KaKao Arena 3회 대회

이주남 | 이화정 | 윤소영

Melon TALK

플레이리스트에 가장 어울리는 곡들을 예측할 수 있을까?

이번 대회에서는 플레이리스트에 수록된 곡과 태그의 절반 또는 전부가 숨겨져 있을 때, 주어지지 않은 곡들과 태그를 예측하는 것을 목표로 합니다.

Kakao Arena 3회 대회

00.Contents

플레이리스트에
가장 어울리는 곡들을
예측할 수 있을까?

Kakao Arena 3회 대회

01.Subject

대회 문제

플레이리스트에 가장 어울리는 곡들을 예측할 수 있을까?

이번 대회에서는 플레이리스트에 수록된 곡과 태그의 절반 또는 전부가 숨겨져 있을 때, 주어지지 않은 곡들과 태그를 예측하는 것을 목표로 합니다.

—

각 플레이리스트별로 원래 플레이리스트에 들어있었을 것이라 예상되는 **곡 100개, 태그 10개를 제출**

점수 산정 방법

예측 곡과 예측 태그의 nDCG의 가중평균 값

score = 평균 nDCG(예측한 곡) * 0.85 + 평균 nDCG(예측한 태그) * 0.15

DCG

할인된 누적 이득, 등급이 매겨진 관련성 값이 결과 위치에 비례하여 대수적으로 감소하므로 검색 결과 목록에서 하위에 나타나는 관련성이 높은 문서에 대해 불이익을 주어야 한다는 것. DCG만 놓고 볼 경우 추천된 아이템 개수에 따라 DCG가 다를 수 있으므로 0~1사이의 값으로 정규화해주어야 한다.

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

nDCG

정규화된 DCG, 추천시스템의 평가 지표로 Precision, Recall의 한계를 극복함. 복수 컨텐츠가 relevance를 가지고 있다고 해도 점수에 따라 관련 정도를 평가할 수 있음. log함수를 통해 순서에 대한 가중치가 주어지므로 추천시스템에 적용하기 매우 적절한 평가 지표.

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}, \qquad \mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} \quad 1}{\log_2(i+1)}$$

02.Data&EDA

Data

Song meta.json
플레이리스트 메타데이터
(707989 * 9)

Genre_gn_all
곡 메타데이터 장르정보

Train
(115071 * 6)

Val
(23,015 * 6)

Test
(10,740 * 6)

EDA

Meta data에 0과 월·일이 0000인
이상치 확인

Playlist update_date보다 수록된 song의 issue_date가 미래의 시간을 가지는
이상치 확인

Meta data에 Genre가 **없는 곡을 확인**

Genre_gn_all 데이터에
없는 GN9000 이라는 **곡을 확인**

주어진 태그, 장르만으로는 유사한 플레이리스트를 찾기 어려우므로
유사한 플레이리스트를 찾아 그들이 가지고 있는 노래와 태그를 추천한다.

Kakao Arena 3회 대회

Project Process

Preprocessing

01
Tokenizing

02
Embedding

03
Clustering

04
Matrix

Modeling

05
Candidate

06
Filter

03.Preprocessing

Tokenizing

문장을 토큰으로 나누는 과정

Sentencepiece(vocab size = 32000)

학습 모델의 크기는 단어의 개수에 영향을 받음.
자연어 처리시 **OOV(단어장에 없는 단어)**문제가 발생하기 때문에 Sub-word 분리 전략을 사용
하나의 단어는 여러 서브워드 조합으로 구성된 경우가 많아서 분리하여 인코딩 및 임베딩 하는 전처리

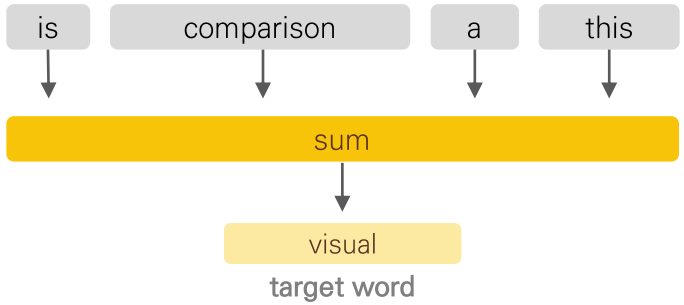
BPE – 빈도수에 따라 subword구성, unknown word 처리에 효과적. sentencepiece 이용하여 가지고있는 text로 모델 생성 가능.

Embedding

album_emp_100
singer_emb_100
song_emb_200
tag_emb_30
tag_gnr_title_emb_100

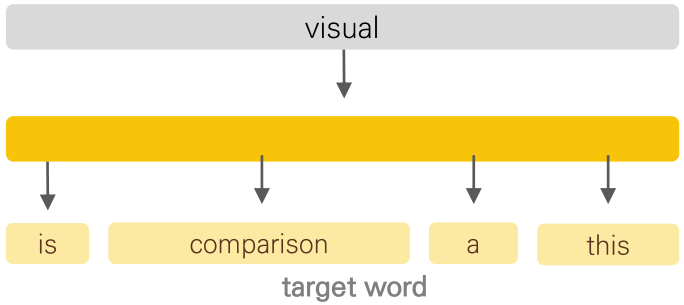
Word2Vec : 파라미터로 SQ(분석 방법론)을 설정할 수 있고, CBOW(0)와 Skip-Gram(1)이 있다.

CBOW : 주변에 있는 단어로 중간에 있는 단어 예측



VS

Skip-Gram : 중간에 있는 단어로 주변에 있는 단어 예측



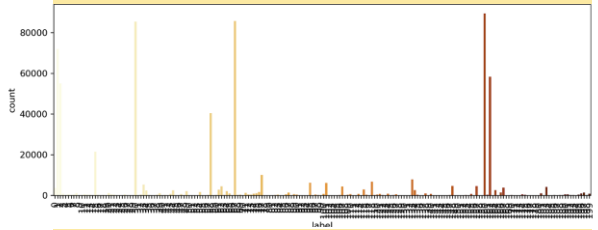
더 우수한 성능을 가지고 있다고 판별된 Skip-Gram 사용

03.Preprocessing

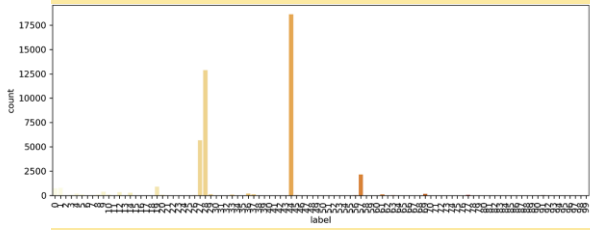
Clustering K-means

6가지 변수를 조합하여 군집을 생성

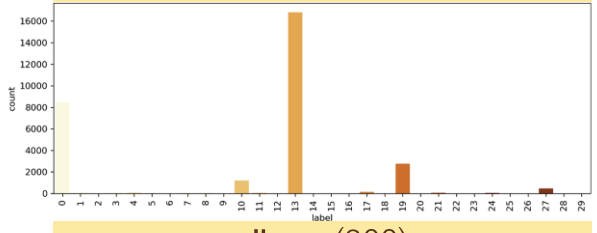
song (200)



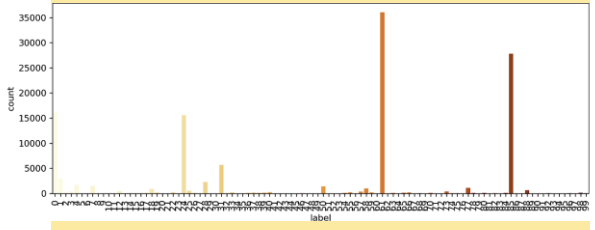
tag + title + genre (100)



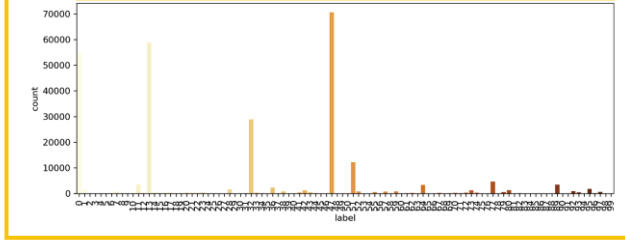
tag (30)



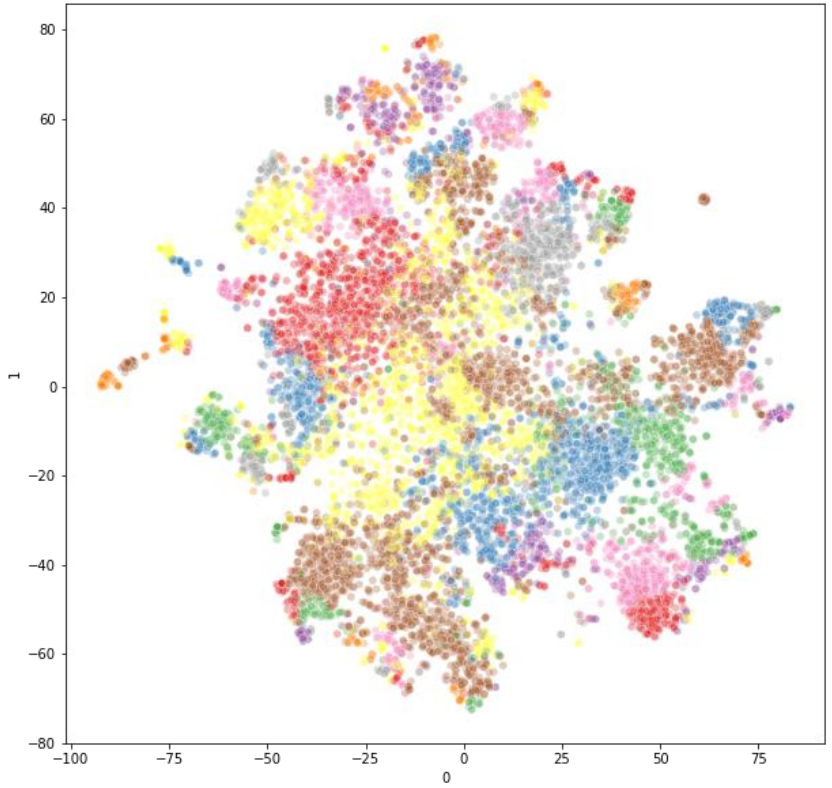
singer (100)



album (200)



detail genre (원-핫인코딩)



03.Preprocessing

Matrix

행렬 matrix 만들기 : 군집화된 속성들을 (태그, 노래, 가수 , 앨범) 플레이리스트 기준으로 집계. 총 13개

- 단일 기준 matrix 생성
  ex, song 기준, tag 기준

- 복합 기준 matrix 생성 : 군집화 A와 군집화 B를 더해서 새로운 플레이리스트 특징(matrix)을 만들어 냄
  ex, song + tag , album + tag 등

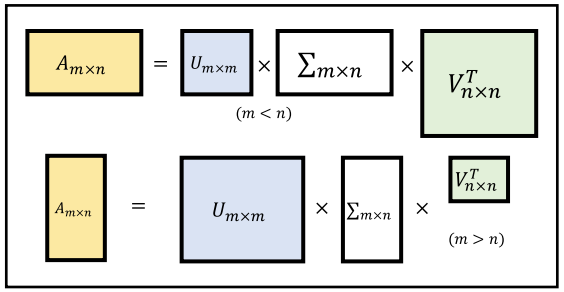
	Palylist_Id	clu0	clu1	clu2	……	……	Clu198	clu199	clu200
0	61281	2	0	1	……	……	0	0	1
1	10532	0	0	1	……	……	0	0	2
2	76951	0	0	0	……	……	5	0	0
……	……	……	……	……	……	……	……	……	……
……	……	……	……	……	……	……	……	……	……
115069	131982	5	0	0	……	……	0	2	1
115070	100389	1	0	0	……	……	1	0	0

Ex, 61281인 playlist_id에는, clu0번에 포함된 노래가 2개 , clu1번에는 0개, clu2개는 1개 있다는 의미.

04.Model

SVD
(특이값 분해)

부적합

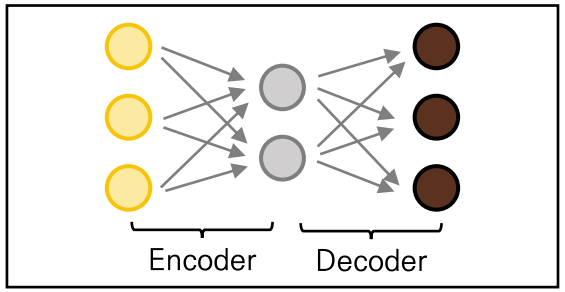


특이값 : 행렬로 표현되는 선형 변환의 스케일 변환을 나타내는 값
행렬을 분해해서 액기스만 뽑아놓고 사용하는 것.

장점 : 행렬이 정방행렬이든 아니든 관계없이 모든 m*n행렬에 대해 적용 가능
단점 : 추천하는 이유를 알 수 없음.

AutoEncoder

부적합

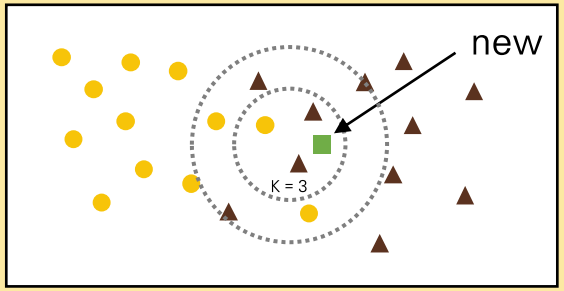


input -> encoder -> decoder -> output
비지도 학습, 정답 데이터가 따로 있는 것이 아닌 입력 값 그 자체가 정답.
self-supervised learning

장점 : 용량도 작고 품질도 더 좋아짐. 차원의 저주 예방
단점 : Feature를 압축 하다 보면, 다른 데이터가 들어와도 training set과 비슷하게 만들어버릴 수 있음.(overfitting)

KNN + Filter

적합



K-최근접이웃 알고리즘.
하이퍼파라미터(k : 탐색할 이웃 수 / 거리측정 방법)
k가 작을 경우 데이터의 지역적 특성을 지나치게 반영(overfitting), k가 매우 클 경우 과하게 정규화 (underfitting)

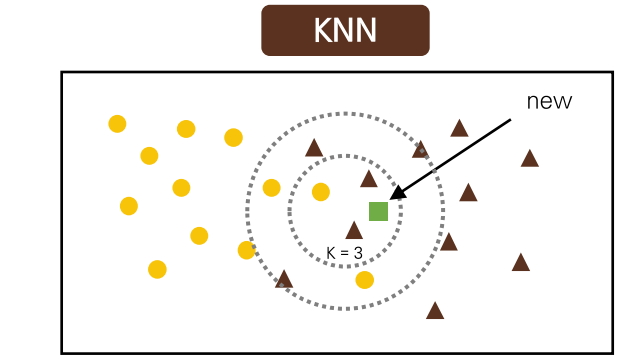
장점 : 노이즈의 영향을 크게 받지 않으며 학습 데이터 수가 많다면 꽤 효과적인 알고리즘.
단점 : k가 분석에 적합한지 불분명해서 연구자가 임의로 선정해야 함. 계산 시간이 오래 걸림.

04.Model

Matrix 13개 생성

- **Matrix 1** song기반
- **Matrix 2** tag기반
- **Matrix 3** album기반
- ⋮
- **Matrix 12** song+tag+title기반
- **Matrix 13** song+tag+title+album기반

KNN알고리즘의 코사인 유사도를 이용하여 Train데이터에 제시된 플레이리스트와 유사한 플레이리스트 7개 선정



코사인 유사도

각도 기반 추천 (cosine similarity)

$$cosine\ similarity = cos\theta = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2}\sqrt{\sum B_i^2}}$$

- playlist 01
- playlist 02
- playlist 03
- playlist 04
- playlist 05
- playlist 06
- playlist 07

플레이리스트의 곡, 태그 count

곡 100개 태그 10개 추출

05.Result

Song & Tag 예측 결과

31번 플레이리스트 정보
제목 : 디즈니와 함께하는 겨울나기

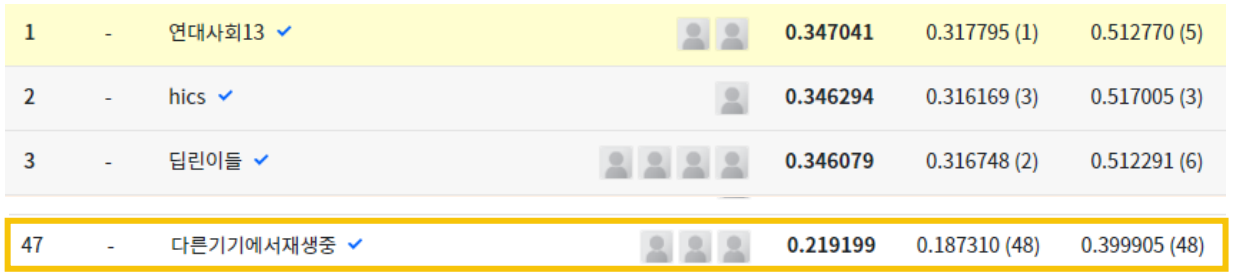
{ 31, 42, 44, …… 55 , 56 }

{ 발라드, 외로움 , 슬픔 …… }

추천된 플레이리스트 노래와 태그 결과

{겨울왕국, 알라딘 … }

{ 발라드, 외로움 , 슬픔 …… }



1	-	연대사회13 ✔	**0.347041**	0.317795 (1)	0.512770 (5)
2	-	hics ✔	**0.346294**	0.316169 (3)	0.517005 (3)
3	-	딥린이들 ✔	**0.346079**	0.316748 (2)	0.512291 (6)
47	-	다른기기에서재생중 ✔	**0.219199**	0.187310 (48)	0.399905 (48)

곡 nDCG : 0.187310 (48 등)
태그 nDCG : 3.99905 (48 등)

최종 스코어 : 0.219199 (47등 / 785 팀)