

JPOP

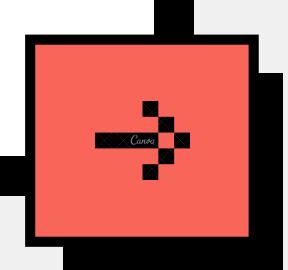


JPOP MUSIC RATINGS COMPARISON BETWEEN YOUTUBE AND SPOTIFY

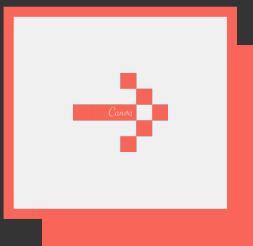
การเปรียบเทียบเรตติ้งเพลง JPOP ระหว่าง youtube และ spotify

SC312105 SEC 1

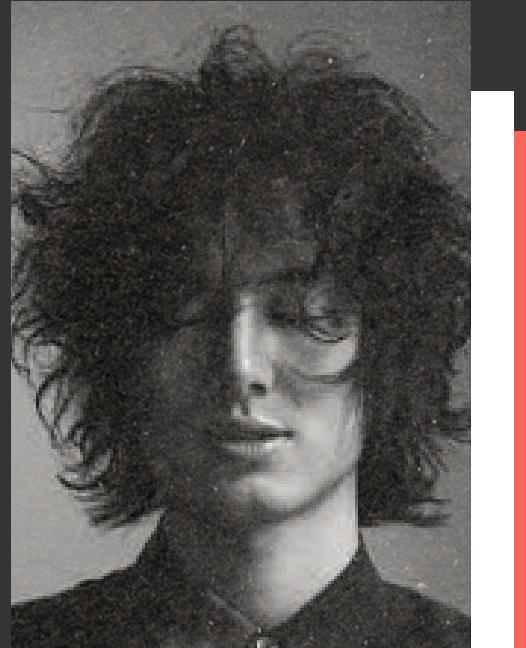
• • •



Member



...

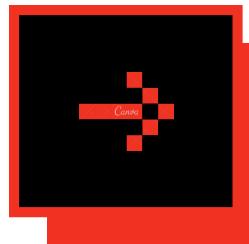


01

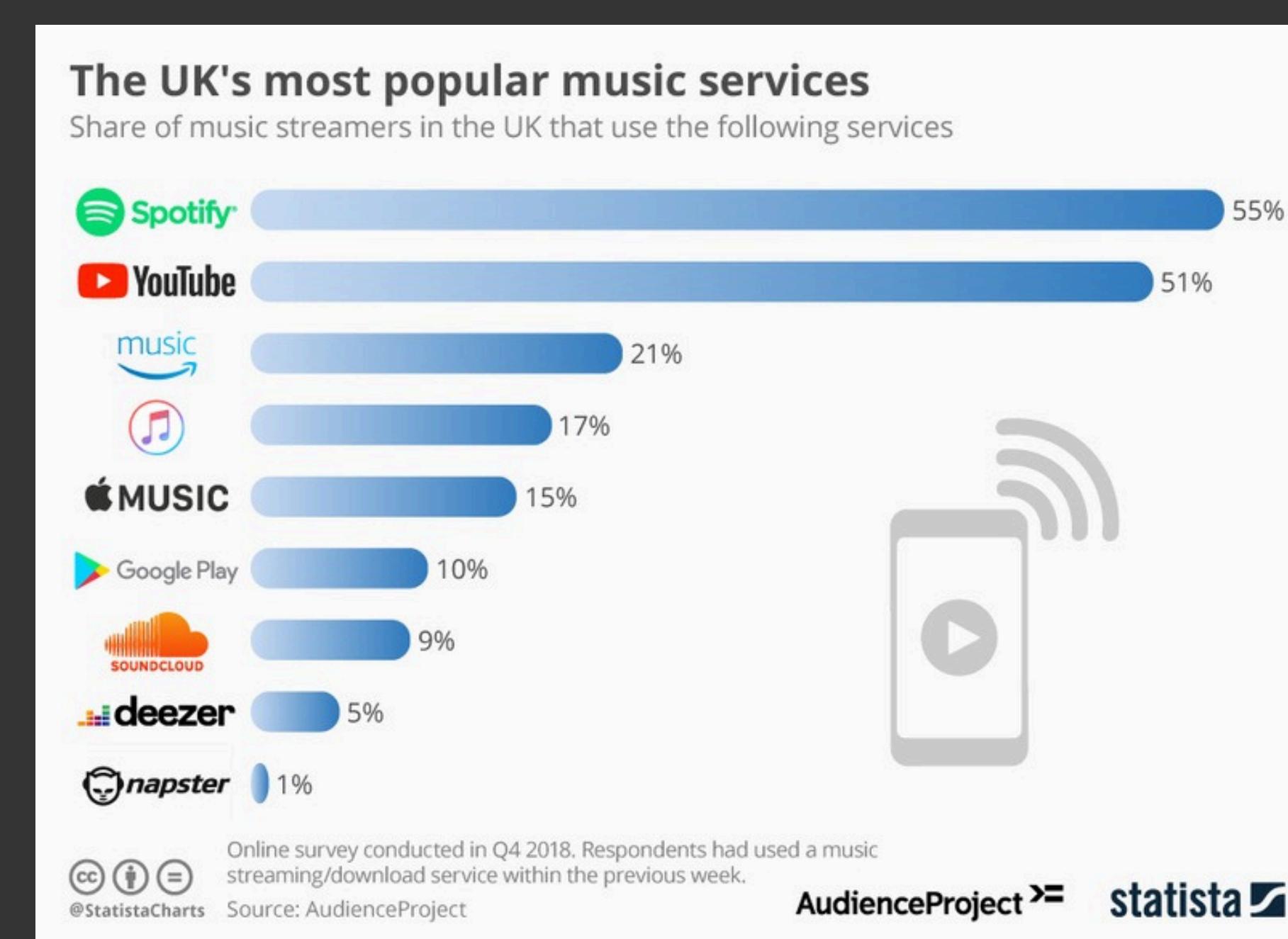
นางสาวรุ่งตะวัน สุกโธ²
643020489-0

02

นางสาวศุภารัตน์ แจ่มกลาง
643021253-4



Get to know Music Platform



ปัจจุบันผู้คนนิยมฟัง
เพลงผ่านทาง[แอปพลิเคชัน](#)

- spotify
- youtube
- apple music
- joox

OBJECTIVES

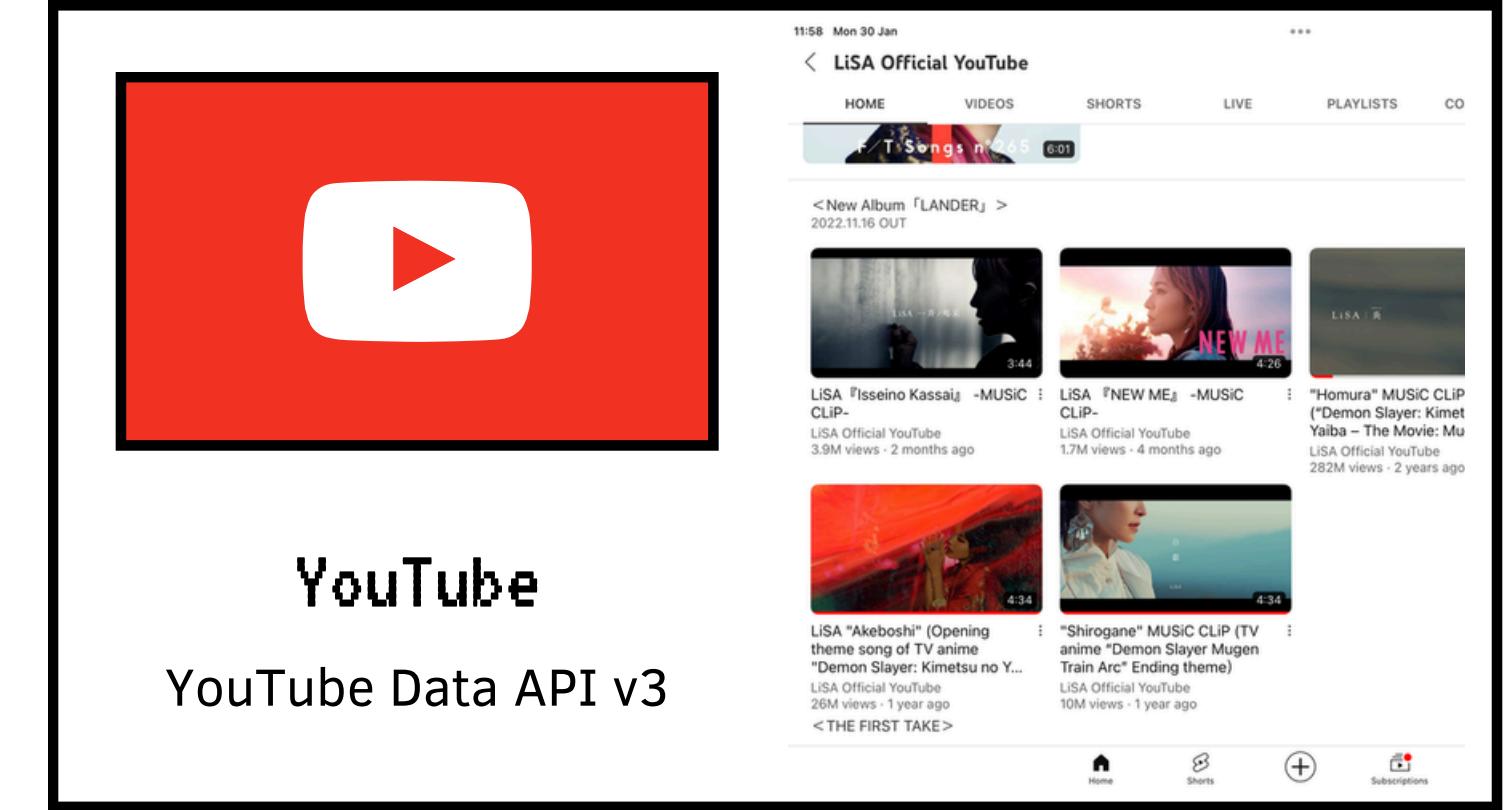
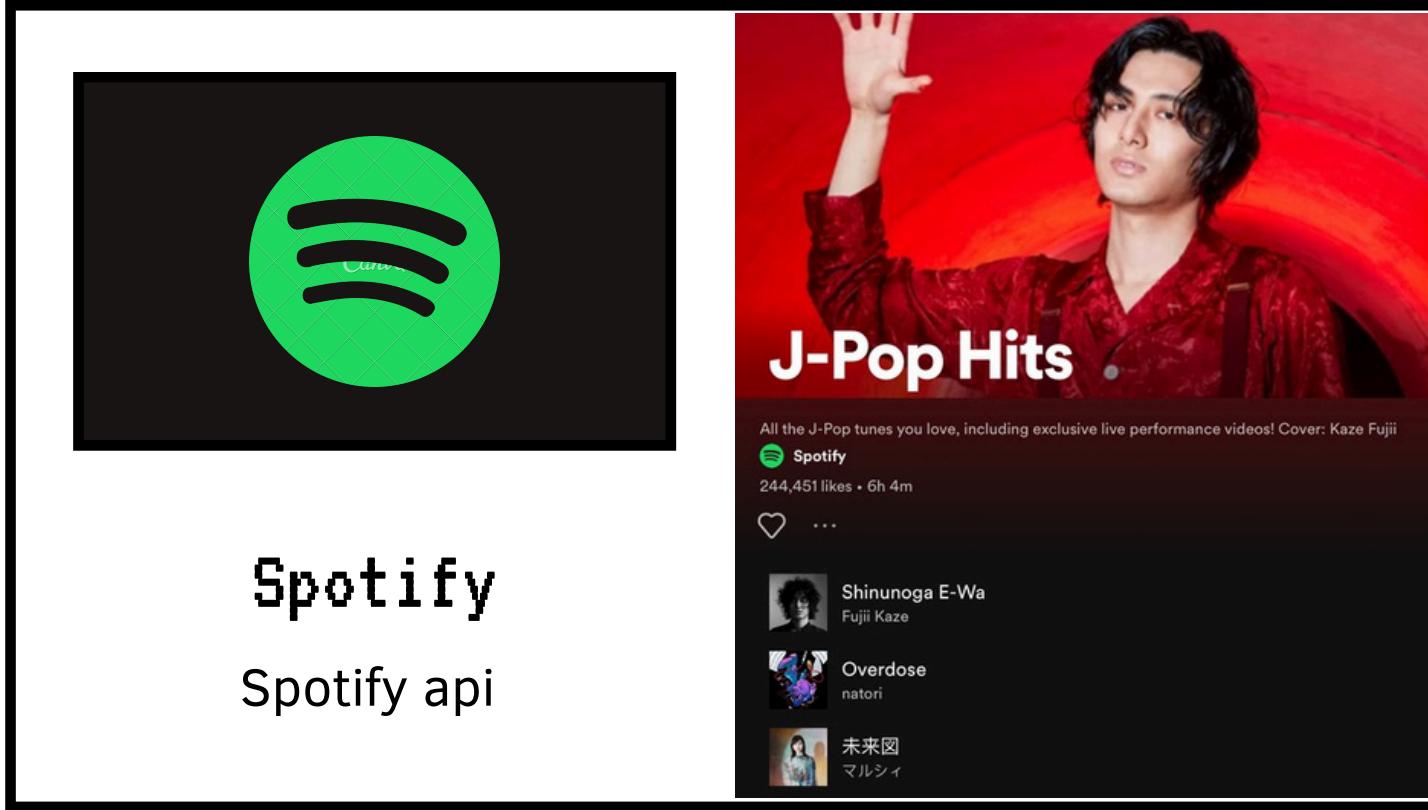
01

เพื่อเตรียมเก็บความนิยมในการฟังเพลงแนว JPOP ระหว่างใน youtube และ spotify



JPOP

Data Source



Database



mongoDB

Database Design

Collection

- channels (Youtube)
- videos (Youtube)
- Tracks (Spotify)

Youtube

document in channels

```
_id: ObjectId('64136ede894b83f0ddd4e428')
channelId: "UCUCeZaZeJbEYAAzvMgrKOPQ"
uploads: "UUUCeZaZeJbEYAAzvMgrKOPQ"
chTitle: "Kenshi Yonezu 米津玄師"
viewCount: 472623273
subscriberCount: 6710000
videoCount: 94
```

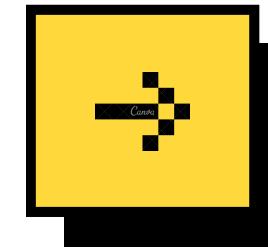
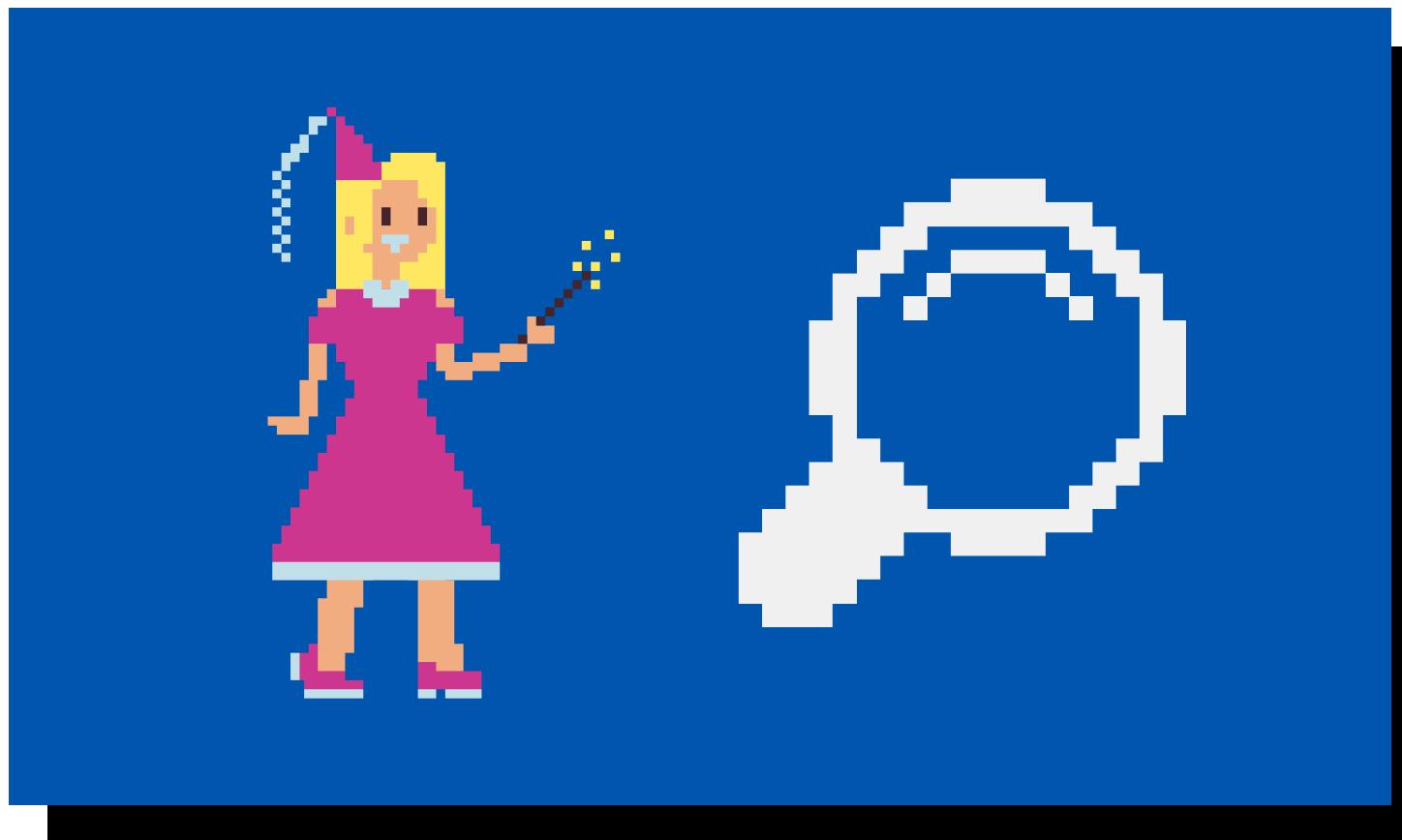
document in videos

```
_id: ObjectId('64139365894b83f0ddd6b4ae')
videoId: "HSNWJbkIsmo"
publishedAt: "2022-11-22T13:36:14Z"
channelId: "UCUCeZaZeJbEYAAzvMgrKOPQ"
title: "Kenshi Yonezu - KICK BACK (Live with Daiki Tsuneta) "Chainsaw Man" OP"
categoryId: 10
defaultAudioLanguage: "ja"
viewCount: 13613826
likeCount: 319786
favoriteCount: 0
commentCount: 9093
licensedContent: "True"
```

Spotify document in tracks

```
_id: ObjectId('64132b8f894b83f0ddd3a16e')
  : "0"
track_name: "New Genesis"
track_id: "6A8NfypDHuwilWbo4alyca"
artist_name: "Ado"
artist_id: "6mEQK9m2krja6X1cfsAjfl"
popularity: 64
track_number: "1"
type: "track"
preview_url: "None"
duration_ms: 226813
disc_number: 1
is_local: "False"
uri: "spotify:track:6A8NfypDHuwilWbo4alyca"
href: "https://api.spotify.com/v1/tracks/6A8NfypDHuwilWbo4alyca"
external_ids: "{ 'isrc': 'JPP002201806' }"
available_markets: "[ 'AR', 'AU', 'AT', 'BE', 'BO', 'BR', 'BG', 'CA', 'CL', 'CO', 'CR', 'CY', 'DE', 'DK', 'DO', 'ES', 'FI', 'GB', 'GR', 'HK', 'HU', 'ID', 'IL', 'MX', 'NL', 'NO', 'PE', 'PT', 'RO', 'SE', 'SI', 'SV', 'TR', 'VE', 'ZA' ]"
```

Data Preparation



ตรวจสอบว่ามีการซ้ำของข้อมูล ตรวจสอบค่าของข้อมูลว่า มีความผิดปกติหรือไม่ ตรวจสอบข้อมูลที่เป็นค่าว่าง

```
isna()
value_counts()
info()
YOUTUBE
```

		Data columns (total 12 columns):		
		#	Column	Non-Null Count Dtype
		0	---	-----
_id		0	_id	58712 non-null object
videoId		0	videoId	58712 non-null object
publishedAt		0	publishedAt	58712 non-null object
channelId		0	channelId	58712 non-null object
title		0	title	58712 non-null object
categoryId		0	categoryId	58712 non-null int64
viewCount		8	viewCount	58704 non-null float64
likeCount	1509	7	likeCount	57203 non-null float64
favoriteCount	0	8	favoriteCount	58712 non-null int64
commentCount	3041	9	commentCount	55671 non-null float64
licensedContent	0	10	licensedContent	58712 non-null object
defaultAudioLanguage	27032	11	defaultAudioLanguage	31680 non-null object
dtype:	int64		dtypes: float64(3), int64(2), object(7)	
			memory usage: 5.4+ MB	
			None	

collection - videos

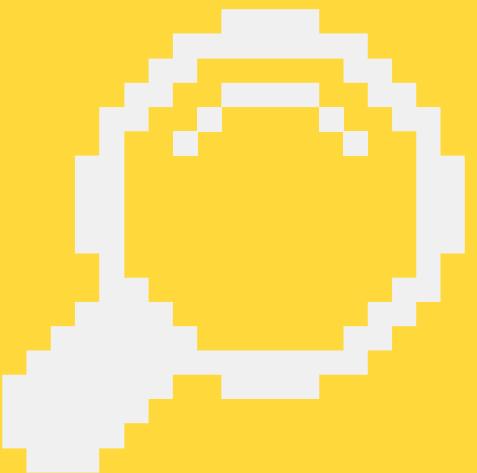
ຕຽບສອບຈາກ ISO 639-1 standard language codes

en	13457	pt-BR	188	id	58
en-US	5266	de	169	es-ES	55
ja	3906	pt	163	or	33
hi	1787	zh-Hant	158	ur	33
en-GB	1284	ru	155	el	20
ko	794	ta	150	wo	20
es-419	747	th	150	en-CA	12
fr	499	pl	150	ar	5
en-IN	489	zh-HK	149	zh-CN	4
es	393	zh-TW	148	zh-Hans	2
zxx	365	pa	141	sa	1
bn	282	fil	136	ne	1
vi	247	sdp	59	su	1
				zh	1
				it	1
				es-MX	1

isna()
value_counts()
info()
YOUTUBE

defaultAudioLanguage

- defaultAudioLanguage
1. zxx -> japan
 2. und -> en
 3. zh -> zh-Hans
 4. es-ES -> es
 5. sdp -> japan

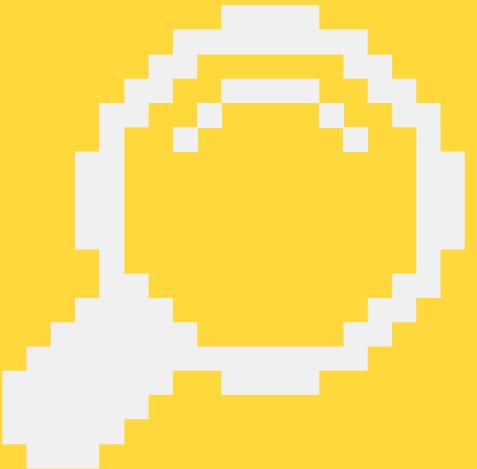


เพิ่ม not defined สำหรับ ข้อมูล defaultAudioLanguage
ที่เป็นค่าว่าง

isna()
value_counts()
info()
YOUTUBE

defaultAudioLanguage, tags, defaultLanguage,
regionRestriction

```
videos.update_many(  
    {  
        "defaultAudioLanguage" : { '$exists': False }  
    },  
    {  
        '$set': { "defaultAudioLanguage": 'not defined'  
    }  
},  
    upsert=False  
)
```



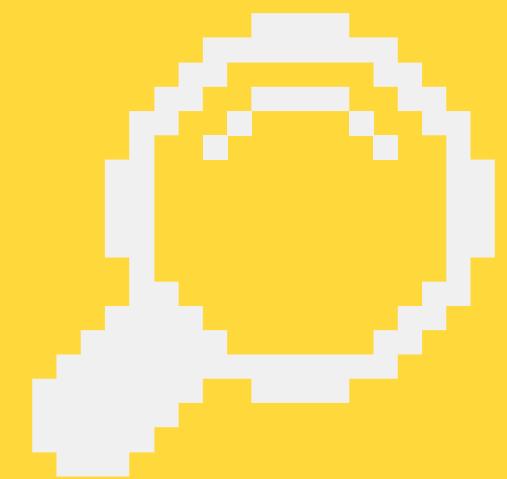
ตรวจสอบว่ามีการซ้ำของข้อมูล ตรวจสอบค่าของข้อมูลว่า มีความผิดปกติหรือไม่ ตรวจสอบข้อมูลที่เป็นค่าว่าง

isna()
value_counts()
info()
YOUTUBE

viewCount, likeCount, commentCount

```
# calculate mean likeCount each channel
from pymongo import aggregation

results = videos.aggregate([
    {
        "$group": {
            "_id": "$channelId",
            "avg": { "$avg": "$likeCount" }
        }
    }
])
results = list(results)
results
```

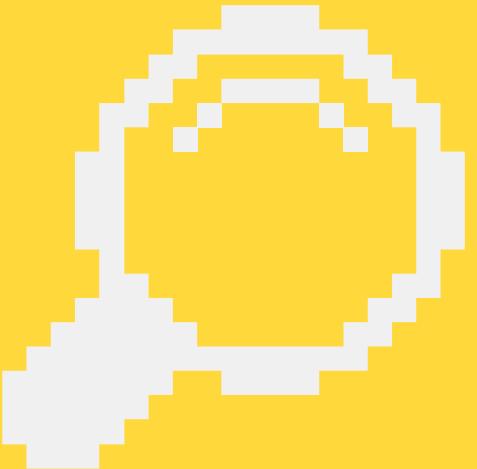


```
isna()
value_counts()
info()
YOUTUBE
```

ตรวจสอบว่ามีการซ้ำของข้อมูล ตรวจสอบค่าของข้อมูลว่า
มีความผิดปกติหรือไม่ ตรวจสอบข้อมูลที่เป็นค่าว่าง

_id	0	_id	0
videoId	0	videoId	0
publishedAt	0	publishedAt	0
channelId	0	channelId	0
title	0	title	0
categoryId	0	categoryId	0
viewCount	8	viewCount	0
likeCount	1509	likeCount	0
favoriteCount	0	favoriteCount	0
commentCount	3041	commentCount	0
licensedContent	0	licensedContent	0
defaultAudioLanguage	27032	defaultAudioLanguage	0
dtype: int64		dtype: int64	

collection - videos



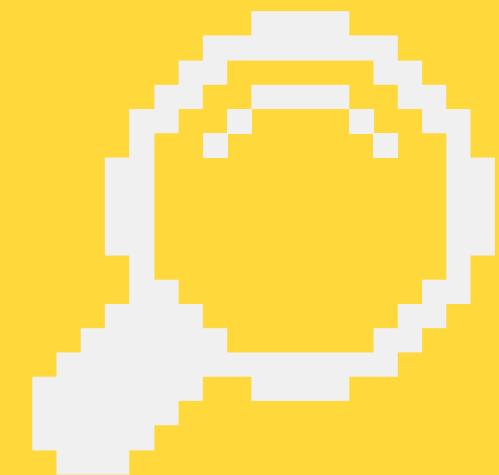
ตรวจสอบว่ามีการซ้ำของข้อมูล ตรวจสอบค่าของข้อมูลว่า มีความผิดปกติหรือไม่ ตรวจสอบข้อมูลที่เป็นค่าว่าง

info()
SPOTIFY

```
[ ] print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21427 entries, 0 to 21426
Data columns (total 16 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   track_name       21427 non-null object 
 1   track_id         21427 non-null object 
 2   artist_name      21427 non-null object 
 3   artist_id        21427 non-null object 
 4   popularity       21427 non-null int64  
 5   track_number     21427 non-null int64  
 6   type             21427 non-null object 
 7   preview_url      21427 non-null object 
 8   duration_ms      21427 non-null int64  
 9   disc_number      21427 non-null int64  
 10  is_local          21427 non-null bool   
 11  uri              21427 non-null object 
 12  href              21427 non-null object 
 13  external_ids     21427 non-null object 
 14  external_ids     21427 non-null object 
 15  available_markets 21427 non-null object 
dtypes: bool(1), int64(4), object(11)
memory usage: 2.6+ MB
None
```

นับจำนวนชื่อเพลงและชื่อศิลปินที่ซ้ำ

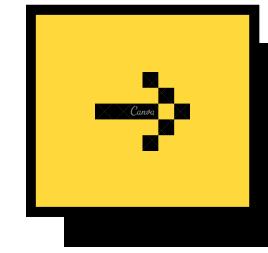
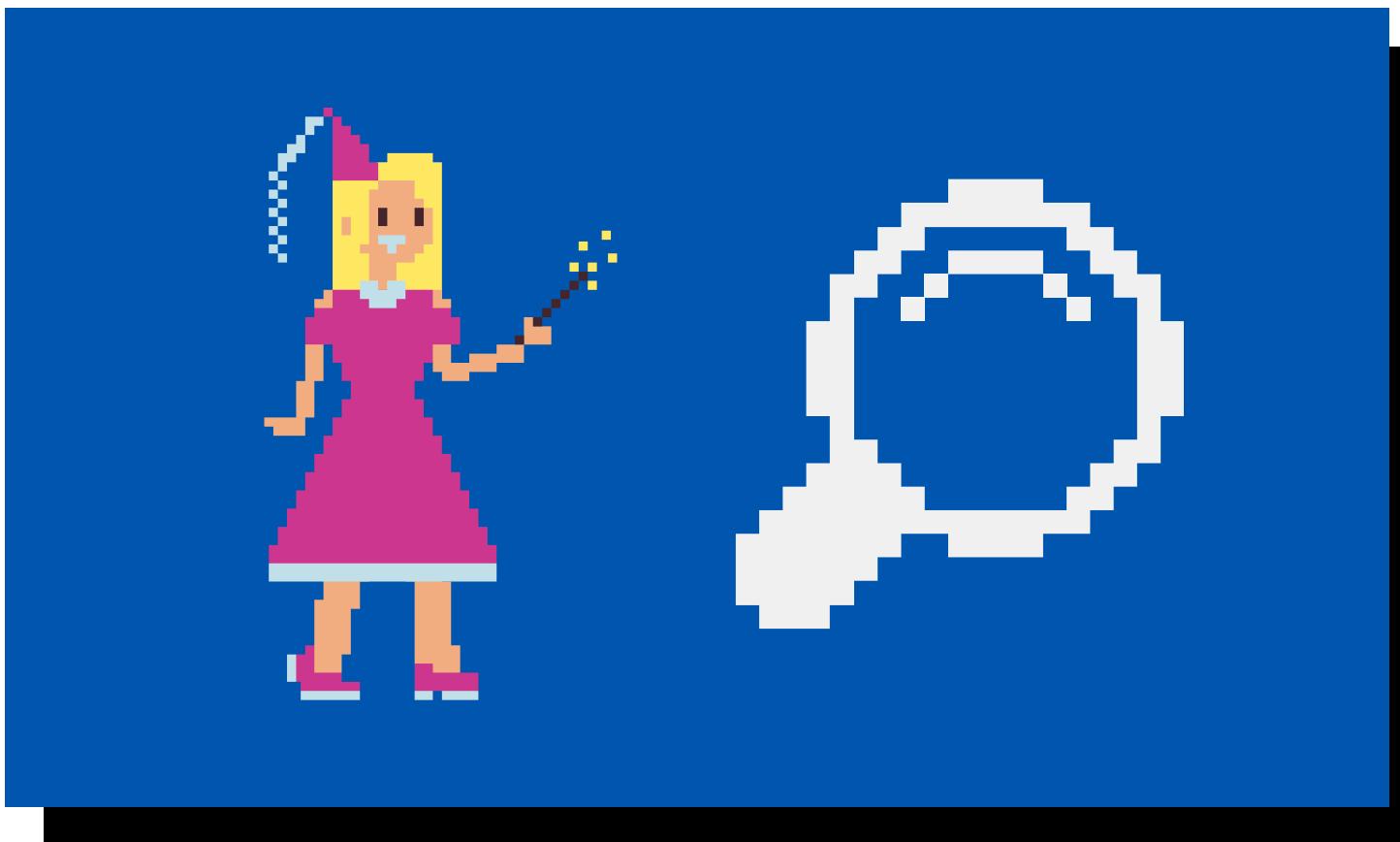


value_counts()
SPOTIFY

```
0  df["track_name"].value_counts()
  ...
  Timber - DANCE COVER REMIX      27
  Power In Your Soul - DANCE COVER REMIX  24
  Something Stupid - PARTY HITS EDIT     21
  Lovefool - PARTY HITS EDIT          21
  Unstoppable - DANCE COVER REMIX      20
  ...
  All We Get - [DJ Mix Ver.]        1
  All Night - [DJ Mix Ver.]         1
  Poolside Vibe - [DJ Mix version]   1
  Heartbreaker - [DJ Mix Ver.]       1
  WATARIDORI - Cover Ver. - Mixed    1
Name: track_name, Length: 12843, dtype: int64
```

```
0  df["artist_name"].value_counts()
  ...
  Yutaka Yamada      646
  Lal                 635
  PARTY HITS PROJECT 459
  Ohashi Trio        459
  Party Town          414
  ...
  August Rigo          1
  Steel Pulse          1
  Island Sunset Sessions 1
  Kynan Russell        1
  You Hee Yeol          1
Name: artist_name, Length: 1095, dtype: int64
```

Data Processing



หาซื้อช่องใน youtube จากชื่อศิลปินจาก spotify

```
ch = pd.DataFrame()
for index, row in temp_sp.iterrows():

    artist_name = row['artist_name']
    # , case=False, regex=False
    matching_rows = temp_yt[temp_yt['chTitle'].str.contains(artist_name)]

    if not matching_rows.empty:
        print(f"artist name '{artist_name}' found in yt:")
        print(matching_rows['chTitle'])
        ch = ch.append(matching_rows, ignore_index =True)
        # break

2005      POST Malone
25866    Post Malone
25867    Post Malone
Name: chTitle, Length: 66, dtype: object
<ipython-input-234-8fc51514ce39>:11: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Please use concat instead.
    ch = ch.append(matching_rows, ignore_index =True)
artist name 'PEDRO' found in yt:
2515    PEDRO the Budgie
2516    PEDRO the Budgie
```

Youtube

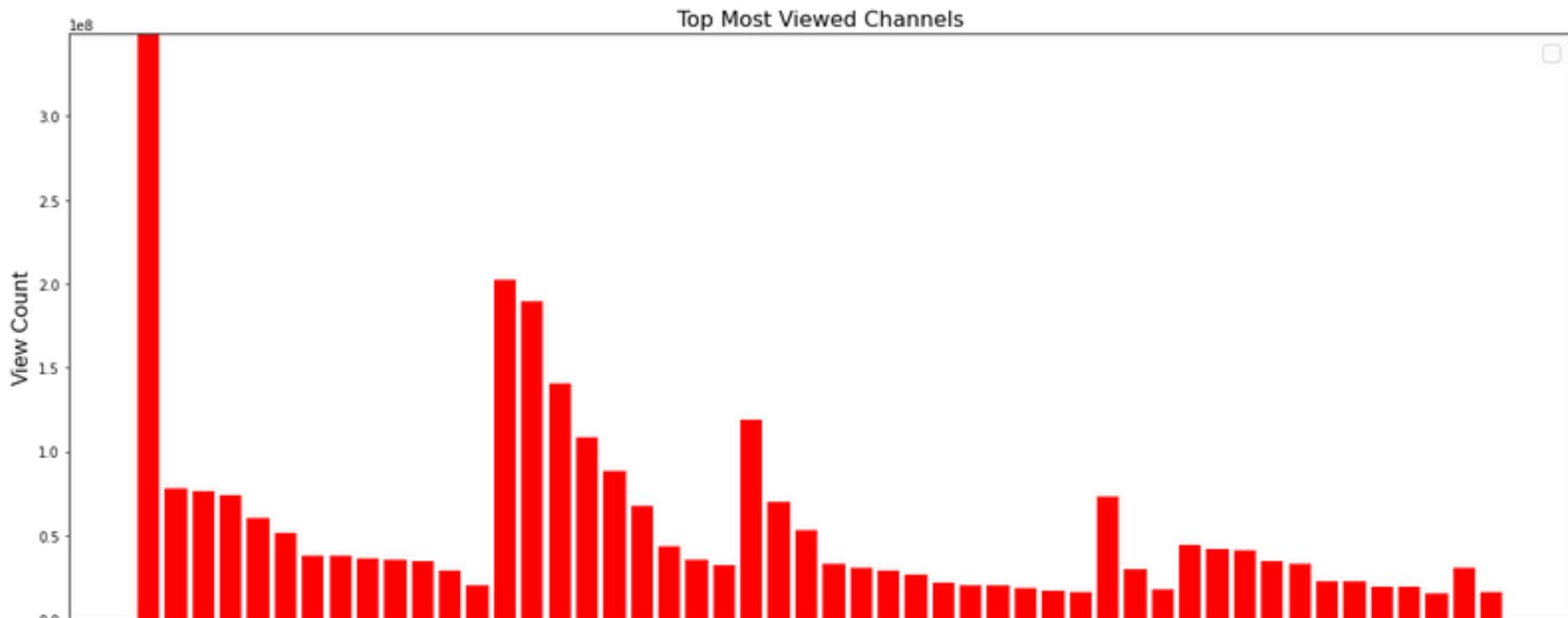
หาชื่อช่องใน youtube จากชื่อศิลปินจาก spotify

```
result_first50 = videos.aggregate([
    {"$match": {"channelId": {"$in": chId},
                "defaultAudioLanguage" : "ja",
                "licensedContent": "True"}
     },
    {"$group": {
        "_id": "$videoId",
        "viewCount": {"$max": "$viewCount"}
    }},
    {"$sort": {"viewCount": -1}},
    {"$group": {
        "_id": None,
        "top50": {"$push": {"_id": "$_id", "viewCount": "$viewCount"}}
    }},
    {"$project": {"_id": 0, "top50": {"$slice": ["$top50", 50]}}}
])

top50 = list(result_first50)
```

Youtube

Top most viewed videos



Youtube



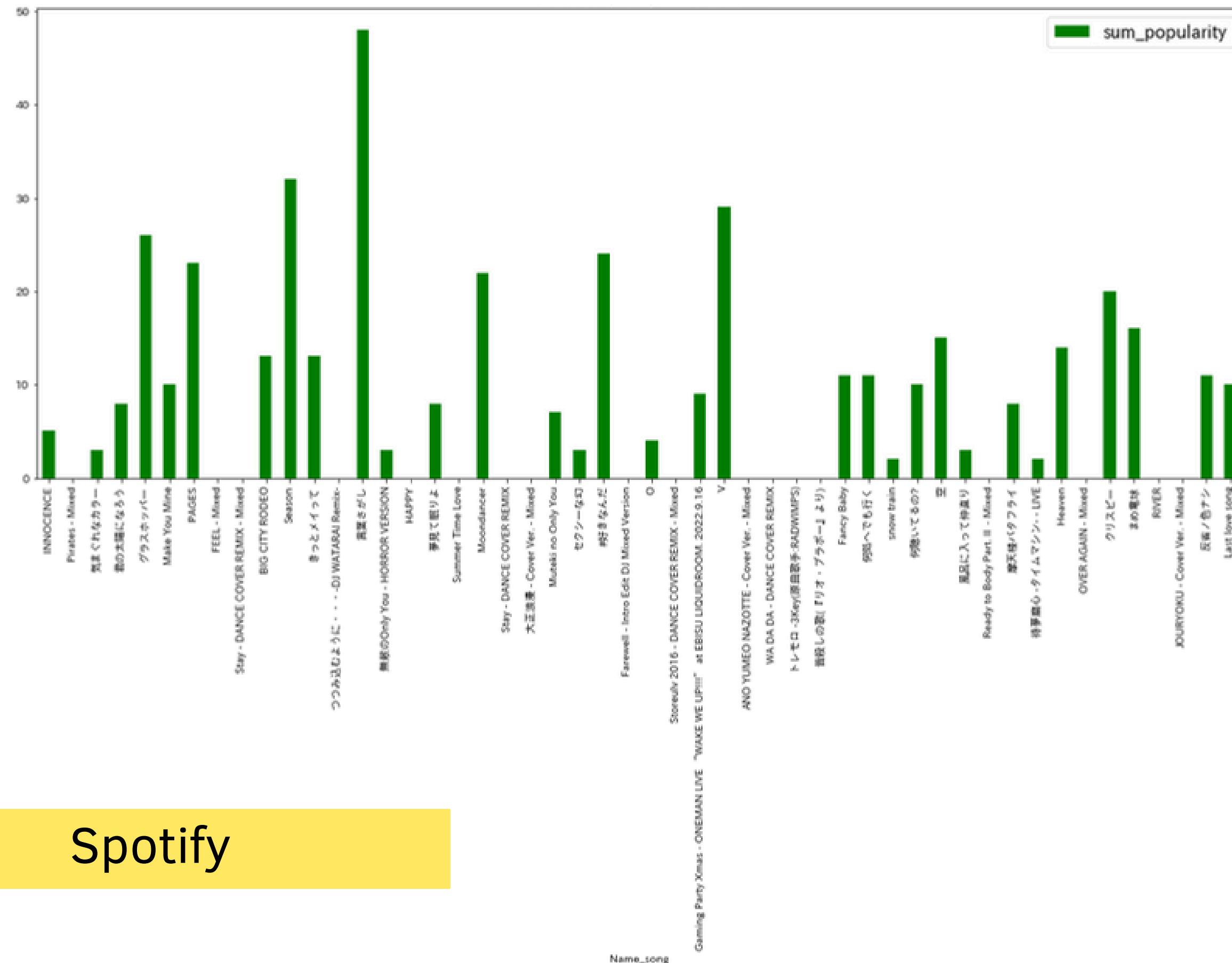
```
from pymongo import aggregation
results = tracks.aggregate([
    {"$group": {"_id": "$track_id",
                "Name_song": {"$first": "$track_name"},
                "artist": {"$first": "$artist_name"},
                "sum_popularity": {"$sum": "$popularity"}
    }
}, {"$sort": {"sum": -1}},
{"$limit": 50}
])
re = list(results)
print(re)
```

...

Spotify

Example Spotify popularity's song 50 song

23

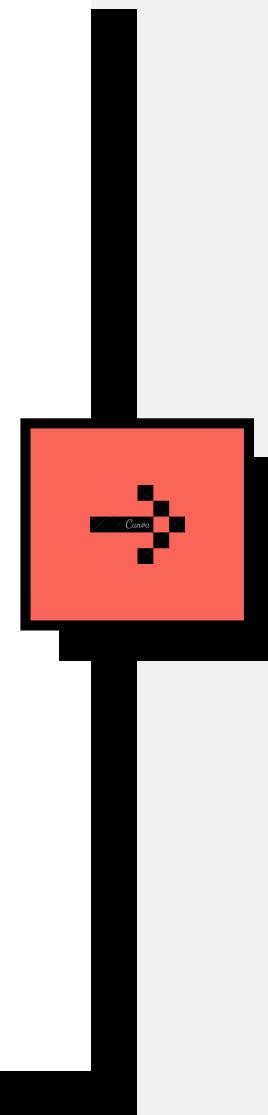


Spotify



THANK YOU !

Any question?



Request Youtube

```
!gdown 1PEW5qLidwul3FlFVtMPVoVGMZuEWS0fD
↳ Downloading...
From: https://drive.google.com/uc?id=1PEW5qLidwul3FlFVtMPVoVGMZuEWS0fD
To: /content/id_and_namesong_spotify.csv
100% 1.67M/1.67M [00:00<00:00, 174MB/s]

[ ] df_artist_spotify = pd.read_csv("/content/id_and_namesong_spotify.csv")
print(df_artist_spotify.shape)
df_artist_spotify.head()

(19439, 5)

  Unnamed: 0          track_id    track_name      artist_id artist_name
0           0  6A8NfypDHuwiLWbo4a1yca  New Genesis  6mEQK9m2krja6X1cfsAjfl      Ado
1           1  7yQnZkAxSiAavC0zFRB1NI  I'm invincible  6mEQK9m2krja6X1cfsAjfl      Ado
2           2  3WY0iazRPHOllq6CdbKLZ6     Backlight  6mEQK9m2krja6X1cfsAjfl      Ado
3           3  1pnXMEGyw85jtg9fFZf4GP   Fleeting Lullaby  6mEQK9m2krja6X1cfsAjfl      Ado
4           4  4eChbtoFoOmWQt03aRDzaJ      Tot Musica  6mEQK9m2krja6X1cfsAjfl      Ado
```

```
[ ] nameArtists = []
for id in df_artist_spoftify['artist_name']:
    if id not in nameArtists:
        nameArtists.append(id)
print(len(nameArtists))
# for id in nameArtists:
#     print(id)
#     print("---")
```

1112

```
[ ] chunks = [nameArtists[id:id+10] for id in range(0, len(nameArtists), 10)]
print(len(chunks))
chunks
```

112

```
[[ 'Ado',
  'Gen Hoshino',
  'GReeeeN',
  '緑とたけだ',
  'Gukii',
  '8LOOM',
  'Stereopony',
  '歌っちゃ王',
  'JO1',
  'CHEMISTRY'],
```

▼ request search from name - youtube

```
[ ] api_key = "AIzaSyCZpcAQ5_y8HCo5tZk9_K5HP1EnPYypJj8"
```

```
[ ] list_json = []
count = 0
url = "https://www.googleapis.com/youtube/v3/search"
for chunk in chunks :
    nameAt = '|'.join(str(id) for id in chunk)
    # print(nameAt)
    params = {"key" : api_key,
               "q" : nameAt,
               "type": "video, channel",
               "part": "snippet",
               "maxResults" : 50,
               "videoLicense" : "youtube",
               "videoDuration" : "medium",
               "pageToken" : "CGQQAA"}
    # print(params)
    r = requests.get(url, params=params)
    json_data = r.json()
    list_json.append(json_data)
print(r)
```

```
<Response [200]>
```

search from channel ID

```
[ ] api_key = "AIzaSyD8L-I3q4DCZXG17et7L8Q1TzHuHIfi2tM"

[ ] list_json = []
url = "https://www.googleapis.com/youtube/v3/channels"
for chunk in chunks :
    chIDs = ', '.join(str(id) for id in chunk)
    # print(chIDs)
    params = {"key" : api_key,
               "part": "id, snippet, contentDetails, statistics, topicDetails",
               "id" : chIDs,
               "maxResults" : 50}
    # print(params)
    r = requests.get(url, params=params)
    json_data = r.json()
    list_json.append(json_data)
print(r)
```

```
<Response [200]>
```

```
<Response [200]>
```

▼ request search from name - youtube

```
[ ] api_key = "AIzaSyCZpcAQ5_y8HCo5tZk9_K5HP1EnPYypJj8"
```

```
[ ] list_json = []
count = 0
url = "https://www.googleapis.com/youtube/v3/search"
for chunk in chunks :
    nameAt = '|'.join(str(id) for id in chunk)
    # print(nameAt)
    params = {"key" : api_key,
               "q" : nameAt,
               "type": "video, channel",
               "part": "snippet",
               "maxResults" : 50
               "videoLice
               "videoDura
               "pageToken"
# print(params)
r = requests.get(url, params=params)
json_data = r.json()
list_json.append(json_data)
print(r)
```

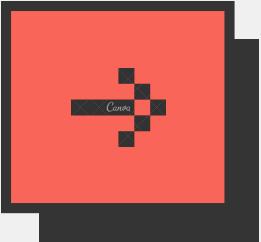
```
<Response [200]>
```

	videoId	publishedAt	publishTime	title	channelId	channelTitle	description
0	jhOVibLEDhA	2016-09-20T10:30:01Z	2016-09-20T10:30:01Z	星野源 - 念 (Official Video)	UCPK0rgZXnnb89nSeITvTdGA	星野源	星野源 0th Single 「念」 https://jvcmusic.link.to/ko1...
1	96QCkbE7WpI	2021-09-03T03:00:04Z	2021-09-03T03:00:04Z	[PROCESS JO1] EP.10 「踊/Ado」 カバー/パフォーマンスの裏側	UCsmXiDP8S40uBeJYxvyulmA	JO1	[PROCESS JO1] EP.10 「踊/Ado」 カバー/パフォーマンスの裏側
2	HG0M_eZduxY	2018-12-12T13:51:13Z	2018-12-12T13:51:13Z	星野源 - Pop Virus (Official Video)	UCPK0rgZXnnb89nSeITvTdGA	星野源	星野源 5th Album 「POP VIRUS」 https://jvcmusic.link....
3	deIS1BmSfZQ	2020-01-16T13:00:11Z	2020-01-16T13:00:11Z	星野源 - Ain't Nobody Know (Official Video)	UCPK0rgZXnnb89nSeITvTdGA	星野源	Directed by Kyotaro Hayashi Starring Gen Hoshi...
4	RlUb2F-zLxw	2018-08-19T19:00:08Z	2018-08-19T19:00:08Z	星野源 - アイデア (Official Video)	UCPK0rgZXnnb89nSeITvTdGA	星野源	星野源 12th Single 「アイデア」 https://jvcmusic.link.to/...
...
11211	x8A9Y1Dq_cQ	2009-10-25T20:32:53Z	2009-10-25T20:32:53Z	Willie Nelson - Seven Spanish Angels (Official...)	UCIGWQ7CZipZIHxwu5EMINPw	WillieNelsonVEVO	"Seven Spanish Angels" by Willie Nelson and Th...
		2016-10-	2016-10-	Florence + The Machine -			Dance Fever - the new album - out now

popularity()

The popularity of the track. The value will be between 0 and 100, with 100 being the most popular. The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity. Note that the popularity value may lag actual popularity by a few days: the value is not updated in real time

Request Spotify



5/9



ขอข้อมูลของ spotify ผ่าน postman
เสิร์ช gern = jpop, type = artist

```
# Opening JSON file
f = open('/content/search-spotify-genre-pop-typeArtist-limit50-offset0')
# returns JSON object as
# a dictionary
data1 = json.load(f)
data1

{'artists': {'href': 'https://api.spotify.com/v1/search?query=genre%3Djpop&type=artist&offset=0&limit=50',
  'items': [{"external_urls": {"spotify": "https://open.spotify.com/artist/6mEQK9m2krja6X1cfsAjfl"},
    'followers': {'href': None, 'total': 2232016},
    'genres': ['japanese teen pop'],
    'href': 'https://api.spotify.com/v1/artists/6mEQK9m2krja6X1cfsAjfl',
    'id': '6mEQK9m2krja6X1cfsAjfl',
    'images': [{"height": 640,
      'url': 'https://i.scdn.co/image/ab6761610000e5ebf3bb04995cb61f04936424ee',
      'width': 640},
     {'height': 320,
      'url': 'https://i.scdn.co/image/ab67616100005174f3bb04995cb61f04936424ee',
      'width': 320},
     {'height': 160,
      'url': 'https://i.scdn.co/image/ab6761610000f178f3bb04995cb61f04936424ee',
      'width': 160}],
    'name': 'Ado',
    'popularity': 76,
    'type': 'artist',
    'uri': 'spotify:artist:6mEQK9m2krja6X1cfsAjfl'},
   {'external_urls': {'spotify': 'https://open.spotify.com/artist/1S2S00lgLYLGHWA44qGEUs'},
    'followers': {'href': None, 'total': 1567858},
    'genres': ['japanes...']}
```

```
data1 = pd.DataFrame.from_dict(data1.get('artists').get('items'))
print(data1.columns)
print(data1.shape)
data1
```

```
Index(['external_urls', 'followers', 'genres', 'href', 'id', 'images', 'name',
       'popularity', 'type', 'uri'],
      dtype='object')
(50, 10)
```

	external_urls	followers	genres	href		id	images	name	pop
0	{'spotify': 'https://open.spotify.com/artist/6... 2232016}	{'href': None, 'total': 2232016}	[japanese teen pop]	https://api.spotify.com/v1/artists/6mEQK9m2krja6X1cf8Ajl...		6mEQK9m2krja6X1cf8Ajl	[{'height': 640, 'url': 'https://i.scdn.co/ima...}]	Ado	100
1	{'spotify': 'https://open.spotify.com/artist/1... 1567858}	{'href': None, 'total': 1567858}	[j-acoustic, j-pop, japanese singer-songwriter]	https://api.spotify.com/v1/artists/1S2S00lgLYLGHWA44qGEUs		1S2S00lgLYLGHWA44qGEUs	[{'height': 640, 'url': 'https://i.scdn.co/ima...}]	Gen Hoshino	100

```
data2 = pd.DataFrame.from_dict(data2.get('artists').get('items'))
print(data2.columns)
print(data2.shape)
data2
```

```
Index(['external_urls', 'followers', 'genres', 'href', 'id', 'images', 'name',
       'popularity', 'type', 'uri'],
      dtype='object')
(50, 10)
```

	external_urls	followers	genres	href		id	images	name	pop
0	{'spotify': 'https://open.spotify.com/artist/5... 4603}	{'href': None, 'total': 4603}	[]	https://api.spotify.com/v1/artists/5qJ1s1VYuUL5e6HOoiG58d		5qJ1s1VYuUL5e6HOoiG58d	[{'height': 640, 'url': 'https://i.scdn.co/ima...}]	J-Pop Factory	100
1	{'spotify': 'https://open.spotify.com/artist/4... 5336}	{'href': None, 'total': 5336}	[]	https://api.spotify.com/v1/artists/4QetRFFnJpfUtu0sLRaQZy		4QetRFFnJpfUtu0sLRaQZy	[{'height': 640, 'url': 'https://i.scdn.co/ima...}]	Sarina	100
2	{'spotify': 'https://open.spotify.com/artist/1... 1704}	{'href': None, 'total': 1704}	[]	https://api.spotify.com/v1/artists/1C6w6vaTW2SngOe8YG1LIG		1C6w6vaTW2SngOe8YG1LIG	[{'height': 640, 'url': 'https://i.scdn.co/ima...}]	河野純喜(JO1)	100

จัดการ S

concat data1 and data2

```
data = pd.concat([data1, data2], ignore_index=True)
# drop column external_url href image
data.drop(columns=['external_urls', 'href', 'images'], inplace=True)
data.columns
data
```

	followers	genres	id	name	popularity	type	uri
0	{href: None, total: 2232016}	[japanese teen pop]	6mEQK9m2krja6X1cfuAjl	Ado	76	artist	spotify:artist:6mEQK9m2krja6X1cfuAjl
1	{href: None, total: 1567858}	[j-acoustic, j-pop, japanese singer-songwriter]	1S2S00lgLYLGHWA44qGEUs	Gen Hoshino	65	artist	spotify:artist:1S2S00lgLYLGHWA44qGEUs
2	{href: None, total: 2097714}	[j-pop, j-pop boy group, j-poprock, j-rock]	14T8NkbwXVZgbOvwnuGV89	GReeeN	63	artist	spotify:artist:14T8NkbwXVZgbOvwnuGV89
3	{href: None, total: 112169}	[j-pop]	0EmfXhT2Bhc53cyK3hrZMS	BLOOM	63	artist	spotify:artist:0EmfXhT2Bhc53cyK3hrZMS
4	{href: None, total: 65370}	[anime, j-poprock, okinawan pop]	35VNJ4BFDvs9XkLf6ghU6	Stereopony	42	artist	spotify:artist:35VNJ4BFDvs9XkLf6ghU6
...
95	{href: None, total: 2291}	[]	6b2avTcQHdmOBEQVaxQE5P	北谷 瑞喜	41	artist	spotify:artist:6b2avTcQHdmOBEQVaxQE5P
96	{href: None, total: 0}	[]	6cPpozCFvG43JeBJtJDXW	からいば	21	artist	spotify:artist:6cPpozCFvG43JeBJtJDXW
97	{href: None, total: 18840}	[japanese teen pop]	7qfEFJ6L2AaZ5iuoqTNAZI	KooP	46	artist	spotify:artist:7qfEFJ6L2AaZ5iuoqTNAZI
98	{href: None, total: 209}	[]	1zAxRDFzbhi0CFKtd4M9U	她	20	artist	spotify:artist:1zAxRDFzbhi0CFKtd4M9U