

# Jesse's Jukebox

---

## 추천팀

July 28, 2022

### 1 배경 설명

제시는 사용자의 음악 성향을 고려하여 각 사용자가 좋아할 음악을 추천하는 음악 추천 시스템을 만들고자 합니다. 이를 위해 제시는 다양한 소스로부터 여러 명의 사용자가 들은 음악과 그 음악을 들은 횟수를 수집하였습니다. 이제 제시가 수집한 데이터를 이용하여 각 사용자가 좋아할것 같은 음악을 맞춰봅시다.

### 2 실험에 필요한 데이터

실험에 필요한 데이터의 포맷과 의미에 대한 설명입니다. 실험에 필요한 파일은 주어진 것과 주어지지 않는 것이 있습니다. 주어지지 않는 데이터는 실험 과정 중에 직접 생성하셔야 합니다.

## 2.1 LISTEN\_COUNT.TXT

각 유저가 어떤 음악을 몇 번 들었는지를 저장하고 있습니다. 파일에는  $\langle user\_id \rangle$   $\langle music\_id \rangle$   $\langle count \rangle$  가 공백 문자를 이용하여 구분되어 있습니다.

예시:

```
18091 6716 3
15772 1251 1
```

위 예시에서는 user\_id가 18091인 사용자가 music\_id가 6716인 음원을 3회, user\_id가 15772인 사용자가 music\_id가 1251인 음원을 1회 들었다는 의미입니다.

## 2.2 TEST\_DATA.TXT

결과 품질 평가에 쓸 파일입니다. 각 줄마다  $\langle user\_id \rangle \langle music\_id\#1 \rangle \langle music\_id\#2 \rangle \dots \langle music\_id\#n \rangle$  형태로 저장되어 있어야 합니다. 이 파일은 주어지지 않습니다. 주어진 listen\_count.txt 파일을 적당히 가공해서 생성하세요.

예시:

```
15722 200 210
2716 100
3759 250
```

## 3 문제

제시가 수집한 listen\_count 데이터를 이용하면 특정 user가 좋아할 만한 음악을 예측해 볼 수 있습니다. 이제 당신은 각 user마다 해당 user가 좋아할 것 같은 음악 K개를 예측해서 추천하려고 합니다. 각 user별로 가장 좋아할 것 같은 음악부터 순서대로 보여 질 수 있도록 만들어 주세요. 단, 해당 user가 이미 들었던 (listen\_count 파일에 해당 user가 들은적이 있는) 음악은 제외합니다.

## 4 결과 품질 평가

결과의 품질 평가 기준은 NDCG입니다.

NDCG: [https://en.wikipedia.org/wiki/Discounted\\_cumulative\\_gain#Normalized\\_DCG](https://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG)

참고로 주어진 데이터에서 가장 인기 많은 노래 100곡을 모든 user에게 추천하였을 때의 NDCG는 0.1입니다. 제출하신 결과물은 특정 user에 대한 개인화된 추천 결과이므로 그 2배인 0.2 이상의 성능을 내는것을 목표로 해봅시다.

## 5 필요 수행 과정

### 5.1 TRAIN

- 주어진 listen count 데이터를 이용하여 추천 모델을 생성합니다.
- 자신의 방법이 모델이 필요하지 않은 경우 이 과정을 생략해도 됩니다.
- 실행 방법: `./train <listen_count> <output>`

실행 예:

```
> ./train listen_count.txt my_model
```

### 5.2 RECOMMEND

- 5.1에서 생성한 model에 기반하여 각 user\_id에 대한 추천 music\_id를 K개씩 생성합니다.
- 5.1을 생략한 경우 model 대신에 listen count 파일을 입력합니다.
- 입력에 쓰일 user\_id.txt는 각 라인마다 user\_id를 담고있는 파일입니다.
- 실행 방법: `./recommend [<model | listen_count>] <user_id.txt> <K>`
- 결과 형식: `<user_id><music_id#1><music_id#2>...<music_id#K>`

실행 예:

```
> cat user_id.txt
```

```
15722
2716
3759
```

```
> ./recommend my_model user_id.txt 5
15772 201 200 1819 215 203
2716 514 184 1235 1184 1752
3759 514 534 1249 6341 4623
```

위 결과의 첫 번째 줄을 해석하면 user\_id가 15772인 사용자는 music\_id 201, 200, 1819, 215, 203 를 순서대로 좋아할 것이다 가 됩니다.

### 5.3 EVALUATE

- 5.2에서 생성한 추천 결과에 대한 품질 평가를 합니다.
- test data 파일을 제작하고 자신의 추천 결과와 비교하여 NDCG를 계산합니다.
- 실행 방법: `./evaluate <recommend_result> <test_data>`

실행 예

```
> ./evaluate my_result.txt test_data.txt
NDCG: 0.200000
```

참고로, 주어진 listen\_count 데이터에서는 평가를 위해 일부 user가 좋아하는 music\_id가 누락되어 있습니다. 또한, 실제 정답 데이터에서는 각 user마다 추천된 음악의 갯수가 서로 다를 수 있습니다.

## 6 제약 조건

- CPU: 8 core 사용 가능
- 메모리: 8GB 사용 가능
- 전체 수행 시간 (Train, Recommend, Evaluation): 20분 이내
- 사용 가능한 언어: C, C++, C#, Java, Scala, Python (이외의 다른 언어 사용을 원하는 경우 사전 협의 가능), **오픈소스 사용 가능**

## 7 제출 파일

- (a) source code
- (b) 실험 결과 (NDCG 값 등)
- (c) 실험에 사용한 training data, test data, model 파일 (자신의 실험이 재현 가능해야함)
- (d) 구현한 방법(알고리즘)에 대해 간단한 설명 (참고한 논문이 있다면 써주세요)
- (e) source code 컴파일 및 실행할 때 필요한 프로그램
- (f) source code 컴파일 및 실행 방법을 기술한 설명서

## 8 평가 기준

- (a) 제약 조건 내 source code 컴파일 및 실행 가능 여부
- (b) 특정 user\_id들에 대해 100개씩 recommend 한 후 NDCG 값 측정
- (c) 사용된 알고리즘과 그에 대한 이해도 (필요에 따라 전화 면접 등을 통해 평가)
- (d) Coding skill