



# 음원차트 순위 예측

김예진 박보영 박지운 이서영

# 목차

---

#1 주제 소개

#2 데이터 준비

#3 순위 예측 회귀 모형

#4 순위 추이 예측 (clustering)

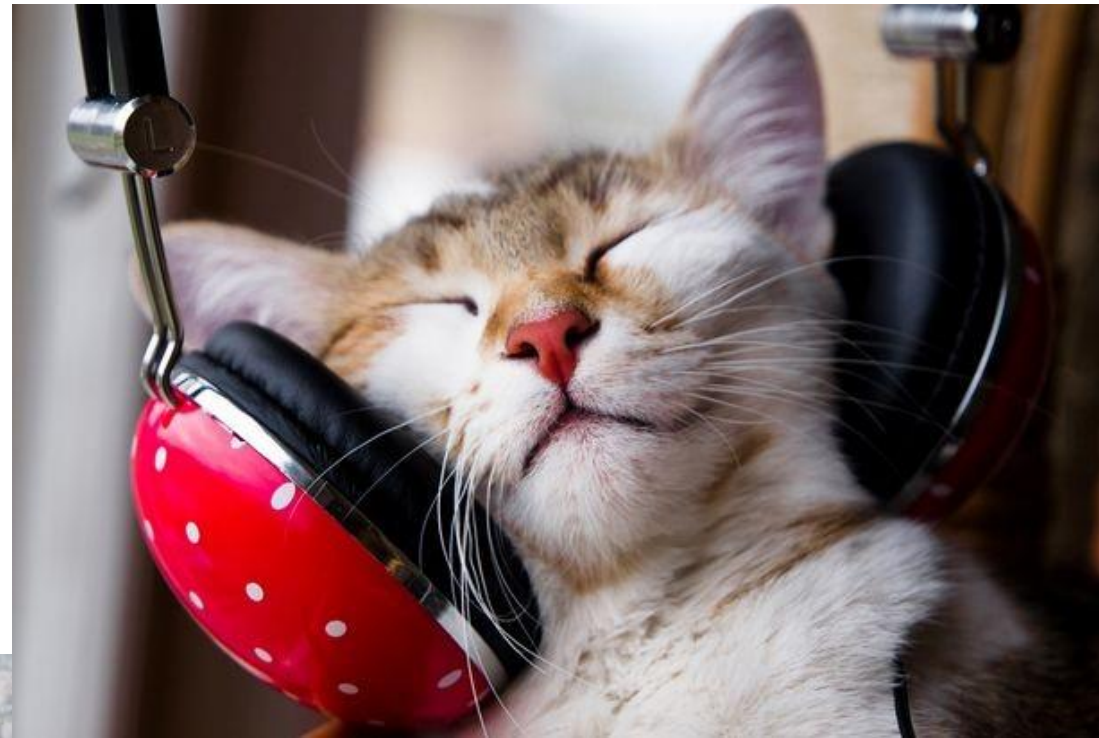
#5 결론 및 제언



# #1 주제 소개



# #1.1 주제 소개



## 주제 : 음원차트 순위 예측

음원차트 순위는 음원의 흥행 여부 평가에 있어 중요한 지표이다.

- ✓ 음원차트 순위에 많은 영향을 미치는 요소로는 어떤 것들이 있을까?
- ✓ 이를 바탕으로 현재의 순위와 미래의 순위 변화 추이를 예측할 수 있을까?

# #1.1 주제 소개

## 회귀 분석

- 해당 곡의 현재 정보를 바탕으로 현재 순위를 예측할 수 있을까?
- RMSE와 그래프를 바탕으로 정확도 확인

## 클러스터링 분석

- 곡의 기본 정보를 바탕으로 클러스터링 분석 진행
- 동일한 군집에 속한 곡들은 동일한 순위 변화 추이를 나타낼까?
- 해당 곡의 4주 뒤 순위가 어떻게 변화해 있을지 예측할 수 있을까?









































## #2 데이터 준비



# #2.1 크롤링

## Melón

2022.07.04 ~ 2022.07.10 장르종합 주간차트

전체선택 ▶ 들기 ▶ 다운 + 담기 선물 ▶ TOP 100 들기				
<input type="checkbox"/> 순위	곡정보		좋아요	뮤비 다운
<input type="checkbox"/> 1 -0		▶ +  LOVE DIVE IVE (아이브)   LOVE DIVE	♡ 179,244	 
<input type="checkbox"/> 2 -0		▶ +  TOMBOY (여자)아이들   I NEVER DIE	♡ 207,278	 
<input type="checkbox"/> 3 -0		▶ +  That That (prod. & feat. SUGA of BTS) 싸이 (PSY)   싸다9	♡ 116,729	 
<input type="checkbox"/> 4 ↑13		▶ +  POP! 나연 (TWICE)   IM NAYEON	♡ 74,179	 
<input type="checkbox"/> 5 -0		▶ +  정이라고 하자 (Feat. 10CM) BIG Naughty (서동현)   정이라고 하자	♡ 166,411	 
<input type="checkbox"/> 6 ↑2		▶ +  사랑인가 봐 멜로망스   사랑인가 봐 (사내맞선 OST 스페셜 트랙)	♡ 151,316	 
<input type="checkbox"/> 7 ↑1		▶ +  나의 X에게 경서   나의 X에게	♡ 110,692	 
<input type="checkbox"/> 8 ↑1		▶ +  봄여름가을겨울 (Still Life) BIGBANG (빅뱅)   봄여름가을겨울 (Still Life)	♡ 280,670	 
<input type="checkbox"/> 9 ↑1		▶ +  LOVE me BEO (비오)   LOVE me	♡ 106,998	 
<input type="checkbox"/> 10 ↑1		▶ +  FEARLESS LE SSERAFIM (르세라핌)   FEARLESS	♡ 66,558	 

음원 사이트 크롤링을 통해 얻은 데이터 사용  
주간차트 1위~50위

기간 :  
2017년 1월 첫째 주 ~ 2022년 7월 첫째 주

데이터 :

주간차트	가수정보
순위	가수명
곡명	성별
가수명	활동 타입
발매일	팬 수
좋아요 수	
앨범명	
장르	
댓글 수	



# #2.1 크롤링

## 주간차트

rank	title	artist	Album_date	LIKE	Album_name	Album_Genre	replies
1	당신의 밤	황광희 X 7	2016.12.31	125,153	무한도전 9	랩/힙합	562
2	에라 모르지	BIGBANG	2016.12.13	185,501	MADE	랩/힙합	699
3	Beautiful	Crush	2016.12.17	230,966	도깨비 OST	발라드, R&B	332
4	좋다고 말해	볼빨간사춘기	2016.12.21	225,161	Full Album	인디음악,	284
5	Stay With Me	찬열 (CHANGYOL)	2016.12.03	220,989	도깨비 OST	발라드, 랩,	555
6	오랜 날 오랜 밤	AKMU (악뮤)	2017.01.03	221,595	사춘기 하	포크/블루즈	480
7	쏘아	하하 X MILLIEB	2016.12.31	45,597	무한도전 9	랩/힙합	258
8	이쁘다니까	에디킴	2016.12.24	102,326	도깨비 OST	발라드, 국	91
9	LAST DANCE	BIGBANG	2016.12.13	166,152	MADE	발라드	843
10	저 별	헤이즈 (Heize)	2016.12.05	147,639	저 별	R&B/Soul	196
11	Decalcomania	마마무 (MAMAMU)	2016.11.07	156,213	MEMORY	댄스	527
12	Who Are You	샘김 (Sam Kim)	2016.12.25	106,493	도깨비 OST	발라드, 국	217
13	TT	TWICE (트와이스)	2016.10.24	196,698	TWICEcoastline	댄스	1,065
14	처럼 (Feat. J)	유재석 X J	2016.12.31	25,739	무한도전 9	랩/힙합	119
15	만세	양세형 X ELO	2016.12.31	32,841	무한도전 9	랩/힙합	195
16	I Miss You	소유 (SOY)	2016.12.31	120,121	도깨비 OST	발라드, 국	175
17	우주를 줄게	볼빨간사춘기	2016.08.29	290,915	Full Album	인디음악,	579
18	나만 안되는 사랑	볼빨간사춘기	2016.08.29	207,254	Full Album	인디음악,	307
19	이 바보야	정승환	2016.11.29	136,393	목소리	발라드	256
20	불장난	BLACKPINK	2016.11.01	195,931	SQUARE TWO	댄스	434

## 가수정보

artist	sex	act_type	fan
신예영	여성	솔로	4
Knock	남성	솔로	48
우디 (Woo Do)	남성	솔로	648
박명수 X 7	남성	그룹	845
유재석 X J	남성	그룹	1,019
WSG워너비	여성	그룹	1,058
토요태	혼성	그룹	1,187
양세형 X ELO	남성	그룹	1,314
하은요셉	남성	그룹	1,321
Ryan Gosling	혼성	콜라보	1,369
정준하 X 2	남성	그룹	1,376
황광희 X 7	남성	그룹	1,595
갯츠키 (GOT7)	남성	그룹	1,828
유두래곤	남성	솔로	1,919
MSG워너비	남성	그룹	1,925
비룡	남성	솔로	2,401
김대명	남성	솔로	2,413
WSG워너비	여성	그룹	2,491
하하 X MILLIEB	남성	그룹	2,562
진민호	남성	솔로	2,854



# #2.2 변수 생성 및 전처리

## 수치형, 범주형 변수가 아닌 변수 처리

- 곡명이 한글인지
- 곡명 길이
- 앨범명은 제거

## 날짜 관련 변수 처리

- 발매일 ~ 첫 차트인까지 기간
- 발매일 ~ 해당 차트 주차까지 기간
- 해당 차트 주차의 계절

✓ 주간차트 -> 날짜 계산을 위해 해당 주차의 마지막 날짜를 사용

✓ 발매 이후 발매일이 갱신되어 계산 결과가 음수로 나온 경우, 0으로 처리

# #2.2 변수 생성 및 전처리

## 장르가 여러 개인 경우 존재

- 원-핫 인코딩 : 해당 장르는 모두 1, 나머지는 0으로 처리
- 라벨 인코딩 : 첫 번째를 대표장르로 간주하고 나머지는 무시

## 새로운 변수 추가

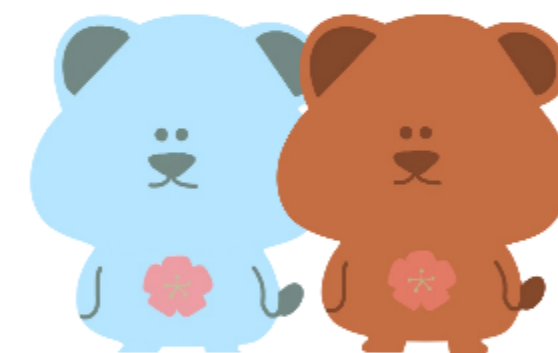
- 전 주차의 순위 (없는 경우 가장 하위인 50위로 처리)
- 해당 곡의 총 1위 횟수
- 해당 가수의 차트인 곡 수

✓ 총 1위 횟수, 첫 차트인 날짜 등의 계산은 새로 추가된 모든 변수는 확보된 데이터 안에서 이루어짐

# #2.3 최종 사용 변수

term	해당 차트의 주차
rank	순위
is_kor	곡명이 한글인지 (1, 0)
title_len	곡명의 길이
LIKE	좋아요 수
replies	댓글 수
previous rank	전 주차의 순위
chartin_cnt	해당 가수의 차트인 곡 수
first_cnt	해당 곡의 총 1위 횟수
rel_chartin	발매일 ~ 첫 차트인까지 기간
rel_term	발매일 ~ 해당 차트 주차까지 기간
season	해당 차트 주차의 계절 (봄, 여름, 가을, 겨울)
sex	가수의 성별 (여성, 남성, 혼성)
act_type	가수의 활동 타입 (그룹, 솔로, 콜라보)
fan	가수의 팬맺기 수

# #3 순위 예측 회귀 모형



# #3.1 AutoML

직접 모델 피팅을 진행한 결과, 좋은 수치 및 그래프 결과를 얻기 힘들어 최적 파라미터를 효율적으로 구할 수 있는 AutoML 사용

Automated Machine Learning

: ML 파이프라인에서 반복되는 수작업을 자동화하는 프로세스

	설명	
데이터셋	2017.1~2022.6 주간 차트 곡 및 아티스트 관련 데이터	
피쳐	numerical	LIKE, replies, first_cnt, rel_chartin, rel_term, title_len, previous_rank, chartin_cnt, fan_반올림
	categorical	is_kor, season, sex, act_type, main_genre
타겟	rank	
파라미터 세팅	train_size = 0.7, remove_multicollinearity = True, feature_selection = True	
	설명	사용 기법
스케일링	데이터 분포가 불균등하므로 로그 변환 후 스케일링을 한번 더 사용하여 완화 (raw data로는 좋지 않은 결과 보임)	log 변환
		Standard Scaling
		MinMax Scaling

# #3.1 AutoML

: AutoML로 모든 모델 탐색 후 좋은 성능을 보인 모델 확인

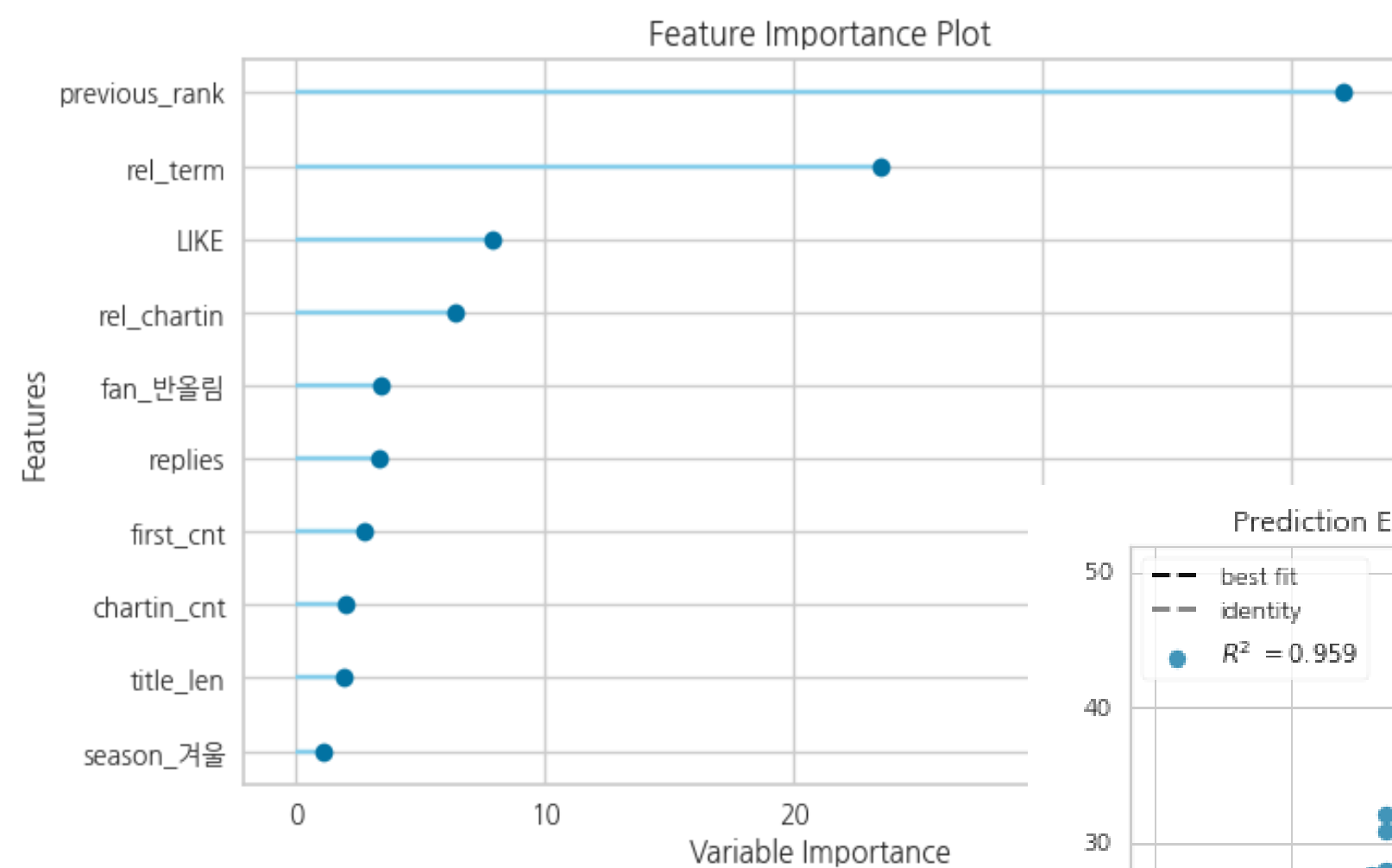
Scaling	Model (top 3)	RMSE	R2	RMSLE
Log & Standard Scaling	CatBoost	5.0600	0.8766	0.2675
	LightGBM	5.3073	0.8642	0.2816
	XGBoost	5.3251	0.8633	0.2810
Log & MinMax Scaling	CatBoost	5.1043	0.8737	0.2677
	XGBoost	5.2830	0.8645	0.2761
	LightGBM	5.3040	0.8636	0.2818

# #3.2 CatBoost with standard data

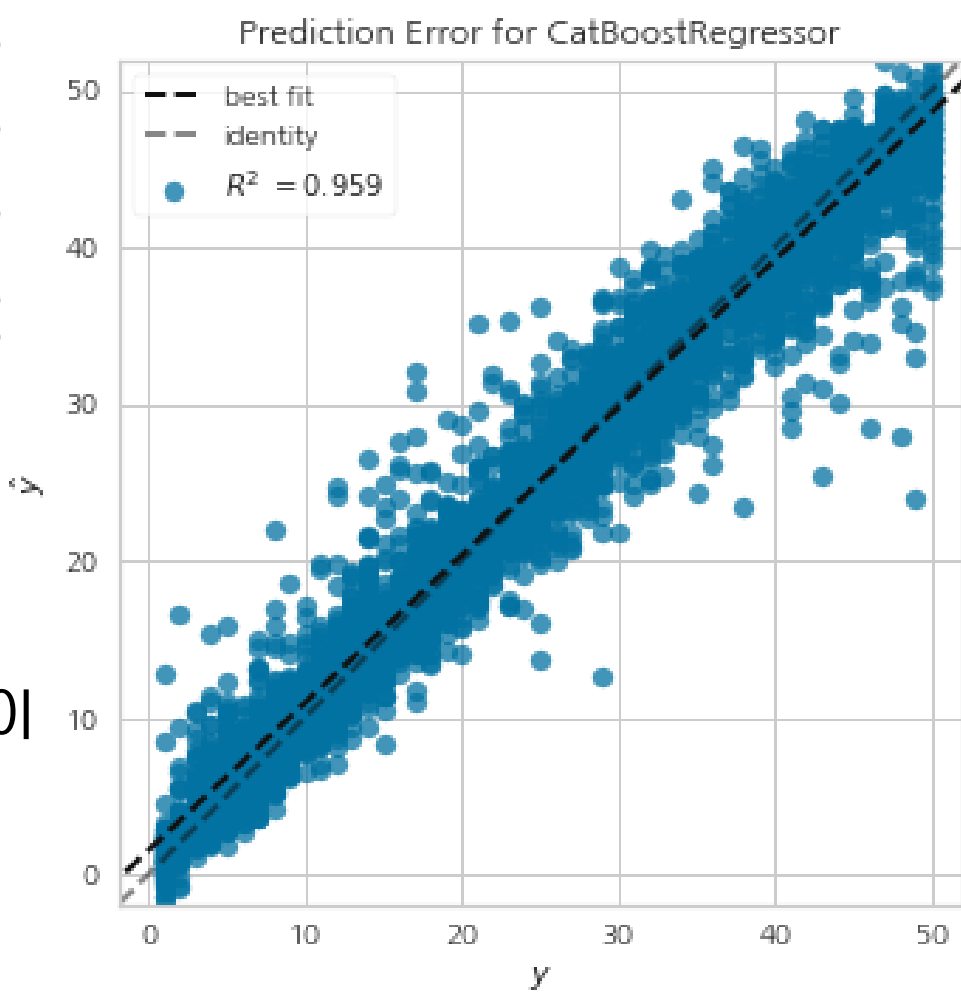
: 가장 성능이 좋았던 CatBoost 모델 Kfold 진행 (K=10)

	MAE	MSE	RMSE	R2	RMSLE	MAPE
Fold						
0	3.0954	23.3529	4.8325	0.8815	0.2664	0.2471
1	3.0604	22.3937	4.7322	0.8909	0.2855	0.2495
2	3.2489	28.2890	5.3187	0.8652	0.2959	0.2709
3	3.1904	26.3774	5.1359	0.8785	0.2600	0.2092
4	3.1068	25.1552	5.0155	0.8781	0.2760	0.2244
5	3.1972	26.6287	5.1603	0.8714	0.2689	0.2393
6	3.4051	28.1812	5.3086	0.8653	0.3009	0.2779
7	3.2170	28.1737	5.3079	0.8671	0.2941	0.2597
8	3.2396	28.5545	5.3436	0.8621	0.3032	0.2807
9	3.1532	24.9572	4.9957	0.8791	0.2667	0.2239
Mean	3.1914	26.2064	5.1151	0.8739	0.2818	0.2483
Std	0.0933	2.0778	0.2053	0.0087	0.0153	0.0231

- 최적 파라미터 탐색 결과 없음



- 변수 중요도 탐색 결과,  
⇒ 중요 변수 : 저번 주 순위, 해당 주와 발매일 간 차이
- 큰 이상치 없이 피팅 잘 된 것 확인



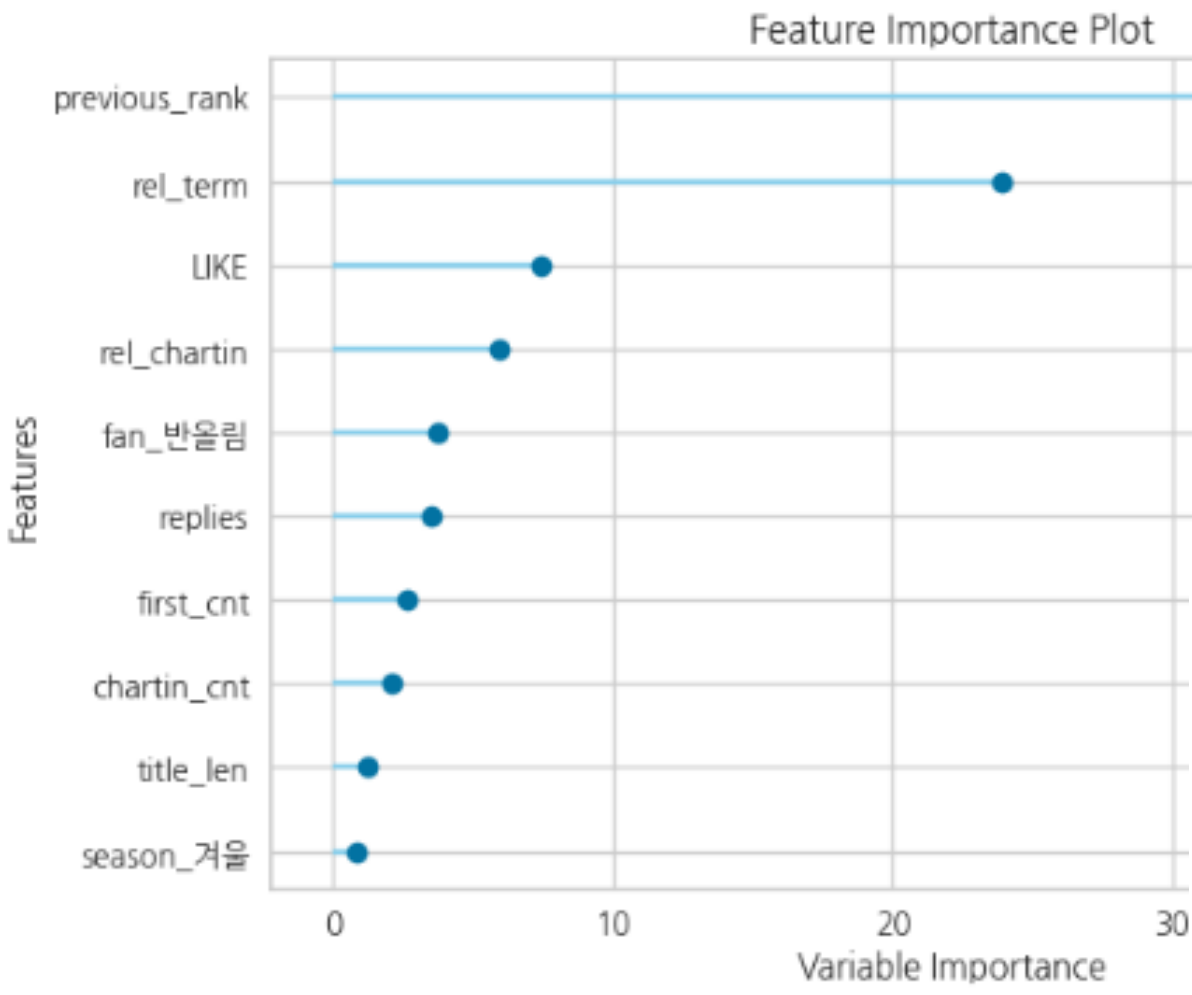


# #3.2 CatBoost with normalized data

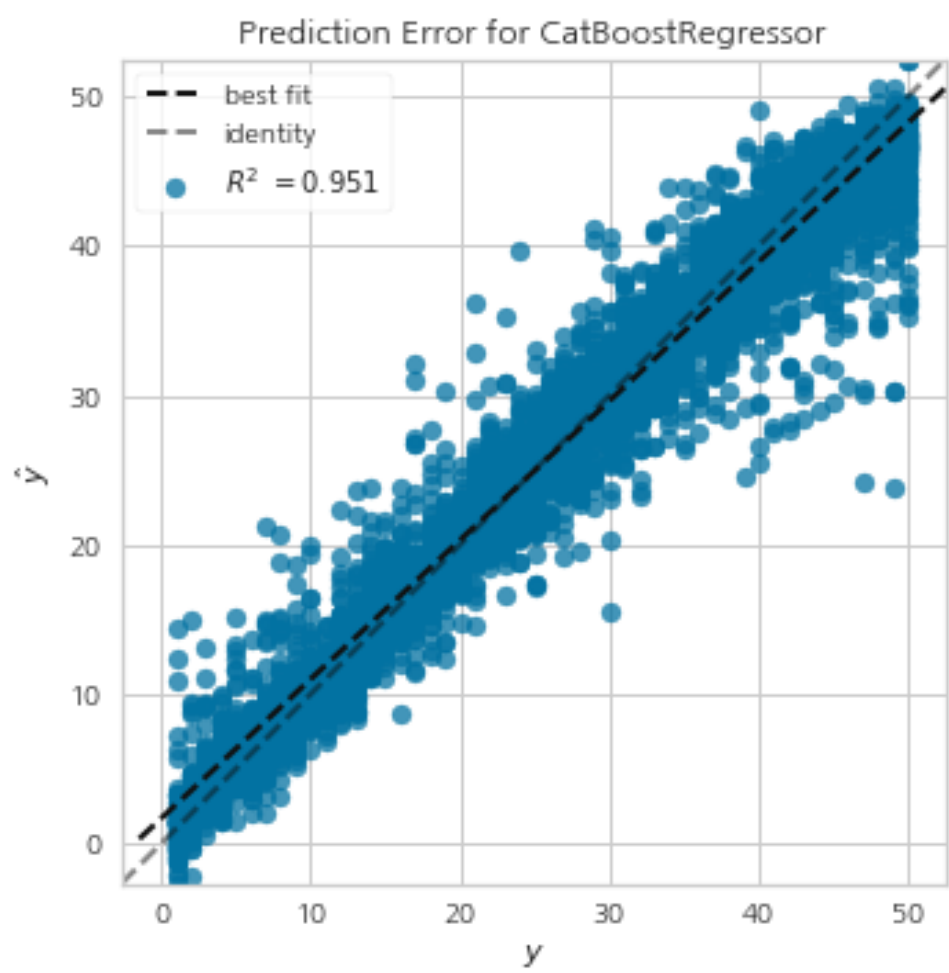
: 가장 성능이 좋았던 CatBoost 모델 Kfold 진행 (K=10)

	MAE	MSE	RMSE	R2	RMSLE	MAPE
Fold						
0	3.0988	27.1930	5.2147	0.8636	0.2444	0.1886
1	3.2458	26.0386	5.1028	0.8776	0.2777	0.2402
2	3.0242	25.0321	5.0032	0.8840	0.2685	0.2319
3	3.1181	25.7771	5.0771	0.8772	0.2665	0.2259
4	3.2545	29.5437	5.4354	0.8579	0.2961	0.2425
5	2.8256	20.4850	4.5260	0.8998	0.2430	0.2098
6	3.1841	28.1506	5.3057	0.8647	0.2801	0.2493
7	2.9717	25.6941	5.0689	0.8737	0.2686	0.2272
8	3.2516	31.1966	5.5854	0.8460	0.2641	0.2034
9	3.0913	24.7908	4.9790	0.8799	0.2899	0.2761
Mean	3.1066	26.3902	5.1298	0.8724	0.2699	0.2295
Std	0.1314	2.7757	0.2738	0.0143	0.0164	0.0237

- 최적 파라미터 탐색 결과 없음



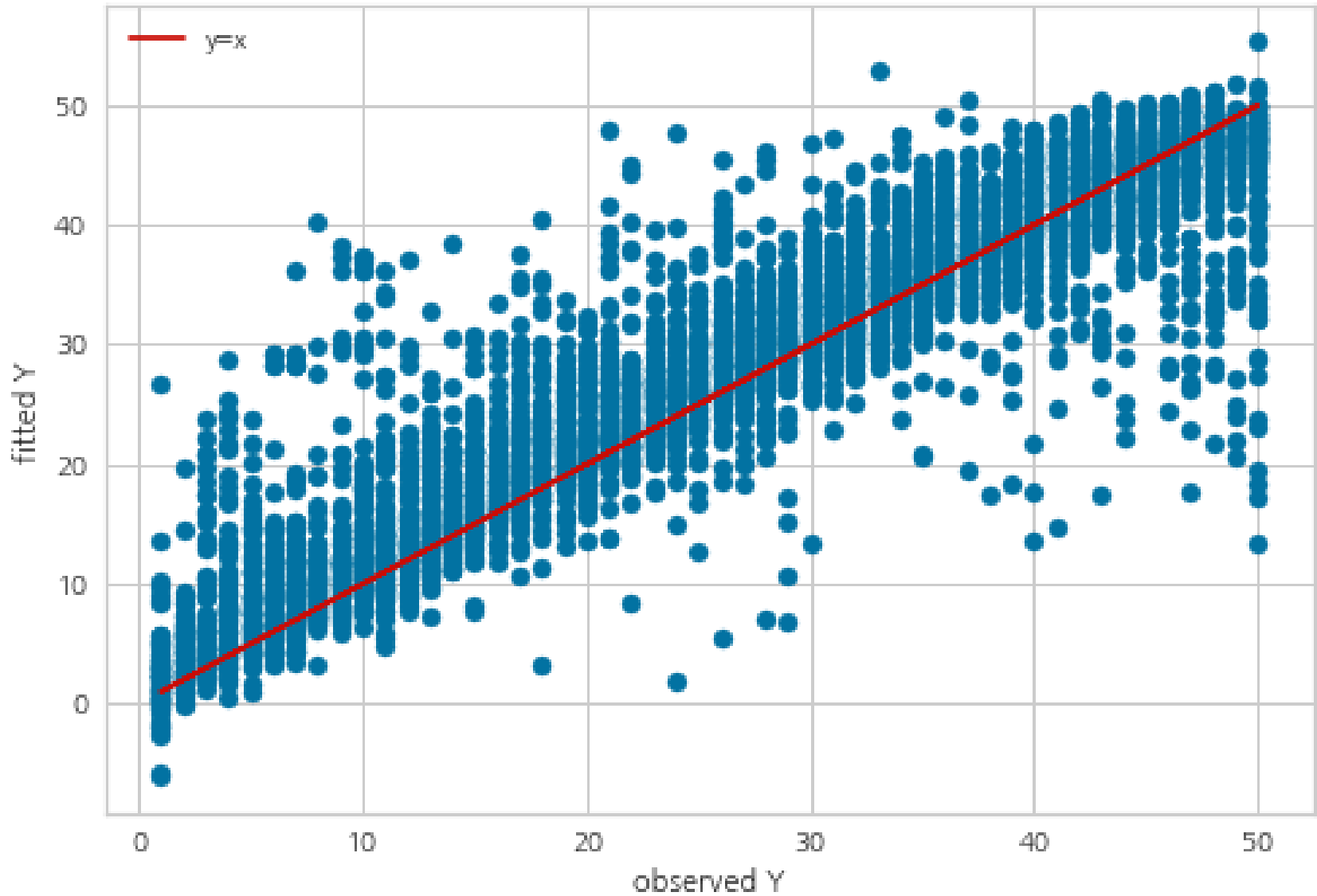
- 앞의 결과와 거의 일치하나, 미세하게 Standard Scaling한 데이터에서 좋은 성능 보임



# #3.3 단일 모델로 순위 예측

: 범주형 변수만 One-hot Encoding 처리 한 원본 데이터로 모델 예측 결과

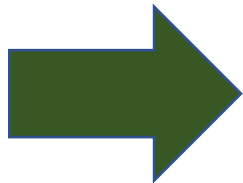
파라미터 세팅	CatBoost는 모델 자체 기본 파라미터 최적화가 잘 되어있어 최적 파라미터 탐색 결과, 튜닝 필요 없음 확인	
RMSE		5.9391
R2		0.8301



# #3.3 단일 모델로 순위 예측

: 7월 1주차 주간 차트 곡 중 하나 pick

term	rank	title	artist	LIKE	replies
2022071	10	FEARLESS	LE SSERAFIM (르세라핌)	54999	1599
first_cnt	rel_chartin	rel_term	title_len	previous_rank	chartin_cnt
0	13	69	8	9	1
fan_반올림	is_kor	season	sex	act_type	main_genre
7730	1?	여름	여성	그룹	댄스



10

▼1



FEARLESS

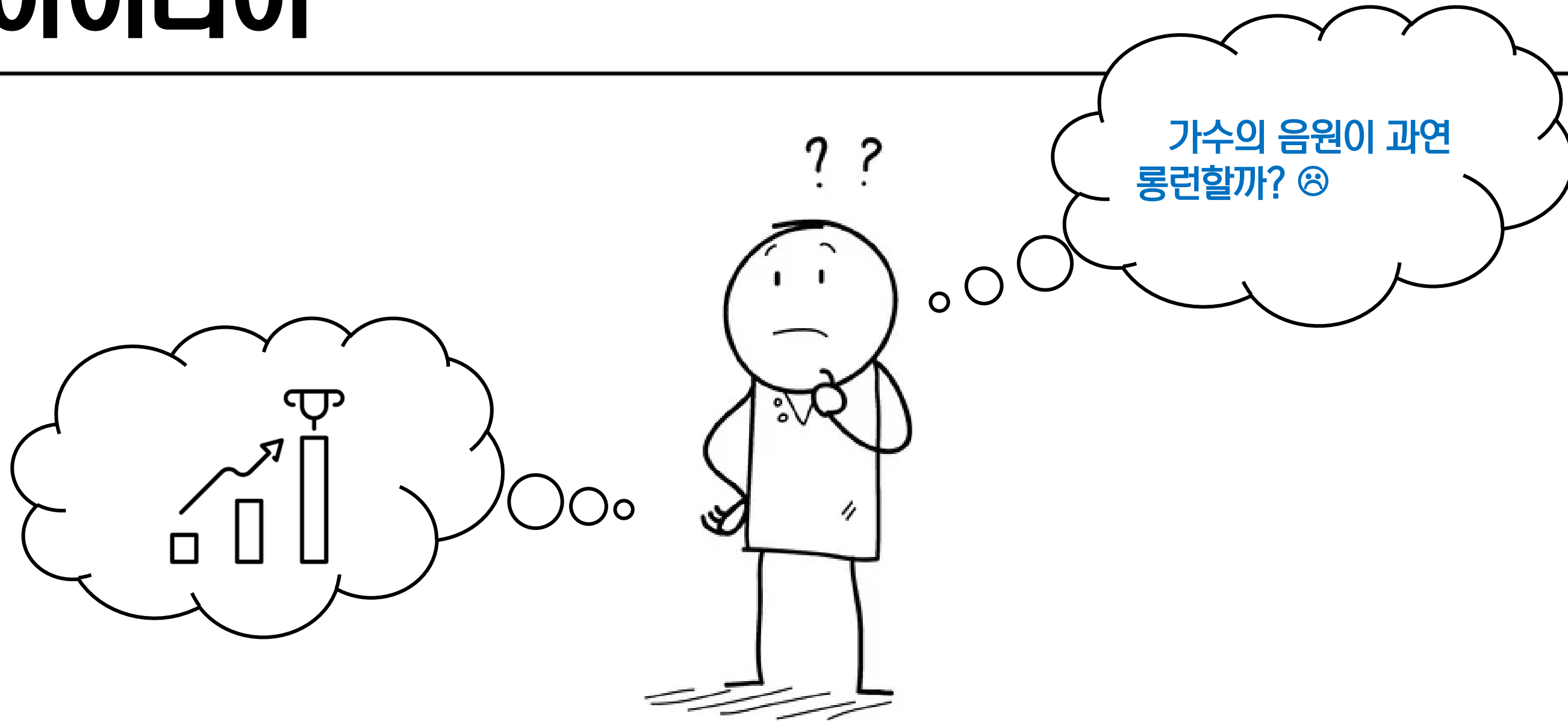
LE SSERAFIM (르세라핌)

예측 등수
11.1

## #4 순위 추이 예측 (clustering)



# #4.1 아이디어



INPUT

음원, 가수의 기본정보

Ex) 제목 길이, 곡 장르, 가수 성별, 활동 타입  
등

OUTPUT

4주뒤 해당 음원의 순위 추이 예측

Ex) 4주뒤 "Butter"는 순위를  
"안정적으로 유지"할 것이다.

# #4.2 데이터 전처리

## 1) 원 데이터를 ‘곡 이름’ 및 ‘가수’ 를 기준으로 pivot

	title	artist_name	LIKE	replies	first_cnt	rel_chartin	is_kor	title_len	fan_반 올림	chartin_cr	... week164
0	25	볼빨간사춘기	37009.00000	190.00000	0	5.0	1	2	272000.0	29	... 50
1	0310	백예린 (Yerin Baek)	99175.00000	664.00000	0	5.0	1	3	174000.0	10	... 50
2	2002	Anne-Marie	382053.76360	967.00000	0	240.0	1	4	59000.0	2	... 50
3	134340	방탄소년단	154052.00000	1337.00000	0	9.0	1	6	784000.0	51	... 50
4	#첫사랑	볼빨간사춘기	126538.26670	462.00000	0	4.0	1	4	272000.0	29	... 50
...	...	...	...	...	...	...	...	...	...	...	...
1012	흔들리는 꽃들 속에서 네 샴푸 향이 느껴진거야	장범준	304576.67710	1462.53125	1	9.0	1	19	108000.0	6	... 50
1013	흔한 이별	허각	81707.66667	161.00000	0	4.0	1	4	60000.0	4	... 50
1014	흰눈	먼데이 키즈 (Monday Kiz)	37072.84615	52.00000	0	7.0	1	2	41000.0	5	... 50
1015	힘든 건 사랑이 아니다	임창정	113153.60000	747.00000	0	6.0	1	9	98000.0	6	... 50
1016	11:11:00 AM	태연 (TAEYEON)	193200.00000	692.00000	0	68.0	1	5	257000.0	17	... 50

가장 오랜기간 차트인한 곡( ‘모든 날, 모든 순간 ‘)을 기준으로 Week1~Week173생성, 각 주에 해당하는 RANK값 대입



# #4.2 데이터 전처리

## 2) 필요없는 피쳐 제거

	title	artist_name	LIKE	replies	first_cnt	rel_chartin	is_kor	title_len	fan_반 올림	chartin_cnt	...	week164
0	25	볼빨간사춘기	37009.00000	190.00000	0	5.0	1	2	272000.0	29	...	50
1	0310	백예린 (Verin Baek)	99175.00000	664.00000	0	5.0	1	3	174000.0	10	...	50
2	2002	Anne-Marie	382053.76360	967.00000	0	240.0	1	4	59000.0	2	...	50
3	134340	방탄소년단	154052.00000	1337.00000	0	9.0	1	6	784000.0	51	...	50
4	#첫사랑	볼빨간사춘기	126538.26670	462.00000	0	4.0	1	4	272000.0	29	...	50
...	...	...	...	...	...	...	...	...	...	...	...	...
1012	흔들리는 곳들 속에서 네 삼푸 향이 느 껴진거 야	장범준	304576.67710	1462.53125	1	9.0	1	19	108000.0	6	...	50
1013	흔한 이 별	허각	81707.66667	161.00000	0	4.0	1	4	60000.0	4	...	50
1014	흰눈	먼데이 키즈 (Monday Kiz)	37072.84615	52.00000	0	7.0	1	2	41000.0	5	...	50
1015	힘든 건 사랑이 아니다	임창정	113153.60000	747.00000	0	6.0	1	9	98000.0	6	...	50
1016	11:11:00 AM	태연 (TAEYEON)	193200.00000	692.00000	0	68.0	1	5	257000.0	17	...	50

```
my_clus = my_clus.drop(['first_cnt', 'rel_chartin', 'chartin_cnt', 'rel_term', 'rel_min_max'], axis=1)
my_clus
```

executed in 60ms, finished 20:29:41 2022-08-09

	LIKE	replies	is_kor	title_len	fan_반 올림	genre_POP	genre_R&B/Soul	genre_국내드 라마	genre_국외영 화	genre_댄스	...	sex_남 성	sex_여 성	sex_...
0	37009.00000	190.00000	1	2	272000.0	0	0	0	0	0	...	0	1	0
1	99175.00000	664.00000	1	3	174000.0	0	1	0	0	0	...	0	1	0
2	382053.76360	967.00000	1	4	59000.0	1	0	0	0	0	...	0	1	0
3	154052.00000	1337.00000	1	6	784000.0	0	0	0	0	0	...	1	0	0
4	126538.26670	462.00000	1	4	272000.0	0	0	0	0	0	...	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1012	304576.67710	1462.53125	1	19	108000.0	0	0	1	0	0	...	1	0	0
1013	81707.66667	161.00000	1	4	60000.0	0	0	0	0	0	...	1	0	0
1014	37072.84615	52.00000	1	2	41000.0	0	0	0	0	0	...	1	0	0
1015	113153.60000	747.00000	1	9	98000.0	0	0	0	0	0	...	1	0	0
1016	193200.00000	692.00000	1	5	257000.0	0	0	0	0	0	...	0	1	0

1017 rows × 32 columns

클러스터링에 필요없는 title, artist\_name 을 제거 & 기간 및 시간과 관련된 first\_cnt, rel\_chartin, chartin\_숫, rel\_term, rel\_min\_max 제거



# #4.2 데이터 전처리

3) 더미변수를 제외한 나머지 피처들에 대해 **Standard Scailing** 적용

```
yes_sca_df = pd.DataFrame(my_clus_scaled, columns = ['like_sca', 'replies_sca', 'title_len_sca', 'fan_sca'])
yes_sca_df
```

executed in 31ms, finished 20:29:46 2022-08-09

	like_sca	replies_sca	title_len_sca	fan_sca
0	-1.007983	-0.511805	-1.051544	0.433303
1	-0.205184	-0.260565	-0.936959	-0.012019
2	3.447857	-0.099962	-0.822374	-0.534591
3	0.503487	0.096154	-0.593204	2.759884
4	0.148180	-0.367634	-0.822374	0.433303
...	...	...	...	...
1012	2.447333	0.162690	0.896398	-0.311930
1013	-0.430753	-0.527176	-0.822374	-0.530047
1014	-1.007159	-0.584951	-1.051544	-0.616385
1015	-0.024667	-0.216572	-0.249450	-0.357371
1016	1.009037	-0.245724	-0.707789	0.365142

1017 rows × 4 columns

4) 스케일링 적용하지 않은 **더미 변수들을 옆에 붙여** clustering 데이터셋 완성

# #4.3 모델 생성

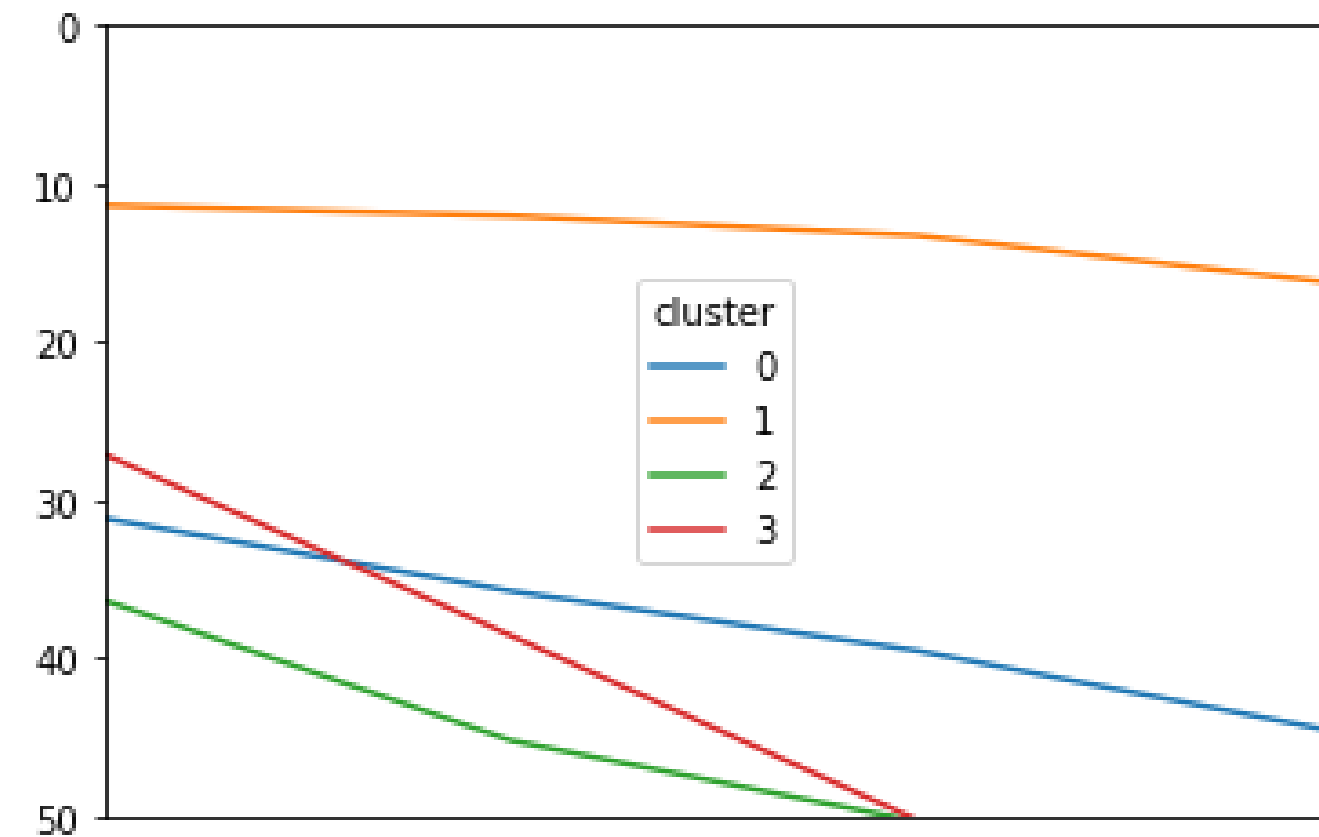
## 모델1) 차원축소(PCA) ▶ K-MEANS 클러스터링

```
def pca_kmeans(scaled_data, n_compo, n_clus, max_it):  
    #pca 수행  
    pca_ = PCA(n_components=n_compo)  
    pca_.fit(scaled_data)  
    my_clus_pca = pca_.transform(scaled_data)  
    #데이터프레임으로 변환  
    df_pca = pd.DataFrame(my_clus_pca)  
    #kmeans 수행  
    kmeans = KMeans(n_clusters=n_clus, init='k-means++', max_iter=max_it, random_state=0)  
    kmeans.fit(df_pca)  
    return kmeans.labels_
```

n\_components = 3, n\_clusters = 4,  
max\_iter = 300

### ▶ 각 클러스터에 해당하는 음원들 순위의 평균을 시각화

Cluster 0 : 완만 하강형, 661개  
Cluster 1 : 안정 유지형, 69개  
Cluster 2 : 비교적 낮은 순위에 있다가 급격 하강형, 189개  
Cluster 3 : 비교적 높은 순위에 있다가 급격 하강형, 98개



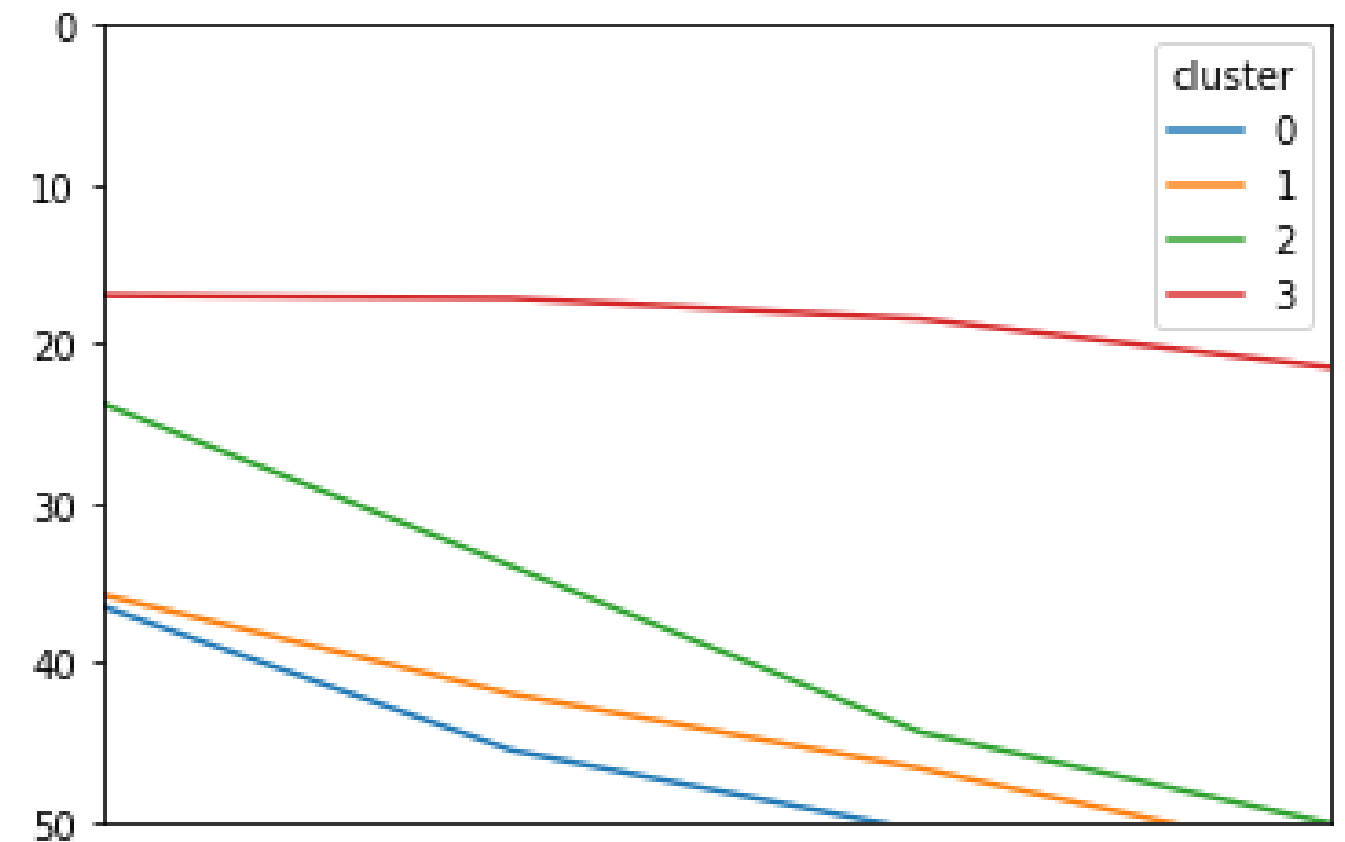
# #4.3 모델 생성

모델2) 차원축소(Truncated PCA) ► K-MEANS 클러스터링

```
from sklearn.decomposition import TruncatedSVD
# n_components = 3
trun_3 = TruncatedSVD(n_components=5)
trun_3.fit(my_clus_scaled)
my_clus_trun = trun_3.transform(my_clus_scaled)
```

```
#KMeans clustering (n_clusters=4)
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4, init='k-means++', max_iter=300, random_state=0)
kmeans.fit(df_trun)
```

실루엣계수: 0.16



Cluster 0 : 비교적 낮은 순위에 있다가 급격 하강형, 162개  
Cluster 1 : 완만 하강형, 519개  
Cluster 2 : 비교적 높은 순위에 있다가 급격 하강형, 111개  
Cluster 3 : 안정 유지형, 225개

# #4.3 모델 생성

모델3) 차원축소(PCA) ▶ GMM(GaussianMixture)클러스터링 ▶ 각 클러스터에 해당하는 음원들 순위의 평균을 시각화

```
my_data = pd.read_csv('rank99.csv')
#clustering에 사용할 변수 분리 (title, artist_name, week@변수 제외)
my_clus = my_data.iloc[:, 2:39]

my_clus = my_clus.drop(['first_cnt', 'rel_chartin', 'chartin_cnt', 'rel_term', 'rel_min_max'], axis=1)
no_scale = my_clus.drop(['LIKE', 'replies', 'title_len', 'fan_반올림'], axis=1)
yes_scale = my_clus[['LIKE', 'replies', 'title_len', 'fan_반올림']]

from sklearn.preprocessing import StandardScaler
stan_scaler = StandardScaler()
my_clus_scaled = stan_scaler.fit_transform(yes_scale)
yes_sca_df = pd.DataFrame(my_clus_scaled, columns = ['like_sca', 'replies_sca', 'title_len_sca', 'fan_sca'])
my_clus_scaled = pd.concat([yes_sca_df, no_scale], axis=1)
my_clus_scaled
```

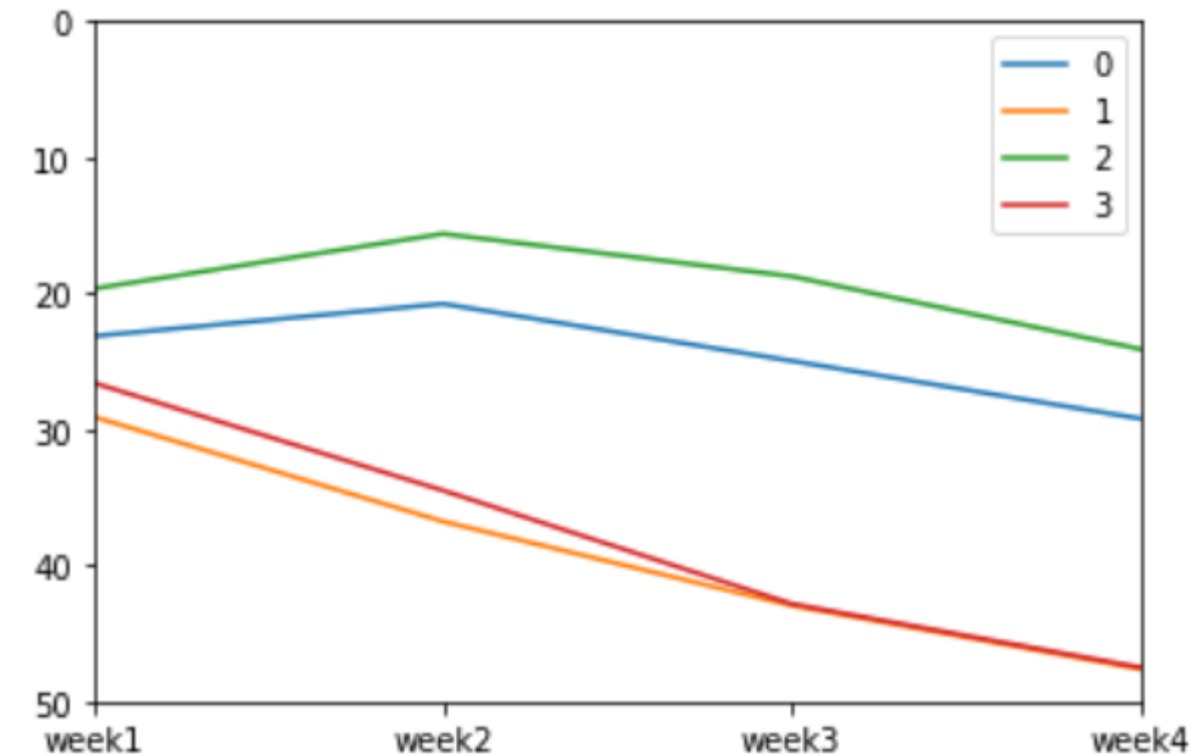
```
def make_pca(n_component, data):
    pca=PCA(n_components=n_component)
    pca.fit(data)
    top_pca=pca.transform(data)
    return top_pca
#print(top_pca.shape)
```

Min-Max scaler로 진행했을 때와 동일한 결과

```
ranknc=my_data_rank.copy()
gmm=GaussianMixture(n_components=4,verbose=False, random_state=0)
ranknc['cluster']=gmm.fit_predict(pca99)

for i in range(4):
    plt.plot(ranknc.groupby('cluster').mean().transpose()[i], label=(str)(i))
plt.legend()
plt.ylim([50, 0])
plt.xlim([0, 3])

plt.show()
```



Cluster 0 : 상승 후 완만 하강형, 412개

Cluster 1 : 비교적 낮은 순위에 있다가 급격 하강형, 309개

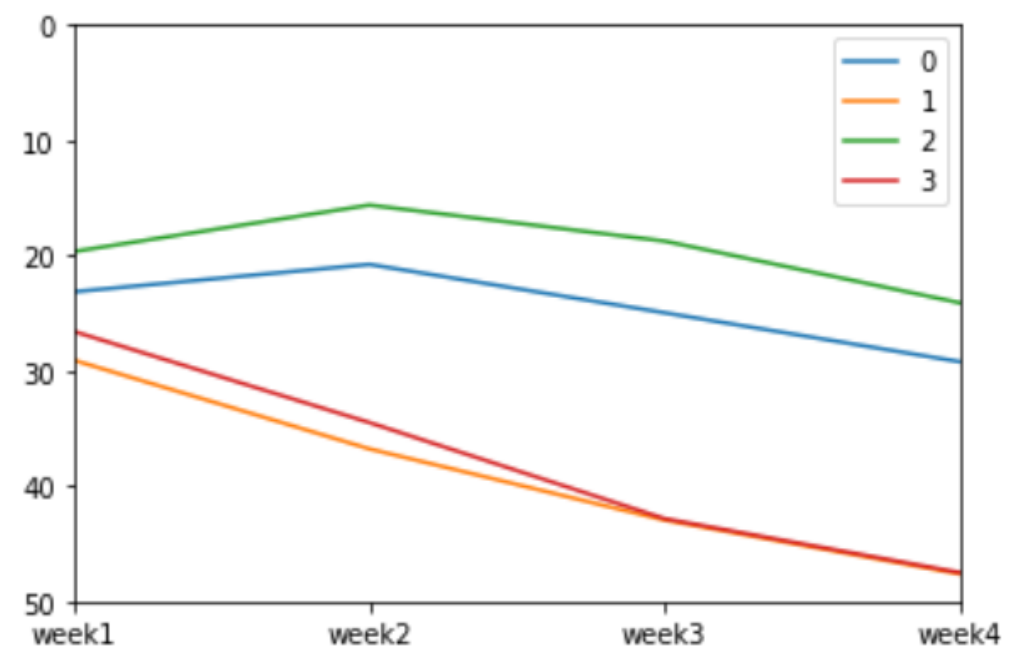
Cluster 2 : 상승 후 완만하강형, 241 개

Cluster 3 : 비교적 높은 순위에 있다가 급격 하강형, 55개

# #4.3 모델 생성

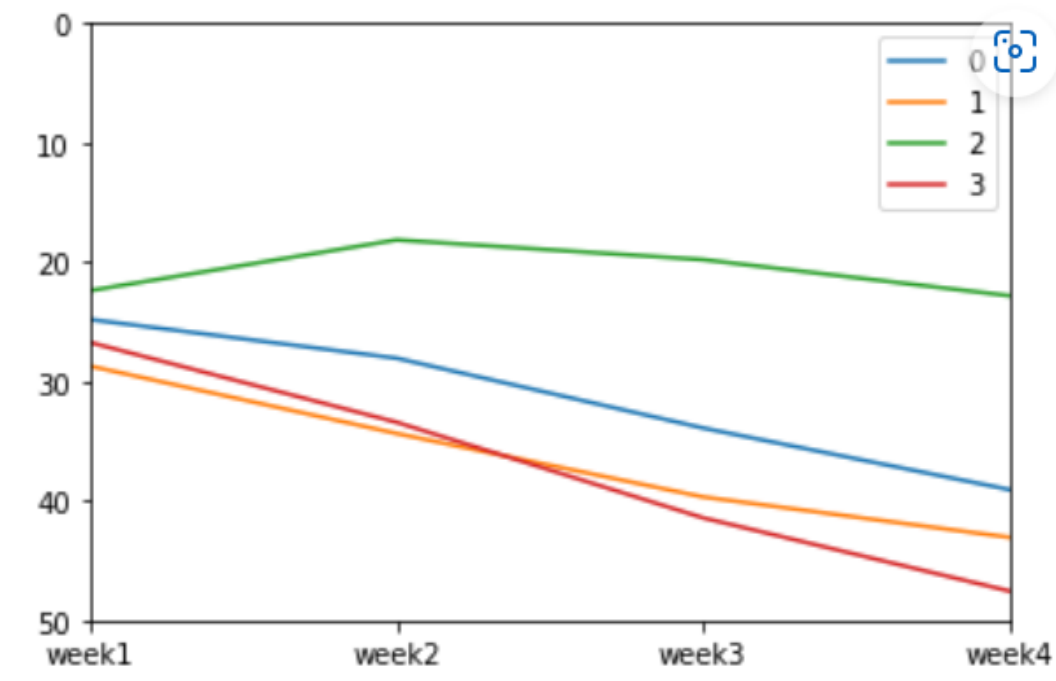
모델4) 차원축소(PCA) ▶ TIMESEIRESKMEANS 클러스터링 ▶ 각 클러스터에 해당하는 음원들 순위의 평균을 시각화

```
rankc=rankc=my_data_rank.copy()
kmeans=TimeSeriesKMeans(n_clusters=4, verbose=False, random_state=0)
rankc['cluster']=kmeans.fit_predict(pca99)
for i in range(4):
    plt.plot(rankc.groupby('cluster').mean().transpose()[i], label=(str)(i))
plt.legend()
plt.ylim([50, 0])
plt.xlim([0, 3])
plt.show()
```



- Cluster 0 : 상승 후 완만 하강형, 485개
- Cluster 1 : 비교적 낮은 순위에 있다가 급격 하강형, 286개
- Cluster 2 : 상승 후 완만하강형, 175 개
- Cluster 3 : 비교적 높은 순위에 있다가 급격 하강형, 71개

Min-Max scaler로 진행했을 때



- Cluster 0 : 완만 하강형, 326개
- Cluster 1 : 비교적 낮은 순위에 있다가 급격 하강형, 312개
- Cluster 2 : 상승 후 안정유지형, 244 개
- Cluster 3 : 비교적 높은 순위에 있다가 급격 하강형, 135개

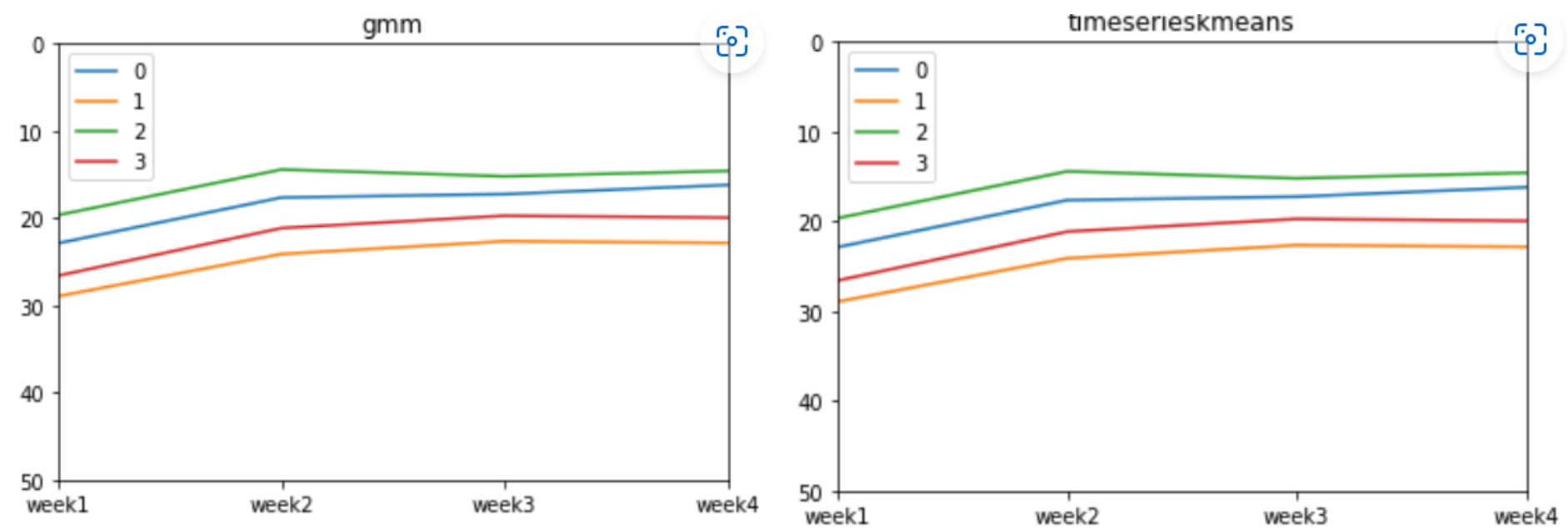
# #4.3 모델 생성

모델5) 차트인 이후 rank=np.NaN처리 ▶ 차원축소(PCA) ▶ TIMESEIRESKMEANS & GMM

	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	...
0	45.0	46.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
2	42.0	38.0	27.0	22.0	17.0	13.0	10.0	7.0	5.0	3.0	...
3	25.0	35.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
4	3.0	2.0	4.0	5.0	7.0	10.0	12.0	14.0	19.0	22.0	...
...	...	...	...	...	...	...	...	...	...	...	...

Rank=99 에서 rank=NaN으로 변경

▶ 각 클러스터에 해당하는 음원들 순위의 평균을 시각화



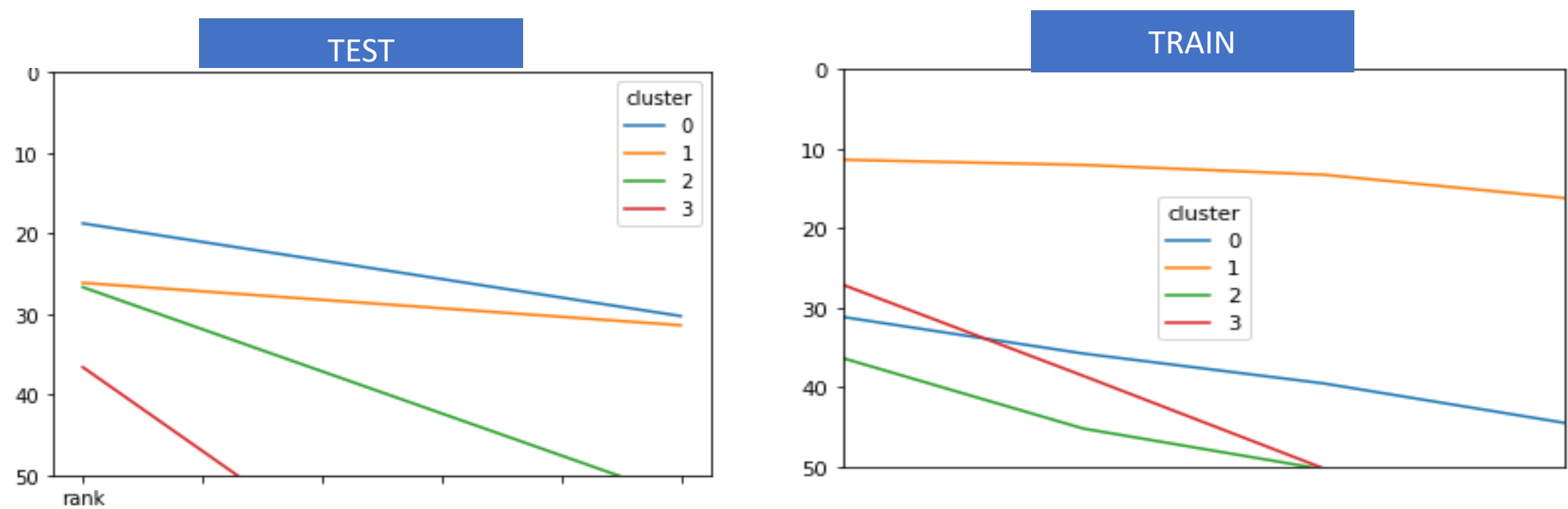
모델4와 클러스터 분포 개수가 동일하다  
Cluster0~3 : 상승 후 안정유지형

Min-Max scaler로 진행했을 때와 동일한 결과

# #4.4 결론 및 시연

최종모델 선정 기준 : 그래프 시각화의 분류 적절성 & 실루엣 계수  
시연 : 7월 1주차 데이터를 바탕으로 최종모델을 수행했을 때, 4주 뒤인, 8월 1주차 순위 데이터가 어떻게 변화할지 시각화

**모델1** ( StandardScaler ▶ PCA ▶ K-MEANS) 은 그래프 시각화 분류의 적절성으로 선정

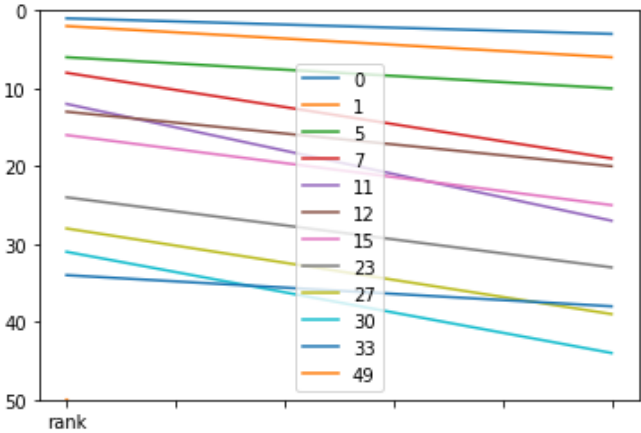


실루엣계수 0.19  
Cluster 0 : 완만하강형(0), 23개  
Cluster 1 : 안정유지형(1), 12개  
Cluster 2 : 비교적 높은 순위에 있다가 급격 하강형(2), 9 개  
Cluster 3 : 비교적 낮은 순위에 있다가 급격 하강형(3), 5개

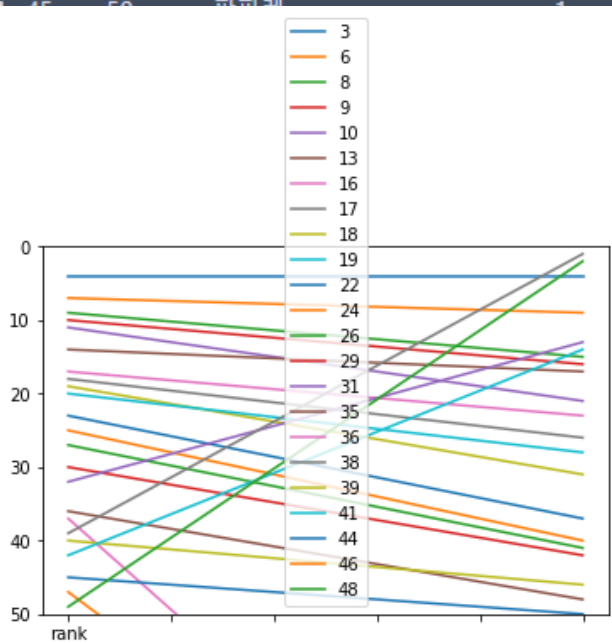


# #4.4 결론 및 시연

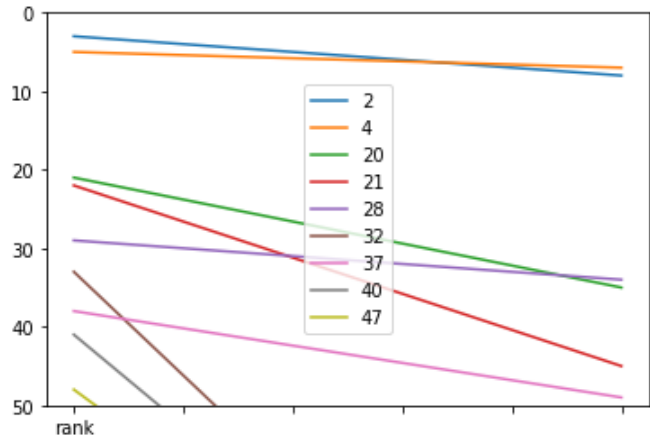
	rank	rank_2	title	cluster
0	1	3	LOVE DIVE	0
1	2	6	TOMBOY	0
5	6	10	사랑인가 봐	0
7	8	19	봄여름가을겨울 (Still Life)	0
11	12	27	Feel My Rhythm	0
12	13	20	사랑은 늘 도망가	0
15	16	25	신호등	0
23	24	33	STAY	0
27	28	39	Next Level	0
30	31	44	Weekend	0
33	34	38	너의 모든 순간	0
49	50	99	OHAYO MY NIGHT	0



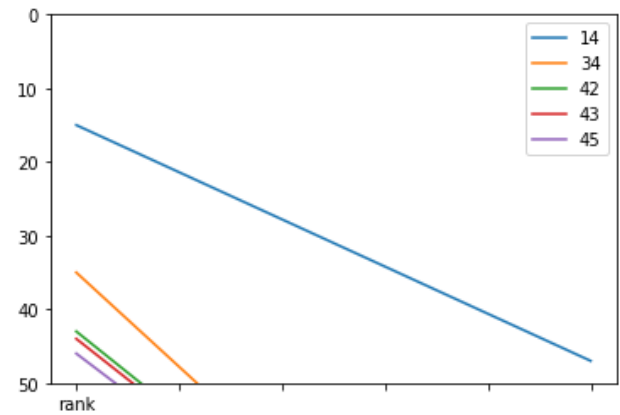
	rank	rank_2	title	cluster
3	4	4	POP!	1
6	7	9	나의 X에게	1
8	9	15	LOVE me	1
9	10	16	FEARLESS	1
10	11	21	취중고백	1
13	14	17	내가 아니라도	1
16	17	23	ELEVEN	1
17	18	26	우리들의 블루스	1
18	19	31	아무래도 난	1
19	20	28	다정히 내 이름을 부르면	1
22	23	37	MY BAG	1
24	25	40	INVU	1
26	27	41	늦은 밤 헤어지긴 너무 아쉬워	1
29	30	42	듣고 싶을까	1
31	32	13	그라데이션	1
35	36	48	너를 생각해	1
36	37	99	ZOOM	1
38	39	1	그때 그 순간 그대로 (그그그)	1
39	40	46	다시 만날 수 있을까	1
41	42	14	Love story	1
44	45	50	판타지	1
46	47	99	사랑은 늘 도망가	1
48	49	99	OHAYO MY NIGHT	1



	rank	rank_2	title	cluster
2	3	8	That That (prod. & feat. SUGA of BTS)	2
4	5	7	정이라고 하자 (Feat. 10CM)	2
20	21	35	GANADARA (Feat. 아이유)	2
21	22	45	Left and Right (Feat. Jung Kook of BTS)	2
28	29	34	That's Hilarious	2
32	33	99	고백하는 취한밤에 (Prod. 2soo)	2
37	38	49	SMILEY (Feat. BIBI)	2
40	41	99	회전목마 (Feat. Zion.T, 원슈타인) (Prod. Slom)	2
47	48	99	리무진 (Feat. MINO) (Prod. GRAY)	2



	rank	rank_2	title	cluster
14	15	47	Yet To Come	3
34	35	99	Butter	3
42	43	99	Permission to Dance	3
43	44	99	드라마	3
45	46	99	내 손을 잡아	3



Cluster0 : 순위가 완만하게 하강한다

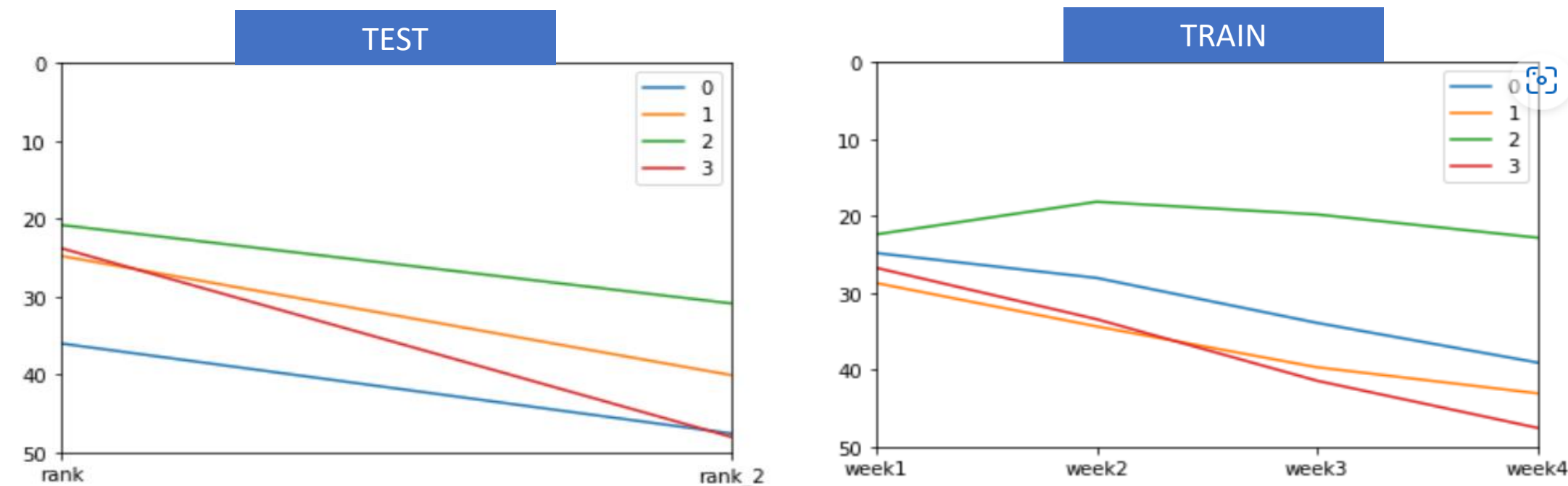
Cluster1 : 순위가 안정하게 유지한다

Cluster2 : 비교적 높은 순위에서 급격하강한다

Cluster3 : 비교적 낮은 순위에서 급격하강한다

# #4.4 결론 및 시연

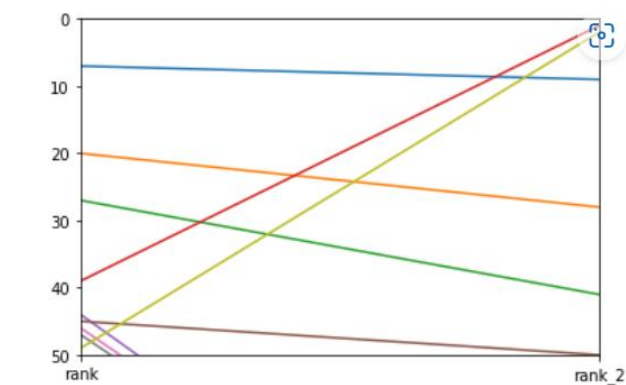
**모델4** ( MinMaxScaler ▶ PCA) ▶ TIMESEIRESKMEANS) 은 실루엣 계수의 우수성으로 선정



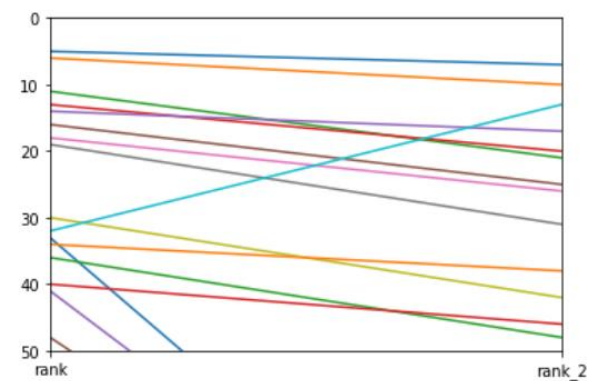
실루엣 계수 0.62  
Cluster 0 : 완만하강형(0), 15개  
Cluster 1 : 비교적 낮은 순위에 있다가 급격 하강형(1) 13개  
Cluster 2 : 안전유지-완만하강형(2), 13 개  
Cluster 3 : 비교적 높은 순위에 있다가 급격 하강형(3) ,9개

# #4.4 결론 및 시연

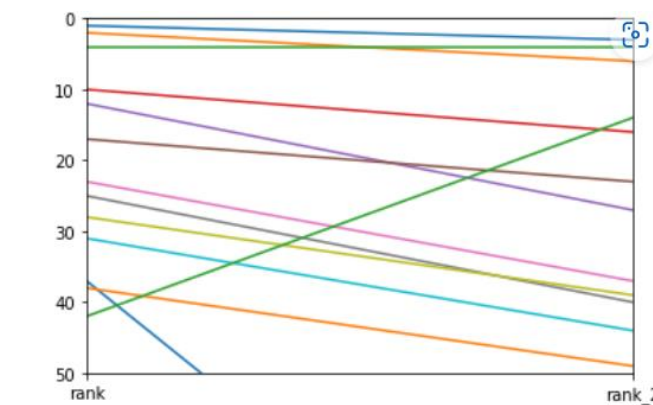
rank	rank_2		title	cluster
6	7	9	나의 X에게	0
19	20	28	다정히 내 이름을 부르면	0
26	27	41	늦은 밤 헤어지긴 너무 아쉬워	0
38	39	1	그때 그 순간 그대로 (그그그)	0
43	44	99	드라마	0
44	45	50	팡파레	0
45	46	99	내 손을 잡아	0
46	47	99	언제나 사랑해	0
48	49	2	보고싶었어	0



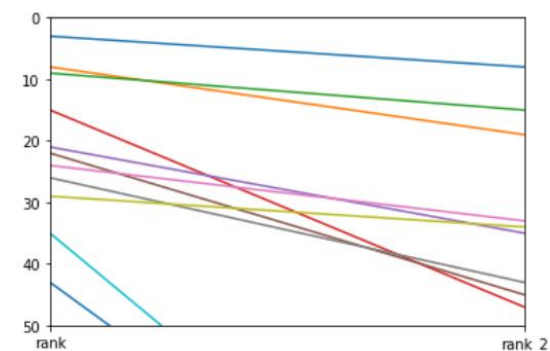
rank	rank_2		title	cluster
4	5	7	정이라고 하자 (Feat. 10CM)	1
5	6	10	사랑인가 봐	1
10	11	21	취중고백	1
12	13	20	사랑은 늘 도망가	1
13	14	17	내가 아니라도	1
15	16	25	신호등	1
17	18	26	우리들의 블루스	1
18	19	31	아무래도 난	1
29	30	42	듣고 싶을까	1
31	32	13	그라데이션	1
32	33	99	고백하는 취한밤에 (Prod. 2soo)	1
33	34	38	너의 모든 순간	1
35	36	48	너를 생각해	1
39	40	46	다시 만날 수 있을까	1
40	41	99	회전목마 (Feat. Zion.T, 원슈타인) (Prod. Slom)	1
47	48	99	리무진 (Feat. MINO) (Prod. GRAY)	1



rank	rank_2	title	cluster	
0	1	3	LOVE DIVE	2
1	2	6	TOMBOY	2
3	4	4	POPI	2
9	10	16	FEARLESS	2
11	12	27	Feel My Rhythm	2
16	17	23	ELEVEN	2
22	23	37	MY BAG	2
24	25	40	INVU	2
27	28	39	Next Level	2
30	31	44	Weekend	2
36	37	99	ZOOM	2
37	38	49	SMILEY (Feat. BIBI)	2
41	42	14	Love story	2



rank	rank_2	title	cluster	
2	3	8	That That (prod. & feat. SUGA of BTS)	3
7	8	19	봄여름가을겨울 (Still Life)	3
8	9	15	LOVE me	3
14	15	47	Yet To Come	3
20	21	35	GANADARA (Feat. 아이유)	3
21	22	45	Left and Right (Feat. Jung Kook of BTS)	3
23	24	33	STAY	3
25	26	43	Dynamite	3
28	29	34	That's Hilarious	3
34	35	99	Butter	3
42	43	99	Permission to Dance	3
49	50	99	OHAYO MY NIGHT	3



Cluster0 : 대부분 순위 완만하강형으로 잘 분류되었다

Cluster1 : 비교적 낮은 순위에서 급격하강한다 순위가 완만하게 하강한다

Cluster2 : 순위가 완만하게 하강, 안정유지한다

Cluster3 : 비교적 높은 순위에서 급격하강한다

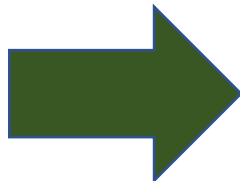
## #5 결론 및 제언



# #5.1 새로운 데이터로 예측

: 수집한 데이터셋 기간(2017년 1월 첫째 주 ~ 2022년 7월 첫째 주)에 전혀 속하지 않는 새로운 음원의 순위를 예측해보면!

term	rank	title	artist	LIKE	replies
2022082	2	Attention	NewJeans	93301	1603
first_cnt	rel_chartin	rel_term	title_len	previous_rank	chartin_cnt
0	6	13	9	12	3
fan_반올림	is_kor	season	sex	act_type	main_genre
29000	0	여름	여성	그룹	댄스



2	▲ 10		Attention NewJeans
예측 등수			
10.9			

: 실제 등수와는 다소 다른 결과를 얻음

4주 후 음원 순위 추이를 예측해보면!

: 클러스터2에 해당하는 것으로, 비교적 높은 순위에 있다 급격하강할 것으로 예측됨

# #5.2 결론 및 제언

## 의의

- 데이터 수집부터 전처리, 가공까지 주도적으로 해냄~~~~!!! (㉸)
- 현재의 음원 순위를 성공적으로 예측함.
- 4주 뒤 순위 변화 추이 예측이라는 기존의 음원 사이트에서 제공되지 않던 서비스를 개발함.
- 음악 업계의 종사자들이 미래 상황에 대해 대처할 수 있는 유의미한 정보 제공이 가능할 것으로 기대됨.

## 개선 방향

- 소속사에 대한 정보 등 유의미한 변수가 부족함.
- 역주행 등 급상승하는 곡에 대한 예측이 어려움.

# THANK YOU

