습 시간 차이 보임
- 결과 정리 및 아쉬운점
  - 하이퍼파라미터를 일부 수정하는 정도로는 성능에 큰 영향을 주지 못함(실제로 리더보드 0.14이상의 성능이 나오지 못함)
  - L2 정규화 등 loss 함수 수정, activation function 변경 등을 시도하지 못해서 아쉬움



## MultiVAE

VAE를 기반으로 사용자와 아이템 간의 잠재적인 관계를 학습하는 추천 시스템

- 특징
  - 비선형 확률 모델
  - Multinomial Likelihood 사용
  - 정규화 파라미터 도입( **Annealing** )
    - KL Divergence 제어 → 학습 과정에서 점진적으로 발산의 중요성을 증가 시킴 → 보다 좋은 잠재 변수 표현 학습
    - Negative Sampling을 별도로 수행할 필요가 없다.
- 결과 정리 및 아쉬운 점
  - 구현한 모델이 RecBole과 validation score은 유사했으나 test score에서 큰 차이를 보임.  
validation dataset에 과적합했다고 판단함.  
⇒ L2 정규화 도입, dropout 비율 높이기(0.5→0.7), learning rate(0.001→0.0005), 차원 조절을 해보았으나 큰 효과가 없었다.  
모델 구현시 잘못 구현된 부분을 끝까지 찾아내지 못했다.
  - 문제를 해결하느라 더 다양한 방법(스케줄러 사용 등)을 적용해보지 못한 것이 아쉬움.



## GRU4Rec

RNN 계열인 GRU(Gated Recurrent Unit)를 기반으로 순차적 사용자 행동 데이터를 모델링하는 추천 시스템.

- 특징
  - GRU를 활용하여 시퀀스 데이터의 장기 종속성(long-term dependency)을 효과적으로 처리
  - Negative Sampling 없이 softmax를 사용하여 다음 아이템을 예측
  - 사용자 세션에 대한 개인화된 추천 제공 가능
- 실험 결과
  - RecBole에서 제공된 GRU4Rec을 활용하여 실험 수행.
  - 하이퍼파라미터 튜닝을 통해 recall@10 점수를 0.9에서 1.2로 개선.
  - 다른 HGN이나 SASRec과 같은 시퀀셜 모델보다 성능이 우수.
  - EASE와 ADMM-SLIM과 같은 간단한 모델에 비해서는 성능이 떨어짐.
  - 앙상블용으로 사용되었으나, 앙상블 시에도 기대만큼 효과를 보지 못함.
- 결과 정리 및 아쉬운 점
  - GRU4Rec을 직접 구현하는 대신 RecBole의 구현을 사용함.
  - 하이퍼파라미터 튜닝을 통해 성능 향상을 시도했으나, 기본적인 모델 구조를 개선할 기회는 부족.
  - 시간이 부족하여 다양한 실험이나 구현을 하지 못한 점이 아쉬움.





## EASE

input에 대해 간단한 선형 변환을 통해 재생성된 output을 출력하는 얇은 모델

- 특징

- hidden layer가 존재하지 않는 모델
- closed form으로 파라미터 추정 가능하다 → 빠른 모델 적합 시간
- sparse한 데이터에 robust하다

- 실험 결과

- 정규화에 사용되는 weight를 높일수록 성능이 향상함
- 현재 데이터에서 가장 안정적으로 높은 성능을 보임

- 추가 시도 (Multi-EASE)

- 기존의 EASE 모델 square loss를 multinomial loss로 수정함으로써 Multi-VAE와 유사한 원리로 성능 개선을 시도해봄
  - 기존 버전 :  $\min_B \|X - XB\|_F^2 + \lambda \cdot \|B\|_F^2$
  - 수정 버전 :  $\min_B -\log P(X|B) + \lambda \cdot \|B\|_F^2$ 
    - closed form으로 정리되지 않으므로 iteration을 돌려 학습
- 결과적으로 validation recall@10이 EASE에 비해 0.02정도 낮게 나타남

- 결과 정리

- MovieLens 데이터의 경우 인기도에 기반한 추천이 다른 데이터에 비해 유효한 편 (EASE 논문 참고)
- 현재 데이터는 sequence 중간 중간 샘플링되어 test set으로 빠지면서 더욱 sparse한 형태

→ EASE와 같이 간단한 모델이 인기 영화(대세 패턴)를 잘 학습하고, sparse한 데이터에 잘 작동한 것으로 보임



## LRML

Metric Learning에 관계 벡터 개념을 도입한 implicit feedback collaborative ranking 추천 모델

### • 특징

- negative sample과 pairwise 데이터로 학습
- Netflix Prize, MovieLens20M과 같은 큰 데이터셋에서 더 좋은 성능을 보임
- L2 거리 기반 Metric Learning
- hinge loss를 통해, 충분히 학습한 샘플에 대해서는 학습하지 않음,  $L =$

$$\sum_{(p,q) \in \Delta} \sum_{(p',q') \notin \Delta} \max(0, s(p,q) + \lambda - s(p',q'))$$

### • 실험 결과

- L2 정규화 가중치에 따라, recall@10의 수렴값 차이가 큼.
- Negative sampling 방식에 따라 성능차이 발생.

### • 추가 시도

- Negative sampling 방식 변경

방식 1: Train의 positive sample의 개수 500만개 대비 4배 Negative sample을 cache에 저장하고 random으로 사용

- 장점: negative 샘플링 과정이, 학습 이전에 한번만 진행되어, 학습 속도가 빠름. epoch 당 40초.
- 단점: 방식 2보다 최종 loss가 높음. 학습 가능한 데이터의 부족이 원인.

방식 2: 매 mini batch마다 negative sample을 새롭게 sampling

- 장점: 매번 새로운 sample을 사용하기 때문에, 다양한 정보를 학습 가능하고 방식 1보다 성능이 좋음
- 단점: 방식 1에 비해 속도가 느림. epoch 당 2분.

최종적으로 모델의 성능 극대화를 위해, 방식 2 채택.

- embedding vector L2 norm Clipping 방식 변경

embedding weight에 clip을 적용하는 방식에서, vector에 clip을 적용하는 방식으로 변경. weight의 과적합은 L2 정규화를 통해 통제. → 약간의 성능 향상

### • 결과 정리

- 논문의 실험결과 대비, 낮은 성능을 보임. 기존 데이터에서 dense하게 가공을 했기 때문에, 강점을 보이지 못한 것 같음. 유사하게 FM과 같은 Embedding weight를 학습하는 모델들의 성능도 낮았음.



## ADMMSLIM

사용자-아이템 간의 희소한 행렬을 기반으로 한 협업 필터링(Collaborative Filtering) 기법

- 특징

- 희소성(Sparsity)을 활용한 정규화
- ADMM(Alternating Direction Method of Multipliers) 사용
- 사용자-아이템 간의 선형 관계를 학습하며, 다른 모델에 비해 상대적으로 간단한 구조를 가지고 있음

- 실험 결과

- Valid score 기준 0.24로 상당히 준수한 성능을 보임.
- 학습하는데 걸리는 시간도 짧아 다양한 시도를 해볼 수 있었음.
- 하이퍼 파라미터 튜닝은 모델 성능에 크게 영향을 미치지 못했음.

- 결과 정리 및 아쉬운 점

- 파라미터 튜닝으로 0.22~0.24 사이의 valid score가 나옴.
- 리더보드에서는 단일 모델 기준 가장 좋은 성능을 보임.
- 모델의 구조를 이해하여 개선을 시도해보지 못 한 점이 아쉬움.

## ADMMSLIM 파라미터 튜닝

Grid Search를 통해 하이퍼파라미터 튜닝

```
parameter_dict = {
    'learning_rate': 0.001,
    'lambda1': 3.0,
    'lambda2': 200.0,
    'alpha': 0.5,
    'rho': 5000.0,
    'k': 50,
    'positive_only': False,
    'center_columns': True
}
```

## 6) 앙상블

괜찮은 성능의 모델(EASE, ADMMSLIM, RECAE, MultiVAE, CDAE)과 sequential 모델(GRU4Rec) 앙상블 진행

### soft voting

- 단순 sum기반 Soft Voting: 모델의 predict probability를 더하는 방식 시도
  - 아쉬운점: 모델 선택에 따라 성능이 크게 변함, Best 모델의 성능 이상을 보여주지 못함
- 가중치 기반 Soft Voting: 모델별 성능에 따른 가중치를 부여한 Soft Voting 방식 시도
  - 아쉬운점: 제한된 제출 횟수로 인해 다양한 가중치 조합에 대한 충분한 실험은 수행하지 못함
- Model(EASE + ADMMSLIM)의 predict probability에 대한 후처리 적용 및 비교
  - 적용한 방법: Min-Max Scaling, Softmax

- 결과: 3가지 케이스(후처리 적용 X, Min-Max, softmax)에 성능차이가 미비함

## Hard Voting

- **Top100 Hard Voting:** 각 모델마다 유저별로 100개의 영화가 추천된 6개의 csv가 있는 상태에서 진행

### 1. count 기반 기본 하드 보팅

각 모델의 추천 결과에서 가장 많이 추천된 항목들을 10개씩 선택

### 2. ranking에 따른 가중치 첨가

각 모델의 추천 항목에서 랭크에 따라 가중치를 부여 → 유저별로 10개씩 선택

- 가중치 : 로그 스케일 가중치 사용, 최대 1.5로 제한

```
weight = 1 + math.log(total_items + 1 - rank)
weight = min(weight, 1.5)
```

- 상위 랭크에 더 높은 가중치
- 하위 랭크도 완전히 무시되지 않음
- 완만하고 자연스러운 가중치 감소

6개 모델 weighted hard voting 결과: 리더보드 0.0919

- **모델 순서 최적화:** A모델의 top5 + B모델의 unique items + A모델의 나머지로 채우는 방식 시도. 하지만 성능이 저하되었는데, 분석 결과 상위 4-5개 추천은 대부분의 모델이 유사한 아이템을 추천하고 있었음을 확인.
- **다중 모델 Top5 앙상블:** 여러 모델의 Top5에서 순차적으로 중복 제거하며 채우는 방식 시도. 하지만 유사한 유형의 모델이 많고, 시퀀셜 모델의 기여도가 낮아 다양한 관점의 추천이 부족했음.
- **Top-20 Voting:** 각 모델의 상위 20개 추천에 대해 투표 방식으로 앙상블 시도. 단일 모델과 유사한 성능을 보였으며, Top-k 값을 15, 10으로 변경해도 동일한 결과가 나와 실질적으로는 상위 10개 추천이 중요함을 확인.

**결과 분석** → 앙상블의 성능이 낮았던 이유 분석

- 선택된 대부분의 모델들이 유사한 collaborative filtering 기반
- 제출 횟수 제한으로 다양한 가중치 조합에 대한 충분한 실험 불가
- Min-Max Scaling, Softmax 등 후처리 방법들이 성능 향상에 큰 영향을 주지 못함

## 7) 최종 결과

No.	Model	Result
1	<b>ADMMSLIM 5-fold</b>	0.1622
2	EASE, ADMMSLIM, RecVAE	0.1613
3	<b>private score 기준 0.16 이상 Hard Voting</b>	0.1622
4	EASE, ADMMSLIM(3), RecVAE Soft Voting	0.1540
5	EASE, ADMMSLIM(3), RecVAE Soft Voting (Softmax)	0.1412
6	EASE, ADMMSLIM, GRU4REC (가중치 0.4, 0.4, 0.2)	0.1617

- 단순한 선형 모델(ADMMSLIM)이 가장 안정적이고 높은 성능
- 복잡한 앙상블 기법보다 단순한 Hard Voting이 더 효과적

## 추가로 시도해본 것들

최종 예측에 EDA 기반 가중치 부여

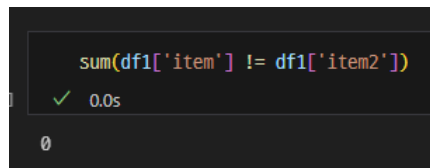
- EDA: 인기 많은 영화를 본 유저는 인기 많은 영화를 좋아한다

- 로직 순서

1. 아이템 별 상호작용 횟수를 기반으로 인기도 계산
2. 로그 스케일로 조정(지나치게 큰 영향 안 가게)
3. 사용자별 평균 인기도 계산
4. 사용자별 선호도를 정규화하여 가중치를 계산
5. 아이템의 인기도 점수에 이 가중치를 곱해 사용자 맞춤형 인기 보너스를 생성
6. 사용자가 이미 상호작용한 아이템은 제외
7. 기존 모델의 예측값에 인기 보너스를 가중치로 추가하여 최종 점수를 계산
8. 최종 점수에서 Top-K 아이템을 추천

- 결과

- 가중치를 부여하기 전과 후, 결과 변화가 없다.



item : 기존 예측, item2 : 인기도 반영 예측

- 모델이 이미 인기도를 충분히 고려하고 있다고 판단

## 자체 평가 의견

- 잘했던 점

- RecBole을 통해 다양한 추천 모델을 시도해봄
- 단순히 RecBole 사용에 그치지 않고 직접 모델을 구현함

- 시도하였으나 잘 되지 않았던 점

- 영화 정보를 반영할 수 있는 context-aware 모델을 발전시키고자 하였으나 전반적으로 낮은 성능으로 활용해보지 못함
- 여러 앙상블 기법을 시도해보았으나 성능 개선에 실패함

- 아쉬운 점

- 구현한 모델들을 하나로 합치지 못함
- 후반부에 제출 횟수가 부족하여 다양한 시도를 못 해봄 → 초반에 조금 더 하기

- 발전시킬만한 요소

- 모델에 따른 추천 결과를 분석하여 적합한 앙상블 기법 적용

## 2. 개인 회고

### 1) 강현구\_T7502

#### 초기 목표와 달성한 정도

벌써 4번째 프로젝트다. 처음에 어버버하며 GPU서버도 어케 연결하지 공공대던게 엇그제 같은데 엇그제긴 하다. 그래도 지나온 시간에 비하면 좋은 커리큘럼과 좋은 팀원들 덕분에 빠르게 걸음마를 댔 것 같다. 이번 초기목표는 논리적으로 생각하고 수행하는 것. 달성했나? 잘 모르겠다.

#### 내가 가장 신경 썼던 점

논리적으로 생각하고 수행하기.

RecBole을 다음에도 사용할 수 있도록 배우기.

## 구현하지 못한 아이디어

양상블기법에 시간을 많이 투자한 것치고 결과값이 아쉬웠다. 좀더 논리적으로 할 수 있을 것 같아 아쉬웠고, 다른 조들의 시퀀셜을 구현할때 다양한 논문을 적용하거나 더 한 차원 높이는 걸 보고 모델링쪽에 아쉬움이 많이 남는 것 같다.

## 한계와 아쉬웠던 점

내 인생에 생성형AI를 이제 뭘 수 있을까? 당장 겨우 이거 쓰는것도 AI도움받고있는데 앞으로 나아가면서 이걸 어떻게 해야할지 잘 모르겠다. AI가 있어서 성장곡선이 빨라진 것 맞지만 그럼에도 이게 항상 진짜 나는 아니라고 생각하기에 언젠가에 미래에 독이 될 걸 알기에 조심하며 사용해야 할 것 같다.

## 다음에 시도해볼 것

이젠 Kaggle형 프로젝트가 끝나서 뭘 시도 해야하지? 라는 질문에 뭐라 해야할지 모르겠다. 이제 프로젝트가 끝나고 1달동안 프로젝트가 없는만큼 물론 다른 공부할 게 엄청 많겠지만, 이론적기초나 좀 다져둘까 생각중이다.

## 마지막 한마디

프로젝트를 4개를 쉬는 시간없이 하다보니까 힘들긴했는데 돌아보니까 많이 배운 것 같다.

## 2) 서동준\_T7527

### 초기 목표와 달성한 정도

최종 모델의 성능과 모델 이해 둘 다 중요하게 생각하고 대회에 임했다. 초반에는 빠르게 recbole을 활용하여 높은 순위를 유지하다가, 후반부에 순위가 많이 떨어졌다. 다른 팀들과 사용한 모델에서는 큰 차이가 없었지만, 마지막에 하이퍼파라미터 튜닝과 양상블 기법에서 약간의 성능차이가 있었다. 프로젝트 후반부에는 모델의 이해를 우선으로 하고 각자 모델을 구현하는 시간을 가졌다. 때문에 양상블에서 좀 더 다양한 시도를 할 시간이 부족했다. 결국 둘 다 얻어가는 건 실패했지만, 배움을 목적으로 하는 프로젝트였기 때문에 높은 순위보단 모델에 대한 이해가 더 큰 수확이라고 생각한다.

### 내가 가장 신경 썼던 점

- 데이터의 성격을 정확하게 이해하는 것을 중요하게 생각했다.
  - sequential한 데이터와 static한 데이터를 동시에 예측해야 하는 특이한 Task였다. 하지만, 데이터셋의 timeStamp가 영화 시청 날짜가 아닌, 평가 날짜라는 점에서 Train 데이터의 연속성 정보에 noise가 많을 것으로 예측했다.
  - 결과적으로 sequential 모델의 성능이 낮았고, 양상블에서도 성능 향상을 보이지 않았다.
- 논문의 모델을 이해하고 직접 구현해보는 것을 가장 중요하게 생각했다.
  - LRML 모델 구현에 가장 많은 시간을 썼다. 특히 negative sampling의 성능과 효율성 개선에 집중했다.

### 구현하지 못한 아이디어

LRML 모델 구조를 개선해보는 작업을 시도해보지 못했다. LRML 모델 구현을 성공했지만, 기존의 EASE 모델 대비 낮은 성능을 보였다. 추가적인 작업을 통해 성능을 올린다고 해도, 이번프로젝트에서는 사용하기 어려운 성능이라, LRML 모델은 개선은 중단했다.

### 한계와 아쉬웠던 점

많은 시간을 투자해서 LRML 모델을 구현했지만, 성능이 생각보다 낮았다. 이번 데이터셋에서 임베딩 유사도 기반 모델들의 성능이 낮다는 것을 인자하고 처음부터 다른 모델을 고려했으면 좋았겠다는 아쉬움이 있다. 또한 구현 과정에서 체계가 부족했던 것에 대한 아쉬움이 있다. 중간 테스트 없이 모델을 구현하다 보니, 모델 성능의 문제가 구현상의 문제인지, 모델 자체의 문제인지 판단하기가 어려웠다. 멘토님의 조언대로 작은 데이터로 모델을 테스트하거나, train데이터에 과적합 후, 모델의 성능을 확인하는 방법 등으로 모델을 테스트 하는 과정이 필수적인 것 같다.

### 다음에 시도해볼 것

다음에는 pytorch lightning을 활용해보고 싶다. 모델 구현 단계에서 실수 하나하나가 구현 속도를 매우 늦춘다는 것을 알게되었다. 빠르게 모델을 구현하고 실험할 수 있도록 모델 구현 과정을 체계화해보고 싶다.

## 마지막 한마디

처음으로 데이터셋부터, 학습, 추론까지 직접 torch로 구현해본 프로젝트였다. 나름 경험치를 많이 쌓은 것 같아서 뿌듯하고 재밌었다.

## 3) 양시영\_T7534

### 초기 목표와 달성한 정도

초기 목표는 feature를 선택하거나 model을 선택할때 합리적인 근거에 기반한 선택을 하는 것이었다. 하지만 제한된 시간과 이것저것 다해보려고하는 욕심 때문에 일부분만 실천 할 수 있었다.

### 내가 가장 신경 썼던 점

이번 프로젝트에서 가장 신경쓴 부분은 적응이다. 기존에 있던 팀에 새로 합류하게된 만큼 지금까지 하던 방식이 아닌 이곳의 방식에 맞추는게 맞다고 생각해서 조금은 한걸음 물러서서 상황을 보거나 적응을 하려고 노력했다.

### 구현하지 못한 아이디어

처음 이 영화추천이라는 키워드를 들었을때 가장 먼저 생각난것은 구현한 모델을 이용해서 간단한 웹 서비스를 만들어볼까 였다. 이러한 생각을 하게된 이유 중 하나는 성능지표 0.01을 높이려고 노력을 하는것도 중요하지만 그 차이가 실제로 어떻게 느껴지는지만 들어서 비교체험을 해보면 더 좋지 않을까 생각했다.

### 한계와 아쉬웠던 점

정해진 기간과 리더보드 제출 횟수 등 제한된 상황으로인해 원하는 만큼 다양한 방법을 모두 다 해보지 못해서 아쉽고 좀 더 선택과 집중을 잘했으면 어땠을까하는 아쉬움이 있다.

### 다음에 시도해볼 것

초기에 하고싶은것에 대한 리스트업과 WBS를 잘 짜서 좀 더 특정 부분에 대한 집중도를 높이는 프로젝트를 하고싶다.

## 마지막 한마디

다음에는 어떻게 하면 더 좋을지 알게되는 기회였다. 다음 프로젝트는 더 힘내보자!

## 4) 이도걸\_T7540

### 초기 목표와 달성한 정도

다양한 모델 사용해보며 경험치 쌓기

- 실제 모델들을 다양하게 사용해보며 해당 Task에서 발전 가능성 있는 모델을 알아두기
- 다양한 모델을 사용해보았다.

### 내가 가장 신경 썼던 점

- 데이터의 특성을 파악하여 효율적인 계획 세우기
  - 처음 문제를 접했을 때와 달리 sequential 모델이 잘 작동하지 못하겠다고 생각했다.
  - 데이터가 영화를 본 시간이 아닌 영화에 대한 평가를 내린 시간이란 점이 데이터의 순서가 크게 영향을 주기는 어려웠다.
- ADMMSLIM 모델 성능 높이기
  - 처음 테스트로 모델을 돌려볼 때, 가장 좋은 성능을 보였다.

- 파라미터 튜닝을 통해 더 좋은 성능을 낼 수 있도록 노력했다.

### 구현하지 못한 아이디어

- 일반 Boosting 모델을 해당 Task에 적용했다면 어떤 결과가 나왔을까?
  - 학습과 pred 부분에 대한 코드 작성에서 어려움을 겪어 결과를 보지 못했다.
  - CatBoost와 같은 카테고리형 모델이 왠지 성능이 괜찮았을 것 같다.

### 한계와 아쉬웠던 점

- Recbole을 메인으로 사용해서 직접 모델 코드를 작성하지 못한 점이 아쉽다.
  - 실제 모델에 대해 깊게 파고들며 공부하지 못했다.
  - 파라미터 튜닝을 제외하고는 성능을 높일 방법을 탐색하지 못했다.

### 다음에 시도해볼 것

- 성능이 잘 나오는 Recbole 모델을 바탕으로 해당 모델을 발전시키는 작업을 해보고 싶다.
- Recbole처럼 한 라이브러리 내에서 다양한 시도를 해볼 수 있는 Baseline 코드를 작성해보고 싶다.

### 마지막 한마디

4번의 대회 동안 많이 성장했을까...

## 5) 이수미\_T7541

### 초기 목표와 달성한 정도

좋은 성능과 모델 공부 두 가지를 목표로 하였다.

- RecBolt 코드로 학습만 시키지 말고 하나의 모델이라도 제대로 이해하고자 하였다.
- 모델 구현에 시간을 많이 썼기 때문에 하이퍼파라미터 튜닝과 앙상블에 시간을 적게 투자했고, 이로 인해 좋은 성능이 나오지 못한 것 같다.

### 내가 가장 신경 썼던 점

MultiVAE 공부와 구현

- VAE부터 시작해서 MultiVAE 논문도 찾아서 읽었다.
- 구현도 했으나 끝까지 하자가 있었다. 내 코드는 대체 어디가 잘못된 것인가...

### 구현하지 못한 아이디어

오류 찾다가 구조 개선을 못해봤다...

### 한계와 아쉬웠던 점

- 초반부에 프로젝트에 대한 감이 잡히지 않아 하면서 정하자~ 했던 부분이 많다. 그럼에도 불구하고 목표를 확실하게 세웠더라면 우왕좌왕 하는 시간이 줄일 수 있을 것 같다.
- MultiVAE에 꽂혀서 다른 것들에 신경을 거의 쓰지 못했다. 빠른 구현이 이루어지지도 않았고 끝까지 test dataset에 대한 예측 성능이 떨어져서 뭐 하나 제대로 해낸 느낌이 들지 않는다. 며칠 동안 구현 코드 문제해결만 했던 것이 대회형 프로젝트에는 적합하지 않았던 것 같다.
- 1등 팀을 보니, 20개만 예측해서 앙상블을 진행하였다. 우리 팀은 100개로 진행하였는데 20개로 했으면 비슷한 성능을 가질 수 있었는지 궁금하다.

### 다음에 시도해볼 것

- 멘토님께서 추천해주신 파이토치 라이트닝을 사용해보고 싶다.



- 계획적인 프로젝트 진행도 하고 싶다.

## 마지막 한마디

하나 남았다.....

## 6) 최윤희\_T7549

### 초기 목표와 달성한 정도

- 직접 모델 구현해보기
  - EASE와 EASE에서 파생시킨 Multi-EASE 모델을 직접 구현해봤다. Multi-EASE 모델의 경우, 기대했던 성능이 나오지 않았고 이에 대한 원인을 제대로 파악하지는 못한 점이 다소 아쉽다.

### 내가 가장 신경 썼던 점

- 논리적으로 분석 방향성 설정하기
  - 영화의 제목 유사성과 사용자군 사이의 연관성을 EDA를 통해 확인한 후, context를 잘 활용해보고자 RecBole에서 context-aware 모델을 담당하였다.
  - 이 후, context 모델의 성능이 현저히 떨어지는 것을 확인하고, 현재 높은 성능을 내고 있는 EASE 모델의 논문을 보며 EASE 모델이 잘 나오는 이유에 대해 탐색했다.
  - 논문에서 언급된 multinomial loss를 적용해보기 위해 직접 EASE 및 Multi-EASE 모델을 구현하여 성능을 확인하였다.

### 구현하지 못한 아이디어

- 영화 제목 임베딩 변수 활용
  - 시리즈물과 같이 제목 유사성이 높은 영화의 경우, 평점을 부여한 사용자군도 유사하다는 것을 EDA를 통해 파악하였다. 이를 기반으로 제목에 대한 임베딩 값을 context로 사용하여 DeepFM과 같은 모델에 적용해보고자 하였다. 하지만 기본적으로 너무 낮은 모델의 성능과 복잡한 RecBole 구조로 인해 들어가는 자원 대비 기대되는 아웃풋은 높지 않다고 판단하여 직접 시도해보지 못하고 넘어갔다.

### 한계와 아쉬웠던 점

RecBole이 있어서 편했지만 그만큼 제대로 된 분석보다는 모델을 돌리고 결과 확인하는 수준의 가벼운 프로젝트를 했다는 생각이 든다. 개인적으로 EASE 모델을 구현해보기는 했지만, 이것 또한 시간 부족으로 결과가 잘 나오는 것만 확인하고 넘어간 것 같다. 단순히 모델의 성능을 보고 사용 여부를 결정하기 보다는 모델의 추천 결과와 현 데이터의 특징들을 함께 분석해서 새로운 개선 방향을 도출해내는 식의 분석을 했다면 어땠을까하는 아쉬움이 남는다.

### 다음에 시도해볼 것

당분간 좀 쉬어야겠습니다..

## 마지막 한마디

힘들었지만 재밌었습니다!

