PRI66 Project Report
Spotify lyrics Data

André Santos, Edgar Torre, Miguel Amorim
Faculdade de Engenharia, Porto

ABSTRACT

Information has never been so available as nowadays. That's why the processing of acquiring information and preparing that information is as important as never before. This report intends to document the creation of a good database and an information processing in the context of 2022/2023 Project for Information Processing and Retrieval course.

Keywords: Information Processing and Retrieval, Music, Lyrics, datasets, M.EIC, statistics, music's data

1 INTRODUCTION

As part of course PRI, a project, to develop during the whole semester, was requested. In the end, this project will be an information search system, which means work on data collection and preparation, information querying and retrieval, and retrieval evaluation.

As per the chosen topic, at first we did not know what would be the best option like food, movies or music. Our decision was made based on three reasons, we wanted a dataset with a large amount of data, with at least thousands rows, we wanted different information in the dataset, with at least 10 rows. We also wanted different types of data. With all of that, we decided on a music database.

2 DATASET

2.1 Dataset choice

At the moment we decided to work on music, we start looking for a dataset with a lot of different data, like release year, artist, album, popularity and the most important, the lyrics. We found a dataset with more than 4 million rows with the lyrics of each one and some more data of each music. Although it was difficult to work with so large amount of data we decided to work with this one found on Kaggle (link below).

However, this dataset had some problems, information that we do not need (which we deleted), too much rows to work with (which we deleted and kept just a part of the rows) and lacking different type of information. With that problem in mind, the solution we found was finding some API that could give us the data we need. We found that the Web Spotify API (link below) could give us what we were looking for, like popularity, albums, duration of musics.

In conclusion, we selected a dataset of musics' data collected by someone else and added to it a lot information with the Spotify API.

Kaggle link: https://www.kaggle.com/datasets/nikhilnayak123/5-million-song-lyrics-dataset .

API link: https://developer.spotify.com/documentation/web-api/ .

2.2 Dataset Content

The chosen dataset cointains the following:

- 4 million musics with attributes as artist name, release year, lyrics, title (which we used a 80000 musics, picked randomly).

From the API, for each music we got the following data:

- track number, album's name, album's number of tracks, track's qualification, duration, popularity, release year.

2.3 Data Quality and Source

Kaggle is a well-known community for data searches, the post author states that all the data comes from genius.com, a well-known music platform with 5.5 million followers on social media, with this in mind, we can infer that this data is reliable. About Spotify, since it's an app that is used all around the world and already receives some awards, we can also state it's reliability.

About the quality of the data, we can say that met our criteria, the date we needed was there and with no unexpected values.

3 PIPELINE

Our Data Preparation Pipeline is done, mostly, on Python scripts, where pandas and matplotlib libraries were fundamental to clean and manipulate the data. With that we manage to create a structured and clean SQL database system.

3.1 Pipeline Scheme

After all the analysis of the problem the final version of the pipeline process was the following:

As we can see, the initial data come from the Kaggle dataset ds2.csv and we refined this data, eliminating the duplicated ones and eliminating the columns that we did not need (year, features, id, views). Furthermore, we picked a small population of data from the bigger one we had on our hands and added more data that came from the API (duration, track's number, popularity, release year, track qualification, album's number of tracks and album's name).

The join was made based on the title of the music and the name of the artist. After that the process consists on plotting

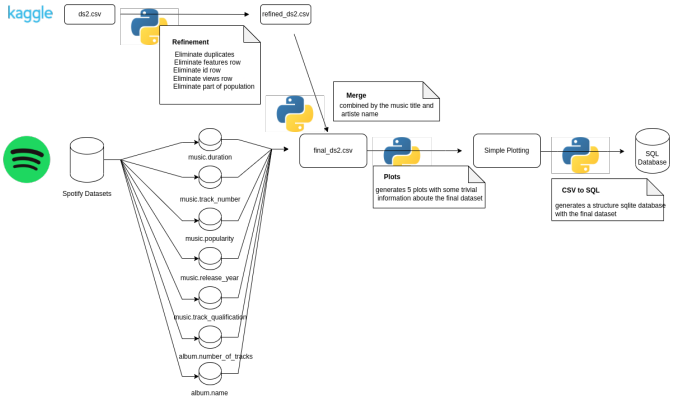

Figure 1: Pipeline Scheme

some graphs and turning the final version of the dataset on a structured SQL database system.

3.2 Data characterization

3.3 Domain Conceptual Model

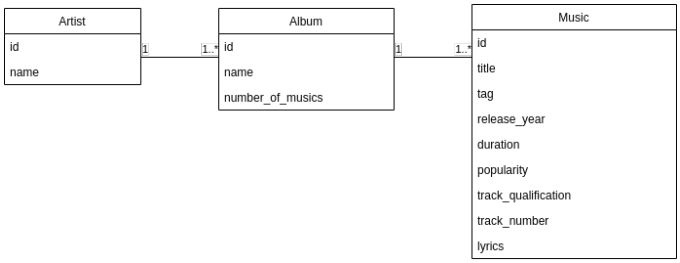

Figure 2: Domain Conceptual Model

The Database consists on three tables. The table Artist that stores the name of every artist. The table Album that stores the general information about the albums. And a Music table, where all the information about the musics is stored, that's the main class of our database. With this in mind, and using the final version of dataset and another script of Python. we created our SQL database.