SPOTIFY AND RADIO STATIONS

Data Management report

Baghrous Tariq 904027
Balbin Canchanya Gianni Eduard 901609
Galante Fabio 851242

Master's Degree in Data Science
Università degli Studi di Milano-Bicocca
2022/2023 Academic Year

Contents

1 Project Division: Roles

Each project member has performed at least one key task for the completeness of the work, resulting in obtaining the final database. The primary roles are as follows:

- Gianni Eduard Balbin Canchanya: data integration and data quality
- Tariq Baghrous: data acquisition, preparation and exploration
- Fabio galante: data integration, data quality and data storage

2 Introduction

2.1 Project Idea

Spotify is a popular music streaming service that has transformed the music industry. Launched in 2008, it offers an extensive library of music accessible through a freemium model. Thanks to personalized recommendations and social integration, Spotify provides a user-friendly experience, shaping the way we discover and enjoy music in the digital age.

The popular music streaming platform has a strong influence on the ARIA charts in Australia and the Billboard charts in the United States. This is due to its large user base and the data collected through music listening, which influences the positions of songs in the charts. This correlation has improved the accuracy of the charts and reflects the shift in music consumption towards streaming.

Similarly, radio stations play a crucial role in the music industry by promoting and broadcasting new music releases. They curate playlists, collaborate with record labels, and provide visibility to artists, helping them gain recognition and reach a broader audience. Radio broadcasting has a significant impact on music sales and chart positions. Despite the rise of digital platforms changing the music landscape, radio stations remain influential in shaping popular tastes and sparking conversations about music, continuing to play a fundamental role in the discovery and appreciation of music.

In the modern world, music has become an essential part of many people's lives, and the availability of favorite songs on local radio stations is a crucial aspect. However, finding accurate information about radio stations broadcasting preferred songs can be complex.

In this context, the project we present aims to create an innovative database that allows users to identify radio stations in a specific country based on the genre of their favorite and/or most popular songs on the Spotify platform. Consequently, the project offers users an effective solution to discover radio stations where they can listen to the music they love.

This project leverages data from two main sources: data on the Top 10,000 songs on Spotify according to the ARIA and Billboard charts and data on over 66,000 radio stations worldwide.

3 Data Acquisition

3.1 Top 10000 Songs on Spotify 1960-Now

This dataset was downloaded from Kaggle [1] and consists of a comprehensive collection of the top 10,000 songs that have dominated the music scene from 1960 to the present day.

The dataset was curated based on the rankings of both ARIA (Australian Recording Industry Association) and Billboard, ensuring a diverse representation of songs that have achieved immense commercial success and significant cultural importance.

After importing this dataset, we decided to select only some of the available variables, considering them more useful and relevant to our purposes.

The selected attributes are as follows:

- **Track URI:** The unique identifier for the track in the Spotify database.
- **Track Name:** The name of the track.
- **Artist URI(s):** The unique identifier of the artist in the Spotify database.
- **Artist Name(s):** The name of the artist.
- **Album URI:** The unique identifier of the album in the Spotify database.
- **Album Name:** The name of the album.
- **Popularity:** The popularity of the track. The value ranges from 0 to 100, with 100 being the most popular.
- **Artist Genres:** A list of genres associated with the artist. If not yet classified, the array is empty.

3.2 Spotify Web API

The Spotify Web API [2] is a set of tools and services provided by Spotify that can be used to integrate Spotify's functionalities into other platforms. Through a developer account, we were able to acquire various attributes to enrich the initial database:

- **artist genres:** A list of genres associated with the artist. If not yet classified, the array is empty.
- **available markets:** A list of countries where the track can be played, identified by their ISO 3166-1 alpha-2 code.
- **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 the least danceable, and 1.0 is the most danceable.
- **energy:** Ranges from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
- **loudness:** QTotal loudness of the track in decibels.
- **speechiness:** Presence of spoken words in a track. The more exclusively spoken the recording is, the closer the attribute value is to 1.0.
- **acousticness:** A confidence measure from 0.0 to 1.0 indicating whether the track is acoustic. 1.0 represents high confidence that the track is acoustic.

- **instrumentalness:** The closer the instrumentalness value is to 1.0, the greater the likelihood that the track does not contain vocal content.

- **liveness:** Detects the presence of an audience in the recording. Higher liveness values represent a greater likelihood that the track was performed live.

- **valence:** A measure from 0.0 to 1.0 describing the musical positivity conveyed by a track.

- **tempo:** The overall estimated tempo of a track in beats per minute (BPM).

- **track_href:** A link to the Web API endpoint providing all the details of the track.

- **duration_ms:** The duration of the track in milliseconds.

3.3 Radio stations API

The Radio World API [3] is a REST API that allows access to an extensive collection of radio stations from various countries and genres. With approximately 64,000 available radio stations, users can explore a wide range of musical choices that best match their preferences. The attributes extracted from this API are as follows:

- **radio_name:** Name of the radio station.

- **radio_url:** URL link of the radio station.

- **genre:** Genre of the radio.

- **country_name:** Country of origin.

4 Data Preparation

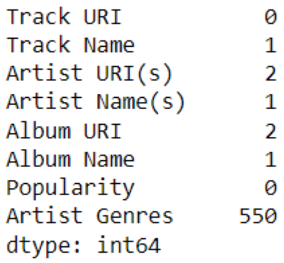
The entire process was carried out on the Jupyter Notebook platform using Python as the programming language.

The process begins with the importation of the first dataset, ”Top 10000 Songs on Spotify 1960-Now,” with the name ”top10k,” which contains 9951 rows and 35 columns. We select 8 columns of interest using the *usecols* command.

Through the Radio World API, we create the ”radio_metadata” dataset consisting of 64230 rows and 4 columns, considered the most relevant, and export a CSV file so that it is available in the directory for subsequent uses.

4.1 Data Cleansing

The next step involves cleaning the datasets. In this regard, we checked for the presence of null data and obtained the following output:



```
Track URI            0
Track Name           1
Artist URI(s)        2
Artist Name(s)       1
Album URI            2
Album Name           1
Popularity           0
Artist Genres      550
dtype: int64
```

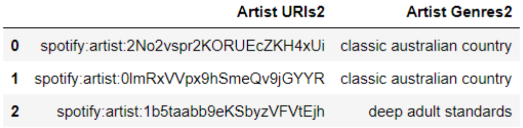
Figure 1: missing values on ”Top 10000 Songs on Spotify 1960-Now”

Therefore, we went on to investigate the nature of the missing values and, if possible, use the Spotify web API to replace them. As for the track with the missing name, we traced it back, through the Track URI, to the track’s URL on Spotify and noticed that it does not exist, which is why we subsequently deleted it.

Extracting the other two tracks with missing Artist URI(s) and Album URI, we noticed that the format of the Track URI starts with ”spotify:local:”, which means they are local files—music tracks not directly available in the Spotify library but added from the personal library within the application.

After removing these latter tracks, we focused on identifying and removing duplicates using the *.drop_duplicates()* command, finding approximately 48 duplicate rows.

The last obstacle is due to the small number of songs with missing Artist Genres. For this issue, we attempted to use the Spotify web API to find the missing genres by entering the Track URI as input. As seen in the image below, we found only three matches, and the remaining rows will be subsequently discarded.



	Artist URIs2	Artist Genres2
0	spotify:artist:2No2vspr2KORUEcZKH4xUi	classic australian country
1	spotify:artist:0lmRxVVpx9hSmeQv9jGYYR	classic australian country
2	spotify:artist:1b5taabb9eKSbyzVFVtEjh	deep adult standards

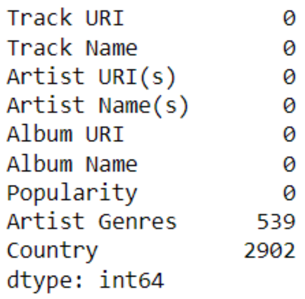
Figure 2: Artist Genres

4.2 Data Enrichment

In this phase of the process, we enriched the initial dataset with additional information that will allow us to both integrate with the *radio_metadata* and extract all the musical features for each track.

In particular, we added a column named ”Country” in which, for each track, a list of countries where the song can be played is reported, identified by their ISO 3166-1 alpha-2 code. This was possible through the Spotify Web API *”sp.tracks”*, entering the various track URIs as input and obtaining the *available_markets* as output.

Now that we have all the desired attributes for top10k, we proceed to identify the rows with null/empty values for Artist Genre and Country and remove them from the dataframe using the *dropna()* command (COMPLETENESS).



```
Track URI             0
Track Name            0
Artist URI(s)         0
Artist Name(s)        0
Album URI             0
Album Name            0
Popularity            0
Artist Genres       539
Country            2902
dtype: int64
```

Figure 3: missing artist genres and countries

Using the Spotify Web API *”sp.audio_features”* and the track URIs from the dataset, we created an additional dataframe called *top10k_features*, which includes all the interesting features for each song (previously described in section 2.2).

Regarding the radio dataset, we encountered a CONSISTENCY issue with the top10k dataset, as countries are reported with their full names in one dataset while in the musical tracks dataset, they are reported with their ISO 3166-1 alpha-2 code, making subsequent integration impossible. We decided to add a column in the *radio_metadata* dataframe called ”country_code,” which contains the ISO 3166-1 alpha-2 code based on the country_name. To perform this mapping, we use a Python library called *”pycountry”*. The matching was not 100% due to the unusual format in which some countries were written (CONSISTENCY), so we had to manually add the codes for countries that did not provide one.

We have prepared three datasets named **top10k_spotify.csv**, **radio_metadata.csv**, and **top10k_features.csv**, ready for the next phase of the project.

4.3 Data Integration

Using the two obtained datasets, *top10k_spotify* and *radio_metadata*, we sought a way to integrate them, with the ultimate goal of having a dataframe containing all the songs with their respective associated radio stations.

To achieve this, we created intermediate datasets to streamline the process due to the large volume of data we need to work with.

Firstly, we created from the *top10k_spotify* dataset:

- the file *genres2* containing the information we deemed important for all songs. Consequently, we omitted the Spotify URI addresses and divided the *ArtistGenres* column into multiple columns, equal to the number of possible musical genres in the *top10k_spotify* dataset, which is 32. It is noteworthy that some columns for many songs may be incomplete as not every song necessarily has exactly 32 musical genres.

From the *radio_metadata* dataset, we observed that it, too, could generally have multiple genres, up to a maximum of 3. Consequently, we created:

- The file *radio1genre* which has the same information as the initial dataset but is grouped for those radios that have only one genre;
- The file *radio2genre* where in this case, there are only those with two genres; and,
- The file *radio3genre* which contains only radios that include exactly three musical genres.

Given that all the musical genres of the songs are in lowercase, we have also changed the letter case to lowercase for the genre of the radios in *radio1genre*.

We then proceeded to find the match between songs and radios both by genre and by country and inserted it into *listaok*.

Once we obtained *listaok*, which presents for each song a vector containing the numbers associated with the radios that can play it, we want to transform this list into a dataframe so that it can be exported and used when needed.

We note that different songs have the same vector of radios, so we identify all the different vectors from each other, obtaining a total of 15 vectors. This means that only 15 different combinations of radios can be processed and then assigned to the corresponding songs.

We then begin our process of converting the list into a dataframe, obtaining *match_finale3*.

This dataset has dimensions 3114x5979, where 3114 is the number of tracks, 5975 is the maximum number of radios that can play a certain song. We then add another 4 columns to indicate the name and URI of the artist and track. This dataset contains within each cell that does not belong to the first four columns the URL of a radio on which the corresponding song can be played. However, not all tracks can be played on 5975 radios. Therefore, those with X radios will present the URL in columns 5 to X+5 and will contain 0s in the following columns.

Since this dataset is large, we had problems importing it into SQL, which cannot read so many columns. Therefore, we decided to split it into 4 datasets, each with 1500 columns (except the last one, which has 1479). In this way, we created the following tables:

match1_1,*match1_2*,*match1_3*,*match1_finale4*.

5 Data Storage

Once we obtained our final dataset with integrated radios and songs, we split it into the 4 datasets mentioned earlier and loaded them into SQL along with *generi1radio* and *generi2*, thus obtaining our database.

We decided to run some brief queries to show how our data can be explored and various information can be obtained.

Query 1

With query 1, we aim to retrieve data on radio stations from the dataset *match1_1*, which contains, for each song, all the radio stations on which it can be played.

To do this, we join this dataset with *radio1genere*, finding the match between the radio URL contained in *radio1genere* and the fifth column of *match1_1*, which contains the URL of the first identified radio station on which the song of the corresponding row can be played.

Therefore, as a result, we will obtain a dataset that presents in the first column the name of the song, in the second the name of the radio station, in the third the genre of the radio, and in the fourth the country where this radio station can be listened to.

This table contains fewer than 3114 songs (the number of songs in *match1_1*) because it removes duplicates, keeping only one observation for each individual song.

Query 2

In this query, instead, we start from *radio1genere* and find out which of these radio stations are identified first as stations that can play the songs in *match1_1*). Then, we want to know what is the first song they are associated with, and finally add details about this song.

To do this, we find the match between URLs in *radio1genere* and *match1_1*). After completing this part, we find the match between the song's name in the result obtained from the query and the song's name contained in the dataset *generi2*, from which we obtain further information about the songs (album and artist).

6 Data Exploration

By importing the Python libraries *matplotlib.pyplot* and *seaborn* we explored the three datasets **top10k_spotify.csv**, **radio_metadata.csv** and **top10k_features.csv**.

In the first one, we identified and visualized the top 10 artists who contributed the most tracks to the dataset, as shown in the horizontal bar plot below:

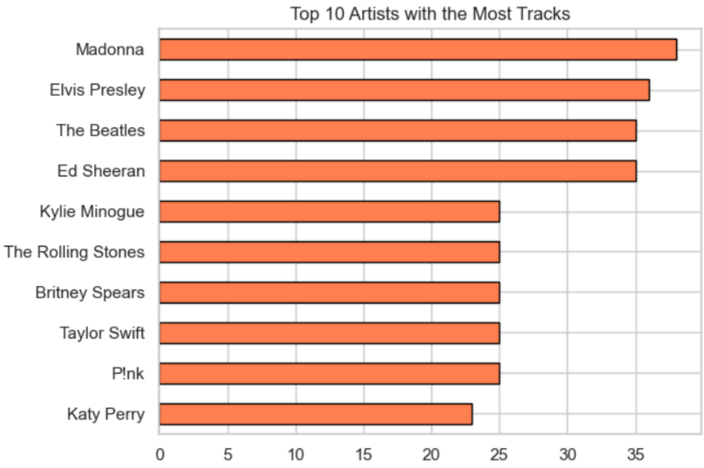


Figure 4: Top 10 Artists with the Most Tracks

From the data analysis, it's evident that Madonna is the artist with the highest number of tracks in our dataset, followed by Elvis Presley and The Beatles.

In the analysis of radio station data, we identified the top 10 countries with the highest number of radio stations:

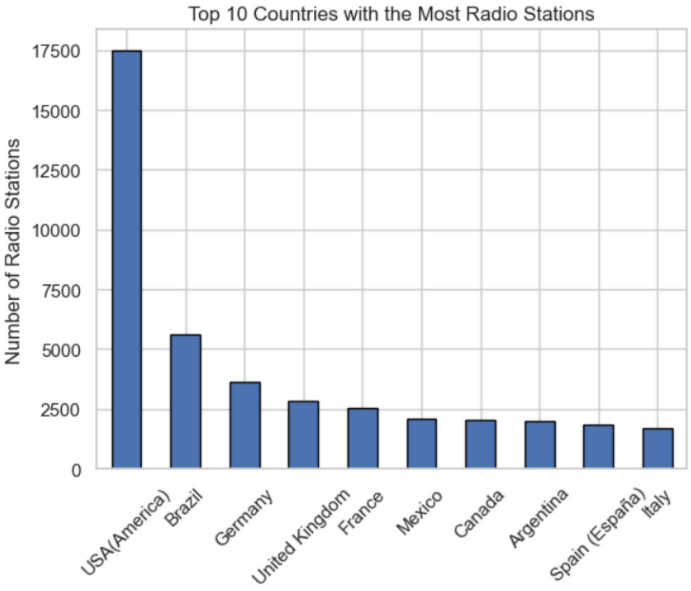


Figure 5: Top 10 Countries with the Most Radio Stations

From observing the graph, it's apparent that the United States of America leads the list, followed by Brazil and Germany. These results highlight the predominant localization of radio stations in the United States, around 17,500 out of 64,230, providing a starting point for further analysis and research in the radio sector.

In the dataset **top10k_features.csv**, we examined the correlations between various musical features and the level of track popularity. This was represented through a heatmap graph of the correlation matrix:

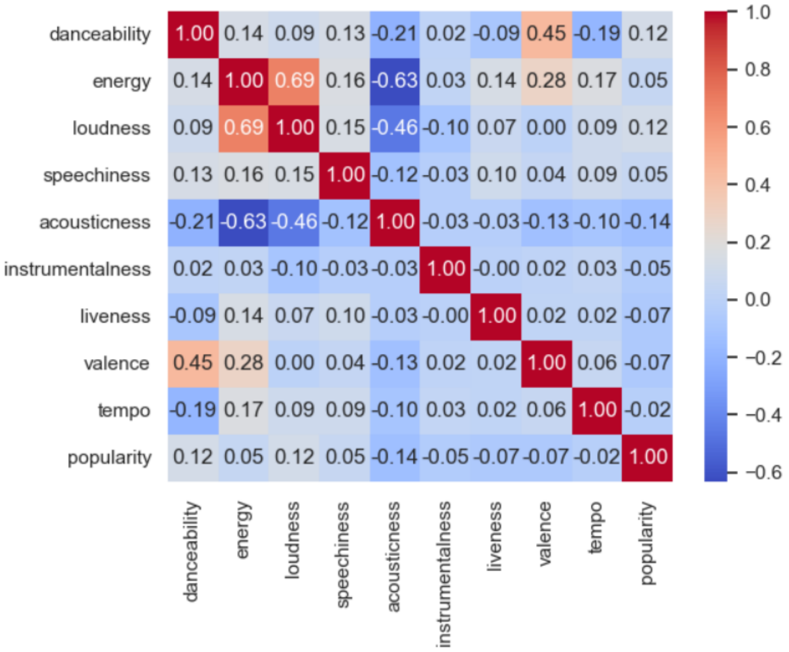


Figure 6: Top 10 Countries with the Most Radio Stations

From observing the graph, we can notice that most of the correlations between these musical features and track popularity hover around 0. This indicates that, in general, there isn't a strong correlation between these features and popularity. However, it's interesting to note that some pairs of features may show a positive or negative correlation, which can provide further insights into the influence of musical features on track popularity.

This type of analysis is crucial for better understanding the factors contributing to the success of a music track and can be used to uncover trends or relevant insights within the top tracks of recent years.

7 Conclusions

The project has successfully achieved the goal of creating an innovative database that allows users to identify radio stations in a specific country using the genre of their favorite songs or the most popular ones on Spotify.

Looking ahead, our project offers wide room for innovation and expansion. Some possible directions include implementing advanced search based on musical preferences, such as allowing users to search for radio stations based on a combination of musical genres, specific decades, countries of origin, and more. This functionality could be implemented through advanced filters and an intuitive user interface.

To further enhance the user experience, a recommendation system based on users' musical preferences could be developed, particularly using machine learning algorithms to analyze listening habits and suggest related radio stations or tracks that the user may enjoy.

An interesting area of development could involve monitoring musical trends, detecting and constantly analyzing emerging trends in music. Such a system could leverage sophisticated algorithms to track new music releases, streaming charts, and user behaviors, integrating the analysis of conversations and social media shares related to music, to provide a comprehensive picture of ongoing trends.

In summary, the evolution of the project could be driven by the desire to improve the user experience, offering more advanced features and a more comprehensive service in the field of music discovery.

References

[1] Kaggle, *Top 10000 Songs on Spotify 1960-Now*, `https://www.kaggle.com/datasets/joebeachcapital/top-10000-spotify-songs-1960-now?resource=download`

[2] Spotify, *Web API*, `https://developer.spotify.com/`

[3] Rapid API Hub, *Radio World - 75,000+ Worldwide FM Radio stations*, `https://rapidapi.com/dpthapaliya19/api/radio-world-75-000-worldwide-fm-radio-stations`