

[Programmers DE6 Team 5Re]

# Music Ranking Charts

- | 팀장 한기중
- | 팀원 오주언, 장현우, 최정우
- | Github: [DE6-ProjectPart3-5Re](#)

# CONTENTS

01

팀원소개/역할분담

02

프로젝트 개요

03

시스템 구조

04

데이터흐름

05

대시보드

06

개선점 및 결과

# 팀원소개/역할분담

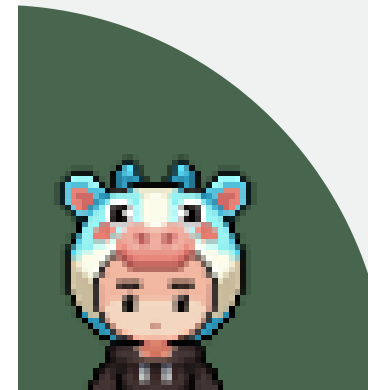
## 팀장 한기중

- Snowflake구축
- genie ELT 구축
- 차트 ELT, 시각화 작업



## 팀원 최정우

- genie ELT 구축
- 차트 ELT, 시각화 작업



## 팀원 오주언

- genie ELT 구축
- 차트 ELT, 시각화 작업



## 팀원 장현우

- genie ELT 구축
- 차트 ELT, 시각화 작업



# 프로젝트 개요



genie

Bugs!

billboard



다양한 음악의 랭킹차트데이터를 모아,  
시기별 선호 음악 모니터링

국내음원 차트와 외국음원 차트의 순위를 비교해  
국내외 음악 선호도 차이 비교

Docker,Airflow 등 다양한 기술을 활용한 경험

# 프로젝트를 개요



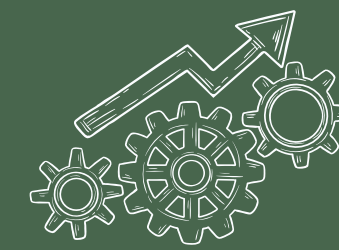
## Docker 활용

Docker를 활용하며,  
기존 개발 환경과 Docker  
개발환경의 차이점과  
운영방식 등에 대한  
다양한 경험을 위해



## Airflow 기반의 파이프라인 구성

Airflow의 전반적인  
처리방식과 활용법 등  
Airflow 학습을 위해



## 기술적인 경험

Docker, Airflow 등  
평소에 사용해 보지 못한  
기술을 경험하며 좀 더  
성장하기 위해

# 사용 기술 스택

데이터 적재/ELT



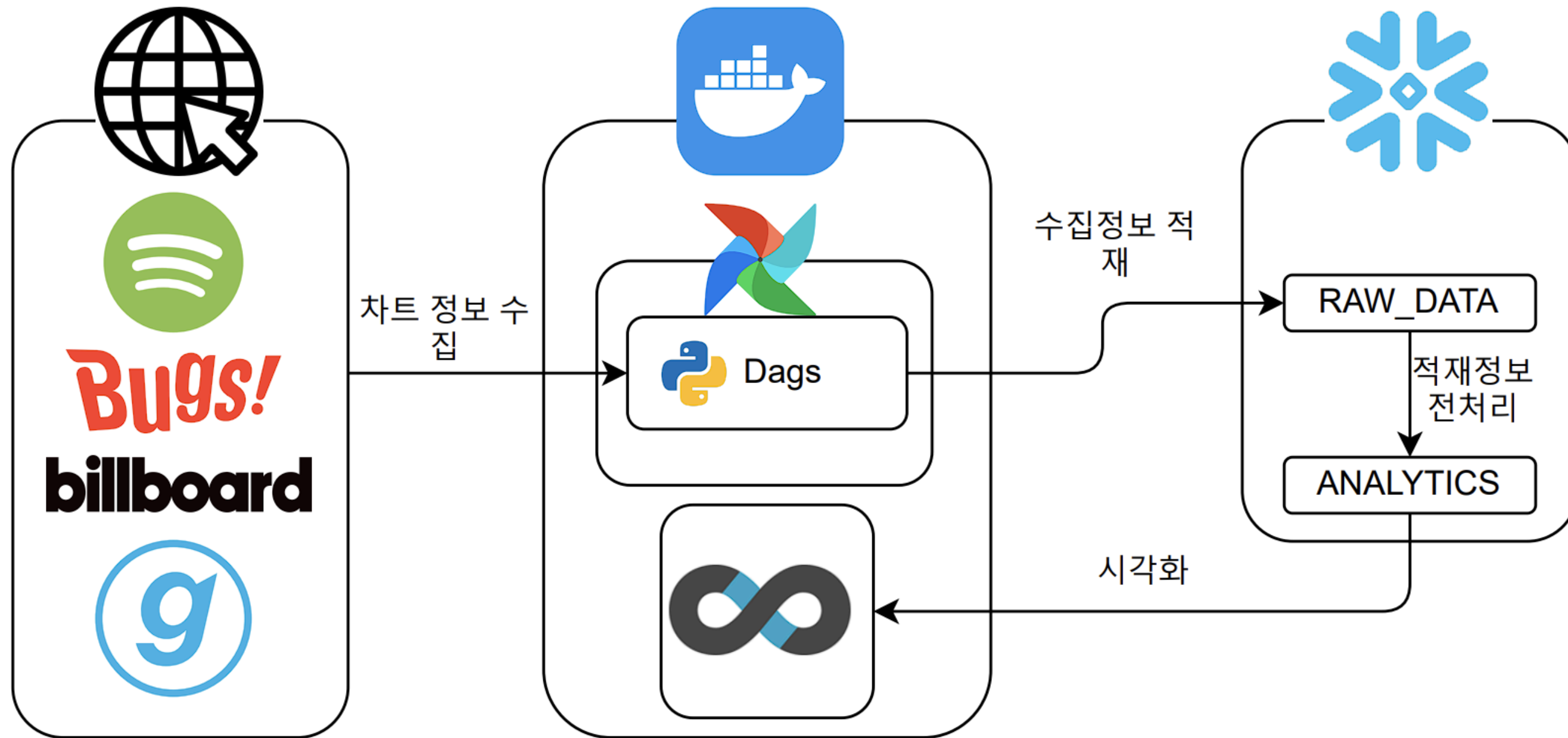
데이터 ELT/시각화



협업 tool



# 시스템 구조



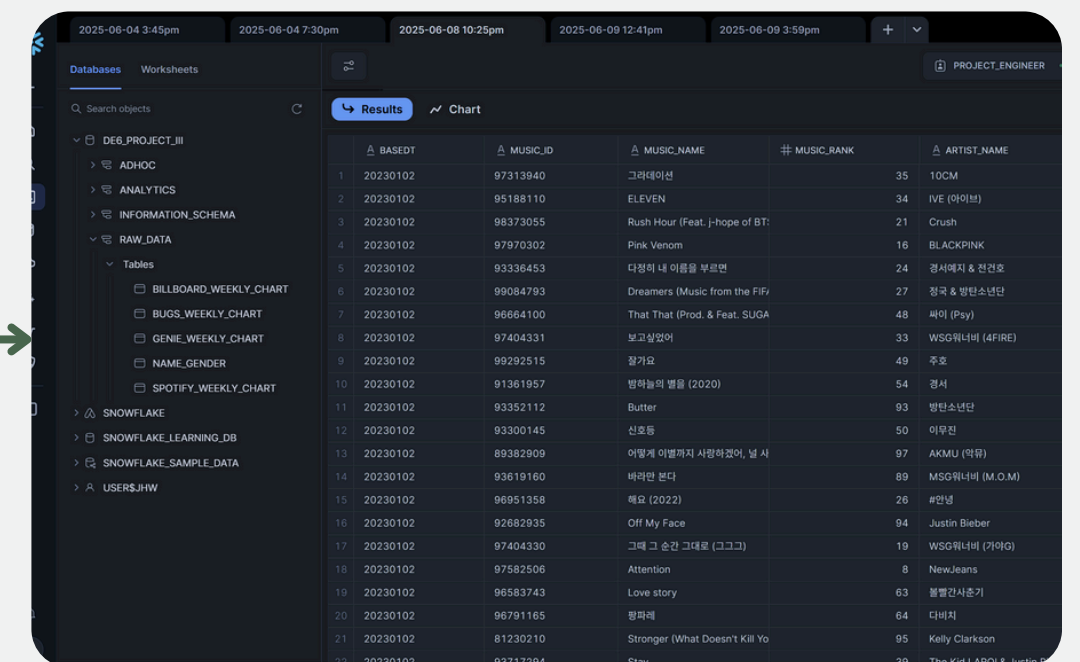
# 데이터 흐름

```
def scrape_billboard(date):  
    url = f"https://www.billboard.com/charts/hot-100/{date}"  
    headers = {"User-Agent": "Mozilla/5.0"}  
    response = requests.get(url, headers=headers)  
    soup = BeautifulSoup(response.text, "html.parser")  
    chart_items = soup.select("div.o-chart-results-list-row-container")  
  
    result = []  
    for item in chart_items[:100]:  
        try:  
            rank = int(item.select_one("span.c-label.a-font-primary-bold-1").text.strip())  
            title = item.select_one("h3.c-title").text.strip()  
            artist = item.select_one("span.c-label.a-no-truncate.a-font-primary-s").text.strip()  
  
            result.append({  
                "basedt": basedt_str,  
                "music_rank": rank,  
                "music_name": title,  
                "artist_name": artist  
            })  
        except Exception:  
            continue  
    return result  
  
logging.info(f"{chart_date_str} 기준 주간 차트 수집 중...")  
return scrape_billboard(chart_date_str)
```

크롤링을 통해 데이터 수집

```
def load_to_snowflake(records):  
    hook = SnowflakeHook(snowflake_conn_id="snowflake_conn")  
    conn = hook.get_conn()  
    cur = conn.cursor()  
    try:  
        cur.execute("BEGIN")  
  
        if records:  
            basedt = records[0]["basedt"]  
            cur.execute(f"DELETE FROM DE6_PROJECT_III.RAW_DATA.BILLBOARD_WEEKLY_CHART WHERE BASEDT = '{basedt}'")  
  
            for row in records:  
                sql = """  
                    INSERT INTO DE6_PROJECT_III.RAW_DATA.BILLBOARD_WEEKLY_CHART  
                    (BASEDT, MUSIC_ID, MUSIC_NAME, MUSIC_RANK, ARTIST_NAME)  
                    VALUES (%s, %s, %s, %s, %s)  
                    """  
                cur.execute(sql, (  
                    row["basedt"],  
                    None,  
                    row["music_name"],  
                    row["music_rank"],  
                    row["artist_name"]  
                ))  
  
            cur.execute("COMMIT")  
            logging.info(f"len(records))건 적재 완료 (basedt = {records[0]['basedt']})")  
        except Exception as e:  
            logging.error(f"적재 실패: {e}")  
            cur.execute("ROLLBACK")  
            raise  
    finally:  
        cur.close()  
        conn.close()
```

크롤링 데이터 적재



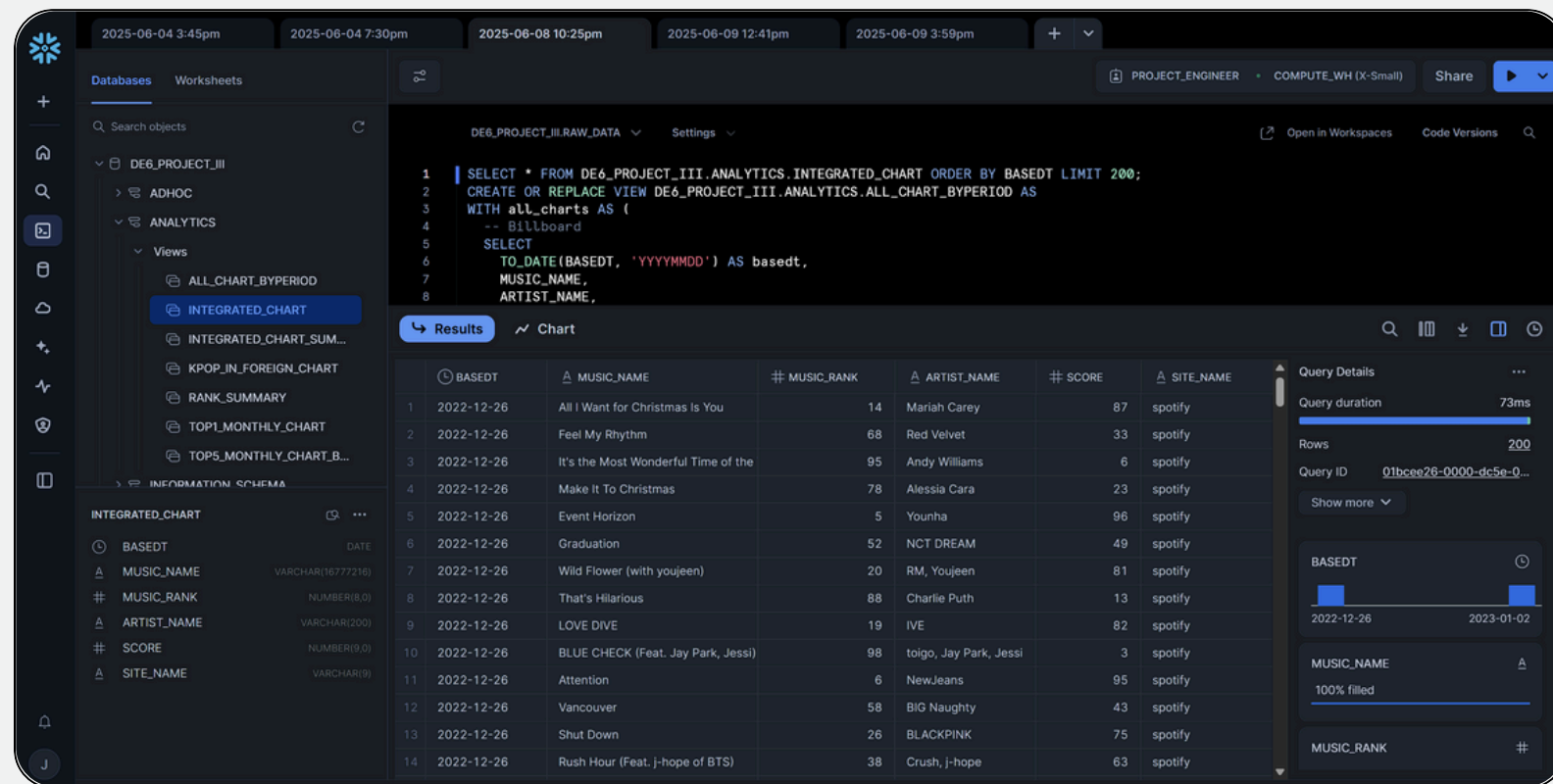
BASEDT	MUSIC_ID	MUSIC_NAME	MUSIC_RANK	ARTIST_NAME
20230102	97313940	그래데이션	35	10CM
20230102	95188110	ELEVEN	34	IVE (아이브)
20230102	98373055	Rush Hour (Feat. J-hope of BT)	21	Crush
20230102	97970302	Pink Venom	16	BLACKPINK
20230102	93336453	다정히 내 이름을 부르면	24	경사제지 & 전진호
20230102	99084793	Dreamers (Music from the F/F)	27	장국 & 영만소년단
20230102	96664100	That That (Prod. & Feat. SUGA)	48	싸이 (Psy)
20230102	97404331	보고싶었어	33	WSG워너비 (4FIRE)
20230102	99292515	잘가요	49	주요
20230102	91361957	밤하늘의 별들 (2020)	54	경서
20230102	93352112	Butter	93	방탄소년단
20230102	93300145	신호등	50	이무진
20230102	89382909	어떻게 아별까지 사랑했어, 널 시	97	AKMU (악뮤)
20230102	93619160	바라만 본다	89	MSG워너비 (M.O.M)
20230102	96951358	해요 (2022)	26	#안녕
20230102	92682935	Off My Face	94	Justin Bieber
20230102	97404330	그때 그 순간 그대로 (그그그)	19	WSG워너비 (가이)
20230102	97582506	Attention	8	NewJeans
20230102	96583743	Love story	63	불멸간사초기
20230102	96791165	황하래	64	다비치
20230102	81230210	Stronger (What Doesn't Kill Yo	95	Kelly Clarkson
20230102	97313940	Star	39	Tha Kidz (ADN & Justin)

SNOWFLAK  
RAW\_DATA에 적재

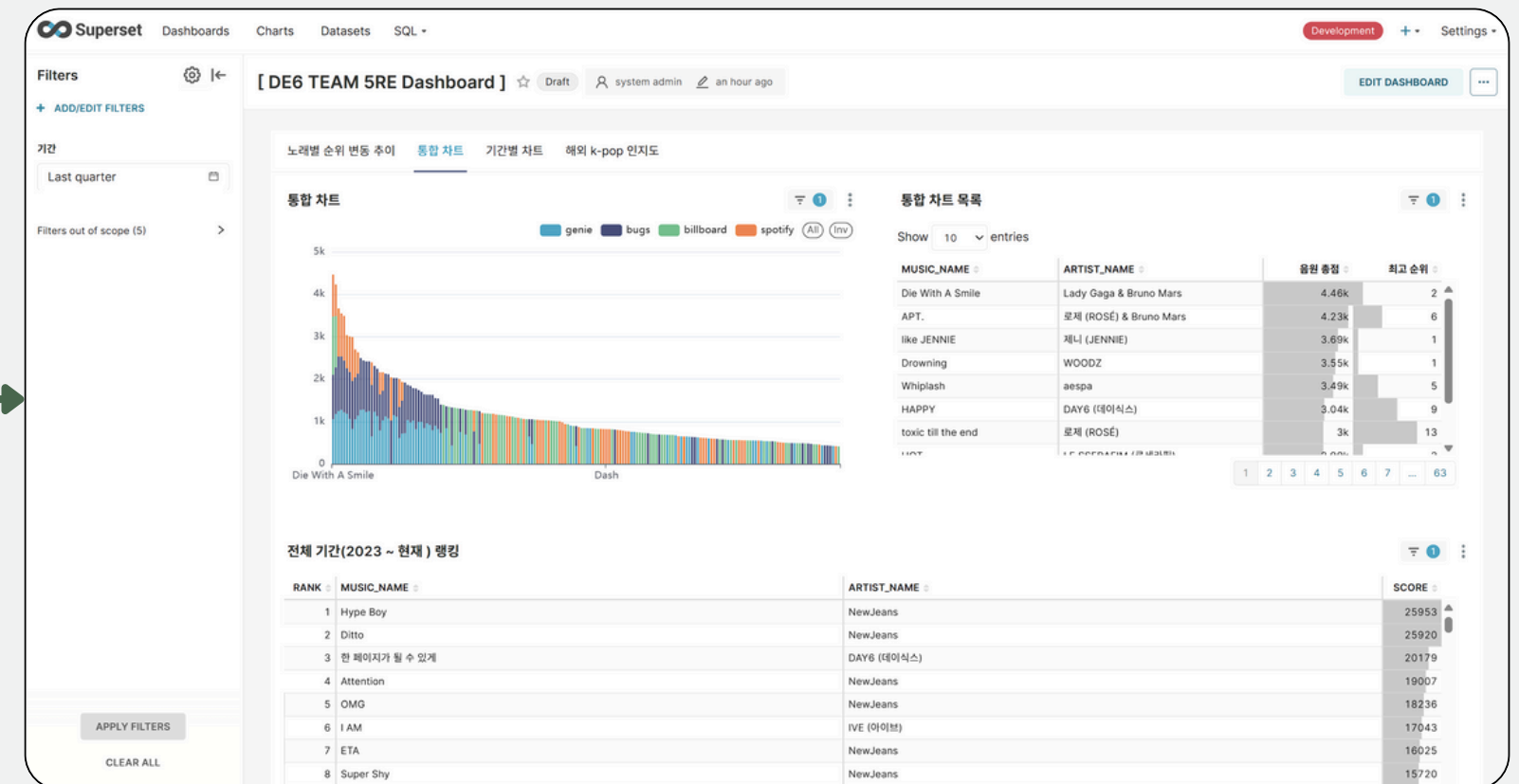
airflow dag로 구성해서 주기적으로 데이터 수집



# 데이터 흐름



SQL을 활용해 시각화 목적에 맞게  
ANALYTICS 스키마에 View 구성.



VIEW기반으로 SUPERSET에서 dashboard 작성

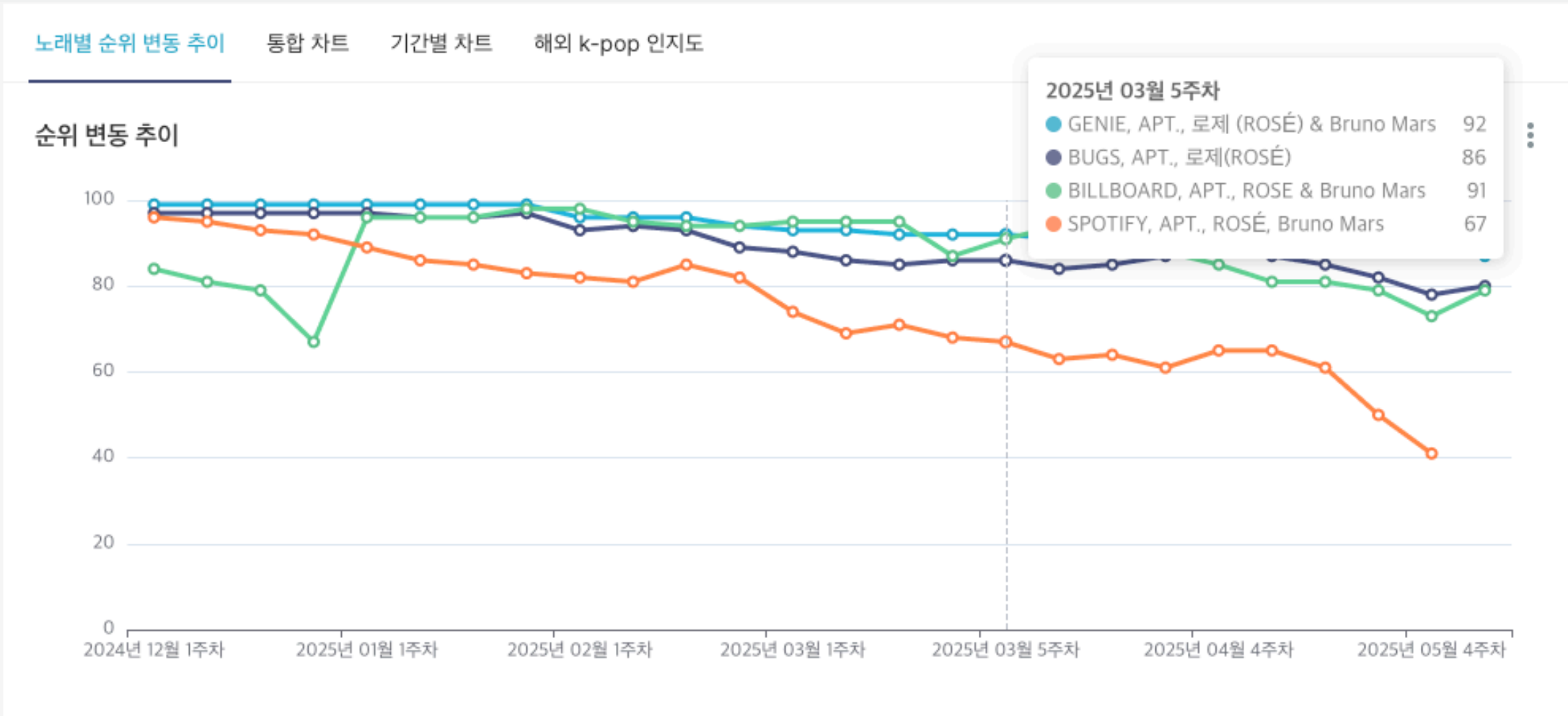
# 대시보드

## 노래별 순위 변동 추이

.....

특정 곡의 음원 사이트(billboard, bugs, spotify, genie)별 순위 변동 추이를 확인할 수 있습니다.

곡 명을 기준으로 필터링이 가능합니다.

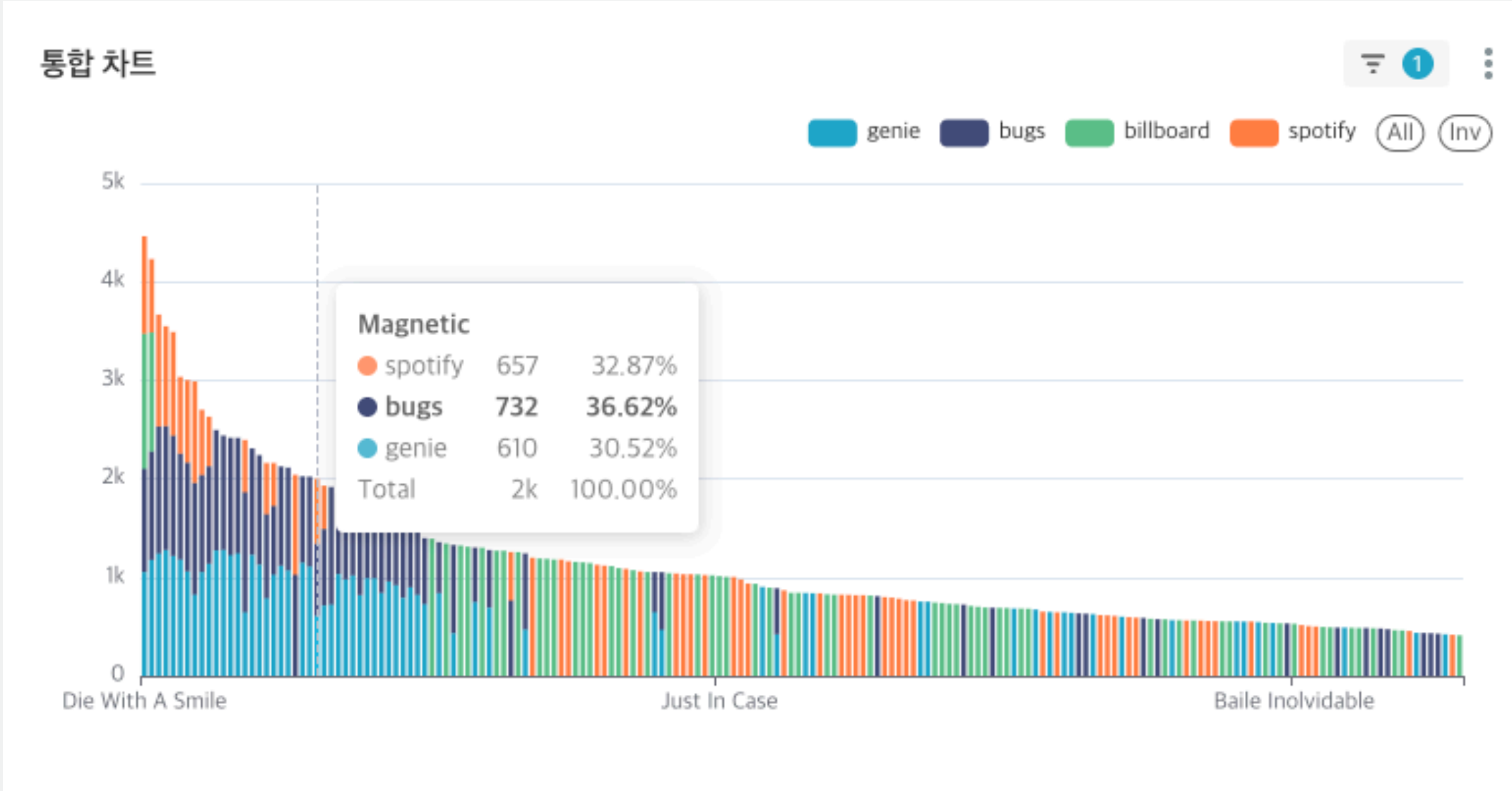


# 대시보드

## 통합 차트 - 막대그래프

수집한 4개의 음원 사이트의 순위 정보를 합산한 정보를 막대 그래프로 시각화했습니다.

어떤 차트에서 어떤 음원 점수를 기록했는지를 알 수 있습니다.



# 대시보드

## 통합 차트 - 테이블

수집한 4개의 음원 사이트의 순위 정보를 합산한 정보를 테이블로 나타냈습니다.

기간 필터링을 통해 특정 기간동안의 통합 차트 순위를 알 수 있습니다.

통합 차트 목록

Show

10

▼

entries

MUSIC_NAME	ARTIST_NAME	음원 총점	최고 순위
Die With A Smile	Lady Gaga & Bruno Mars	4.46k	2
APT.	로제 (ROSÉ) & Bruno Mars	4.23k	6
like JENNIE	제니 (JENNIE)	3.69k	1
Drowning	WOODZ	3.55k	1
Whiplash	aespa	3.49k	5
HAPPY	DAY6 (데이식스)	3.04k	9
toxic till the end	로제 (ROSÉ)	3k	13
HOT	LE SSERAFIM (르세라핌)	2.88k	2

1

2

3

4

5

6

7

...

63

# 대시보드

## 통합 차트 - 전체기간 테이블

.....

수집한 4개의 음원 사이트의 순위 정보를 합산한 정보를 테이블로 나타냈습니다.

데이터가 수집된 전체기간의 순위를 확인할 수 있습니다.

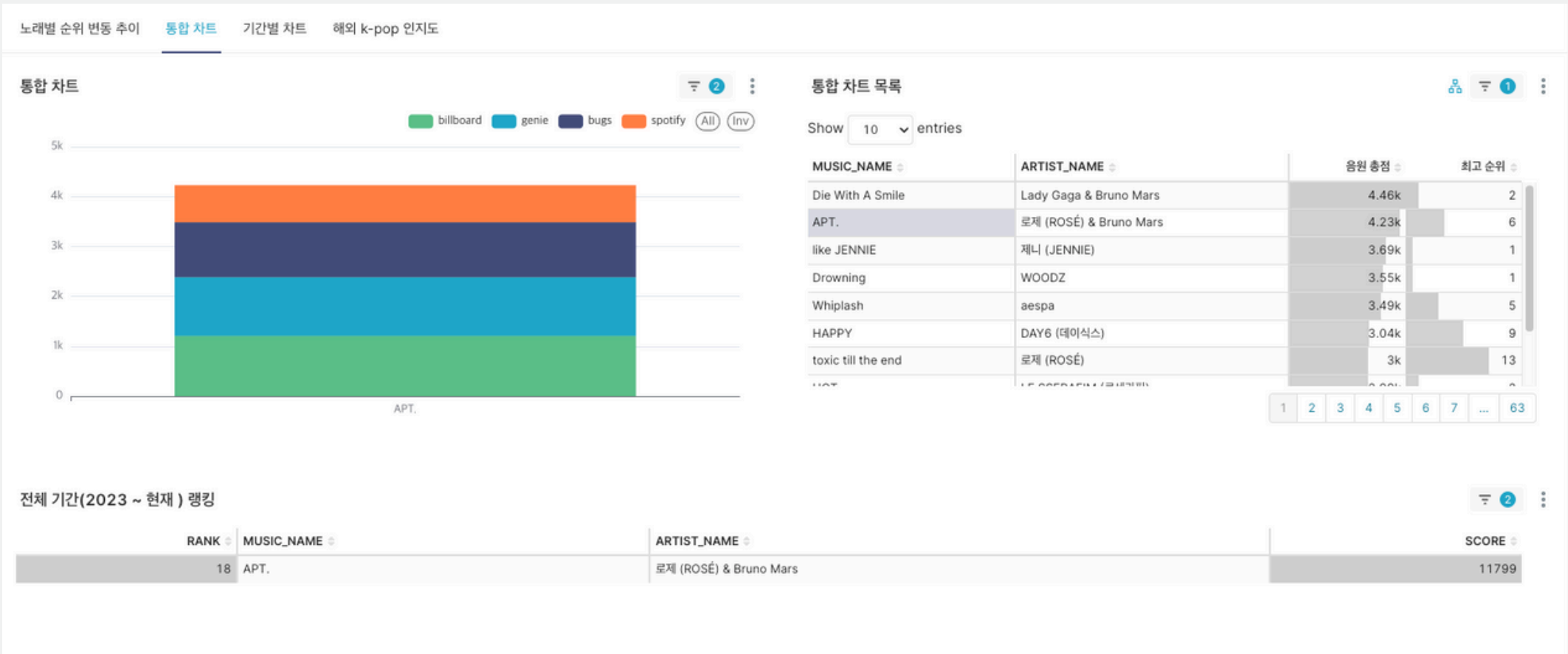
전체 기간(2023 ~ 현재 ) 랭킹				
RANK	MUSIC_NAME	ARTIST_NAME	SCORE	
1	Hype Boy	NewJeans	25953	
2	Ditto	NewJeans	25920	
3	한 페이지가 될 수 있게	DAY6 (데이식스)	20179	
4	Attention	NewJeans	19007	
5	OMG	NewJeans	18236	
6	I AM	IVE (아이브)	17043	
7	ETA	NewJeans	16025	
8	Super Shy	NewJeans	15720	
9	예뻐어	DAY6 (데이식스)	15559	
10	헤어지자 말해요	박재정	14778	

# 대시보드

## 통합 차트 필터링

통합 차트 목록에서 특정 곡을 클릭하면, 해당 곡이 어떤 차트에서 높은 점수를 기록했는지와 전체 기간 랭킹을 확인할 수 있습니다.

오른쪽 차트는 APT. 곡이 최근 3개월 통합차트에선 2위를 기록했고, 전체기간 랭킹에선 18위를 기록한 모습입니다.

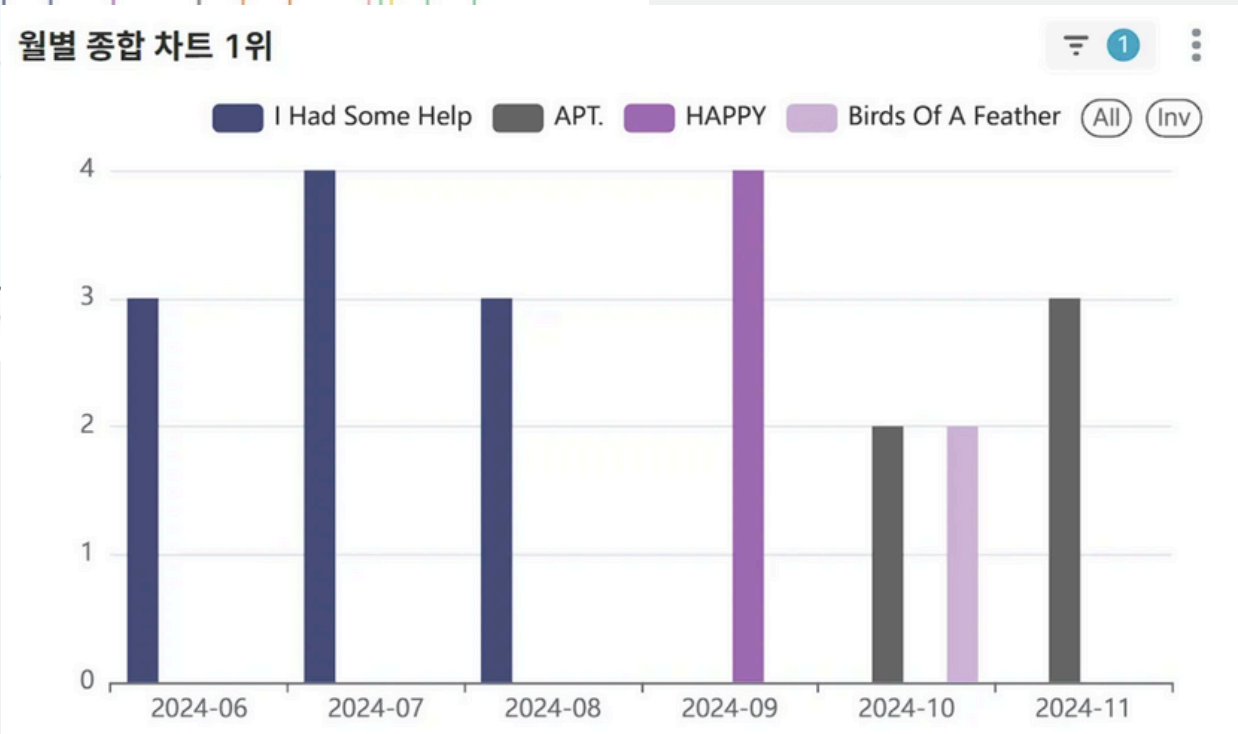
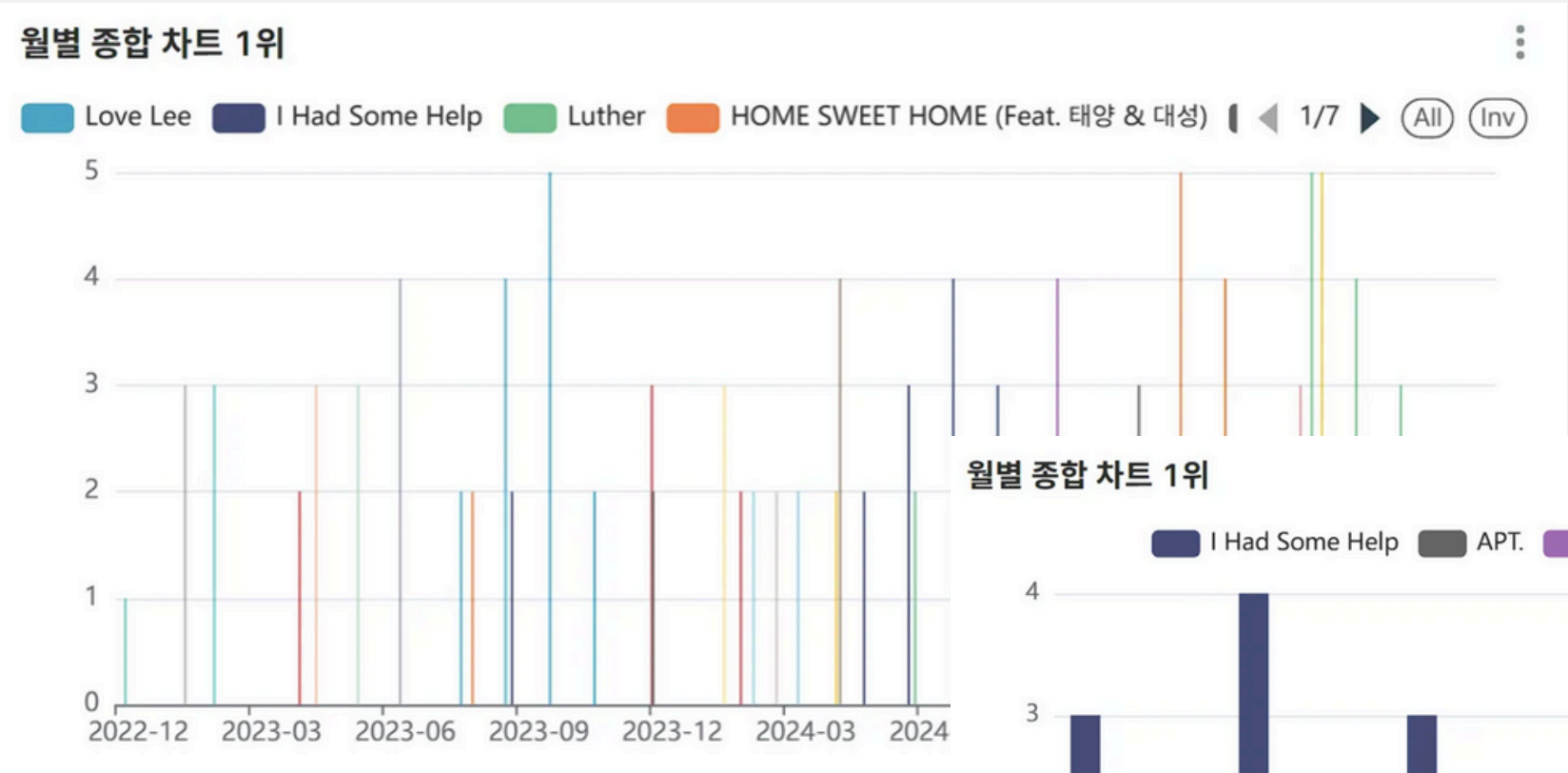


# 대시보드

## 기간별 차트 - 월별 종합 1위

월별로 종합차트 1위를 어떤 곡이 제일 많이 했는지를 보여줍니다.

기간 필터링으로 6개월 단위로 끊어서 해당 월에 어떤 곡이 1등을 가장 많이 한 지 확인할 수 있습니다



# 대시보드

## 기간별 차트 - 일주일별 종합

일주일별 4개의 사이트 종합 차트를 나  
타낸것입니다.

일주일 단위로 필터링이 가능하며  
INTEGRATED\_RANK는 101 - RANK를  
통해 얻은 점수를 월별 평균값으로 매긴  
등수입니다.

일주일별 종합 차트

Show 20 entries

Search 20 records...

MUSIC_NAME	INTEGRATED_RANK
Ordinary	1
너에게 닿기를	2
Never Ending Story	3
What I Want	3
Drowning	3
Just In Case	6
모르시나요 (Prod. by 로코베리)	6
...	...

1

2

3

4

5

6

7

...

13



# 대시보드

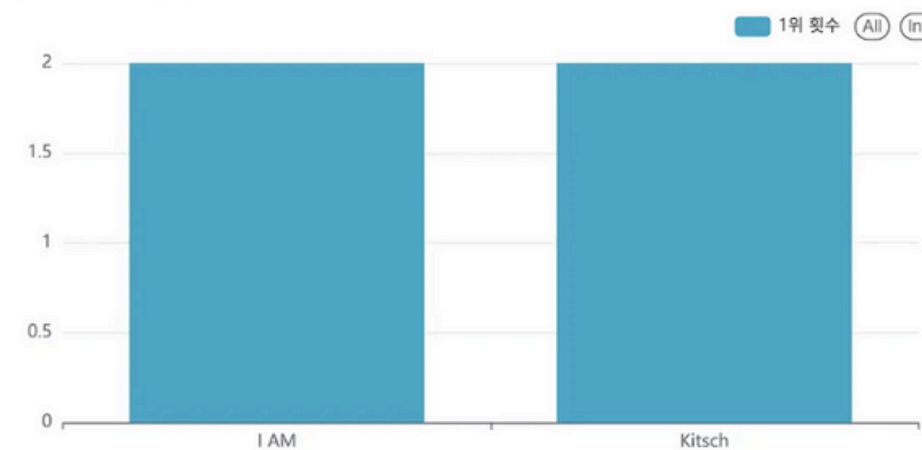
## 기간별 차트 - 차트 별 1위 음원

.....

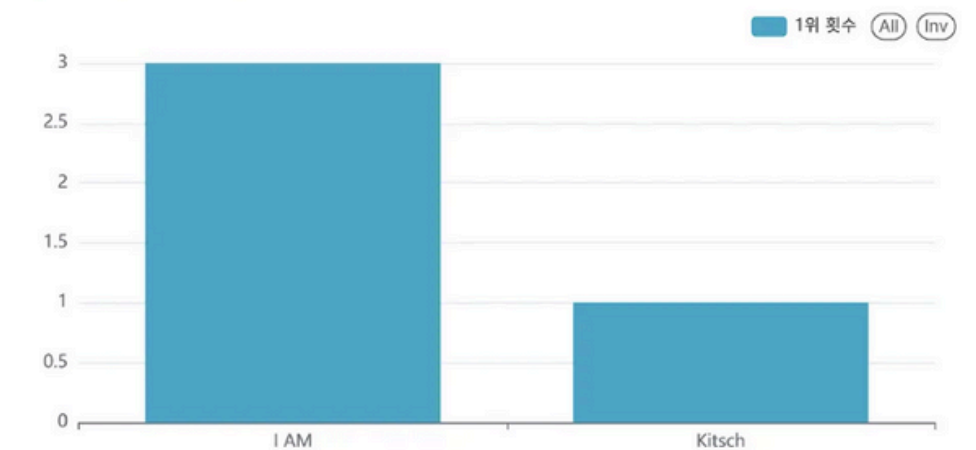
각 사이트별 월별 1등을 많이한 음원을 나타내는 차트입니다. 최대 TOP5까지 표기합니다.

월별 필터링을 통해 특정 월의 1위를 많이 차지한 음원들을 알 수 있습니다.

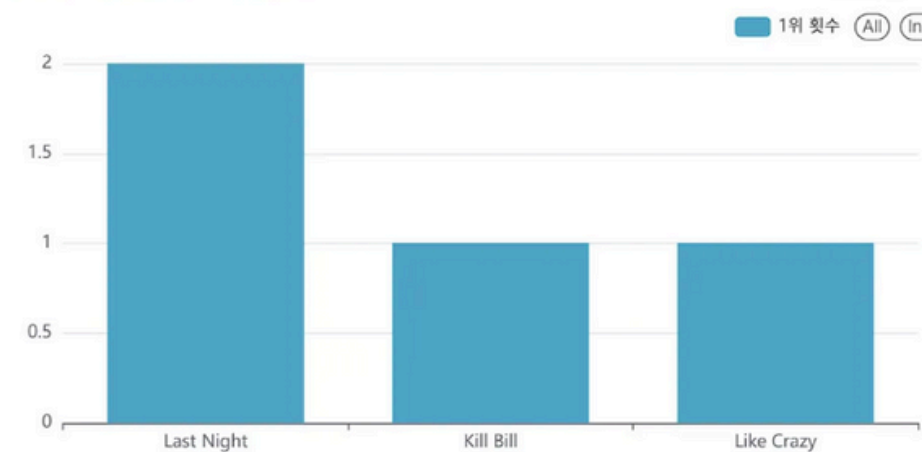
월별 Ranking Top5(genie)



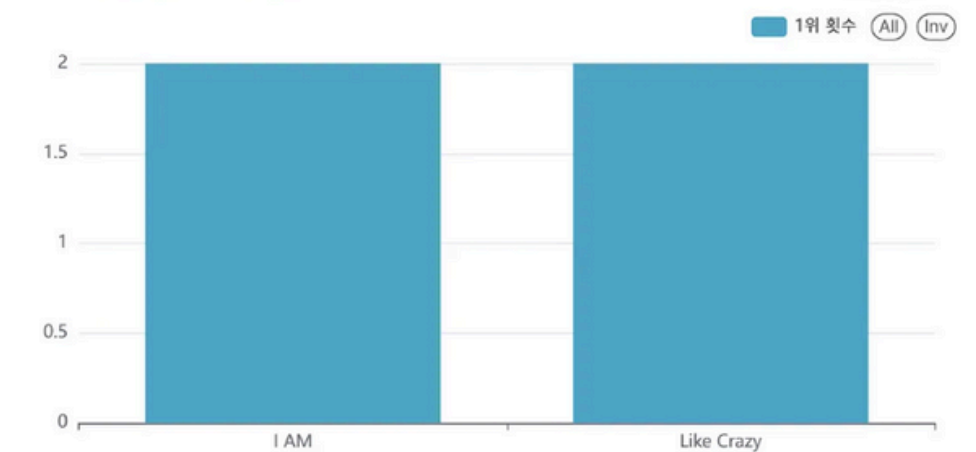
월별 Ranking Top5(Bugs)



월별 Ranking Top5(Billboard)



월별 Ranking Top5(Spotify)

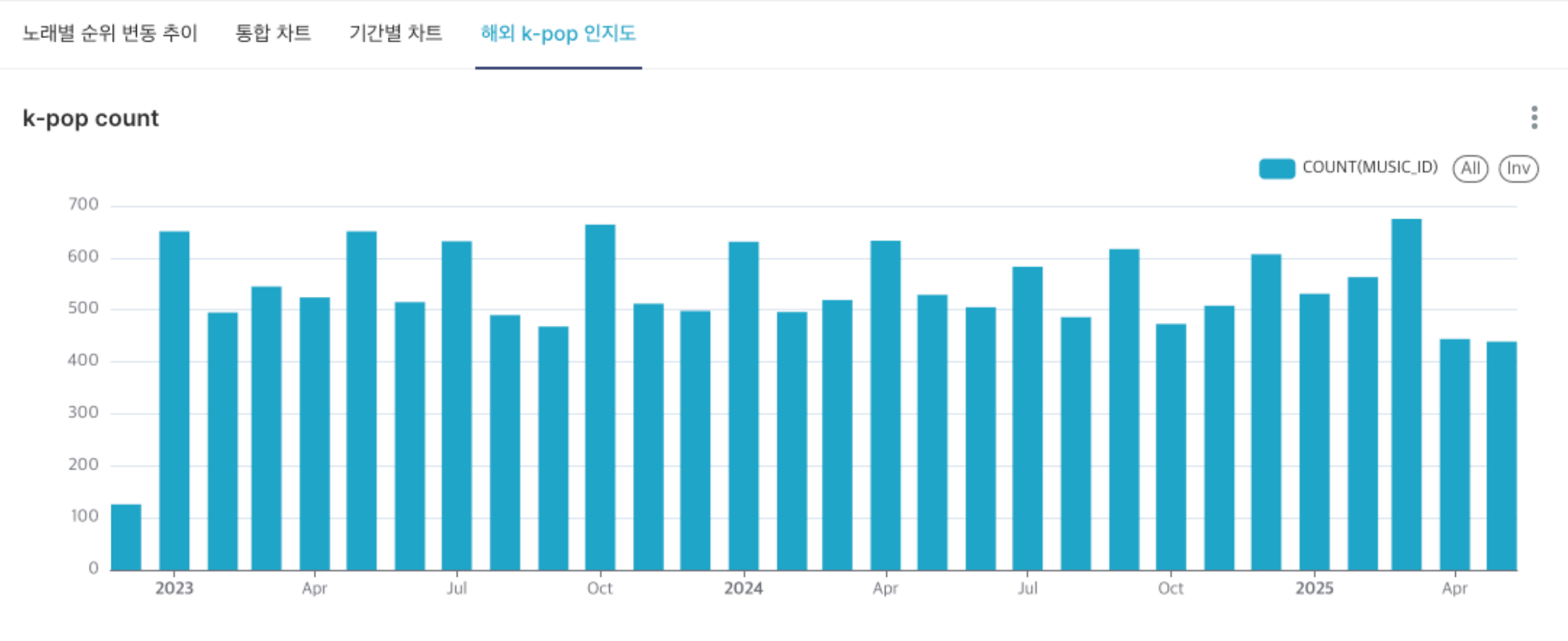


# 대시보드

## 해외 k-pop 인지도

spotify의 미국 차트에서 k-pop이 몇 번 차트에 올랐는지 막대 그래프로 시각화 하였습니다.

해당 그래프를 통해 미국 차트에서 k-pop이 강세를 띄었던 기간을 확인할 수 있습니다



# 결과 리뷰

## 프로젝트 성과

다양한 사이트의 차트정보를 수집하는 자동화된 파이프 라인 구축

데이터가 최신상태로 유지됨으로써 시각화의 결과의 정확성 및 활용성 증가

Airflow및 docker에 대한 전반적인 이해도 향상

# 향후 개선 방향

에러 핸들링 및  
모니터링 강화

Github Action 및 Slack 연결을 통해 CI/CD 모니터링 및 에러 감지/처리  
고도화

곡명·아티스트명  
통합 정제

일부 노래/아티스트의 경우 사이트마다 다른형태로 등록되어 있어  
정규화 필요

문제 해결 시도1: Python에서 특정 패턴을 정하고 패턴에 따라 정규화  
→ 원본 노래제목의 구분이 어려워져 중단

문제 해결 시도2: Spotify API를 활용해 검색 결과로 정규화  
→ 검색 시 오차 범위가 생각보다 커져서 중단

# 프로젝트 한줄 소감

주기적으로 데이터를 수집하는 Airflow DAG에 대해 많이 배울 수 있었고, 데이터를 어떻게 시각화할지 고민했던 과정도 의미 있는 경험이었습니다.

오주언



airflow를 이용한 ETL의 자동화를 직접 설계하고 스케줄링하는 것을 통해 airflow의 핵심을 배우게 되어 좋았습니다.

장현우



Airflow를 이용한 ETL 파이프라인 설계부터 Snowflake에서의 ELT, Superset을 활용한 시각화까지 직접 경험하고 이해도를 높일 수 있었습니다.

최정우



Docker 환경을 구성하고, 해당 환경에서 개발하며 Docker에 대한 다양한 경험을 하게되어 좋았습니다.

한기중



Thank you

감사합니다