

Melon Playlist Continuation

CUAI 4기 메로나

권송아(소프트웨어학), 김소은(응용통계학), 김태윤(소프트웨어학), 김효민(소프트웨어학), 최은서(소프트웨어학), 홍지호(기계공학)

목차



01. 프로젝트 설명





02. EDA





03. MODEL

a. song

b. tag





04. 결론

01. 프로젝트 설명



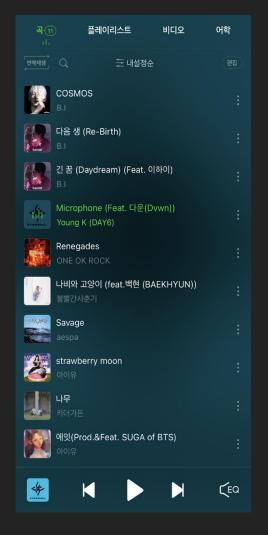
Melon Playlist Continuation

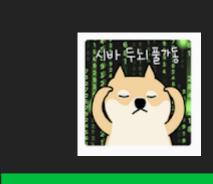


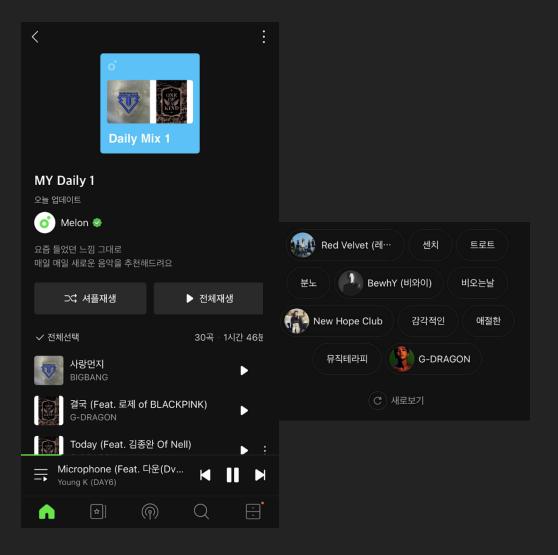
- 주제 선정 배경
- 세계 최대 음원 스트리밍 플랫폼 '스포티파이 ' 기준 6000만곡, 40억개 이상의 플레이리스트
- 각종 음원 스트리밍 업체의 출현으로 인한 <mark>업계</mark> 레드오션
- 추천 시스템의 중요성 부각



플레이리스트에 가장 어울리는 곡들을 예측할 수 있을까?







플레이리스트, 태그

새로운 노래, 태그

이 데이터 세트



Melon Playlist Continuation 플레이리스트에 있는 곡들과 어울리는 곡들을 찾아주세요 일 년 전 시작 2020.09.23

• 플레이리스트 메타데이터

- 플레이리스트 제목
- 플레이리스트에 수록된 곡
- 플레이리스트에 달려있는 태그 목록
- 플레이리스트 좋아요 수
- 플레이리스트가 최종 수정된 시각

• 곡 메타데이터

- 곡 제목
- 앨범 제목
- 아티스트명
- 장르
- 발매일

• 학습 데이터

- 플레이리스트 ID
- 플레이리스트 제목
- 태그 리스트
- 곡 리스트
- 좋아요 개수
- 수정 날짜

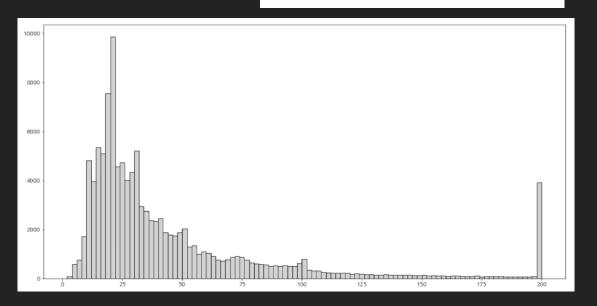
02. EDA



song

- 플레이리스트에는 평균 약 46개의 꼭 수록

	min	25%	50%	mean	75%	max	std
songs	1.0	19.0	30.0	45.94	54.0	200.0	43.95



tag

		중복 매핑 여부	태그수	비율(%)
(0	단일 매핑	17431	59.78
	1	중복 매핑	11729	40.22

- 태그의 매핑 비율은 약 40.2%
- 기분전환, 계절 등 중복을 나타내는 태그가 상위권



03-a. Model(song)

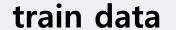


min_pc: 노래가 min_pc 개 미만인 플레이리스트 삭제
if min_pc > 0:
 ply_count = get_count(data, 'id')
 data = data[data['id'].isin(ply_count.index[ply_count['size'] >= min_pc])]

115,071개의 플레이리스트 707,989개의 노래

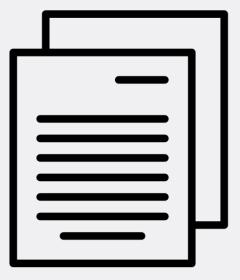
노래가 10개 미만인플레이리스트는 학습 데이터에서 제외





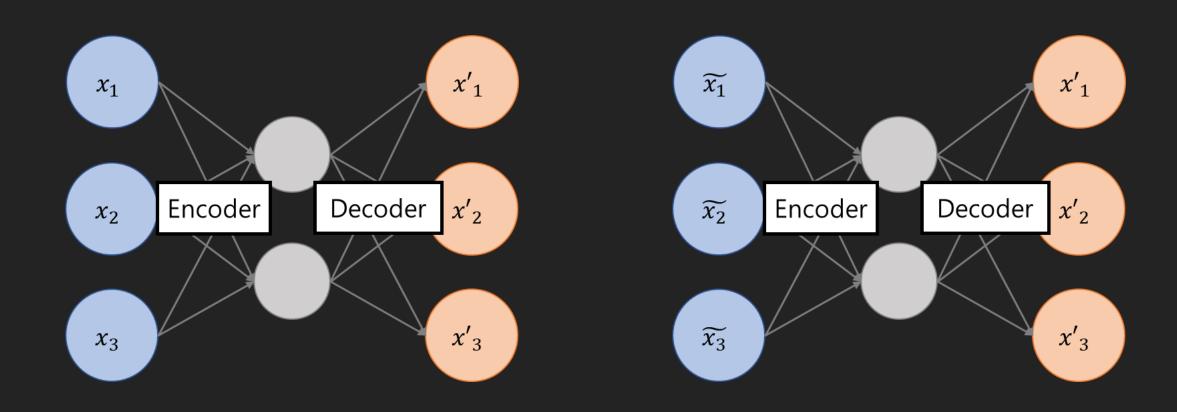


validation data: 20,000

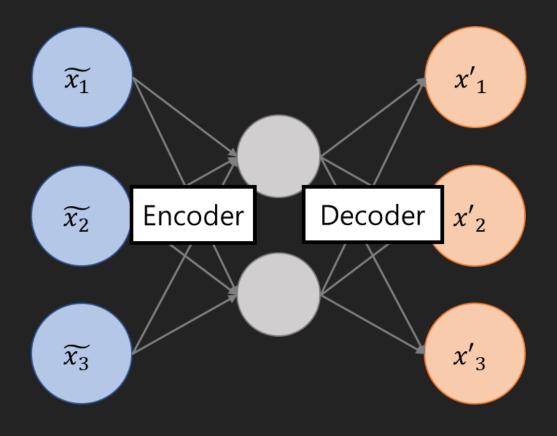


test data: 10,000





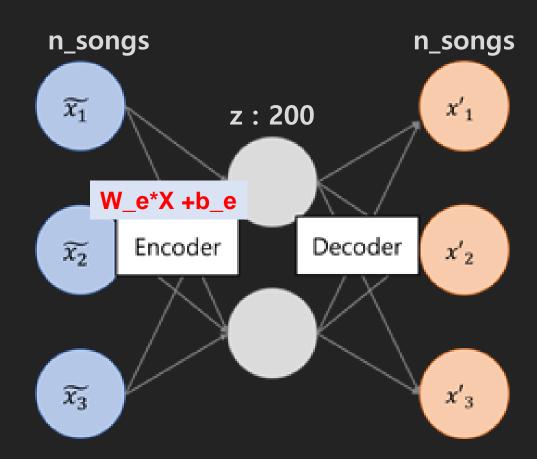
AE DAE



MultiDAE

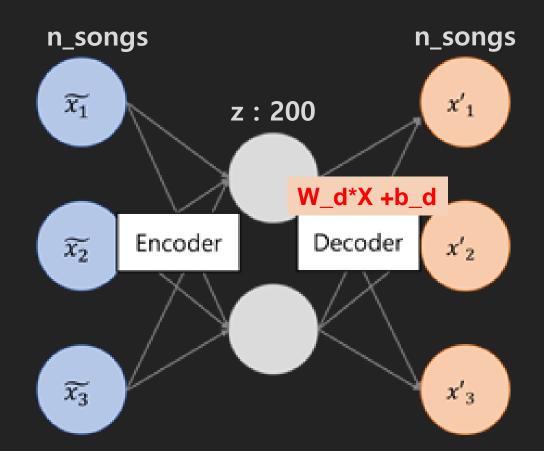
$$\mathcal{L}_{u}(\theta, \phi) = \log p_{\theta}(\mathbf{x}_{u} | g_{\phi}(\mathbf{x}_{u}))$$





MultiDAE





		노래		
플 레		0	1	
이	0	0.121	0.998	
리	1	0.015	0.232	
트				



학습

```
X_t = np.transpose(np.nonzero(X))
   song_meta.loc[X_t[:,1]]['song_name']
                         사랑, 참...
   17839
   25260
                             그래요
   46162
                             북극성
                          오늘은 가지마
   51612
                             응급실
   88503
                             상록수
   97412
                           오빠는...
   109500
   117184
                             그리움
   118150
                        사랑할 것 같아서
                             울지마
   121946
   138319
                            옛날여자
                              낮달
   142777
                          가난한 사랑
   171699
```

예측

```
[126] song_meta.iloc[idx_topk_part]['song_name']
     0
                  Feelings
                      두근두근
     471988
     471989
                       흡연해
     471990
                Last Dance
     471991
                  Memories
     235997
                      떠나갈래
     235998
                 Gang Gang
     235999
                 On My Way
     235991
                      좋을텐데
     707988
               Queen 명곡 멜로디
     Name: song_name, Length: 707989, dtype: object
```



@n:n개의 값 이용

Test NDCG@3000=0.00521 (0.00016)
Test Recall@3000=0.01782 (0.00038)

Test NDCG@1000=0.00365 (0.00015)

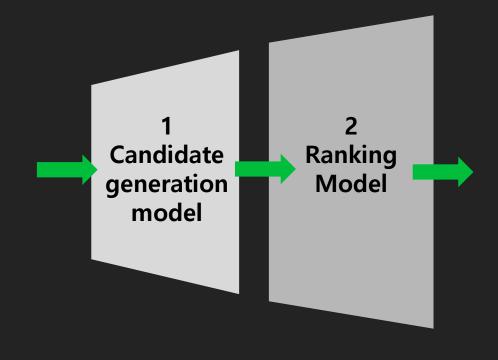
Test Recall@1000=0.00999 (0.00027)

Test NDCG@500=0.00285 (0.00014)

Test Recall@500=0.00636 (0.00022)

Test NDCG@100=0.00192 (0.00014)

Test Recall@100=0.00275 (0.00016)



03-b. Model (tag)

tag 전처리

- Fasttext에서 제공하는 pre-train model을 사용

```
1 print(tag_dic['우클렐레'])
2 print('---')
3 print(tag_dic['폭우'])

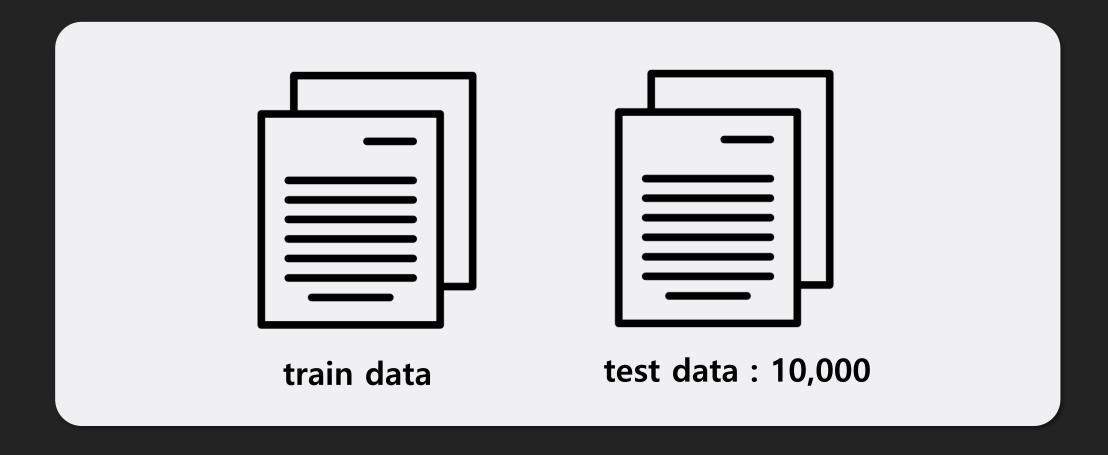
['재즈피아노', '우클렐레']
---
['한파', '폭우', '폭염']
```

△ 태그간 유사도 분석을 마친 결과

- 1. 영어, 숫자 태그 제거
- 2. 태그간 유사도 분석
- 3. 라벨링
- 4. 결측치 제거
- 5. 태그, 플레이리스트, 노래 데이터프레임



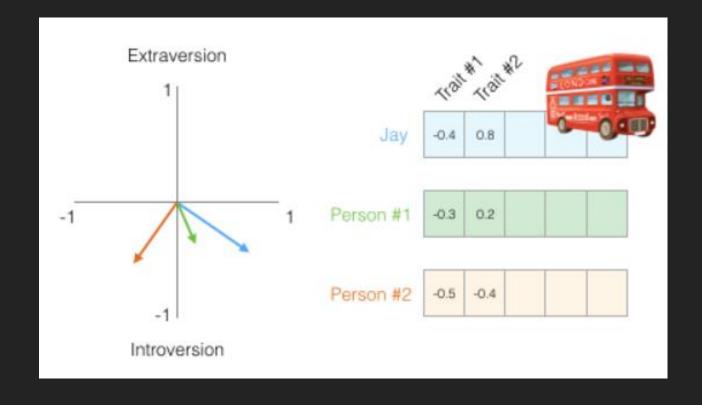
111,131개의 플레이리스트 16,451개의 태그





제목 벡터를 이용한 태그 추천시스템

- Word2Vec 활용





1) 플레이리스트 제목 토크닝

plyIst_title
여행같은 음악
요즘 너 말야
편하게, 잔잔하게 들을 수 있는 곡.-



[['여행', '같다'], ['요즘', '너', '말', '야'], ['편하다', '잔잔하다', '들다', '있다', '곡'], ['크리스마스', '분위기', '흠뻑', '취하'], ['추억', '노래'], (101, 131)



- 1. 결측값 확인
- 2. 한글 외 문자 제거
- 3. 불용어 정의
- 4. 형태소 분석기(Okt)를 통한 토큰화 작업



(95, 819)



2) Word2Vec을 이용한 모델 학습 -> 플레이리스트 벡터값 구하기

```
[['여행', '같다'],
['요즘', '너', '말', '야'],
['편하다', '잔잔하다', '들다', '있다', '곡'],
['크리스마스', '분위기', '흠뻑', '취하'],
['추억', '노래'],
```







3) 코사인 유사도를 이용한 태그 추천



{'슬픔': 3,)밤': 2 '이별': 3,)새벽': 2, '기분전환': 3,)댄스': 1, '발라드감성': 1, '휴식': 1, '사랑': 1, '대한민국': 1, '뭉치자': 1, '화이팅': 1,

'화이팅': 1, '일본노래': 1, '버스에서': 1, '국힙': 1, '힙합': 1}

최종 추천 결과 =>

[('슬픔', 3), ('이별', 3)]



tags	plylst_title
[비오는날, 휴식, 밤, 새벽, 감성, 인디뮤직, 까페]	비를 위한 감성인디뮤직(ver.2)
[매장음악, 오후, 달달한, 벚꽃, 외출, 데이트, 노동요, 신나는]	오후텐션을 올려줄 기분전환리스트
[연말결산, 일렉]	EDM Best Track 결산
[노래, 음악, 팝송, 멜론]	#장르불문 명품 Pop
[셋리스트, 공연] 1	년 만에 완전체로 컴백! 인피니트 멜론 프리미어 쇼케이스 현장



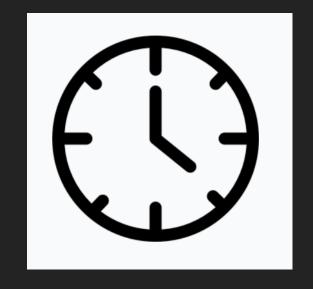
tags [비, 뉴에이지, 감성, 재즈, 매장음악, 밤, 새벽] [기분전환, 스트레스, 휴식, 감성, 힐링, 신나는, 포크, 노동요] [기분전환, 상반기] [기분전환, 설렘, 사랑, 힙합] [기분전환, 셋리스트] 1년 만에 온 plylst_title

비를 위한 감성인디뮤직(ver.2) 오후텐션을 올려줄 기분전환리스트 EDM Best Track 결산 #장르불문 명품 Pop 1년 만에 완전체로 컴백! 인피니트 멜론 프리미어 쇼케이스 현장

0.0549132179783643 Recall:

04. 결론

🧿 결론 - Model(song)



시간 부족



노래 없음



데이터 크기 효과적으로 줄임



Consideration:

• 모델 학습에 대한 시간이 매우 긴 편, 평균 이상의 GPU 성능 요구

Improvements:

• 추천받은 후보군을 랭킹 모델에 넣어 곡의 추천 순서를 매겨 더 발전된 추천 시스템 설계 가능

감사합니다