UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, MODELLISTICA, ELETTRONICA E SISTEMISTICA

DIMES

University degree in

Computer Engineering for the Internet of Things

Course of

Big Data Analytics

Project

Recommendation System for Spotify Users relying on Content-based Filtering approach

GitHub Link

click here

Professors

Andrea Tagarelli

Antonio Caliò

Candidate

Fabio Capparelli
214490

INDEX

INTRODUCTION

The main aim of this project has been to develop a python “Recommendation System” for Spotify songs to Spotify users, using the most known “spotipy” python library that provides to us the “API” to connect our python code to the Spotify environment, and retrieving information about songs, albums, artist, and generic users.

This “Recommendation System” is built using the content-based filtering, to provide the most suitable and affine songs in terms of songs attributes to users according to the same songs attributes obtained considering the songs in the user playlists.

RECCOMENDATION SYSTEM

What is a Recommendation system?

Essentially it is a system that wants to predict the “rating” or possible “preferences” of a user or a group of users to a specific item. These systems could be used in a huge variety of areas, and especially in the Big Data area, for instance to suggest songs, videos on web, social media contents, movies and Tv Series, books and so on. They are used in the most known social media and content streaming platform such as Facebook, Instagram, YouTube, Spotify, Netflix, Amazon etc.

It uses many algorithms and approaches to suggests and predict user preferences, such as:

- Collaborative filtering.
- Content-based filtering.
- Session-based recommender systems.
- Multi-criteria recommender systems.
- Risk-aware recommender systems.
- Mobile recommender systems.
- Hybrid recommender systems.

CONTENT-BASED FILTERING

The principal concept in this kind of approach is the term “content”, so this filter uses item features, attributes, or description to recommend and predict similar item to the user based on user’s likes, previous actions, feedbacks, or interactions.

The steps involved in the building of a recommendation system based on content-filtering are the following:

1. Acquiring items/objects knowledge.
2. Acquiring user profile knowledge.
3. Performing dot product to select similar items.

Acquiring items/objects knowledge

This step relies on selecting proper attributes, features of the specific items and assigning to those a value that can be binary, numerical that will represent the “Content”.

Essentially a generic item is described as a vector, of n components:

$$\boldsymbol{X = \{ x_1, x_2, x_3, \dots, x_n\}}$$

those components are the value (numerical, binary) assigned to the attribute of that item.

Acquiring user profile knowledge

Similarly, to the previous step, a record that defines the user profile, in relation with the same attributes of the item interested. This step depends on the user like, actions and feedbacks provided in the past to the same kind of item.

It is always built a vector with n components (numerical and binary values), with the same structure of the previous one:

$$\boldsymbol{Y} = \{\, \boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \dots, \boldsymbol{y_n} \}$$

The unique difference stands on the assignment of these attributes is for the user and not for a specific item.

Performing dot product

Now it is necessary to gain a recommendation or similarity measure, it could be done performing the dot product among the two vectors, higher is the result more similar are the user profile with the specific item, in terms of contents. And viceversa, lower is this result less similar is the user profile with the specific item.

Recommended measure, R of item k obtained performing the dot product:

$$\boldsymbol{R_k = \langle X_k \cdot Y \rangle = x_{k,1} \cdot y_1 + \cdots + x_{k,n} \cdot y_n = \sum_{i=1}^{n} x_{k,i} \cdot y_i}$$

At the end it is required and necessary to specify a support S, for providing a proper threshold to consider the item “similar” or “affine” to the user and could be recommended, otherwise if the similarity measure is less than this threshold it is not considered similar.

$$\begin{cases} R_k < S \ \rightarrow Not\ Recommended \\ R_k \geq S \ \rightarrow \ Recommended \end{cases} \qquad V \qquad \begin{cases} R_k \leq S \ \rightarrow Not\ Recommended \\ R_k > S \ \rightarrow \ Recommended \end{cases}$$

OUR SOLUTION

Our solution relies on the following three steps, according to what has been explained before:

1. Building the user profile record performing the average among the "audio_features", attributes which characterize the songs inside the user playlists.
2. Getting the same "audio_features" of about 500 tracks of the newest albums uploaded on spotify.
3. Building the final recommender system performing the dot product among the user profile record with all newest tracks and, at the end, selecting the tracks that have the result above the support provided via console by the user. A value inside the [0,1] interval.

As we can notice, the item we are interested in will be the audio tracks, specifically the user playlists tracks and the newest uploaded songs on spotify. The attributes of these items will be the audio features, that explain which characteristics the song have in terms of:

- **Danceability** → describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
- **Energy** → is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale.
- **Liveness** → Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
- **Speechiness** → Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered,

including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

- **Acousticness** → A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- **Instrumentalness** → Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks.
- **Tempo** → The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
- **Valence** → A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Preprocessing phase

It is required a preliminary phase, consisting of a preprocessing/transformation of these audio_features in integer numbers for obtaining consistent results after the "dot product".

Most of these audio_features stand in range from [0, 1], and they are represented by a float number. For converting them to integer value the following operation is performed:

$$\boldsymbol{ATTRIBUTES \rightarrow Integer_{Attr} = round\ (\ Float_{attr} \cdot 5\ )}$$

Where the round operation in python is the approximation of the number obtained by the product to the nearest integer, its floor or ceil. The final value stands in the [0, 5] interval.

Exception is made for "tempo" audio_feature, because this attributes in measured in bpm and its values is obviously greater than 1. So, after a research on spotify tracks, a suitable upper bound for this attribute has been set and it is “190 bpm”, the reason why is that the songs that have the highest tempo value are characterized by 180, 185 bpm a value always less than 190. Dividing the tempo value of each song with this upper bound we restrict the range inside the [0, 1] interval. The operation performed has been the following.

$$\boldsymbol{TEMPO \rightarrow Integer_{Attr} = round\left(\frac{Integer_{attr}}{190} \cdot 5\right) = round\left(\frac{Integer_{attr}}{38}\right)}$$

Building the user profile record

We have considered only the songs that are contained in the user “public” playlists just putting the spotify “user_id” . We have discarded the songs inside the “Favorites” and in the “Saved Albums”, this because these informations are strictly private, and in order to get them we had to log as spotify developer for each user, and not just putting the user_id.

So, the building of the user vector according to the audio features described before is performed using the simple arithmetic mean of each attribute for all tracks. For doing this, we have performed the preprocessing stage (described before) to transform each float attribute to an integer one in the range [0, 5], after it has been computed the average.

Here it is reported this preprocessing for the *Maneskin* song “*Zitti e Buoni*”.

Danceability	Energy	Liveness	Speechiness	Acousticness	Instrumentalness	Tempo	Valence
0.625	0.939	0.424	0.0669	0.00138	0	102.999	0.644

Danceability	Energy	Liveness	Speechiness	Acousticness	Instrumentalness	Tempo	Valence
3	5	2	0	0	0	3	3

The final mean value is approximated o the nearest integer, using always the round python method.

We have analyzed many users, for instance the user profile records got for “Fabio Capparelli”, “Francesco Raco” and “Luigi Rachiele”, have been the following:

Fabio Capparelli, id = ‘11102339711’

Danceability	Energy	Liveness	Speechiness	Acousticness	Instrumentalness	Tempo	Valence
3	3	1	1	1	0	3	2

Francesco Raco, id = ‘prp468n1n5qp2sdr1ps5hk8t0’

Danceability	Energy	Liveness	Speechiness	Acousticness	Instrumentalness	Tempo	Valence
3	3	1	0	1	0	3	3

Luigi Rachiele, id = ‘gixs’

Danceability	Energy	Liveness	Speechiness	Acousticness	Instrumentalness	Tempo	Valence
3	3	1	0	1	1	3	2

Retrieve the newest songs

On this step we have retrieved the newest songs inside the last albums published on spotify, up to 500 songs. Always for each song has been taken the attributes described above and preprocessed in the same way to get concistency among the user record and these tracks.

Example of newest songs retrieve on the July 20:

N, id, Title, Artist, Attributes

```
0 ['1oZq1wMYbEkr7ovRV7OWty', 'Good News', ' Pop Smoke', 3, 3, 1, 1, 1, 0, 3, 3]

1 ['2wzionesqlt3x5Sglb4n91', 'More Time', ' Pop Smoke', 3, 3, 1, 1, 2, 0, 4, 1]

2 ['2UwALqx6yOsXTFt7zRxnts', 'Tell The Vision (feat. Kanye West & Pusha T)', ' Pop Smoke', 3, 3, 2, 2, 1, 0, 4, 3]
```

```
3 ['5EVzgDPaM0LCM3M5exl3z8', 'Manslaughter (feat. Rick Ross & The-Dream)', ' Pop Smoke', 2, 3, 1, 0, 1, 0, 2, 1]

4 ['4LaGu95Ui2s4vprSQYWUAZ', 'Bout A Million (feat. 42 Dugg & 21 Savage)', ' Pop Smoke', 4, 3, 1, 1, 3, 0, 4, 2]

5 ['3MmvQKMKdN3UhVS3eGzJIW', 'Brush Em (feat. Rah Swish)', ' Pop Smoke', 3, 4, 3, 2, 1, 0, 2, 4]

                                        .

                                        .

                                        .

                                        .

                                        -

494 ['6LI6eBUfd7iQy55SaKCtIj', 'FUCKED UP', ' jxdn', 2, 3, 1, 0, 0, 0, 2, 1]

495 ['213zYbLMb3NjOfNl0LFTQr', 'SO WHAT!', ' jxdn', 3, 3, 0, 0, 0, 0, 5, 3]

496 ['2N0FWGaq52dlN6FH4EvmnL', 'ANGELS & DEMONS PT. 2', ' jxdn', 1, 4, 0, 0, 0, 0, 5, 1]

497 ['7MPxEoT36YBCDbrk3ng85S', 'BETTER OFF DEAD', ' jxdn', 3, 3, 1, 0, 1, 0, 2, 2]

498 ['18o8K3rNosL8tOxbCWgMmp', 'DTA', ' jxdn', 3, 4, 1, 0, 0, 0, 2, 3]

499 ['0AFUpHDK01kwapBhVJBtio', 'LAST TIME', ' jxdn', 2, 4, 2, 0, 0, 0, 5, 2]

500 ['0tQPcLmwvAnS5eXSuAUokD', 'NO VANITY', ' jxdn', 2, 4, 1, 0, 0, 0, 4, 2]
```

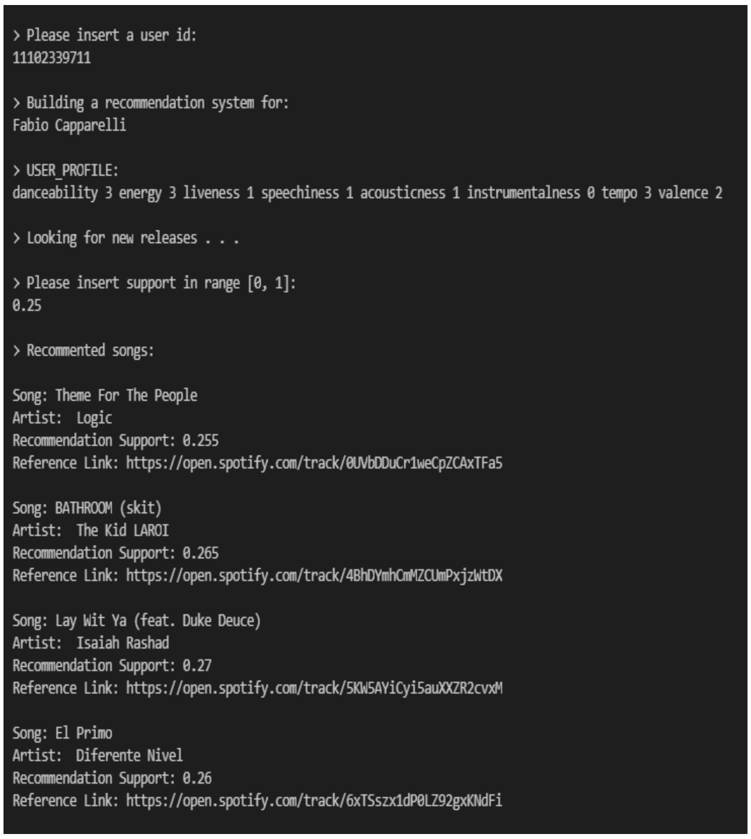
Recommend songs

This is the final step of our content-based solution. Now, as it has been explained before it is performed the dot product among the user profile vector and each new song. The user that interacts with our application through the command line can specify the min_support for the suggestion.

We have user profile vector and songs vector consisting of 8 elements. So, the max_rating obtainable is 200. Obviously, the min_support value should be an integer in [0, 1] interval.

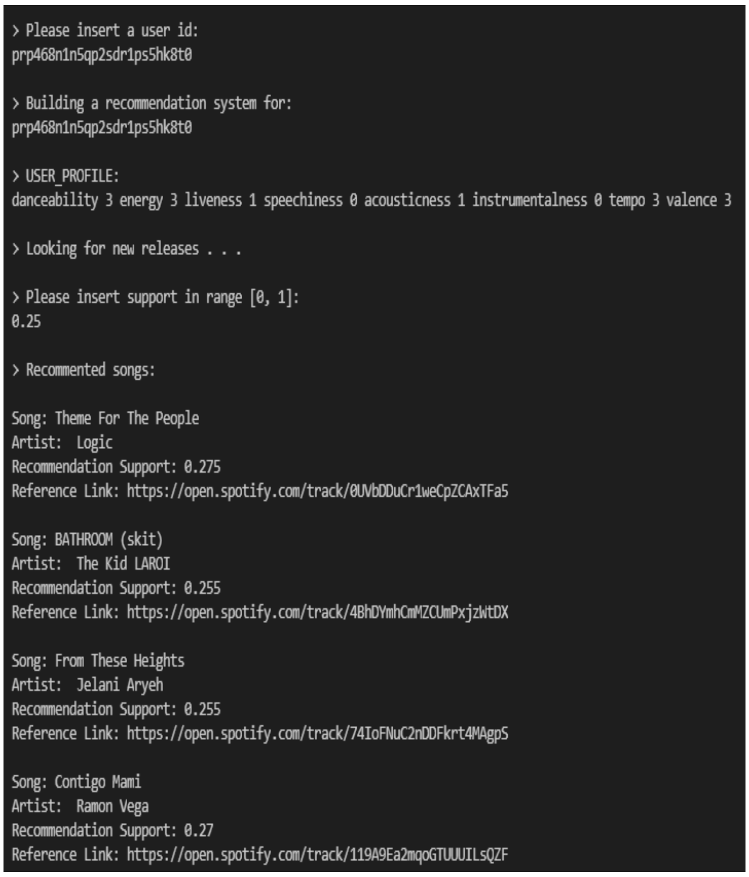
On the following section have been shown some snapshots of our terminal and the songs suggested for the users mentioned before.

Fabio Capparelli

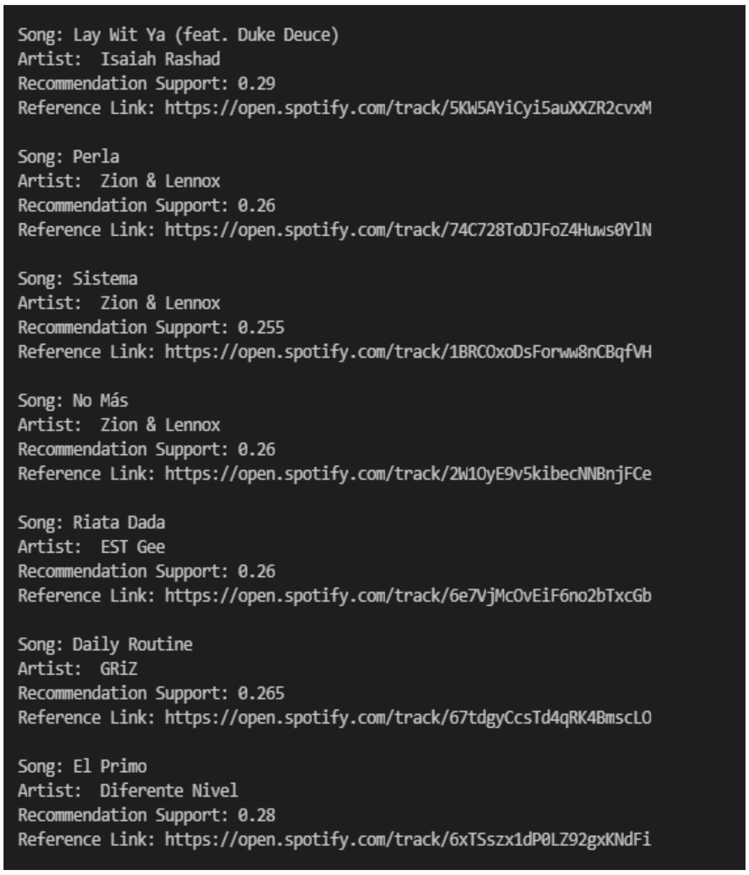


```
> Please insert a user id:
11102339711

> Building a recommendation system for:
Fabio Capparelli

> USER_PROFILE:
danceability 3 energy 3 liveness 1 speechiness 1 acousticness 1 instrumentalness 0 tempo 3 valence 2

> Looking for new releases . . .

> Please insert support in range [0, 1]:
0.25

> Recommented songs:

Song: Theme For The People
Artist:  Logic
Recommendation Support: 0.255
Reference Link: https://open.spotify.com/track/0UVbDDuCr1weCpZCAxTFa5

Song: BATHROOM (skit)
Artist:  The Kid LAROI
Recommendation Support: 0.265
Reference Link: https://open.spotify.com/track/4BhDYmhCmMZCUmPxjzWtDX

Song: Lay Wit Ya (feat. Duke Deuce)
Artist:  Isaiah Rashad
Recommendation Support: 0.27
Reference Link: https://open.spotify.com/track/5KW5AYiCyi5auXXZR2cvxM

Song: El Primo
Artist:  Diferente Nivel
Recommendation Support: 0.26
Reference Link: https://open.spotify.com/track/6xTSszx1dP0LZ92gxKNdFi
```

Francesco Raco

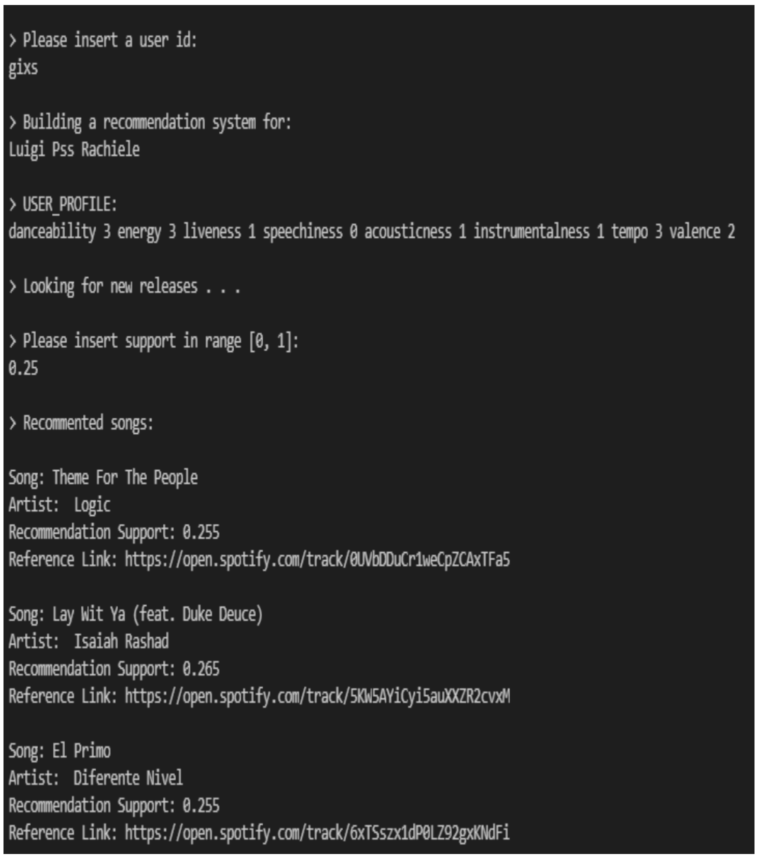


```
> Please insert a user id:
prp468n1n5qp2sdr1ps5hk8t0

> Building a recommendation system for:
prp468n1n5qp2sdr1ps5hk8t0

> USER_PROFILE:
danceability 3 energy 3 liveness 1 speechiness 0 acousticness 1 instrumentalness 0 tempo 3 valence 3

> Looking for new releases . . .

> Please insert support in range [0, 1]:
0.25

> Recommented songs:

Song: Theme For The People
Artist:  Logic
Recommendation Support: 0.275
Reference Link: https://open.spotify.com/track/0UVbDDuCr1weCpZCAxTFa5

Song: BATHROOM (skit)
Artist:  The Kid LAROI
Recommendation Support: 0.255
Reference Link: https://open.spotify.com/track/4BhDYmhCmMZCUmPxjzWtDX

Song: From These Heights
Artist:  Jelani Aryeh
Recommendation Support: 0.255
Reference Link: https://open.spotify.com/track/74IoFNuC2nDDFkrt4MAgpS

Song: Contigo Mami
Artist:  Ramon Vega
Recommendation Support: 0.27
Reference Link: https://open.spotify.com/track/119A9Ea2mqoGTUUUILsQZF
```



```
Song: Lay Wit Ya (feat. Duke Deuce)
Artist:  Isaiah Rashad
Recommendation Support: 0.29
Reference Link: https://open.spotify.com/track/5KW5AYiCyi5auXXZR2cvxM

Song: Perla
Artist:  Zion & Lennox
Recommendation Support: 0.26
Reference Link: https://open.spotify.com/track/74C728ToDJFoZ4Huws0YlN

Song: Sistema
Artist:  Zion & Lennox
Recommendation Support: 0.255
Reference Link: https://open.spotify.com/track/1BRCOxoDsForww8nCBqfVH

Song: No Más
Artist:  Zion & Lennox
Recommendation Support: 0.26
Reference Link: https://open.spotify.com/track/2W1OyE9v5kibecNNBnjFCe

Song: Riata Dada
Artist:  EST Gee
Recommendation Support: 0.26
Reference Link: https://open.spotify.com/track/6e7VjMcOvEiF6no2bTxcGb

Song: Daily Routine
Artist:  GRiZ
Recommendation Support: 0.265
Reference Link: https://open.spotify.com/track/67tdgyCcsTd4qRK4BmscLO

Song: El Primo
Artist:  Diferente Nivel
Recommendation Support: 0.28
Reference Link: https://open.spotify.com/track/6xTSszx1dP0LZ92gxKNdFi
```

Luigi Rachiele



```
> Please insert a user id:
gixs

> Building a recommendation system for:
Luigi Pss Rachiele

> USER_PROFILE:
danceability 3 energy 3 liveness 1 speechiness 0 acousticness 1 instrumentalness 1 tempo 3 valence 2

> Looking for new releases . . .

> Please insert support in range [0, 1]:
0.25

> Recommented songs:

Song: Theme For The People
Artist:  Logic
Recommendation Support: 0.255
Reference Link: https://open.spotify.com/track/0UVbDDuCr1weCpZCAxTFa5

Song: Lay Wit Ya (feat. Duke Deuce)
Artist:  Isaiah Rashad
Recommendation Support: 0.265
Reference Link: https://open.spotify.com/track/5KW5AYiCyi5auXXZR2cvxM

Song: El Primo
Artist:  Diferente Nivel
Recommendation Support: 0.255
Reference Link: https://open.spotify.com/track/6xTSszx1dP0LZ92gxKNdFi
```