

# Fall 2023 Leadership

- President Rebecca Seifert
- Vice President Jenna Abernethy
- Treasurer Kenzo Hubert
- Secretaries Emily Garceau,
   Meghana Ponnapati
- Education Committee Head –
   Danielle Hahn
- Engagement Committee
   Head Sam Brown

# Data Science for Music

DATA SCIENCE CLUB | 2023.09.29 | REBECCA SEIFERT

,			
(max mean min)	Taylor Swift	Drake	BTS
Words	584 433.8	1057 543.1	1370 953.5
Words	336	228	388
Words/Line	5.97	7.09	4.12
Line	93 72.7 53	129 76.6 46	329 231.4 72

## Potential Uses



## **Producers**

What kind of talent to hire?
How to promote?



## **Artists**

Where to advertise? Where to tour?



### Consumers

What artists to try? Who to befriend?

## Sourcing

- amazon music | ARTISTS
- **Spotify** for Developers
  - भी। Chartmetric
  - Soundcharts
    - last.fm

- First party
  - APIs: Amazon Music, Spotify, Apple Music
  - Direct request (GDPR)
- Cross-channel
  - Services: Chartmetric, Soundcharts
  - Open source data sets
- DIY
  - Services: Last.fm, Bandcamp
  - Web scraping

# Common Projects







DASHBOARDS

NATURAL LANGUAGE PROCESSING

METRICS: ENGAGEMENT, DEMOGRAPHICS, AIRPLAY



- Overview
- Getting started
- ▶ Concepts
- **▼** Tutorials

Authorization code

Authorization code PKCE

Client credentials

Implicit grant

▼ How-Tos

Display your Spotify profile data in a web app

#### REFERENCE

- ▶ Albums
- Artists
- ► Audiobooks
- Categories
- Chapters
- ▶ Episodes
- ▶ Genres



Retrieve metadata from Spotify content, control playback or get recommendations

Spotify Web API enables the creation of applications that can interact with Spotify's streaming service, such as retrieving content metadata, getting recommendations, creating and managing playlists, or controlling playback.

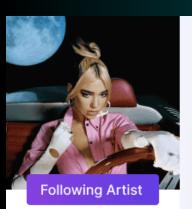
#### **Getting started**

This is where the magic begins! The following steps will help you to get started with your journey towards creating some awesome music apps using the API:

- 1. Log into the dashboard using your Spotify account.
- 2. <u>Create an app</u>. Once you have created your app, you will have access to the app credentials. These will be required for API <u>authorization</u> to obtain an <u>access token</u>.
- 3. Use the access token in your API requests.

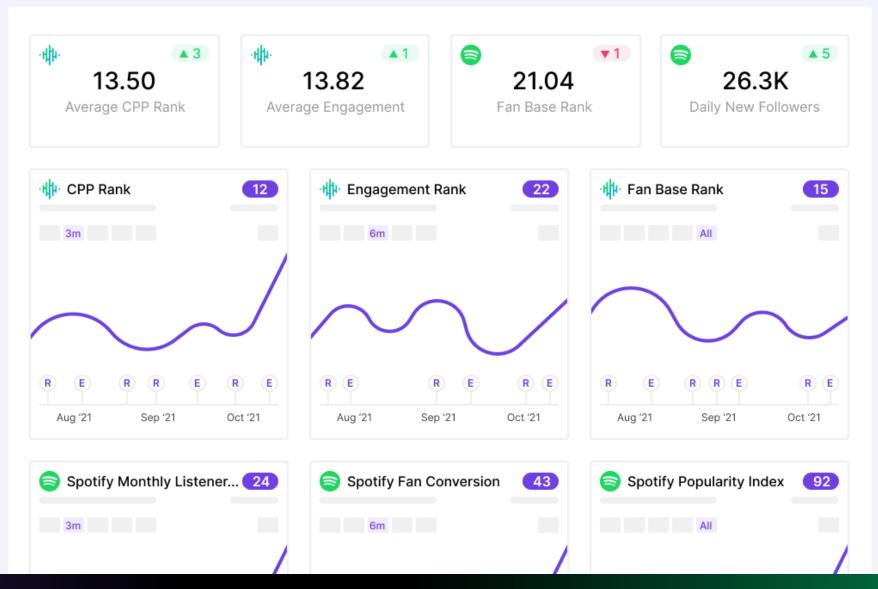
You can follow the Getting started tutorial to learn how to make your first Web API call.

#### **Documentation**



#### Dua Lipa

#### Stats



song	artist	words	lines	lyrics
lovestory	taylorswift	378	64	['We', 'were', 'both', 'young', 'when', 'l', 'first', 'saw', 'you', 'l', 'close', 'my', 'eyes', 'and', 'the', 'flashback', 'starts:', "I'm", 'standing', 'there', '(
blankspace	taylorswift	502	93	['Nice', 'to', 'meet', 'you,', 'where', 'you', 'been?', 'I', 'could', 'show', 'you', 'incredible', 'things', 'Magic,', 'madness,', 'heaven,', 'sin', 'Saw', 'yo
shakeitoff	taylorswift	584	89	['I', 'stay', 'out', 'too', 'late', 'Got', 'nothing', 'in', 'my', 'brain', "That's", 'what', 'people', 'say,', 'mmm-mmm', "That's", 'what', 'people', 'say,',
youbelong	taylorswift	396	81	["You're", 'on', 'the', 'phone', 'with', 'your', 'girlfriend', "She's", 'upset,', "she's", 'going', 'off', 'about', 'something', 'that', 'you', 'said', "'Cause
weareneve	taylorswift	364	53	['I', 'remember', 'when', 'we', 'broke', 'up', 'the', 'first', 'time', 'Saying,', '"This', 'is', 'it,', "I've", 'had', 'enough,"', "'cause", 'like', 'We', "hadn't'
style	taylorswift	377	54	['Midnight', 'You', 'come', 'and', 'pick', 'me', 'up,', 'no', 'headlights', 'Long', 'drive', 'Could', 'end', 'in', 'burning', 'flames', 'or', 'paradise', 'Fade
wildestdrea	taylorswift	358	72	['He', 'said,', '"Let\'s', 'get', 'out', 'of', 'this', 'town', 'Drive', 'out', 'of', 'this', 'city,', 'away', 'from', 'the', 'crowds"', 'I', 'thought,', '"Heaven', "ca
cardigan	taylorswift	336	61	['Vintage', 'tee,', 'brand', 'new', 'phone', 'High', 'heels', 'on', 'cobblestones', 'When', 'you', 'are', 'young,', 'they', 'assume', 'you', 'know', 'noth
lookwhaty	taylorswift	558	83	['I', "don't", 'like', 'your', 'little', 'games', "Don't", 'like', 'your', 'tilted', 'stage', 'The', 'role', 'you', 'made', 'me', 'play', 'Of', 'the', 'fool,', 'no,', 'I'
badblood	taylorswift	485	77	["'Cause,", 'baby,', 'now', 'we', 'got', 'bad', 'blood', 'You', 'know', 'it', 'used', 'to', 'be', 'mad', 'love', 'So', 'take', 'a', 'look', 'what', "you've", 'do
bestieverha	drake	836	96	['You', 'know,', 'a', 'lot', 'of', 'girls', 'be', "Thinkin'", 'my', 'songs', 'are', 'about', 'them', 'This', 'is', 'not', 'to', 'get', 'confused', 'This', "one's", 'fo
forever	drake	1057	129	['It', 'may', 'not', 'mean', "nothin'", 'to', "y'all", 'But', 'understand,', "nothin'", 'was', 'done', 'for', 'me', 'So', 'I', "don't", 'plan', 'on', "stoppin'",
onedance	drake	394	74	['Baby,', 'I', 'like', 'your', 'style', 'Grips', 'on', 'your', 'waist', 'Front', 'way,', 'back', 'way', 'You', 'know', 'that', 'I', "don't", 'play', 'Streets', 'not',
hotlineblin	drake	431	72	['You', 'used', 'to', 'call', 'me', 'on', 'my', 'You', 'used', 'to,', 'you', 'used', 'to', 'Yeah', 'You', 'used', 'to', 'call', 'me', 'on', 'my', 'cell', 'phone', 'Late
over	drake	603	74	['Uh,', 'uh', 'Young', 'Money', 'Yeah,', 'yeah,', 'yeah', 'I', 'know', 'way', 'too', 'many', 'people', 'here', 'right', 'now', 'That', 'I', "didn't", 'know', '
holdonwer	drake	313	56	['I', 'got', 'my', 'eyes', 'on', 'you', "You're", 'everything', 'that', 'I', 'see', 'I', 'want', 'your', 'hot', 'love', 'and', 'emotion', 'endlessly', 'I', "can't", '
passion frui	drake	228	46	['Hold', 'on,', 'hold', 'on,', 'fuck', 'that', 'Fuck', 'that', 'shit', 'Hold', 'on,', 'I', 'got', 'to', 'start', 'this', "motherfuckin'", 'record', 'over', 'again,', 'w
godsplan	drake	347	55	['And', 'they', "wishin'", 'and', "wishin'", 'and', "wishin'", 'and', "wishin'", 'They', "wishin'", 'on', 'me,', 'yeah', 'l', 'been', "movin'", 'calm,', "do
takecare	drake	479	65	['I', 'know', "you've", 'been', 'hurt', 'by', 'someone', 'else', 'I', 'can', 'tell', 'by', 'the', 'way', 'you', 'carry', 'yourself', 'If', 'you', 'let', 'me,', "here's
niceforwha	drake	743	99	['I', 'wanna', 'know', 'who', "motherfuckin'", "representin'", 'in', 'here', 'tonight', 'Hold', 'on,', 'hold', 'on', 'I', 'keep', "lettin'", 'you', 'back', 'in
dynamite	bangtanbo	450	75	["'Cause,", 'ah-ah,', "I'm", 'in', 'the', 'stars', 'tonight', 'So', 'watch', 'me', 'bring', 'the', 'fire', 'and', 'set', 'the', 'night', 'alight', 'Shoes', 'on,', 'ge'
boywithluv	bangtanbo	1294	295	['[Romanized:]', 'Modeun', 'ge', 'gunggeumhae', "How's", 'your', 'day', 'Oh', 'tell', 'me', '(oh', 'yah', 'oh', 'yah,', 'oh', 'yah', 'oh', 'yah)', 'Mwog
butter	bangtanbo	388	72	['Smooth', 'like', 'butter', 'Like', 'a', 'criminal', 'undercover', "Gon'", 'pop', 'like', 'trouble', "Breakin'", 'into', 'your', 'heart', 'like', 'that', '(Ooh
fakelove	bangtanbo	1365	256	['[Romanized:]', 'Neol', 'wihaeseoramyeon', 'nan', 'Seulpeodo', 'gippeun', 'cheok', 'hal', 'suga', 'isseosseo', 'Neol', 'wihaeseolamyeon', 'nan',
dna	bangtanbo	894	266	['[Romanized:]', '[V:]', 'Cheotnune', 'neol', 'araboge', 'dwaesseo', 'Seorol', 'bulleowassdeon', 'geotcheoreom', 'Nae', 'hyeolgwan', 'sok', 'DN,
bloodswea	bangtanbo	819	215	['[Romanized:]', 'Nae', 'pi', 'ttam', 'nunmul', 'Nae', 'majimak', 'chumeul', 'Da', 'gajyeoga', 'ga', 'Nae', 'pi', 'ttam', 'nunmul', 'Nae', 'chagaun', 's
idol	bangtanbo	923	307	['[Romanized:]', 'You', 'can', 'call', 'me', 'artist', '(artist)', 'You', 'can', 'call', 'me', 'idol', '(idol)', 'Anim', 'eotteon', 'dareun', 'mwora', 'haedo', 'I
blackswan	bangtanbo	1194	293	['[Romanized:]', 'Do', 'your', 'thang', 'Do', 'your', 'thang', 'with', 'me', 'now', 'Do', 'your', 'thang', 'your', 'thang', 'with', 'me', 'now', "What
euphoriath	bangtanbo	838	206	['[Romanized:]', 'Neoneun', 'nae', 'salme', 'dasi', 'tteun', 'haetbit', 'Eorin', 'sijeol', 'nae', 'kkumdeurui', 'jaerim', 'Moreugesseoi', 'gamjeongi',
on	bangtanbo	1370	329	['[Romanized:]', 'I', "can't", 'understand', 'what', 'people', 'are', "sayin'", 'Eoneu', 'jangdane', 'matchwoya', 'doelji', 'Han', 'baljaguk', 'ttemye

```
# Import necessary packages
import requests
from bs4 import BeautifulSoup, NavigableString, Tag
import pandas as pd
import csv
# Top 10 (All Time) Songs from Top 3 Artists from last.fm
# https://www.last.fm/charts
  https://www.last.fm/music/Taylor+Swift
    https://www.last.fm/music/Drake
# https://www.last.fm/music/BTS
songs = {'taylorswift': ['lovestory',
                          'blankspace',
                          'shakeitoff',
                          'youbelongwithme'.
                          'weareneverevergettingbacktogether',
                          'style',
                          'wildestdreams',
                          'cardigan',
                          'lookwhatyoumademedo',
                          'badblood'],
         'drake': ['bestieverhad',
                    'forever',
                   'onedance',
                   'hotlinebling',
                   'over',
                   'holdonweregoinghome',
                   'passionfruit',
                   'godsplan',
                   'takecare',
                   'niceforwhat'],
         'bangtanboys': ['dynamite',
                          'boywithluv',
                          'butter',
                          'fakelove',
                          'dna',
                          'bloodsweattears'.
                          'idol',
                          'blackswan',
                          'euphoriathemeofloveyourselfwonder',
                          'on']}
```

```
# Generating corresponding strings following AZLyrics (https://www.azlyrics.com/)
# format and desired file names
# URLs: https://www.azlyrics.com/lyrics/<artistname>/<songtitle>.html
# Raw files: rawHTML/<artistname>/<songtitle>.html
# Final files: lyrics/<artistname>/<songtitle>.txt
urls = []
rawFiles = []
saveFiles = []
for key in songs:
   for title in songs[key]:
       urls.append('https://www.azlyrics.com/lyrics/' + key + '/' + title + '.html')
       rawFiles.append("rawHTML/" + key + "/" + title + ".html")
       saveFiles.append("lyrics_" + key + "_" + title + ".txt")
# Checking urls, raw file names, and final file names
print(urls[0], '\n', rawFiles[0], '\n', saveFiles[0])
print(urls[10], '\n', rawFiles[10], '\n', saveFiles[10])
print(urls[20], '\n', rawFiles[20], '\n', saveFiles[20])
# Testing file creation for one page
html = requests.get(urls[0])
file = open(rawFiles[0], 'wb')
file.write(html.content)
file.close()
# Savings all pages to files
for i in range(len(urls)):
   html = requests.get(urls[i])
   file = open(rawFiles[i], 'wb')
   file.write(html.content)
   file.close()
```

```
# Testing navigation on one page file
file = open(rawFiles[0], 'rb')
soup = BeautifulSoup(file.read(), "html.parser")
 file.close()
# print(soup)
lyrics = soup.body.find('div', {"class": "ringtone"}).next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.nex
print(lyrics[0:200])
# Testing file output for one page
lyrics = soup.body.find('div', {"class": "ringtone"}).next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next sibling.next sibling.get text().strip()
file = open(saveFiles[0], 'w')
file.write(lyrics)
file.close()
 # Reproducing for all songs
 for i in range(len(rawFiles)):
                        # Navigating
                        file = open(rawFiles[i], 'rb')
                         soup = BeautifulSoup(file.read(), "html.parser")
                         file.close()
                        # Exporting (conditional logic to handle different page structures)
                        if (soup.body.find('div', {"class": "ringtone"}).next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sib
                                                   lyrics = soup.body.find('span', {"class": "feat"}).next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_sibling.next_s
                         else:
                                                   lyrics = soup.body.find('div', {"class": "ringtone"}).next sibling.next sibling.nex
                        file = open(saveFiles[i], 'w')
                       file.write(lyrics)
                        file.close()
```

```
# Producing CSV
# Testing one file read and count
file = open(saveFiles[0], 'rb')
lyrics = file.read().strip()
file.close()
words = lyrics.split()
words = [str(word, encoding='utf-8?', errors='strict') for word in words]
countWords = len(words)
print(countWords)
# Creating 2D array to store extracted information
data = [['song', 'artist', 'words', 'lines', 'lyrics']]
index = 0
for key in songs:
    for title in songs[key]:
        # Count lines
        file = open(saveFiles[index], 'rb')
        countLines = len(file.readlines())
        file.close()
        # Extract lyrics
        file = open(saveFiles[index], 'rb')
        lyrics = file.read().strip()
        file.close()
        # Process lyrics
        words = lyrics.split()
        words = [str(word, encoding='utf-8?', errors='strict') for word in words]
        countWords = len(words)
        # Add observation to dataset
        if countWords != 0:
            data.append([title, key, countWords, countLines, words])
        else:
            data.append([title, key, 'error', 'error', words])
        index += 1
# Check
print(data[0], data[1][0], data[1][1], data[1][2])
# Writing array to CSV
file = open('data summary.csv', 'w+')
csvWriter = csv.writer(file, delimiter=',')
csvWriter.writerows(data)
file.close()
```

(max mean min)	Taylor Swift	Drake	BTS
Words	584	1057	1370
	433.8	543.1	953.5
	336	228	388
Words/Line	5.97	7.09	4.12
Line	93	129	329
	72.7	76.6	231.4
	53	46	72

# FAQ

Will we learn data science or just about data science?

If I am just joining now, what do I need to do to catch up?

Do I need prior data science experience to participate?

## Upcoming



2 general meetings in October time/date TBD



Carolina Data Challenge September 30



NCSU Data Science Career Expo October 16



Plant Sciences Initiative Hackathon October 20

