

# [KUBIG CONFERENCE] 아가야 ~ 뭐 보고싶어 ~?

2022 LG 유플러스 AI Ground - 아이들나라 콘텐츠 추천 AI

15기 남정재	16기 박종혁
16기 신인섭	16기 유우혁
16기 임채명	16기 정은미

# Project Overview

---

**2022 유플러스 AI Ground**

## # Competition

# LG U+ AI Ground



The poster for the 2022 LG U+ AI Ground competition features a central title '2022 유플러스 AI Ground' with a subtitle '유플러스 아이들나라 콘텐츠 추천 AI'. It is surrounded by various icons representing AI, creativity, and technology. The bottom section contains detailed information about the competition, including participation dates, eligibility, schedule, prizes, and contact information.

**2022 유플러스 AI Ground**  
유플러스 아이들나라 콘텐츠 추천 AI

- 참가 접수** 2022. 10. 10(월) - 10. 31(월)
- 지원 자격** AI 머신러닝/딥러닝 모델링이나 개발에 관심있는 내국인이자 누구나 참여 가능
- 대회 일정**
  - 대회 참여 기간 (4주)
  - 코드 및 발표 자료 제출
  - 온라인 발표 및 Q&A
  - 최종 수상팀 발표
- 상금 및 특전** 총 1,000만원 상당의 상금  
1등 500만원(1팀), 2등 200만원(1팀), 3등 각 100만원(3팀)  
(시상금은 수상팀 대표에게 지급되며, 세금은 주최측에서 부담합니다.)  
LG 유플러스 신입사원 채용 시 혜택 부여 (최종 리더 보드 상위 10개 팀)
- 주최** LG U+
- 주관** upstage
- 문의사항** ai-edu@upstage.ai
- 모집페이지** <https://github.com/UpstageAI/2022-iguplus-AI-Ground>

QR Code

## # Competition

■ ■ ■

LG U+  
AI Ground

이제 우리 아이 맞춤형 콘텐츠를 추천받을 수 있습니다

아래 선택하신 관심사 기반 콘텐츠는 홈에서 돌아가면서 1개씩 볼 수 있어요.



만들기



숫자/계산



친구/사람



자연탐구



음악예술



수리논리



검색



마이메뉴



생생놀이교실



캐릭터



형빈의 홈



책 읽어주는 TV



영어 유치원



에그스쿨



누리 학습



YouTube

형빈의 관심사



부모님이 선택한 주제



추천 영상



Let's take a look at ...

## Data Overview

No.	구분	파일명	상세 설명
1	데이터	history_data.csv	시청 시작 데이터
2	데이터	watch_e_data.csv	시청 종료 데이터
3	데이터	buy_data.csv	구매 이력 데이터
4	데이터	search_data.csv	검색을 통한 시청 데이터
5	데이터	meta_data.csv	콘텐츠 일반 메타 정보
6	데이터	meta_data_plus.csv	콘텐츠 확장 정보
7	데이터	profile_data.csv	프로필 정보
8	데이터	sample_submission.csv	제출 양식 데이터

## 1) **history\_data.csv** : 시청 시작 데이터

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	profile_id	프로필ID	-	프로필 고유 ID값
2	ss_id	세션 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 세션 시간
3	log_time	행동 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 시간
4	act_target_dtl	행동 분류	MKID + ###	MKID003 : 시청 시작
5	album_id	앨범ID	-	앨범 고유 ID 값
6	payment	지불 금액	원	콘텐츠를 보기 위해 지불한 금액
7	continuous_play	연속 재생 여부	Y/N	연속 재생을 통해 시청한 여부
8	short_trailer	예고편 여부	Y/N	예고편용 콘텐츠 여부

## 2) watch\_e\_data.csv : 시청 종료 데이터

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	profile_id	프로필ID	-	프로필 고유 ID값
2	ss_id	세션 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 세션 시간
3	log_time	행동 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 시간
4	act_target_dtl	행동 분류	MKID + ###	MKID049 : 시청 종료
5	album_id	앨범ID	-	앨범 고유 ID 값
6	watch_time	실제 시청 시간	초	콘텐츠를 실제 시청한 시간
7	total_time	콘텐츠 길이	초	콘텐츠 전체 길이
8	continuous_prev	연속 재생 여부	0/1/2/3	0: 이후 콘텐츠 재생 없음 1: 연속재생 2: 선택에 의한 종료 3: 기타 종료

### 3) **buy\_data.csv** : 구매 이력 데이터

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	profile_id	프로필ID	-	프로필 고유 ID값
2	ss_id	세션 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 세션 시간
3	log_time	행동 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 시간
4	act_target_dtl	행동 분류	MKID + ###	MKID004 : 구매 이력
5	album_id	앨범ID	-	앨범 고유 ID 값
6	payment	지불 금액	원	콘텐츠에 지불한 금액

### 4) **search\_data.csv** : 검색을 통한 시청 데이터

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	profile_id	프로필ID	-	프로필 고유 ID값
2	ss_id	세션 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 세션 시간
3	log_time	행동 시간	YYYYMMDDHHMMSS	해당 이력이 적재된 시간
4	act_target_dtl	행동 분류	MKID + ###	MKID017 : 검색을 통한 시청
5	album_id	앨범ID	-	검색을 통해 시청한 앨범ID



## 5) meta\_data.csv : 콘텐츠 일반 메타 정보

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	album_id	앨범ID	-	앨범 고유 ID 값
2	title	제목	-	앨범 제목
3	sub_title	부제	-	해당 앨범이 속한 카테고리의 상세 명칭
4	genre_large	대분류 장르	-	앨범이 속한 대분류 장르
5	genre_mid	중분류 장르	-	앨범이 속한 중분류 장르
6	genre_small	소분류 장르	-	앨범이 속한 소분류 장르
7	country	국가	-	앨범 제작 국가
8	run_time	콘텐츠 길이	초	콘텐츠 전체 길이
9	onair_date	방영 날짜	YYYYMMDD	콘텐츠가 방영된 날짜
10	cast_1	출연 캐릭터 1	-	출연 캐릭터
11	cast_2	출연 캐릭터 2	-	출연 캐릭터
12	cast_3	출연 캐릭터 3	-	출연 캐릭터
13	cast_4	출연 캐릭터 4	-	출연 캐릭터
14	cast_5	출연 캐릭터 5	-	출연 캐릭터
15	cast_6	출연 캐릭터 6	-	출연 캐릭터
16	cast_7	출연 캐릭터 7	-	출연 캐릭터

## 6) **profile\_data.csv** : 프로필 정보

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	profile_id	프로필ID	-	프로필 고유 ID값
2	age	연령	-	프로필에 해당하는 사용자 나이
3	sex	성별	-	F : 여성, M : 남성
4	pr_interest_keyword_cd_1	부모 관심 키워드 코드 1	P + ##	P01: 과학기술 P02: 정서/사회성 P03: 자연탐구 P04: 바른생활/안전 P05: 활동/운동 P06: 음악예술 P07: 언어논리 P08: 수리논리
5	pr_interest_keyword_cd_2	부모 관심 키워드 코드 2	P + ##	위와 동일
6	pr_interest_keyword_cd_3	부모 관심 키워드 코드 3	P + ##	위와 동일
7	ch_interest_keyword_cd_1	아이 관심 키워드 코드 1	K + ##	K01: 노래/유희 K02: 동물/식물 K03: 동화 K04: 만들기 K05: 숫자/계산 K06: 외국어 K07: 친구/사람 K08: 탈 것/기계 K09: 활동/운동
8	ch_interest_keyword_cd_2	아이 관심 키워드 코드 2	K + ##	위와 동일
9	ch_interest_keyword_cd_3	아이 관심 키워드 코드 3	K + ##	위와 동일

## 7) meta\_data\_plus.csv : 콘텐츠 확장 정보

No	칼럼명 (영문)	칼럼명 (한글)	단위	상세 정의
1	album_id	앨범ID		앨범 고유 ID값
2	keyword_type	태그 코드		태그 코드 고유 값
3	keyword_name	태그 명	-	태그 코드 명
4	keyword_value	태그가 어울리는 정도	0, 1, 2, 3, 4, 5 정수값	0, 1, 2, 3, 4, 5

## 8) sample\_submission.csv : 리더보드 제출 양식

No	칼럼명 (영문)	칼럼명 (한글)	상세 정의
1	profile_id	프로필ID	프로필 고유 ID값
2	predicted_list	예측 리스트	예측값 (리스트)

# Recommender System

---

## Neural Collaborative Filtering

# # Recommender System

■ ■ ■

# Neural Collaborative Filtering

## Neural Collaborative Filtering

Xiangnan He  
National University of  
Singapore, Singapore  
xiangnanhe@gmail.com

Liqliang Nie  
Shandong University  
China  
nieliqliang@gmail.com

Lizi Liao  
National University of  
Singapore, Singapore  
liaolizi.llz@gmail.com

Xia Hu  
Texas A&M University  
USA  
hu@cse.tamu.edu

Hanwang Zhang  
Columbia University  
USA  
hanwangzhang@gmail.com

Tat-Seng Chua  
National University of  
Singapore, Singapore  
dcscts@nus.edu.sg

1v2 [cs.IR] 26 Aug 2017

## ABSTRACT

In recent years, deep neural networks have yielded immense success on speech recognition, computer vision and natural language processing. However, the exploration of deep neural networks on recommender systems has received relatively less scrutiny. In this work, we strive to develop techniques based on neural networks to tackle the key problem in recommendation — collaborative filtering — on the basis of implicit feedback.

Although some recent work has employed deep learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items and acoustic features of musics. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items.

By replacing the inner product with a neural architecture

## 1. INTRODUCTION

In the era of information explosion, recommender systems play a pivotal role in alleviating information overload, having been widely adopted by many online services, including E-commerce, online news and social media sites. The key to a personalized recommender system is in modelling users' preference on items based on their past interactions (*e.g.*, ratings and clicks), known as collaborative filtering [31, 46]. Among the various collaborative filtering techniques, matrix factorization (MF) [14, 21] is the most popular one, which projects users and items into a shared latent space, using a vector of latent features to represent a user or an item. Thereafter a user's interaction on an item is modelled as the inner product of their latent vectors.

Popularized by the Netflix Prize, MF has become the *de facto* approach to latent factor model-based recommendation. Much research effort has been devoted to enhancing MF, such as integrating it with neighbor-based models [21],

## # Recommender System

# Neural Collaborative Filtering

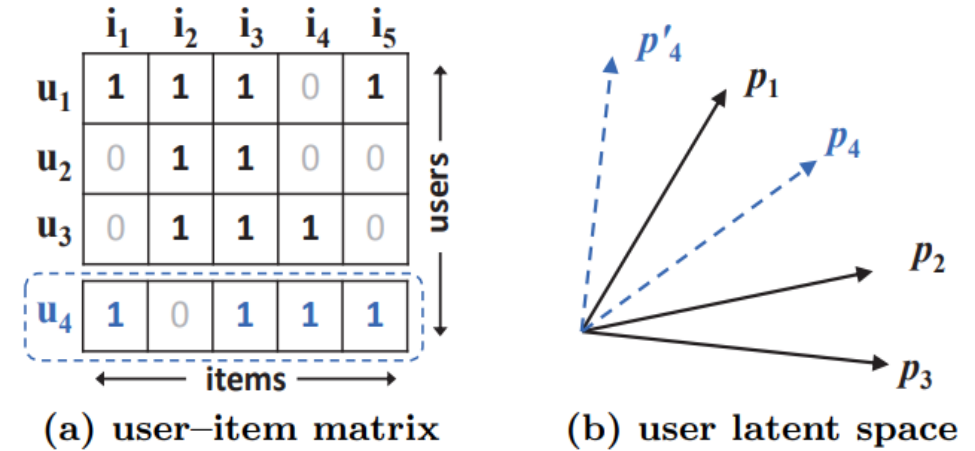


Figure 1: An example illustrates MF's limitation. From data matrix (a),  $u_4$  is most similar to  $u_1$ , followed by  $u_3$ , and lastly  $u_2$ . However in the latent space (b), placing  $p_4$  closest to  $p_1$  makes  $p_4$  closer to  $p_2$  than  $p_3$ , incurring a large ranking loss.

## # Recommender System

# Neural Collaborative Filtering

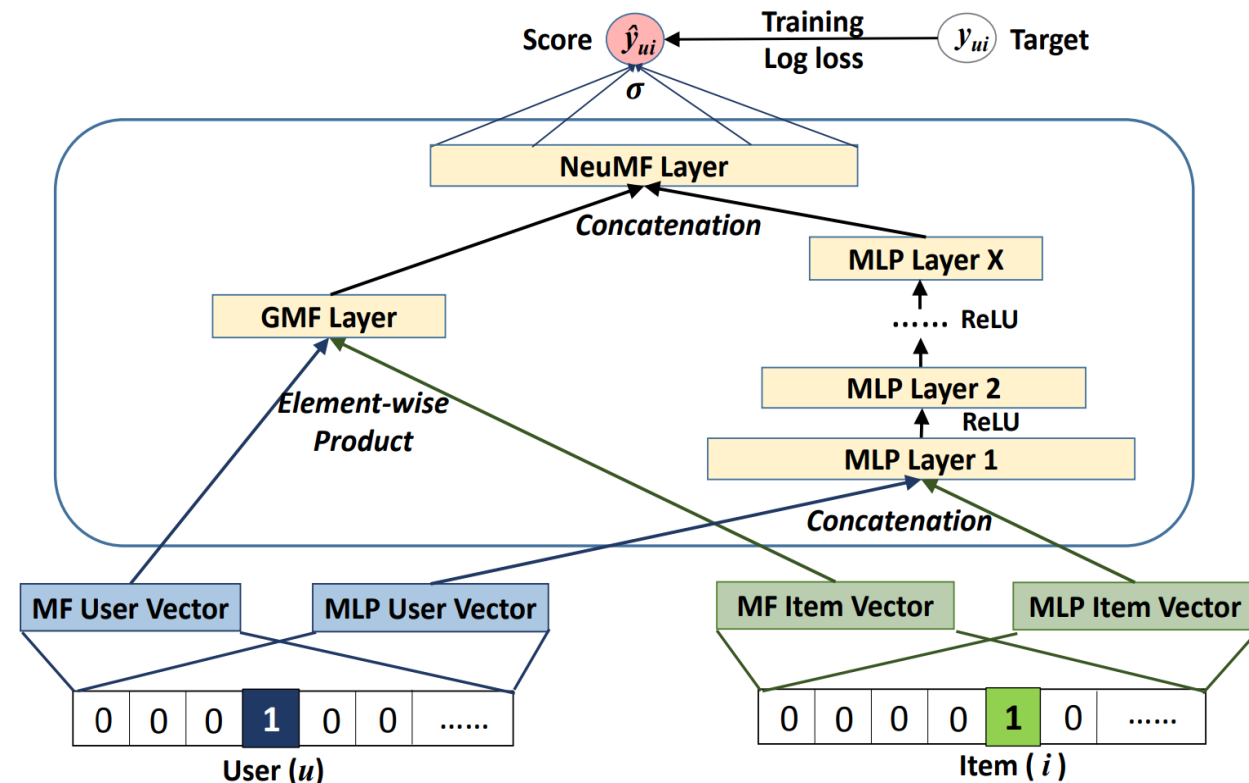


Figure 3: Neural matrix factorization model

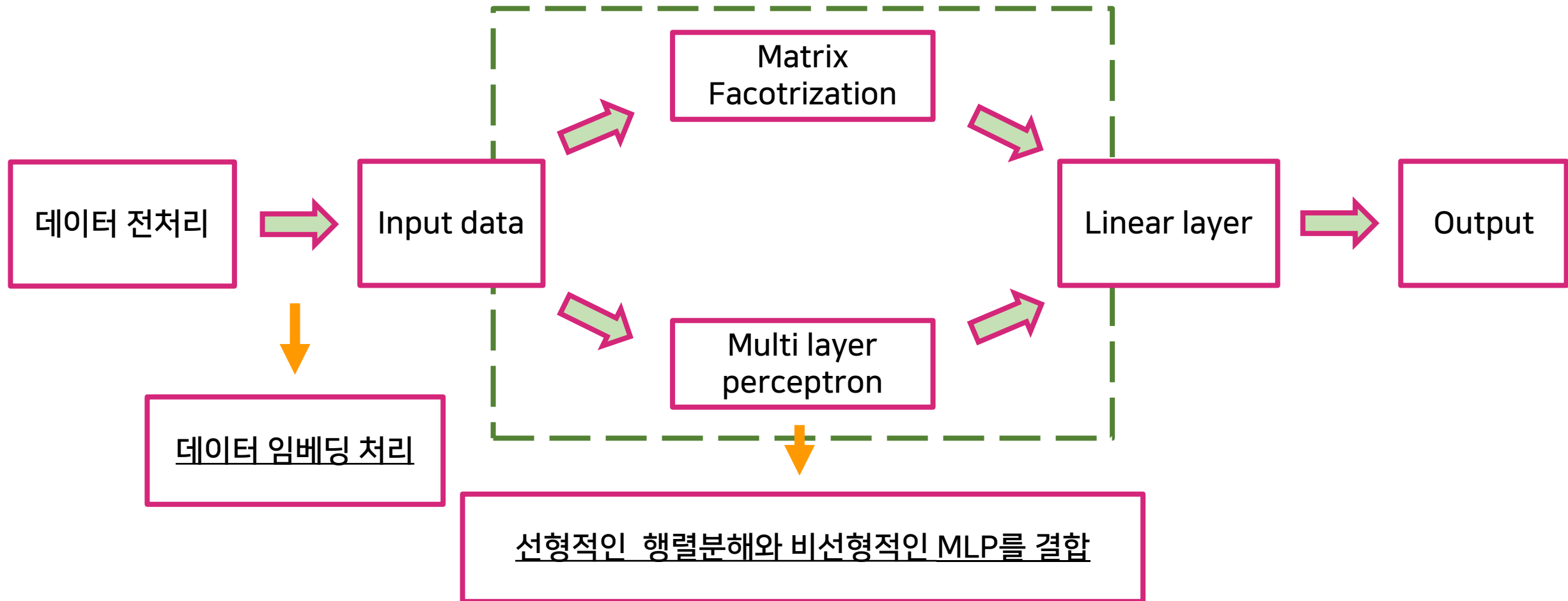
$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G,$$

$$\phi^{MLP} = a_L(\mathbf{W}_L^T(a_{L-1}(\dots a_2(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + \mathbf{b}_2)\dots) + \mathbf{b}_L)),$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix}),$$

- $p_u^G$  : User embedding for GMF
- $p_u^M$  : User embedding for MLP
- $q_i^G$  : Item embedding for GMF
- $q_i^M$  : Item embedding for MLP
- User-item latent 구조를 모델링 하기 위해 Linearity of MF와 non-linearity of DNN 결합한 모델

## 전체 파이프라인 개요



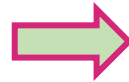


# 전체 파이프라인 개요

## 1. 시청 기록 기반 user item matrix 생성

	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$
$u_1$	1	1	1	0	1
$u_2$	0	1	1	0	0
$u_3$	0	1	1	1	0
$u_4$	1	0	1	1	1

(a) user-item matrix



작품 시청 여부를 기반으로  
User-item matrix에 0, 1을 대입

## 2. Negative sampling 기반 피처 추출

Negative 비율로 시청 여부 데이터 개수 조절



이 때,

User, item 데이터에서 어떤 피처를 사용할지  
정하는 것이 중요함!

연속 시청 여부와 같이 item & user에 모두  
적용되는 피처를 어떻게 사용할지!

조절한 아이템 개수만큼  
user, item feature 저장

데이터 임베딩 처리하기



# TEAM 정재/종혁/채명

## Project Overview

Recommendation system 기초 및 기본 공부

추천시스템 모델 트렌드 동향 파악 & SOTA 모델 및 논문 서칭 및 내용 파악



Neural Collaborative Filtering(NeuMF), Autorec(Autoencoder),  
Pytorch-TorchRec(torchrec.models.deepfm, torchrec.models.dlrm) 등



최근 추천시스템의 동향으로 Autoencoder 기반의 Autorec이라는 모델이 강력한 성능을 지닌 추천 모델로 파악되었음

But,

- ✓ Autorec의 경우, item들은 모두 user의 embedding을 찾기 위해 사용될 뿐, item 자체에 대한 embedding을 찾지 않음  
>> Autorec 계열은 단 한 개의 item이 추가되더라도 학습을 모두 다시 해야 하는 단점
- ✓ NeuMF의 경우, user와 item 모두 embedding을 찾도록 설계되어 있음  
>> 즉, 새로운 콘텐츠가 추가 되었을 때 새로운 item에 대한 embedding을 찾을 수 있다면,  
바로 제작된 모델에 넣어 각 user별로 새로운 item에 대한 선호도를 확인 가능!

>> 따라서 NeuMF 계열의 모델에 적용할 Feature Engineering에 대한 부분과 모델을 구성하고 있는 Architecture에 대해 집중

# TEAM 정재/종혁/채명

## Data pre-processing + Feature Engineering

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 지불금액 >> 결측값 zero
- ✓ 연속재생여부
- ✓ 예고편여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동 시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 실제시청시간 >> 결측값 drop
- ✓ 콘텐츠길이
- ✓ 연속재생여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ Age
- ✓ Sex
- ✓ 부모관심키워드코드1~3
- ✓ 아이관심키워드코드1~3
- ✓ hour >> 시청 시작 데이터 가장 많이 본 시간대 상위 3가지
- ✓ 프로필ID별 총지불금액(구매데이터)
- ✓ 프로필ID별 검색한 album\_id 상위 3가지(검색데이터)

- ✓ 앨범ID
- ✓ 대&중 분류 장르
- ✓ 콘텐츠길이
- ✓ hour >> 시청 시작 데이터 가장 많이 시청된 시간대 상위 3가지
- ✓ 앨범ID별 구매한 사용자의 수(구매데이터)
- ✓ 앨범ID별 검색한 횟수 count(검색데이터)
- ✓ 앨범ID기준으로 meta\_data\_plus 합치기(album\_id갯수=title갯수)

시청 시작  
데이터

시청 종료  
데이터

사용자 기반  
프로필 ID

콘텐츠 기반  
앨범 ID

# TEAM 정재/종혁/채명

## Data pre-processing + Feature Engineering

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 지불금액 >> 결측값 zero
- ✓ 연속재생여부
- ✓ 예고편여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동 시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 실제시청시간 >> 결측값 drop
- ✓ 콘텐츠길이
- ✓ 연속재생여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ Age
- ✓ Sex
- ✓ 부모관심키워드코드1~3
- ✓ 아이관심키워드코드1~3
- ✓ hour >> 시청 시작 데이터 가장 많이 본 시간대 상위 3가지
- ✓ 프로필ID별 총지불금액(구매데이터)
- ✓ 프로필ID별 검색한 album\_id 상위 3가지(검색데이터)

- ✓ 앨범ID
- ✓ 대&중 분류 장르
- ✓ 콘텐츠길이
- ✓ hour >> 시청 시작 데이터 가장 많이 시청된 시간대 상위 3가지
- ✓ 앨범ID별 구매한 사용자의 수(구매데이터)
- ✓ 앨범ID별 검색한 횟수 count(검색데이터)
- ✓ 앨범ID기준으로 meta\_data\_plus 합치기(album\_id갯수=title갯수)

시청 시작  
데이터

시청 종료  
데이터

사용자 기반  
프로필 ID

콘텐츠 기반  
앨범 ID

# TEAM 정재/종혁/채명

## Data pre-processing + Feature Engineering

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 지불금액 >> 결측값 zero
- ✓ 연속재생여부
- ✓ 예고편여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동 시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 실제시청시간 >> 결측값 drop
- ✓ 콘텐츠길이
- ✓ 연속재생여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ Age
- ✓ Sex
- ✓ 부모관심키워드코드1~3
- ✓ 아이관심키워드코드1~3
- ✓ hour >> 시청 시작 데이터 가장 많이 본 시간대 상위 3가지
- ✓ 프로필ID별 총지불금액(구매데이터)
- ✓ 프로필ID별 검색한 album\_id 상위 3가지(검색데이터)

- ✓ 앨범ID
- ✓ 대&중 분류 장르
- ✓ 콘텐츠길이
- ✓ hour >> 시청 시작 데이터 가장 많이 시청된 시간대 상위 3가지
- ✓ 앨범ID별 구매한 사용자의 수(구매데이터)
- ✓ 앨범ID별 검색한 횟수 count(검색데이터)
- ✓ 앨범ID기준으로 meta\_data\_plus 합치기(album\_id갯수=title갯수)

시청 시작  
데이터

시청 종료  
데이터

사용자 기반  
프로필 ID

콘텐츠 기반  
앨범 ID

# TEAM 정재/종혁/채명

## Data pre-processing + Feature Engineering

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 지불금액 >> 결측값 zero
- ✓ 연속재생여부
- ✓ 예고편여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동 시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ 앨범ID
- ✓ 실제시청시간 >> 결측값 drop
- ✓ 콘텐츠길이
- ✓ 연속재생여부
- ✓ hour >> 아이들이 활동하는 시간대 카테고리별로 나눠서 적용

\*\*\* 행동시간이 세션시간보다 빠른 경우 drop

- ✓ 프로필ID
- ✓ Age
- ✓ Sex
- ✓ 부모관심키워드코드1~3
- ✓ 아이관심키워드코드1~3
- ✓ hour >> 시청 시작 데이터 가장 많이 본 시간대 상위 3가지
- ✓ 프로필ID별 총지불금액(구매데이터)
- ✓ 프로필ID별 검색한 album\_id 상위 3가지(검색데이터)

- ✓ 앨범ID
- ✓ 대&중 분류 장르
- ✓ 콘텐츠길이
- ✓ hour >> 시청 시작 데이터 가장 많이 시청된 시간대 상위 3가지
- ✓ 앨범ID별 구매한 사용자의 수(구매데이터)
- ✓ 앨범ID별 검색한 횟수 count(검색데이터)
- ✓ 앨범ID기준으로 meta\_data\_plus 합치기(album\_id갯수=title갯수)

시청 시작  
데이터

시청 종료  
데이터

사용자 기반  
프로필 ID

컨텐츠 기반  
앨범 ID

# TEAM 정재/종혁/채명

Feature 추출 및 학습 적용 (학습 및 추론에 필요한 feature)

```
# feature 추출
features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features1['sex'][user_id])
UIdataset[user_id].append(np.array(features))

features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features2['age'][user_id])
UIdataset[user_id].append(np.array(features))

features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features3['pr_interest_keyword_cd_1'][user_id])
UIdataset[user_id].append(np.array(features))

features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features4['pr_interest_keyword_cd_2'][user_id])
UIdataset[user_id].append(np.array(features))

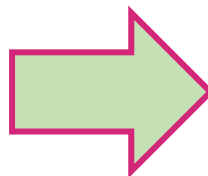
features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features5['pr_interest_keyword_cd_3'][user_id])
UIdataset[user_id].append(np.array(features))

features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features6['ch_interest_keyword_cd_1'][user_id])
UIdataset[user_id].append(np.array(features))

features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features7['ch_interest_keyword_cd_2'][user_id])
UIdataset[user_id].append(np.array(features))

features = []
for item_id in np.concatenate([pos_item_ids, neg_item_ids]):
    features.append(user_features8['ch_interest_keyword_cd_3'][user_id])
UIdataset[user_id].append(np.array(features))
```

배치 데이터로 변환



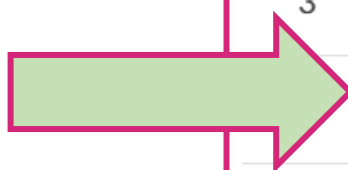
```
batch_feat6 = []
batch_feat7 = []
batch_feat8 = []
batch_feat9 = []
batch_labels = []
for user_id in batch_user_indices:
    item_ids = UIdataset[user_id][0]
    feat0 = UIdataset[user_id][1]
    feat1 = UIdataset[user_id][2]
    feat2 = UIdataset[user_id][3]
    feat3 = UIdataset[user_id][4]
    feat4 = UIdataset[user_id][5]
    feat5 = UIdataset[user_id][6]
    feat6 = UIdataset[user_id][7]
    feat7 = UIdataset[user_id][8]
    feat8 = UIdataset[user_id][9]
    feat9 = UIdataset[user_id][10]
    labels = UIdataset[user_id][11]
    user_ids = np.full(len(item_ids), user_id)
    batch_user_ids.extend(user_ids.tolist())
    batch_item_ids.extend(item_ids.tolist())
    batch_feat0.extend(feat0.tolist())
    batch_feat1.extend(feat1.tolist())
    batch_feat2.extend(feat2.tolist())
    batch_feat3.extend(feat3.tolist())
    batch_feat4.extend(feat4.tolist())
    batch_feat5.extend(feat5.tolist())
    batch_feat6.extend(feat6.tolist())
    batch_feat7.extend(feat7.tolist())
    batch_feat8.extend(feat8.tolist())
    batch_feat9.extend(feat9.tolist())
    batch_labels.extend(labels.tolist())
return batch_user_ids, batch_item_ids, batch_feat0, batch_feat1,
```

# TEAM 정재/종혁/채명

리더보드 제출 결과 및 최종 결과

이름	진행 단계	SCORE (Rank)	Recall@25	NDCG@25
7	Finished	0.2082	0.2124	0.1956
6	Finished	0.2079	0.2122	0.1951
5	Finished	0.2036	0.2080	0.1904
4	Finished	0.2029	0.2075	0.1893
3	Finished	0.2169	0.2216	0.2028
2	Finished	0.2159	0.2205	0.2020
1	Finished	0.2169	0.2213	0.2035










이름	SCORE (Rank)	Recall@25	NDCG@25
7	0.2082 → 0.1576	0.2124 → 0.1627	0.1956 → 0.1423
6	0.2079 → 0.1580	0.2122 → 0.1632	0.1951 → 0.1424
5	0.2036 → 0.1548	0.2080 → 0.1600	0.1904 → 0.1394
4	0.2029 → 0.1534	0.2075 → 0.1586	0.1893 → 0.1380
3	0.2169 → 0.1667	0.2216 → 0.1722	0.2028 → 0.1499
	0.2159 → 0.1657	0.2205 → 0.1714	0.2020 → 0.1488
1	0.2169 → 0.1658	0.2213 → 0.1714	0.2035 → 0.1490





# TEAM 정재/종혁/채명

리더보드 제출 결과 및 최종 결과

					이름	SCORE (Rank)	Recall@25	NDCG@25	
이름	진행 단계	SCORE	Recall@25	NDCG@25					
55	김준태_T2058					0.1667	0.1724	0.1496	11
56	sudokim					0.1667	0.1725	0.1494	7
57	식사는 잡셨어?				  	0.1667	0.1722	0.1499	10
58	베리				 현수 1	0.1665	0.1721	0.1497	5
59	Incheol				 인철  한결  WHYHOW	0.1665	0.1724	0.1486	28

# TEAM 우혁/인섭/은미

ITEM feature >>> 출연자 변수 사용

송년특집으로 방송된 가수 임영웅콘서트가 시청률 16%로 지상파 동시간대 1위를 기록했다.

27일 시청률 조사회사 닐슨코리아 집계 결과, 전날 오후 9시 30분부터 11시 38분까지 방송된 KBS2 '위아 히어로 임영웅'(We're HERO 임영웅) 시청률은 16.1%로 나타났다.



여기서!

뽀롱뽀롱 뽀로로 [1TV] 월,화 오전 7시10분(본)

뽀로로 동화나라 [1TV] 금 오전 7시30분 (본) [1TV] 금 오전 7시30분(본)

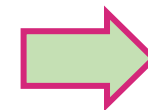
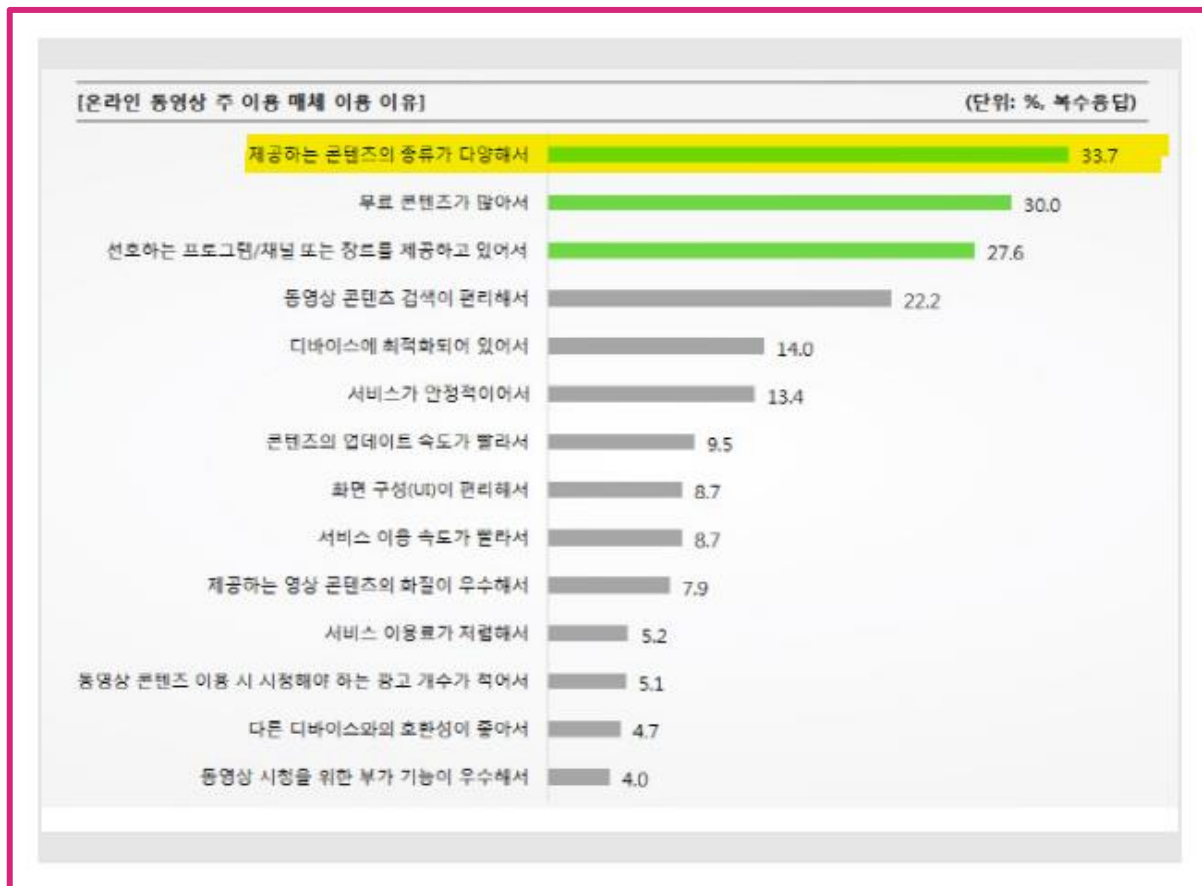
뽀로로와 노래해요 [1TV] 월,화 오전 9시35분(본)

**뽀로로만 해도 여러가지 작품이 존재!!**  
 > 제목보다는 출연자가 중요하다고 생각

아동용 프로그램에 출연하는 캐릭터에 따라 시청 여부가 크게 변할 수 있다고 판단  
 -> cast변수를 label encoding하여 캐릭터들을 구분

# TEAM 우혁/인섭/은미

ITEM feature >>> 장르 / 작품 키워드 변수 사용



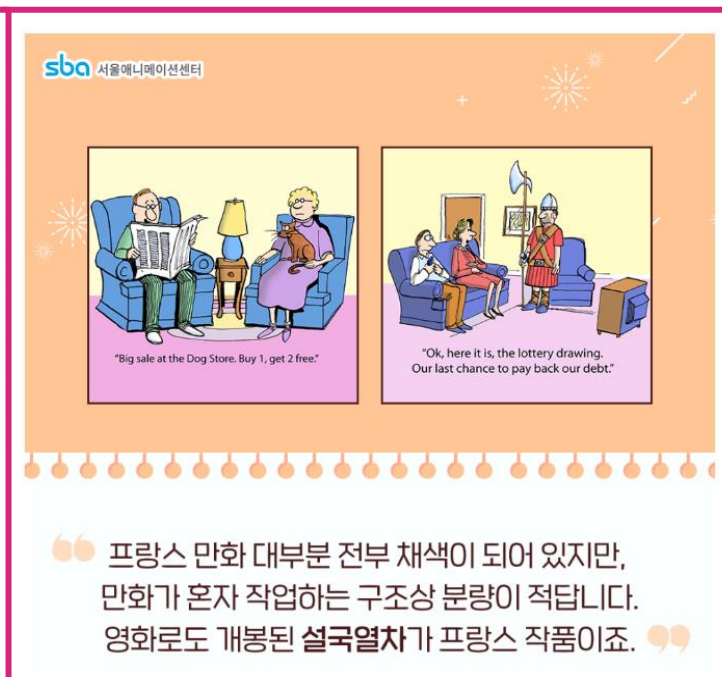
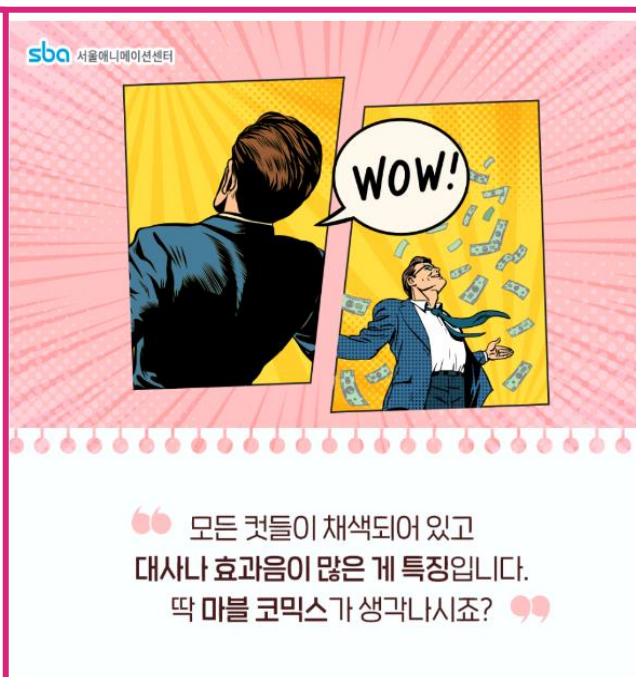
본인이 어떤 취향의 작품을 좋아하는지가 프로그램을  
선택하는데 큰 영향을 미친다고 생각!

>>>

장르 변수와 작품 키워드 변수를 label encoding처리

# TEAM 우혁/인섭/은미

ITEM feature >>> 국가 변수 사용



국가별 만화 차이 존재 >> 애니메이션 특성 변화 가능성 존재 >> 국가변수 label-encoding 진행

# TEAM 우혁/인섭/은미

ITEM feature >>> 시청 시간 / 비용 변수 사용

## 방통위, 2021년 N스크린 시청행태 조사 결과 발표

- 전년 대비 스마트폰 · PC를 통한 방송프로그램 시청시간 감소 -

방통위(위원장 한상혁, 이하 방통위)는 26일 2022년 제4차 미디어 다양성위원회(위원장 김효재, 이하 미디어위) 회의를 개최하여 「2021년 N스크린 시청행태」에 대해 논의하고 조사 결과를 발표하였다.

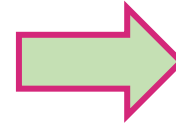
N스크린 시청행태 조사는 스마트폰, PC 등으로 다변화되고 있는 방송프로그램 이용행태 및 시청현황을 파악하기 위해 2017년부터 실시하고 있다.

2021년 주요 조사결과를 살펴보면 다음과 같다.

(스마트폰을 통한 시청) 스마트폰으로 1개월 내 1번 이상 방송프로그램을 시청한 이용자는 70.02%였고, 위드코로나(생활속 거리두기) 시행과 실내 미디어 이용시간 감소 등의 영향으로 월평균 시청시간은 '20년 대비 약 5.29분 감소한 137.37분으로 나타났다.

개인별 월평균 채널 시청시간은 tvN(14.33분), MBC(14.16분), SBS(13.03분), JTBC(12.04분), TV CHOSUN(10.69분) 순 이었고, 장르별로 가장 많이 시청한 방송 프로그램은 오락은 <런닝맨(SBS)>, 뉴스/보도는 <MBC 뉴스데스크(MBC)>, 드라마는 <빈센조(tvN)>로 나타났다.

방송통신위원회도 시청 시간을 기반으로  
시청 현황 파악  
>> 시청 시간 변수의 중요성



작품 러닝타임이 길수록 그렇지 못한 작품에 비해  
시청시간이 상대적으로 길어진다고 생각  
>>

User의 시청 시간을 전체 러닝타임으로 나눠 비율로 판단!

비슷하게 비용 변수 또한 시청자의  
강한 선호도를 엿볼 수 있기에 사용!

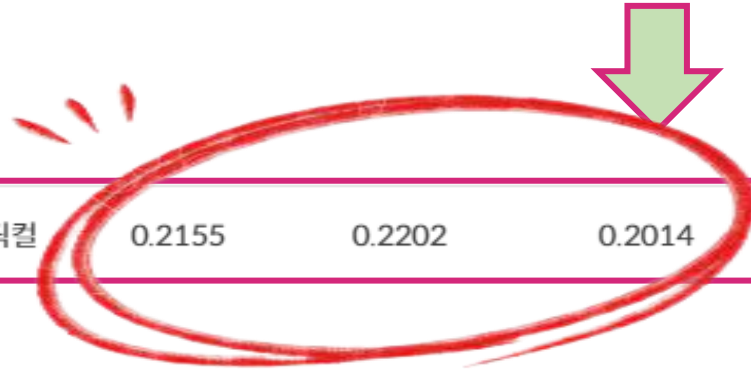
# TEAM 우혁/인섭/은미

ITEM feature 처리 결과

사용한 feature	Feature 처리 방식
Continuos_play	없음
Payment	
Watch_rate	시청 시간 / 러닝타임
Genre_ Large / mid / small	Label encoding
Cast 1, 2, 3, 4	
Keyword_type	

성능변화

ywh0364	0.2131	0.2177	0.1993	상세 보기	2022-11-30 18:18	<input type="checkbox"/>
미다	0.2128	0.2173	0.1992	상세 보기	2022-11-10 21:04	<input type="checkbox"/>



노르딕컬	0.2155	0.2202	0.2014	상세 보기	2022-11-30 00:47	<input checked="" type="checkbox"/>
------	--------	--------	--------	-------	------------------	-------------------------------------

# TEAM 우혁/인섭/은미

USER feature >>> 성별 & 나이 & 취향 변수 사용

성별에 따라 시청 프로그램의 차이가 있다고 판단해  
성별 변수 >> one-hot encoding 진행

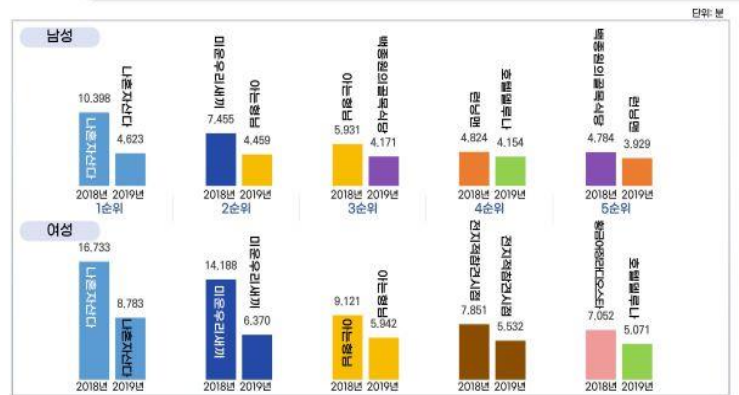
연령대에 따라 시청 프로그램 차이가 커서  
나이 변수 사용 >> 여타 전처리 과정은 없음

개인 취향이 시청 프로그램에 높은 영향을 미치므로  
취향에 대한 변수  
>> one hot encoding을 통해 사용

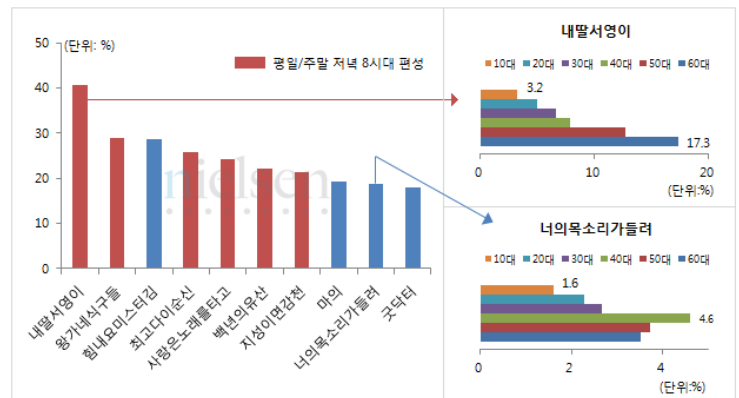
어린 아이들이 시청하기 때문에 아이들의 취향 뿐만 아니라 부모님 취향도 고려!

## • 2019년 고정형TV VOD 시청점유율 보고서 (연간) •

Category 6 성별에 따른 VOD 프로그램 상위 5개 시청시간 전년비교



## 2013 TV 시청률 TOP 10 & 프로그램 연령대별 시청률(AMR)\*



Nielsen Korea TV Behavioral Data, 전국 3,134 가구 (특집, 스페셜, 스포츠, 보도 프로그램 제외)  
(2013.01.01~2013.12.22)

# TEAM 우혁/인섭/은미

## USER feature 처리 결과

사용한 feature	Feature 처리 방식
Age	없음
Sex	One-hot encoding
pr_interest_keyword_1 (부모 취향)	
ch_interest_keyword_1 (아이 취향)	

## 성능변화

ywh0364	0.2131	0.2177	0.1993	상세 보기	2022-11-30 18:18	<input type="checkbox"/>
미다	0.2128	0.2173	0.1992	상세 보기	2022-11-10 21:04	<input type="checkbox"/>



노르딕컬	0.2142	0.2187	0.2004	상세 보기	2022-12-01 20:24	<input type="checkbox"/>
ywh0364	0.2142	0.2186	0.2011	상세 보기	2022-11-28 18:19	<input type="checkbox"/>



# TEAM 우혁/인섭/은미

## ITEM-USER feature 결합 결과

사용한 feature	Feature 처리 방식
Continuos_play	없음
Payment	
Age	
Genre_ Large / mid / small	Label encoding
Cast 1, 2, 3, 4	
Keyword_type	
Watch_rate	시청 시간 / 러닝타임
Sex	One-hot encoding
pr_interest_keyword_1 (부모 취향)	
ch_interest_keyword_1 (아이 취향)	

### 성능변화

ywh0364	0.2131	0.2177	0.1993	상세 보기	2022-11-30 18:18	<input type="checkbox"/>
미다	0.2128	0.2173	0.1992	상세 보기	2022-11-10 21:04	<input type="checkbox"/>



미다	0.2143	0.2189	0.2005	상세 보기	2022-12-02 17:03	<input type="checkbox"/>
미다	0.2142	0.2188	0.2003	상세 보기	2022-12-02 16:11	<input type="checkbox"/>

# Conclusion

---

## # Evaluation

■ ■ ■

LG U+ - AI Ground

# Recommender System

TEAM	SCORE	Recall@25	NDCG@25
정재/종혁/채명	<u>0.1667</u>	0.1722	0.1499
우혁/인섭/은미	<u>0.1647</u>	0.1704	0.1478

$$\text{SCORE} = \text{평균 Recall@K} * 0.75 + \text{평균 NDCG@K} * 0.25$$

(K=25)

### # Recall@K

>> 사용자가 관심있는 전체 아이템 가운데 우리가 추천한 아이템의 비율로서 정확도를 측정하기 위해 적용됨

(전체 평가 점수의 75% 비중)

### # NDCG@K

>> 추천한 아이템에 대하여 순서에 가중치를 두어 평가하는 지표로서 추천 아이템의 우선순위를 측정하기 위해 적용됨

(전체 평가 점수의 25% 비중)

## # Restospective

■ ■ ■

LG U+ - AI Ground

# Recommender System

- ✓ 딥러닝 기반의 방법론으로만 접근하려고 했던 생각에 대한 아쉬움 존재 (리더보드 상위권 솔루션에서 발견할 수 있었던 머신러닝 기반의 방법론 존재)
  - ✓ NeuMF 기반의 모델에 적용하는 feature들에 대한 한계점 존재 (메타데이터를 최대한 프로필 데이터와 콘텐츠 데이터에 적용하려 노력하였지만 데이터 가공 과정에서의 한계 존재)
  - ✓ 추천시스템 대회에 처음 참가하여 해당 분야에 대한 지식 확장과 프로젝트 경험에 대한 만족 (추천시스템 대회 프로세스와 이후 솔루션을 통한 회고로 해당 분야에 대한 경험과 시야를 넓힐 수 있었음)
- 
- ✓ 최종 모델로 Item-User를 결합한 피처 모델의 성능이 나빠 Item Feature 만을 전처리한 모델을 사용
  - ✓ Title과 같은 feature를 NLP를 통해 적용해볼 수도 있음
  - ✓ 어떤 요소를 기반으로 프로그램을 선택하는지에 대해 파악할 수 있었다면 feature engineering 관점에서 더욱 수월하게 진행했을 것
  - ✓ 고정적인 negative ratio가 아니라 사용자의 특성에 맞게 기준을 조정했으면 성능 향상에 도움 되었을 것

**Thank You 😊**

