University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Cybersecurity

User Privacy on Spotify: Predicting Personal Data from Music Preferences

Supervisor
Prof. Mauro Conti
University of Padova

Co-supervisor
Pier Paolo Tricomi
University of Padova

Master Candidate
Jiancheng Ye

Student ID
2016822

Academic Year
2021-2022

"A success doesn't always mean something big, every little thing counts."

Abstract

The way we listen to music has changed drastically in the past decade. Now we can play any kind of music from various artists around the world through our smart devices. Many music streaming providers, if not most, are built with systems to track users' music preferences and suggest new content.

The music we listen to reveals a great deal about who we are. In general, people share their playlists and songs of their favorite artists on the music platform; find people with common music genres and connect with them. It is not always easy to make friends with unknown people, but music is a good way to accomplish that. In spite of that, we must also look at other sides of the coin from a security perspective. Is it a good idea to share music interests with others or will it compromise our privacy? According to privacy experts and developers, there is no purposeless data. Everything can be used to infer private information, even a single like on social media, which seems, at first sight, meaningless, but it can reveal more information than it promises. In the case that our musical tastes reveal our information, we may be profiled for targeted advertisement, by surveillance agencies, or in general, become potential victims of malicious activities Since music is part of our daily lives, and there are many providers that let us listen to music, we are even more at risk of being profiled and having our data sold.

In this research, we demonstrate the feasibility of inferring personal data based on playlists and songs people publicly shared on Spotify. Through an online survey, we collected a new dataset containing the private information of 750 Spotify users and we downloaded around 402,999 songs extracted from a total of 8777 playlists. Our statistical analysis shows significant correlations between users' music preferences (e.g., music genre) and private information (e.g., age, gender, economic status).

As a consequence of significant correlations, we built several machine-learning models to infer private information and our results demonstrated that such inference is possible, posing a real privacy threat to all music listeners. In particular, we accurately predicted the gender (71.7% f1-score), and several other private attributes, such as whether a person drinks (62.8% f1-score) or smokes (60.2% f1-score) regularly.

The purpose of this project is to raise awareness about how seemingly purposeless data can reveal personal information and educate users about how to better protect their privacy.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AIA . . . . . Attribute Inference Attacks

SVM . . . . . Support Vector Machine

1 Introduction

Music has played certainly an important role in human history. Since prehistoric times, cavemen used to play instruments and music to express their sensations, emotions, and feelings. Thousand years later, it still influences our modern society and throughout the last half-century, digitalization has considerably increased the amount of new music produced and available directly to consumers. Nowadays, online services facilitate many of our daily activities, from social interaction to content consumption in the majority of case. Streaming services provide users with the ability to download an application for free, which allows them to access the full service, rather than the main website, and they become increasingly popular as a result of the availability of smartphones with internet connection on a regular basis. Presently, the most prominent and famous streaming services are the following: Spotify, Apple Music, and Tidal [1]. Apparently, Spotify is the most popular one and this may be due to the fact that it offers a freemium service alongside a paid service. Today, Spotify is currently available in 180+ countries with over 422 million monthly active users, including 182 million paying subscribers [2]. Spotify was founded in Sweden in 2006 and was officially launched as a music streaming platform in 2008. It has expanded quickly across the world, offering users the opportunity to listen to the music of their favorite artists without needing to own the music themselves. The platform offers also users who are based in European countries a paid subscription service of €10 a month [3] and it consists of the benefits of no advertisements between songs, unlimited skipping through the tracks, and the ability to use the app on their mobile devices also offline [3].

Contemporary research on musical preferences has applied interactionist theories to music positing that people select musical genres that reflect their psychological traits and needs. An important study conducted by Michal Kosinski et al. [4] shows private traits and attributes are predictable from digital records of human behavior and this leads us to what this research is about: how easily accessible digital records of behavior (e.g. playlists/song) can be used automatically and maliciously.

According to privacy experts and developers, there is no purpose-less data, everything can be used to infer, even a like or comment, which seems, at first sight, meaningless, but it can reveal more information than it promises. In light of this, the question arises: Can users' musical preferences, which seem innocuous, reveal private information?

In this work, we demonstrate the possibility of inferring users' private attributes by exploiting their music preferences, publicly retrievable from streaming services websites. More formally, we set up an Attribute Inference Attack on Spotify from two points of view: first, using the public playlists created by users and then, using public single songs since we want also to find out the impact of singularity over the group.

We used Spotify as a case study to demonstrate the applicability of the attribute inference attack using music data, in fact, with 422 million monthly listeners, exposing such a vulnerability would be detrimental to the privacy of millions of people.

We risk being seen as transparent and being profiled by data brokers who want to make a profit via selling the user attribute information to other parties such as advertisers, banking companies, and insurance industries [5]. Moreover, surveillance agencies can use the attributes to identify users and monitor their activities or even worse case, for minors to be tracked by attackers as victims of malicious activity like cyberbullying [6] and cyberharassment.

In the attack, an attacker collects many users' private information (the ground truth), through social networks or deceiving surveys, to train a classifier that, given in input the public data of a victim, returns their private attributes. We focus on 11 private attributes, such as gender, age, occupation, economic status, or personality traits. The attacker's motivations are multiple. For instance, it could search for particular categories of victims (e.g., kids, rich people,) to perform targeted malicious activities such as stalking or harassment, phenomena that unfortunately are very much present nowadays.

To conduct our experiments, we collected a new dataset containing the private information of 750 Spotify users from 80 different countries aged between 12 and 55 which have at least one playlist in their Spotify public profile and we considered around e 402,999 songs extracted from a total of 8777 playlists. Our results show significant correlations between users' music

preferences and private information, and accordingly, the possibility to infer such private features from the classifiers we built. The success of this attack highlights a crucial privacy threat, not just for Spotify users, but for millions of users worldwide also on other streaming services websites.

Contribution The contribution of this thesis can be summarized as follows:

1. A threat model of Attribute Inference Attack against Spotify users. We describe how to (legitimately) launch an Attribute Inference Attack to infer private information on users while knowing only their Spotify id.
2. we perform an in-depth correlation analysis between Spotify users' music data and different private information, to assess the feasibility of our attack;
3. we conduct our attack on Spotify, by testing several classifiers to infer the private information demonstrating the impact of our attack;
4. we provide a detailed analysis of music streaming services and the applicability of our attack to other streaming music services.

Structure

The rest of the thesis is organized as follows. Chapter 2 presents the related works. Chapter 3 analyzes the music streaming service panorama and gives background information useful to understand the following detailed comments on the correlations found. Chapter 4 presents the threat model and overview of our attack, while the dataset used in the experiments is described in Chapter 5. Correlation analysis and prediction are conducted in Chapters 6 and 7, respectively. Discussions are reported in Chapter 8, and Chapter 9 concludes the thesis.

2
Related works

In this chapter, we focus on related works mainly on attribute inference attacks and privacy in music streaming services.

2.1 Attribute Inference Attacks

An attribute inference attack occurs when an attacker has access to a set of *public data* of a target user and uses them to infer *private attributes* of such target. The public data could be anything present on the web starting from photo [7, 8], posts, or even public ratings or emoji posted on a social platform by a given user can be used to infer their private information such as gender [9, 10, 11].

Most of the time. the attacker first trains a machine learning model, which takes in input the user's public data and outputs their private information. To train such classifiers, the attacker needs first to construct a dataset that contains both public data and private attributes of individuals who have made their private characteristics public. Attribute Inference Attacks are becoming problematic due to the lack of education of most internet users, who publicly share their data while overlooking (or ignoring) the corresponding risks. In general, many of these attacks have been conducted on social networks, given the ease of retrieving public data on these platforms. Nowadays, online social networks facilitate many third-party applications that offer users additional functionality and services. However, they also pose a serious user

privacy risk [12]. Most data published on social networks can be easily retrieved via OSINT [13] and then used to setup an Attribute Inference Attack by developing an ML model.

Indeed, most prior research considers the ecosystem of social networks, due to the ease of retrieving information linking public data with private attributes: Goelbeck et al. [14] infer personality traits of social media users. Jurgens et al. [15] consider Twitter, and predict the location of the users based on their tweets.

Gong and Liu [16] successfully inferred attributes like major, employer, and cities lived from Google+ users by leveraging both their social connections and behavior.

Chaabane et al. [17] exploited users' interests (e.g., music interests) shared on Facebook to predict their private attributes such as ageand gender. Kosinski et al. [4] were able to predict sexual orientation, religious and political views of Facebook users by using their networks, other popular attacks exploit data coming from mobile devices to retrieve information such as the geographical position. Other examples are [18, 19]. All such works show that Attribute Inference Attacks can be enacted in the real world, representing a subtle privacy risk.

To the best of our knowledge, no attacks exploited public Spotify data to infer the private information of users.

2.2 Music and Private Data

Although we did not find too much evidence of Attribute Inference Attacks in Spotify, there are few previous works that analyzed the correlations between music data and users' attributes on other music streaming service platforms.

Most closely connected to our study on predicting listener attributes is the work by Liu et al. [20] using Last.fm data. They determined the gender of Last.fm users based on listening history and in addition, the age is evaluated as being below or beyond 24 years old. The features for the classification are constructed purely from the listening events of the user. They are based on three factors: the listening timestamps, the meta-data of the song, and the artist (e.g., artist and song tags), as well as signal features of the songs. For both tasks, a support vector machine classifier (SVM) with RBF kernel is used and the average of five runs with 80% of the users as a training set is reported. The accuracy for age is 71.1%; the accuracy for gender is 66.1%.

Moreover, in 2017, Thomas Krismayer1 et al. [21] investigated the prediction of personal information of users, such as age, gender, and nationality using the API of the social music platform Last.FM and the set that they used was from the public dataset provided by M.Schedl [22].

Instead, in our work, we focus on Spotify and leverage songs and playlists as our attack vector, which the user publicly releases on the platform. From Spotify, we collected directly through API the information regarding their music preference and we extracted features based on it.

3

Panorama overview

This chapter serves as a background to fully understand the rest of this work since the utilized attributes are strictly related to the music streaming service. We analyze popular music streaming services under multiple general aspects, explain which are the main vectors of our inference attack, and conclude by giving details of Spotify, the main one we chose for our experiment because it is more widely used and can afflict more people.

3.1 Popular Music streaming service

The most common way for people to listen to music now is through streaming services. Customers have unrestricted access to enormous music libraries. These services store the music in a server that users can connect to via their laptops and mobile devices.

In Table 3.1, we aggregated statistics from the four most significant and well-known streaming services [1], stating their release years, monthly active users, subscriber paying users, website, and the accessibility of general user information about songs and playlists, as well as the ability to retrieve them. We will provide a brief overview of each of them in the following sections.

3.1.1 Apple Music

Over the past year, Apple Music has emerged as one of Spotify's top rivals. But without the framework that iTunes offered, this would not have been possible. iTunes was created as a

music platform that allowed users of Apple Mac products to save and play their CDs on their computers. In response to conflicts in the music industry over illegal song downloads, iTunes introduced a music store. iTunes was essential in the proliferation of digital music, but it wasn't until May 2014 that Apple launched its first foray into music streaming with the acquisition of Beats Electronics [23]. This deal eventually led to the formation of the Apple Music streaming service. It aimed to offer the iTunes catalog in its entirety to users for a monthly subscription.

3.1.2 Tidal

An official relaunch of Tidal was held in March 2015, when Jay Z introduced the world to its new owners at a large press conference. The new owners include some of the biggest names in the music industry, including Beyoncé, Madonna, Rihanna, and Coldplay [24].

Although it seemed to the public that this was a new music streaming service, Tidal had already been in existence for many years under a different name. Tidal was founded by Norway's 'largest record-store chain, Platekompaniet [25], as WiMP originally called it. The WiMP streaming service offers a paid subscription service, [25], in order to provide users with a higher quality of music. As it expanded to the UK and the US in 2015, Jay Z purchased it for 56 million dollars and renamed it Tidal.

3.1.3 Youtube music

In 2015, Youtube started as a competitor to Tidal and other music streaming services. In 2018, YouTube Music was launched and it was a relatively new music streaming service that replaced Google Play Music, Google's previous music streaming service. Just like others, YouTube Music has a huge catalog of songs (over 80 Million). It is also a subscription-based service with a free tier, and monthly and yearly packages. It can be used as an app on iOS and Android, but also through the YouTube Music website. It also allows, like Google Play Music did, the upload of your own music collection (see below). There is a free tier for YouTube Music and there is YouTube Music Premium.

3.1.4 Spotify

Spotify was founded in Sweden in 2006 and was officially launched as a music streaming platform in 2008. It has expanded quickly across the world, offering users the opportunity to listen to the music of their favorite artists without needing to own the music themselves. Today, Spo-

tify is currently available in 180+ countries with over 422 million monthly active users, including 182 million paying subscribers [2]. Apparently, Spotify is the most popular one and this may be due to the fact that it offers a freemium service alongside a paid service. The platform offers also users who are based in European countries a paid subscription service of 10 euro [3] a month and it consists of the benefits of no advertisements between songs, unlimited skipping through the tracks, and the ability to use the app on their mobile devices is also offline.

Spotify playlist and song

Spotify is popular with users due to its user-friendly app and online website service which are available across many devices. In general, a user profile is public, and it is easy to understand its music tastes based on artists, albums, genres, and shared playlists. There are 2 types of playlists: the 'discover weekly' playlist, made up of songs and artists that Spotify thinks the user might like with genres and artists they listen to while using the service; and 'custom' playlists that users manually create with their favorite songs and artists and share publicly with their friends.

Table 3.1: Music streaming service panorama table

	Release Year	*Monthly Users*	*Paying Users*	*Public playlist Visible*	*Public S song Visible*	*Retrievable Via API*	*Website*
Spotify	2011	422 M	195 M	yes	yes	yes	open.spotify.com
Apple music	2015	88 M	88 M	yes	yes	yes	apple.com/it/apple-music
Tidal	2015	5 M	5 M	yes	yes	yes	tidal.com
Youtube Music	2015	30 M	30 M	yes	yes	yes	music.youtube.com

4

Attack Design

We now introduce the notation used for the following descriptions of the threat model [26] and the overview of our attack.

4.1 Notation

For simplicity, we express our notation within the scope of playlist. However, the attack can be orchestrated for single song.

- $\mathbb{U}$ is the set of all Users ;
- $p_{i,u}$ is a unique playlist with ID = i and belong by the user $u \in \mathbb{U}$;
- $\mathbb{P}_u$ is the set of all playlist p_i belong to the user u;
- $\mathbf{y}_u$ is a vector of j private information $[y_1, y_2, \ldots, y_j]$ of a user u.
- $Ms_p(u)$ is a generic function of the music streaming service that given a user u, returns their generic playlist data P_u;

4.2 Threat Model

In our scenario, an attacker wants to infer private attributes $\mathbf{y}_u$ of a user u by exploiting their playlist data $\mathbb{P}_u$. The playlists of the users are public and they can be viewed on music streaming service, which publicly releases users' music preferences. The attacker leverages on the function Ms_p of the music streaming to retrieve $\mathbb{P}_u$. We now give more details about the three main participants: the music streaming service, the victim user g, and the attacker.

music streaming service. The music streaming service is the platform on which we perform our attack. This definition fits well including any music streaming present today (e.g., Spotify, LastFM, Apple music) that generates online data. In our attack, we retrieve playlist data P_u of a target user u, if the target has at least one public playlist visible in the profile. Playlist music data P_u can be fetched by Ms_p (i.e., APIs or similar) at any time. We consider this party as trusted, which records and shows playlist music data P_u $\forall u \in \mathbb{U}$ as a main input element in the attack. We assume the collected data is accurate and publicly available to any person. The music streaming service offers mainly the following function: $Ms_p(u)$, is a generic function e that given a user u, returns their generic playlist Data $\mathbb{P}_u$. A casual person can get these data through APIs or by scraping the website.

Victim user. The victim in our scenario is any user u that use a music streaming service and is monitored (voluntarily or involuntarily) by a Tracking Website . u has some private attributes $\mathbf{y}_u$ that does not want to share (e.g., gender, age, personality) with others. Besides that, u is anonymous, thus no name, IP address, or other PII (Personally Identifiable Information) are retrievable. The only identifier available is the Player ID = u, which is used to retrieve their playlist music data through a music streaming service.

Attacker. The attacker could be any entity who has interests in users' attributes, e.g., a cyber criminal, a data broker, an advertiser, or surveillance agency. Cybercriminals can leverage private attributes $\mathbf{y}$ to find their victims for a malicious activity like cyberbullying [27]. Alternatively, a data broker make profit by selling the private data to third parties, such as advertisers, which can use the data for targeted advertisements [28].

4.3 Attack Overview

Figure 4.1 shows our attack overview. The attack runs in two phases. In brief, in *Phase I* the attacker trains a machine learning classifier to infer the private attributes of a victim user giving in input their music preferences. In *Phase II*, the attacker uses the classifier trained in *Phase I* to actually infer the private attributes of one or more user. We now present in details the two phases.

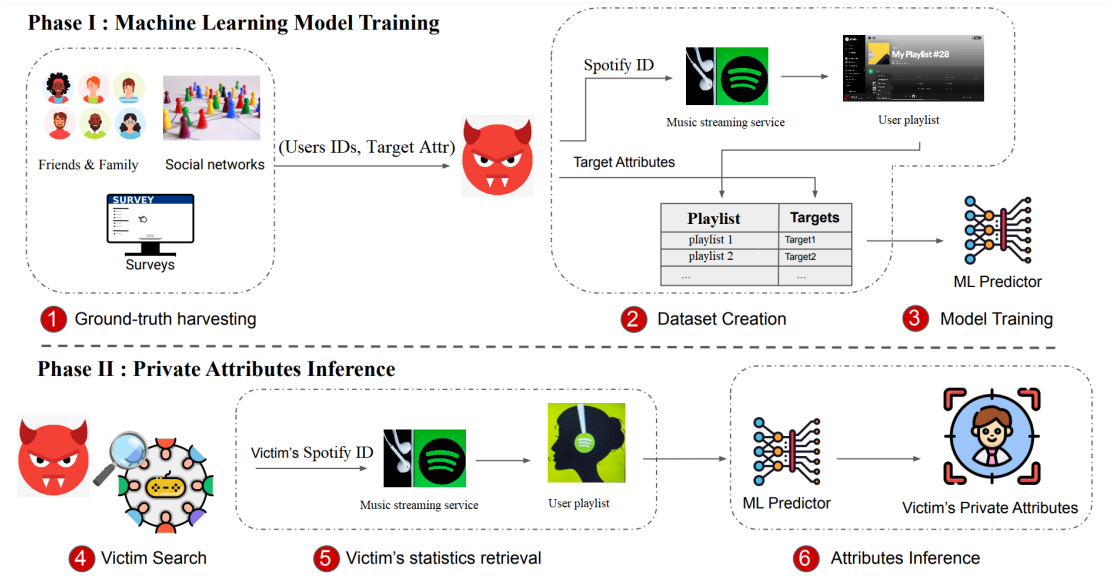


Figure 4.1: Overview of our two-phase attribute inference attack.

4.3.1 Phase I: Machine Learning Model Training

In this phase, the attacker needs to train a machine learning model $C(P_u)$ that given in input music playlists P_u of many users u, returns in output their private attributes $\mathbf{y}_u$. With this aim, the attacker first needs to collect the ground-truth for such tasks, i.e., the private information of a lot of users along their Spotify ID. More formally, the attacker needs many tuples $(u, \mathbf{y}_u)$ as a first step to create a dataset and therefore train $C(P_u)$. These tuples can be obtained in several ways. For instance, the attacker can join social networks groups in which people discuss and promote their music . Within the group, people necessarily use their Spotify ID as identifier. Thus, the attacker can easily get u and retrieve the private attributes $\mathbf{y}_u$ by looking at their social network profile. Alternatively, the attacker can use the private information of friends

and family or conduct a survey (as we did) to ask people their information and their Spotify accounts.

Once the attacker gets the tuples $(u, \mathbf{y}_u)$, all the $\mathbf{y}_u$ go directly into the dataset (after a proper preprocessing) as dependent variables, while the u (User ID) are used to get the playlist data p_u (independent variables). The attacker retrieves p_u by the music streaming function $Ms_p(u)$, and populates the dataset. Last, the dataset is used to train $C(P_u)$, i.e., any state of the art machine learning model (e.g., decision tree, random forest), and the one yielding the best results will be used in *Phase II*

4.3.2 Phase II: Private Attributes Inference

In this phase, the attacker wants to identify a *victim u*'s private attributes $\mathbf{y}_{victim\ u}$. Depending on the attacker's goal, the victim can be found in a variety of ways. The attacker retrieves $p_{victim\ u}$ through the music streaming function $Ms(victim\ u)$, feeds the their playlist to the pre-trained classifier $C(P_u)$, and obtains $\mathbf{y}_{victim\ u}$. Alternatively, the attacker could conduct a large-scale attack in which takes all the users interested, downloads all their playlist, and infers all their private information for selling purposes.

5 Dataset

5.1 Data Collection Procedure

The data collection to create the dataset was performed in two steps. First, we conducted an online survey to find participants and get their personal information, i.e., the ground-truth to test our inference attack. Then, we collected their playlists and songs data released by music streaming services to carry out the attack.

5.1.1 The survey

We found participants for our research using an online survey. The survey was partial anonymous, in the sense we ask for their Spotify profile as ID and in the general, users use a nickname but there are also users who use Facebook accounts and fulfilling it users gave us the consensus to use their data (from the survey answers) related to their Spotify profile. The purpose of the survey was to retrieve the Spotify ID to download users' playlists and songs and we started collecting data from 15/5/2022 and ended 1/9/2022 spreading the survey in different places, i.e., Facebook groups, Reddit, Discord groups, and private messages on different platforms. The estimated time to complete the survey was 4 minutes. We know that the conducted survey cannot be a complete representation of the entire Spotify user community, but we do believe it was fairly enough for a qualitative assessment.

The survey was divided into three sections. In the first one, we asked for general information,

e.g., age, gender, and nationality, for demographic purposes. In the second section, we asked for information about Spotify use experience in general. Finally, ten personality questions[29] were presented in the last part to try to understand how different Spotify users are from each other and as ground truth for the inference phase. We spread the survey in different places, i.e., Facebook groups, Reddit, Discord groups, and private messages on different platforms. The estimated time to complete the survey was 4 minutes. We used best practices developed during the years to protect people's privacy and to filter out invalid answers.

5.1.2 Survey Result

We received a total of 1081 answers. 210 answers were considered invalid, while 121 participants were not visible by API Spotify, because they don't share any public playlist. So, we start with a total of 792 active users from different countries for the research.
541 (68.3%) of them were males, while 221 (27.9%) were females and 30 were non-binary (3.8%). Most of the users were students (46.1%), followed by workers (31.6%), working students (15.8%), and a small fraction of unemployed (6.5%).
The age ranges from 13 to 55, with the majority between 14 and 36. Most of the users are from the United States (26.9%), followed by users in Italy (9.4%), the United Kingdom (7.3%), Canada (5.6%), Germany (5%), and various users from different 71 countries (45.8%). About 40.3% of them use Spotify for 2-3 hours per day, followed by users who use it for 4-5 hours (21.6%), less than 1 hour (20.4%) and over 6 hours (17.7%). Furthermore, 76.9% of them reported having Spotify premium. We also asked about their experience with smoking and drinking and doing sport. Most people don't smoke (78,8%), followed by a few cigarettes per week and on rare occasions (14%) and smoking every day (7.2%). Most people don't drink (45.6%), followed by 1-2 times per week (38.8%), Sometimes 3-4 times per week (11.7%), and Very Often 5+ times per week (3.9%). Most people practice sports occasionally (35.6%), followed by amateurs (31.6%), then competitive users (2.8%), and those who don't practice (30%).

5.1.3 Spotify Data

After we collected all the user's Spotify ID from validated answers, we downloaded all the playlists of such users. We used the official Spotify API from Github[30] to retrieve the JSON file of each playlist. Each JSON file represents a playlist and it contains all information divided into sections. The two main important sections are the following: the owner section and the track section. In the Owner section, we can find information regarding the user eg: Spotify

ID, or nickname. In the tracks section, we can detect all songs added by users over the years. An example is reported in figure 5.1. The format of each track in the JSON file is made by the following :

- Album: this section contains information regarding the song such as id Spotify, song title, available market Spotify market in which we can find it, release date, links of the cover image, etc.
- Artist: this section contains information about the artists of the song, such as their unique Spotify ID, names, and links to their Spotify pages.
- Id: indicates the unique Spotify id in the platform
- Time: indicates the moment when the user added the song in the playlist;
- Duration: indicates the time duration of the song expressed in milliseconds
- Popularity: a number between 0-100 which indicates the popularity of the song in the Spotify market

5.2 Considered Features

Data analysis was dealt with at two levels, one considering the user's information given music preference taking into account a single song (song level), and one taking into account users' playlist (playlist level). In order to distinguish between these cases, two distinct datasets were employed. The playlist dataset was created with data gathered directly from the Spotify platform and the song dataset was created by disaggregating the first one.

5.2.1 Playlist Level Features

In the following, the description of the features utilized in each of these datasets is discussed, together with a render of the private target features defined as our objective.

- **Content features**

  In the first step, we collected the following **features relative to the playlist and its content**:

  - **id playlist**, a unique Spotify string that identifies the playlist in the platform;
  - **name playlist**, a custom name given by the owner attributed to the playlist;

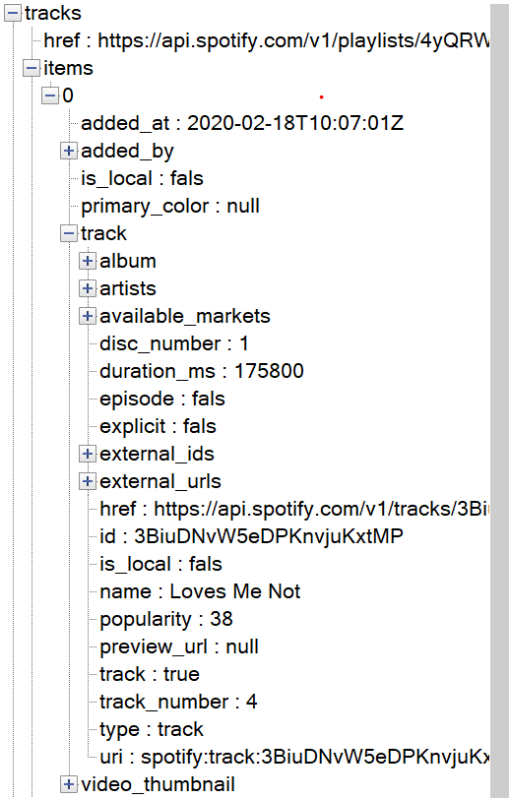


Figure 5.1: Overview of the track section in the Json file

- **id owner**, a unique Spotify string that identifies the owner who created the playlist;
- **name owner**, a custom name assigned to the id by the owner;
- **title language**, indicates in which language is written the name of the playlist;
- **prevalent song language**, indicates the language which is used prevalently in the playlist;
- **is collaborative**, a true/false tag that indicates if the playlist is created in collaboration with other users;
- **average years publication and std**, mean and standard deviation of all the publication years of songs in the playlist;
- **number of songs**, indicates how many songs there are in the playlist;
- **followers**, indicates how many users follow the playlist;
- **number of song solo** , indicates how many songs are performed only by 1 artist;
- **number of song collab** , indicates how many songs is performed by 2 or more artists together(groups);

- **Simpson index**, a measure of diversity which takes into account the number of artist present and it ranges between 0 (no diversity) and 1 (maximum diversity).
- **compare multiple time** , a true/false tag that indicates if exist some artists compare multiple times in the playlist,(which means in the playlist, there are songs that belong to the same artist) ;
- **max time compare** , indicates the max number of some artist that compares in the playlist;
- **ratio male artist** , indicates the percentage of the male artist in the playlist;
- **ratio female artist** , indicates the percentage of the female artist in the playlist;

- **Audio statistical features**

  In the second step, we collected **features relative to the audio statistical** given by directly by Spotify API:

  - **danceability**, describes the suitability of a track for dancing in terms of its tempo and rhythm stability. A value of 0.0 is least danceable and 1.0 is most danceable;
  - **acousticness**,a value from 0.0 to 1.0 of whether the track is acoustic;
  - **energy**, indicates a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy;
  - **instrumentalness**, indicates whether a track contains no vocals;
  - **liveness**, detects the presence of an audience in the recording. Higher liveness values mean a high probability that the track was performed live;
  - **loudness**, indicates the overall loudness of a track in decibels (dB). Values typically range between -60 and 0 db;
  - **speechiness**, which detects the presence of spoken words in a trace. A high value means a high probability that the track was speech-like the recording (e.g. talk show, audiobook, poetry);
  - **tempo**, which indicates the overall estimated tempo of a track in beats per minute (BPM);
  - **valence**, which describes 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).

5.2.2 Song Level Features

We created the song dataset by disaggregating songs from the playlist that was used for the dataset in section 5.3.1. The procedure is very similar, we extracted features from the content level and then from the audio statistical level, but this time considering the singularity of the song. One particularity is for genres, it was expanded in a sort of one-hot encoding fashion. To better explain, a column was created for each possible value that a song can belong it.

5.2.3 Target Features

For each user included in our datasets, was also collected data relative to the eleven target features to study, which are:

- *Gender*, intended as birth gender, either male or female;
- *Age*, split into 3 bins to represent meaningful stages of life for the majority of our population, that are teenagers (0-18), young adults (19-30), and over 30 (30-55).
- *Marital*, indicates if the user is single or in a relationship;
- *Occupation*, either if the player has a job or not;
- *Economic*, user's state of economic well-being, expressed through a number between 1 (lowest) and 3 (highest);
- *Live_with*, either alone or with others;
- *Sport*, indicate the frequency and the level of user in sports activity, reported in the growing scale no, sometimes, amateur, competitive;
- *Drink*, indicate the user drink or not alcohol, reported in the growing scale no, sometimes, often, very often;
- *Smoke*, indicates the user smoke or not cigarettes. reported in the growing scale no, sometimes, often, very often;
- *Continent*, meant as Continent of origin, therefore there are 5 contemplated options;
- *Personality*, classified in function of the Big Five [31] personality traits, i.e., extroversion, agreeableness, conscientiousness, neuroticism, and openness. Each one is expressed through discrete values in a scale from 0 to 100 with a step of 10. For analysis' sake, we grouped such values into 3 categories to represent low (0-25), mid (37.5-62.5), or high (75-100) values of the trait.

Among all of these features, we individuated the five main ones as *gender*, *age*, *economic*, *occupation* and *personality*, since they are characteristic traits that tend to better identify a person. In this work, the study of correlations and predictions will be accomplished for all of the targets mentioned above, but the focus will remain on the most important ones.

5.3 Final dataset

We created 2 datasets: a playlist dataset and song dataset in order to perform our inference attribute attack from two points of view.

5.3.1 Playlist dataset

The initial playlist dataset was created by taking all 1081 users that participated in our survey for the project. The second step involves removing those who don't pass the survey's attention test, which resulted in 210 users being removed. Furthermore, 121 users were removed because they had no playlists. we ended up with 750 users which have at least one playlist in their Spotify profile and we start to collect all of them in the initial user-playlist dataset.

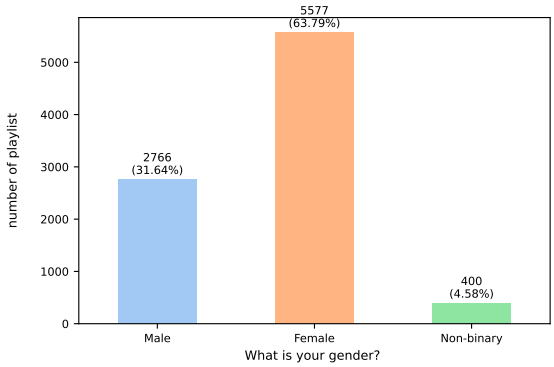
As result, we collected 10217 playlists and we had to remove 1440 ones for various reasons (empty playlists, playlists with 1 song created by mistake, or playlists that are not created by the user). Finally, we end up with 8777 records, where the entry of the database corresponds to the playlist that belongs to a user that we know in prior his private info and it is identified by the pair *(playlist_id)*. In the end, the shape of the obtained dataset is 8777×87. Besides the 15 columns needed to store target features (*personality* is actually expressed through five values, therefore we have one column each).

5.3.2 Song dataset

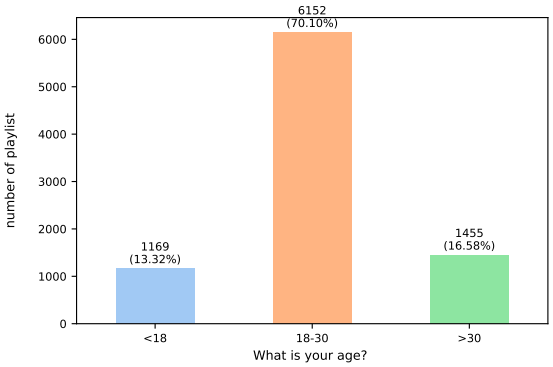
This dataset was created by the disaggregating song from the playlist that we used for the dataset in5.3.1. We extracted each song from the 8777 playlists and then determined the feature vector for each of them and then stored in the final song dataset. In this case, the entries of the dataset are solely identified by the *song_id*, therefore the size of the dataset results to be 402999×65. The number of columns required to store target features is the same as in the playlist-level case. At the player level, we calculated the percentage of songs in each musical genre in the playlist, but at the song level we cannot do that, for this reason, we compute one-hot encoding for the genre that the song belongs to.

5.4 Preliminary Analysis

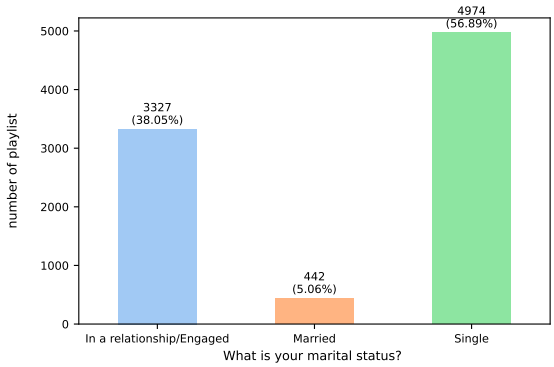
Before proceeding with the research of correlations between the target features and attributes in the different datasets, it is appropriate to perform a study on the user sample collected. The main reason is that we should generally ensure a fairly distributed proportion of the whole population and be aware of possible biases already present in the study data. In the following, we present graphs to visually depict the distributions of each one of the targets presented in Section 5.2.3. For each bar of the barplot are reported both the count and the percentage of the population belonging to such group, at playlist level, and at song level in the following figures respectively.



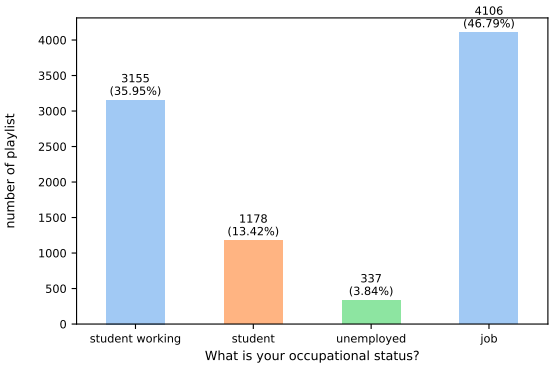
(a) Distribution of *gender* at playlist level



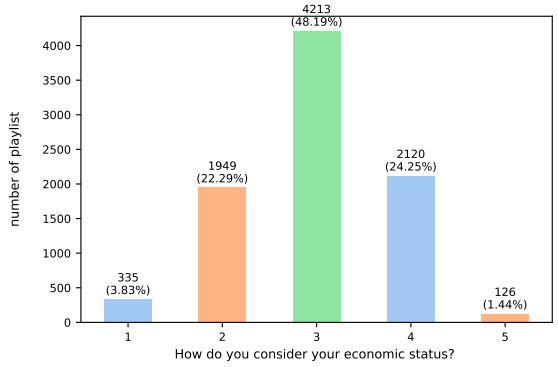
(b) Distribution of *age* at playlist level



(c) Distribution of *marital* at playlist level

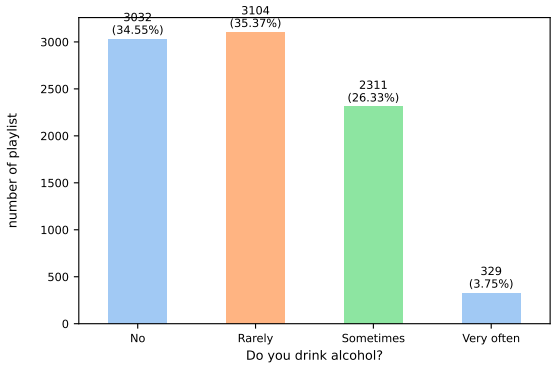


(d) Distribution of *occupation* at playlist level

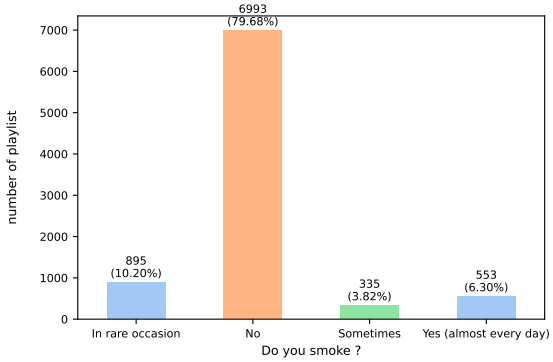


(e) Distribution of *economic* at playlist level

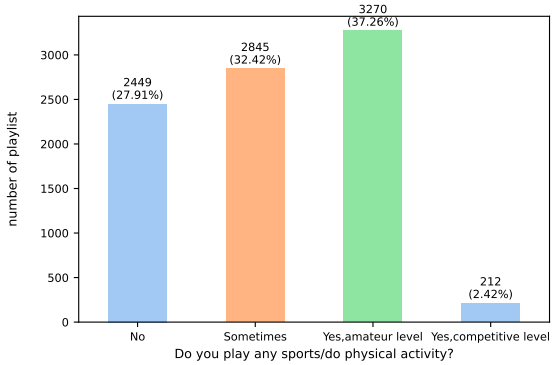
Figure 5.2: Distributions at playlist level of target features explained in Section 5.2.3



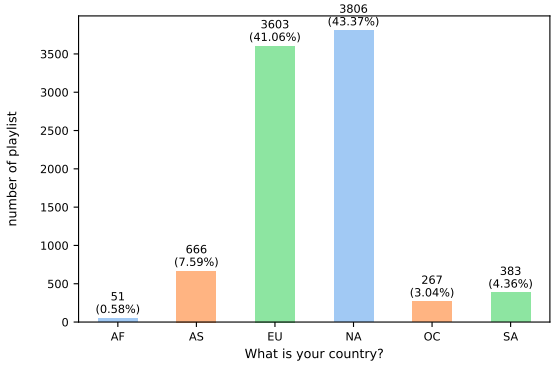
(f) Distribution of *drink* at playlist level



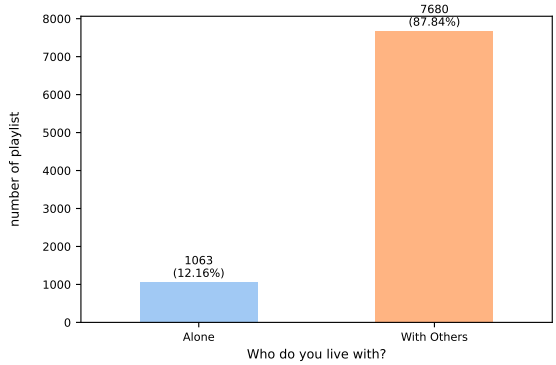
(g) Distribution of *smoke* at playlist level



(h) Distribution of *sport* at playlist level

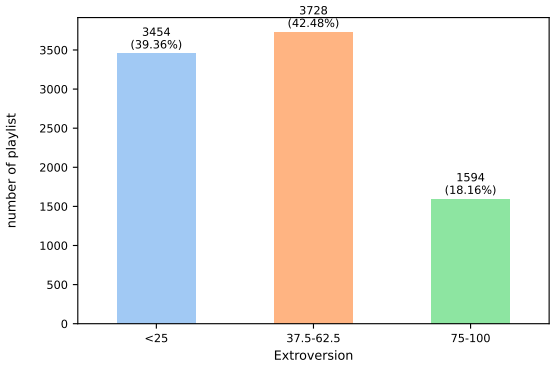


(i) Distribution of *country* at playlist level

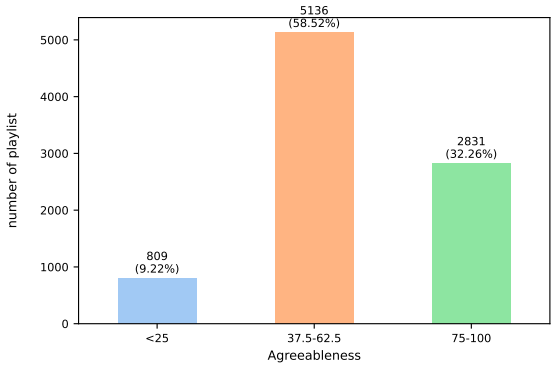


(j) Distribution of *livewith* at playlist level

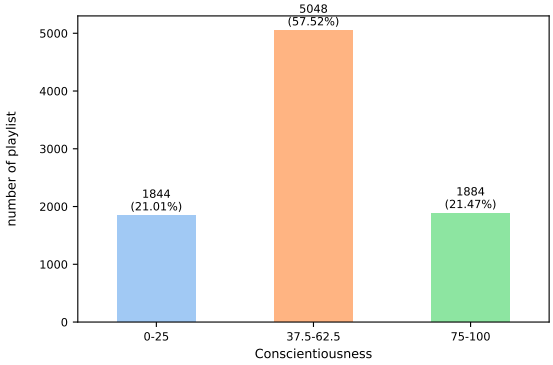
Figure 5.2: Distributions at playlist level of target features explained in Section 5.2.3 (cont.)



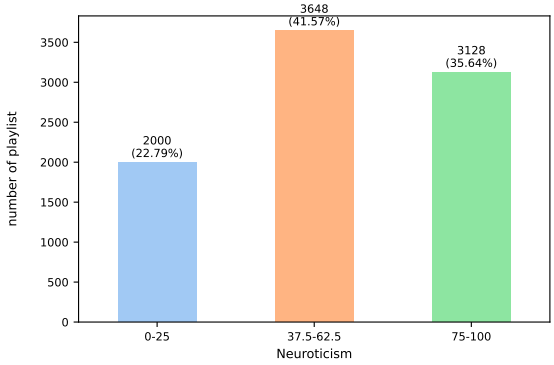
(k) Distribution of *extroversion* at playlist level



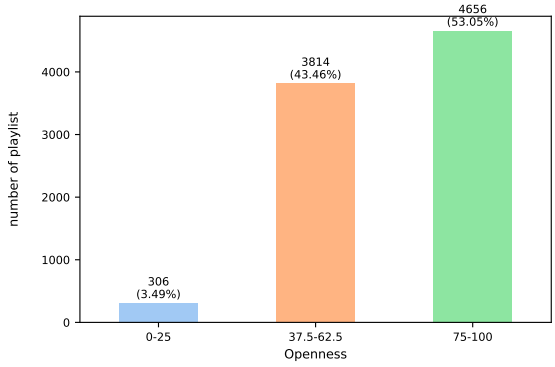
(l) Distribution of *agreeableness* at playlist level



(m) Distribution of *conscientiousness* at playlist level

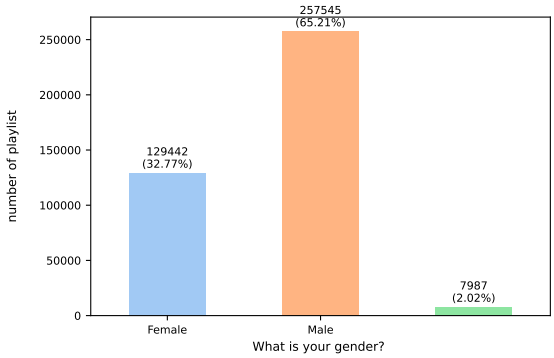


(n) Distribution of *neuroticism* at playlist level

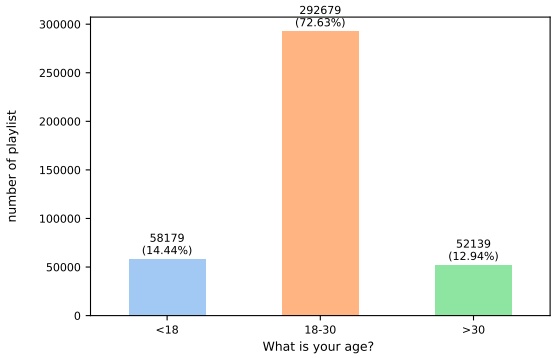


(o) Distribution of *openness* at playlist level

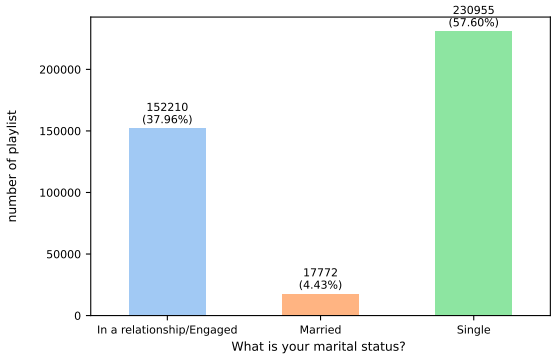
Figure 5.2: Distributions at playlist level of target features explained in Section 5.2.3 (cont.)



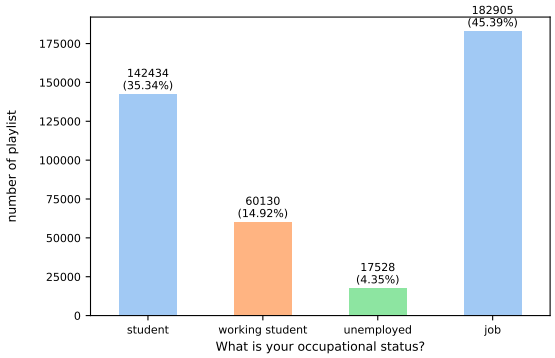
(a) Distribution of *gender* at song level



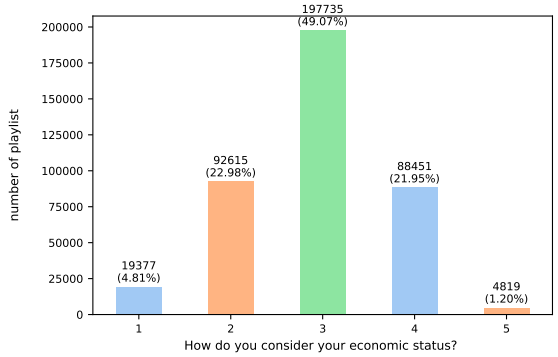
(b) Distribution of *age* at song level



(c) Distribution of *marital* at song level

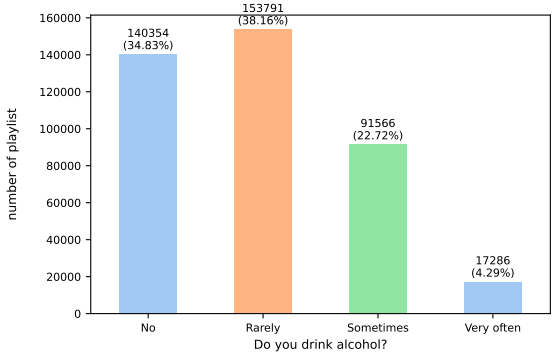


(d) Distribution of *occupation* at song level

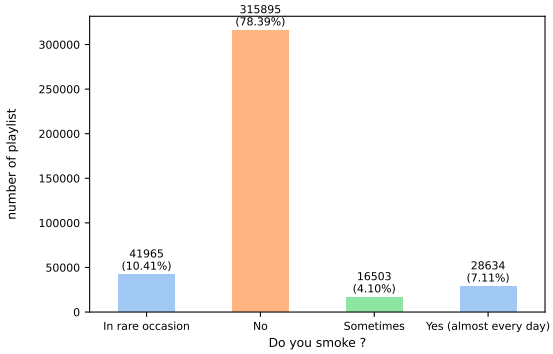


(e) Distribution of *economic* at song level

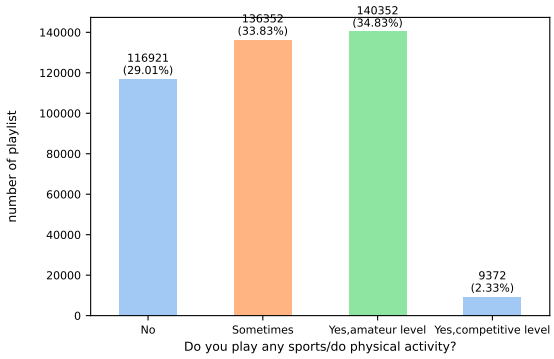
Figure 5.3: Distributions at song level of target features explained in Section 5.2.3



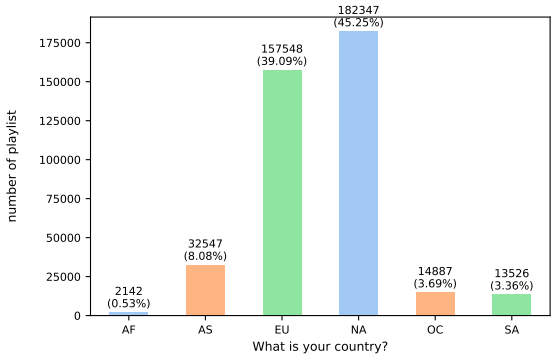
(f) Distribution of *drink* at song level



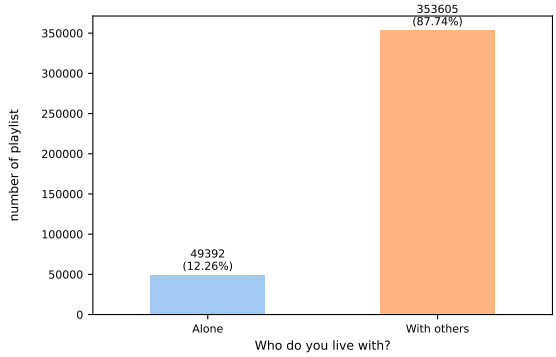
(g) Distribution of *smoke* at song level



(h) Distribution of *sport* at song level

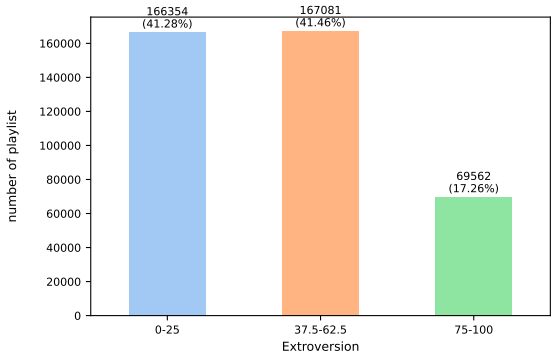


(i) Distribution of *country* at song level

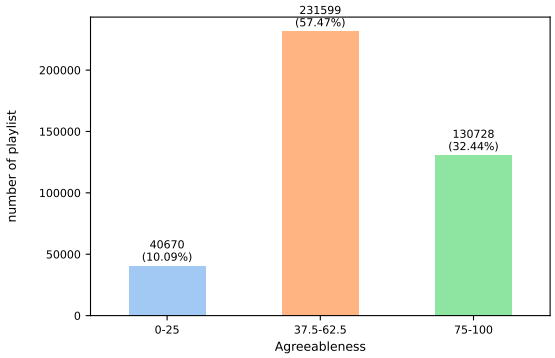


(j) Distribution of *livewith* at song level

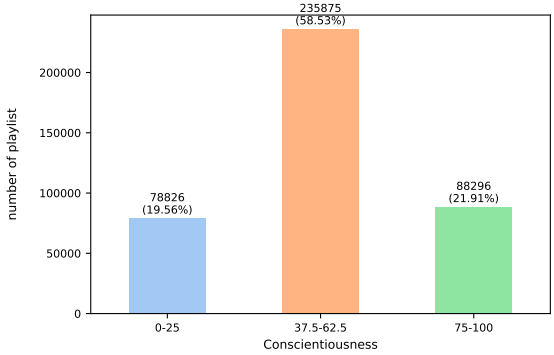
Figure 5.3: Distributions at song level of target features explained in Section 5.2.3 (cont.)



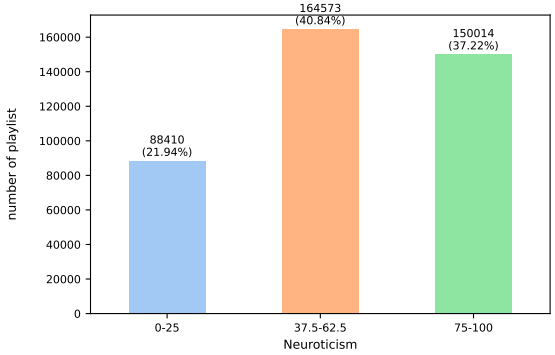
(k) Distribution of *extroversion* at song level



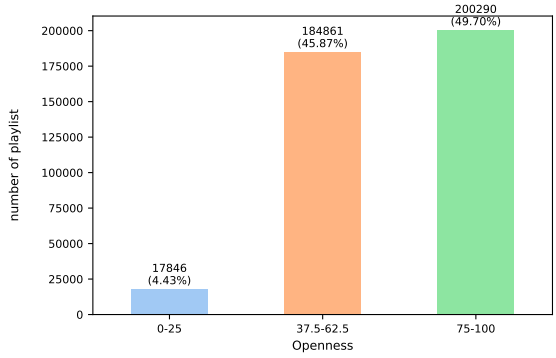
(l) Distribution of *agreeableness* at song level



(m) Distribution of *conscientiousness* at song level



(n) Distribution of *neuroticism* at song level



(o) Distribution of *openness* at song level

Figure 5.3: Distributions at song level of target features explained in Section 5.2.3 (cont.)

6

Correlation Analysis

The goal of correlation is to determine the strength of a relationship between two variables and express it through a normalized value. For any correlation method to be statistically significant, the values should always be compared with their respective p-values.

In this Section the metrics chosen to perform the actual measure of correlation are described in 6.1, then the obtained results are reported and commented at both levels of analysis in 6.2.

6.1 Metrics

It is necessary to have a theoretically correct understanding of the features under examination in order to determine methods for computing correlations. From the statistical point of view, variables can be numerical or categorical. Numerical (or quantitative) variables are those which are expressed through numerical data, either continuous or discrete. On the other hand, categorical (or qualitative) variables are used to describe data that fall into categories. If categories present an intrinsic order or rank, then the categorical variable is ordinal, otherwise, it is nominal.

When it comes to our dataset we can notice that all of our target features are categorical and that six of them — *age*, *economic*, *sport*, *smoke*,*drink* and *personality* respectively — are ordinal. Such observations lead us to use three different methods to evaluate variables correlation, since the appropriate one is different if we are dealing with two numerical variables, two categorical

variables or one categorical and one numerical variable. In sections 6.1.1-6.1.2 we will briefly report the correlation tools identified as most appropriate for each case.

6.1.1 Spearman's ρ

Spearman's test is a non-parametric test to measure correlations between numerical or ordinal variables. It captures the monotonic relationship between data and returns a value $\rho \in [-1, 1]$, where 0 implies that the two variables are actually independent and the sign highlights the direction of the correlation.

We used Spearman rather than Pearson because our data didn't respect the assumptions of the latest, since it does not have a normal distribution. Besides that, Pearson can only be utilized on comparisons between numerical data and, therefore, it would have been useless due to some target features being categorical. furthermore, Spearman is very similar to Pearson itself, meaning that it applies the same concept of measuring the deviation of data from the expected values but computes that on the ranking of the variables.

6.1.2 Logistic Regression

Logistic regression is a classification model which depicts binary dependent variable exploiting logistic functions. It does not properly return a correlation measure but is the only method that can provide an idea of the existence of a relationship between a dependent nominal variable and an independent numerical one. The concept is that if there is some kind of correlation then, the output of the logistic regression model with only the investigated numerical variable as independent component, should denote the correspondent coefficient as significant, i.e., the p-value returned from the z test for significance of the covariate in the model should be low.

To rephrase it, when checking for the existence of a correlation between a target feature and another attribute, we can estimate a logistic regression model for the target feature by providing as input the solely covariate. Then, we can perform the 2-tailed z test over the coefficient that the model estimate for the covariate in order to retrieve the p-value, and hence the significance, of the covariate for the model, i.e, if the covariate captures some behavior of target feature. In this work, when referring to significant levels of a certain value α while commenting on logistic regression's results, we mean that the p-value correspondent to the covariate under examination resulted to be $\leq \alpha$.

6.2 Results

In this section we are going to report and comment the obtained results of correlation research for each target feature at playlist level, trying to interpret them and give a brief explanation for the highest correlation values found as well. Discussions about what we indicated as most relevant features, i.e., *age*, *economic* and *personality*, is exhaustively addressed in the corresponding Subsections 6.2.1-6.2.3, while the inspection of all others is reported in Subsection 6.2.4-6.1. For every target variable, the correlation coefficients were computed against both categorical and numerical covariates.

In the case of numerical and ordinal target features, that are, *age*, *economic*, *personality*, *sport*, *drink* and *smoke*, Spearman formula was used to compute correlations with numerical covariates. For the others, we resort to logistic regression and therefore, we cannot provide a numerical measure but just indicate which correlations are found to be significant (at level α) from a statistical perspective.

The p-values were used to state the statistical significance of the measured correlations. The p-value threshold of 0.01 was deployed for all the computations.

In order to provide a visual representation of the numerically quantifiable correlations that were found significant, the Spearman's indices individuated for *age*, *economic*, *personality* at playlist level are reported in Figure 6.1, and *sport*,*smoke*,*drink* in Figure 6.2.

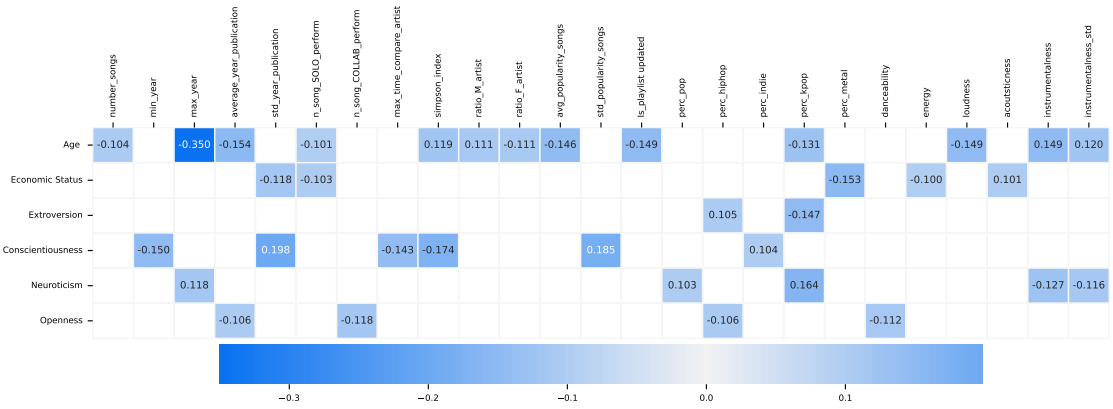


Figure 6.1: Significant (p-value $\leq$ 0.01) Spearman's correlation indices computed for age, economic, personality and dataset's numericial features at playlist level

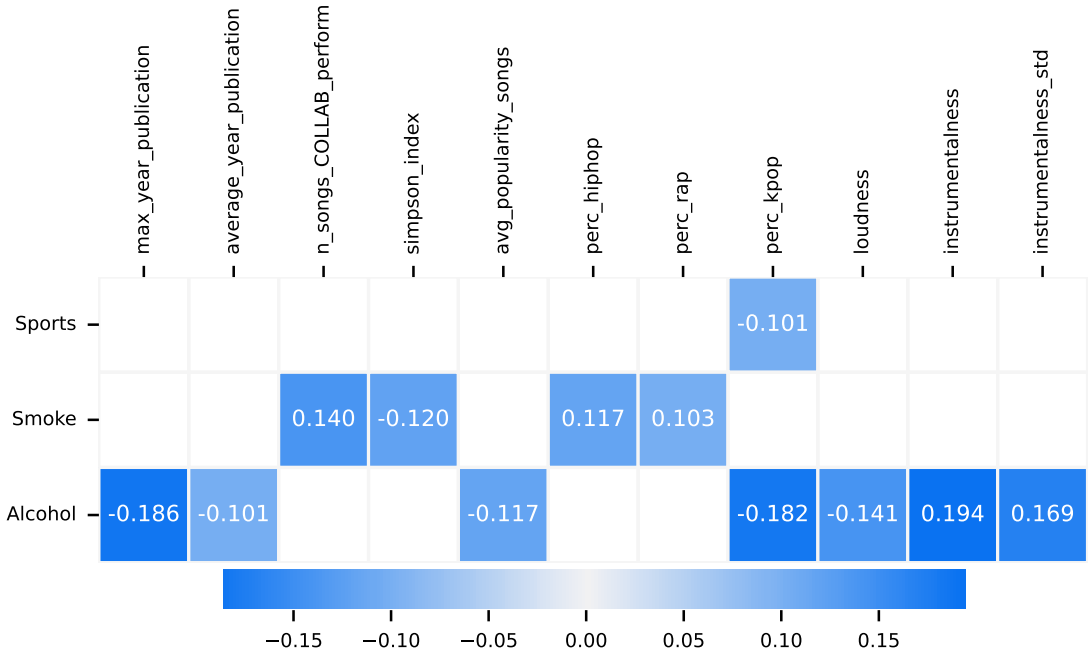


Figure 6.2: Significant (p-value $\leq$ 0.01) Spearman's correlation indices computed for sport, smoke, drink and dataset's numericial features at playlist level

6.2.1 Age

For the age feature at the playlist level, the correlation with the max_year, which denotes the maximum years of the publication of a song present in a playlist, was found significant at level $\alpha = 0.01$ with an index of $\rho = -0.350$.

Not surprisingly, we expected the connection, since it seems reasonable for older people to tend to have songs in their playlists that are published many years ago. Note that Spearman's score reached for the correlation is negative, which means along with the growth of the age, the max_years tends to reduce. Besides that, the average_year_publication comes obviously with $\rho = -0.154$ for the same reason and it seems people tend to add fewer songs when they get old and play the same playlist created and maturated along the years but explored some new content as described from the correlation with number_song($\rho = -0.154$), is_playlist_update($\rho = -0.149$), diversity_index($\rho = 0.119$).

Regarding the correlation with the audio feature, Spearman's analysis outline as noteworthy the relationship with loudness ($\rho = -0.149$) and instrumentalness ($\rho = 0.149$), since old people probably prefer peaceful music after hard work day.

6.2.2 Economic

At the playlist level, it is possible to individuate a negative correlation with n_song_solo_perform($\rho = -0.103$), which denotes rich people prefer songs from artists which are made by bands or concert group. From the analysis of numerical attributes, it emerges the negative relation between perc_metal ($\rho = -0.153$) and energy ($\rho = -0.100$), which seems to suggest that rich people don't like high-energy music like metal and prefer acoustic music as expressed by attribute acoutsticness ($\rho = 0.101$).

6.2.3 Personality

Personality feature is given by the ensemble of the big five personality traits, that are extroversion, agreeableness, conscientiousness, neuroticism, and openness. Each trait is studied singularly and has the same ordinal categories (low, mid, high) and evaluating the correlations at playlist level. We report in following results for four of them (extroversion, conscientiousness, neuroticism, and openness) with statistical significance of the p-value threshold 0.01.

1. **Extroversion**, usually associated with high-energy levels behaviour. Not surprisingly at the playlist level, it resulted to be correlated with perc_hiphop($\rho = 0.105$), an indicator of the proportion of hip-hop songs on the playlist, supporting the notion of Chamorro-Premuzic et al. [32] that extroverted individuals listen to hip-hop music more frequently.

   Another meaningful attribute that resulted highly correlated is perc_kpop_song ($\rho = -0.147$), which indicates how frequently k-pop songs appear in the playlist. It seems a result pretty difficult to interpret logically, which suggests probably extroverted people don't like k-pop music.

2. **Conscientiousness**, is strictly connected to efficiency and diligence. The results of Spearman's analysis suggest a relationship that is not easy to interpret at first glance.

   Observing the graph 6.1, a possible interpretation is the following: a high conscientious people wants to explore more types of genres of music and know in broad for each of them, demonstrated by the negative relationship with max_time _compare_artist ($\rho = -0.143$). When the conscientiousness increases, there are fewer songs per unique artist. In particular, they interest music belonging to more eras, from classic proven by the negative relationship with min_years ($\rho = -0.150$), but also popular songs released recently proven by the relationship of std popularity ($\rho = 0.185$)

3. **Neuroticism**, by being more related to stress levels and bad-tempered, at playlist level resulted to have a higher correlation with the perc_pop ($\rho = 0.103$) and perc_kpop

($\rho = 0.164$) and an inverse correlation with feature instrumentalness ($\rho = -0.127$), which represents the amount of vocals in the song. The closer it is to 1.0, the more instrumental the song is. One possible interpretation is the following: as the level of neuroticism increases, neurotic people tend to listen to high-tempo songs such as pop and kpop music to relieve stress, where the level of the presence of instrumental music is only supportive and backgrounding the song.

4. **Openness**, as suggested by the name, it evaluates the open-mindedness aspect. From our study, when considering playlist, openness has predominantly inverse correlation with average_years_publication ($\rho = -0.106$) and n_song_collab_perform ($\rho = -0.118$).

   One possible interpretation is as follows: as openness increases, average year publication decreases which reflects the playlist of a mature person. In the course of time, people become more aware and openminded, and they open up to diversity, and they explore other song tastes, not only from popular bands, but also from emerging single artists whose music is less popular.

6.2.4 SPORT, SMOKE AND DRINK

Evaluating the correlations at the playlist level of the sport, smoke, and drink target features, we found out that sport target features seem to not be highly correlated with any target in particular. The only notable correlation is with per_kpop ($\rho = -0.101$)and this is due probably to the fact that k-pop music is not suitable mainly as sports music.

Not surprisingly, we found out that the correlation between the target feature smoke and the perc_hip ($\rho = 0.117$) and perc_rap ($\rho = 0.113$), since the average image of rap or hip-hop artists, are accompanied by cigarettes and tattoos in the name of freedom, energy, free-thinking, and this could affect the collective image of their fans.

About the correlation between Alcohol(Drink) and features, Spearman's analysis outline as noteworthy the relationship with max_year_publication ($\rho = -0.1286$)and instrumentalness ($\rho = 0.194$). The main suggestion we deduced is people who tend to drink are adults who have the hobby of attending pubs and therefore lovers of instrumental rhythmic music.

6.2.5 NOMINAL TARGET CORRELATIONS

Considering that we need to find a correlation between numerical and nominal features, logistic regression is an option as explained in 6.1.2. It has been used to determine if a relationship exists between playlist features and gender, occupation, country, marital status, and livewith.

Table 6.1: Significant Correlations at different p-values for nominal target

Target	a<0.01	a<0.005	a<0.001
Marital status	31	29	26
Gender	37	36	33
Occupation	39	37	36
Livewith	12	12	7
Country	40	38	34

For each of them, in table 6.1 , three different p-values are examined and we count how many features are correlated, resulting in $\leq \alpha$. Recall the p-values were used to state the statistical significance of the measured correlations.

For marital status, the target feature resulted correlated with 31 attributes with $\alpha < 0.01$, 29 with $\alpha < 0.005$, and 26 with $\alpha < 0.001$.

For the gender feature at the playlist level, the number of features correlated is 37 considering $\alpha < 0.01$, 36 with $\alpha < 0.005$, and 31 $\alpha < 0.001$.

For occupation, the target feature resulted correlated with 39 attributes with $\alpha < 0.01$, 37 with $\alpha < 0.005$, and 36 with $\alpha < 0.001$.

For livewith, the target feature resulted correlated with 12 attributes with $\alpha < 0.01$, 12 with $\alpha < 0.005$, and 7 with $\alpha < 0.001$.

For country, the target feature resulted correlated with 40 attributes with $\alpha < 0.01$, 38 with $\alpha < 0.005$, and 34 with $\alpha < 0.001$.

7 Predictions

Besides finding the existence of correlations between our datasets and targeted private features, we also intended to perform a simple test to illustrate that a correlation-based technique can be effective in inferring private data in a musical streaming service.

The main purpose is not to build a robust model with excellent performance, but simply to demonstrate that performing feature selection using the correlation metric may help to have better performance than a dummy predictor since this is enough to show that the attributes we collected are actually relevant to infer user's private data.

We conducted test on the 11 target features described in Section 5.2.3 at playlist levels and at song level, using the Python programming language. The employed metrics and techniques are described in Section 7.1 and all of the random states of the case were fixed for the reproducibility of the results.

7.1 Predictive Models

For the experiments we computed the results using four different machine learning models:

- Logistic Regression, due to the fact that is a fairly simple approach that classifies binary data and also presents an extension for the multiclass case;
- Ridged Classifier, which analyzes linear discriminant models in order to discover if there is a linear relationship between the attribute and the modeled feature.

- Support Vector Machine, which analyzes if is it possible to find a linear or no linear hyperplane in an N-dimensional space that distinctly classifies the data points. In our experiment, we used non-linear kern rbf.
- Decision Trees Classifier, because they facilitate interpretability, allowing and easier explanation of the connection of attributes with the modeled feature, which is in line with the concept or correlation itself;
- Random Forest Classifier, to test something more complex with respect to single decision trees, since this is an ensemble of them yet maintaining interpretability.

All of these models were implemented using the correspondent methods of the *scikit-learn*[1].

7.1.1 Cross Validation

For each model, grid search and cross-validation were applied in order to optimize the hyperparameters and obtain more stable results. The main idea is the following: first, shuffle the dataset and split into k number of subsamples and in the first iteration, the first subset is used as the test data while all the other subsets are considered as the training data. Train the model with the training data and evaluate it using the test subset. In the next iteration, select a different subset as the test data set and re-train the model with the training data and test it using the new test data set and keep the evaluation score again. Continue iterating the above k times. Each data subsamples will be used in each iteration until all data is considered. In the end, we will end up with a k number of evaluation scores and the total error rate is the average of all these individual evaluation scores. In both cases, at playlist level and at the song level, the cross-validation technique employed with K-Fold with K=8, meaning that data was divided into 8 chunks maintaining the percentage of samples for each class as in the whole set.

7.1.2 Score Metric

Since the F1-score considers the distribution of data rather than accuracy, it was used to measure the model's performance.

It is strictly correlated with the concepts of precision and recall, which further differentiate correct and misclassified prediction of samples on the base of their actual values. To better explain, let's consider a binary classification problems with two classes: 0 (negative) and 1 (positive); then we can define the notions of:

[1] https://scikit-learn.org/stable/

- True Positives (TP) as number of samples belonging to the positive class that are correctly classified, i.e, number of 1's predicted as 1;
- True Negatives (TN) as number of samples belonging to the negative class that are correctly classified, i.e. number of 0's predicted as 0;
- False Positives (FP) as number of samples belonging to the negative class that are misclassified, i.e., number of 0's predicted as 1;
- False Negatives (FN) as number of samples belonging to the positive class that are misclassified, i.e., number of 1's predicted as 0;

A visual representation of such definitions is reported in Figure 7.1, in order to provide a clean explanation.

	Actual Positives	**Actual Negatives**
Predicted Positives	True Positives (TP)	False Negatives (FN)
Predicted Negatives	False Positives (FP)	True Negatives (TN)

Figure 7.1: Visual representation of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) in a binary classification problem

Now that we have introduced such necessary concepts, we can provide formula descriptions of precision, recall, and F1-score in Equations 7.1, 7.2, and 7.3 respectively.

$$\textit{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7.1}$$

$$\textit{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7.2}$$

$$\textit{F1-score} = \frac{2 \cdot \textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \tag{7.3}$$

This formulation can also be applied to the multiclass case by calculating a score for each class, accounting it as positive if belong to it, and evaluating all other classes as negative. Since

this brings to a number of F1-scores equal to the number of classes, in order to obtain a measure for the overall performance, one should average such values. Module *metrics* of *scikit-learn* provide multiple possible options for the average computation of F1-scores, in the specific we used the macro one, which computes the metric for each label and then provides their unweighted mean.

7.1.3 Feature Selection and Oversampling

Feature selection was performed by trying three different methods, that are:

1. Our correlation extraction procedure, illustrated for the inspection of target features in Chapter 6, together with thresholds in order to keep only the significantly correlated attributes with highest indices;
2. *SelectKBest* univariate technique from *scikit-learn*, which determines the *k* features to select according to a score function.
3. A combination of the two, followed by an elimination procedure to drop eventual duplicate attributes selected by both techniques. In our experiment, we chose k=15.

To resolve the problem of highly unbalanced classes, particularly relevant for some features individuated in Section 5.4, we also applied a combination of data over-sampling for fair learning. In particular, we used SMOTE classes from the *imbalanced-learn*[2] library

The main idea is based on over-sampling data through SMOTE, which is the technique responsible for generating synthetic samples, and then cleaning the noisy space resulting from such creation using a heuristic, which influences the criteria to choose which samples to eliminate. All training set's features were appropriately encoded, scaled, and provided as input to the *imbalanced-learn* pipeline, comprehensive of feature selection, oversampling (when needed) and model-defining steps.

7.1.4 Train, Validation,Test set

We considered prediction results from two points of view building two datasets, one using all features collected in 5.2and the other using the best features from 7.1.4. In both cases, we divided the dataset into three portions: 70% as a train set used to train the model, 10% as validation set, used to tuning model hyperparameters, and 20% as test set, used to provide an unbiased

[2] https://imbalanced-learn.org/stable/

Table 7.1: Playlist level results. For each target features are reported best limiter and type of features used for stratified dummy, svm, logistic regression, decision tree, ridge, and random forest classifiers. Best achieved performance are highlighted

Target feature	best limiter	features used	dummy	svm	LR	DT	RI	RF
Age	8	all features	28.2%	38.3%	39.9%	41.8%	38.8%	**43,5%**
Economic	12	all features	29.3%	**36.3%**	35.4%	30.5%	35.5%	32.3%
Gender	4	best features	47.6%	59.9%	68,1%	61,1%	68,2%	**71,7%**
Occupation	6	best features	30.77%	**40.3%**	32.2%	34.1%	31.9%	33.0%
Country	6	all features	17.7%	25.4%	26.4%	21.0%	23.4%	**30.4%**
Sport	6	best features	45.4%	**54.7%**	46.2%	49.3%	45.5%	46.8%
Smoke	6	best features	45.1%	52.0%	55.2%	55.7%	52.2%	**60.2%**
Drink	4	best features	53.0%	54.9%	59.3%	54.8%	59.8%	**62.8%**
Marital	8	all features	27.4%	36.3%	41.0%	41.7%	40.3%	**45.1%**
Livewith	8	all features	43.4%	51.9%	49.8%	48.3%	49.7%	**54.5%**
Agreeableness	8	best features	28.5%	31.1%	32.9%	30.7%	33.2%	**34.7%**
Conscientiousness	6	best features	33.4%	37.3%	35.9%	33.5%	33.6%	**37.9%**
Extroversion	8	best features	30.4%	30.5%	35.9%	35.3%	35.4%	**37.8%**
Neuroticism	10	best features	32.1%	34.0%	**41.1%**	35.2%	40.4%	41.0%
Openness	10	best features	30.9%	35.9%	32.4%	33.7%	32.6%	**36.5%**

evaluation of a final model. We made a division also for users, more precisely if a user along with his playlists or songs were used in train phase, then the they never appears in validation phase or test phase and vicersa, otherwise it will become recognition problem. In order to prevent the problem of too many playlists/songs belonging to a single user, we decided to place a limiting parameter. We aim to balance people who have lots of playlists with those who have few, so as to avoid bias. We decided on the value of the limiter based on the median of the number of playlists/songs of individual users. At the playlist level, we considered prediction results in relation to multiple values for the limiter and at the song level, we took into consideration only the median as values of the limiter.

7.2 Playlist Level Results

In order to provide a comparison term for our results we also computed the performance of dummies classifiers[3]. In our case we considered stratified ones.

In Table 7.1 we reported the outcome of our experiments at playlist level. For each target feature are indicated mean of all the tested models, together with those of the stratified dummy.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

As can be clearly seen by Table 7.1, at playlist level, Our approach was relatively successful, with an increase in accuracy of about 10% compared to a dummy model in most cases. In contrast, for the personality feature, the mean scores are similar to those of the best dummy classifier in majority of case, and the increase is not particularly significant.

7.3 Song Level Results

In Table 7.2 are depicted the results obtained at song level. For each target feature are reported mean F1-score and standard deviations of grid search results for models illustrated in Section 7.1. For comparison reasons, the dummy classifier with stratified strategy is also shown, since it performed better than the other two considered dummies. As opposed to the playlist level, we computed results using limiter = 250 and the best features described in 2

Table 7.2: song level results. Each target feature is reported by mean F1-score and standard deviations for stratified dummy, SVM, logistic regression, decision tree, ridge, and random forest classifiers. Best achieved performances are highlighted

target feature	dummy stratified		svm		logistic regression		decision tree		ridge classifier		random forest	
	mean	std dev	mean	std dev	mean	std dev	mean	std dev	mean	std dev	mean	std dev
age	26.90%	0.0028	35.78%	0.039	32.12%	0.0096	37.16%	0.0097	31.55%	0.010	**37.72%**	0.007
gender	48.36%	0.0086	57.61%	0.0309	63.51%	0.0195	62.61%	0.0244	62.99%	0.0180	**63.88%**	0.02130
economic	31.83%	0.0055	30.67%	0.0124	31.78%	0.018	33.57%	0.0097	31.66%	0.0183	**33.86%**	0.0107
occupation	28.47%	0.0052	32.17%	0.0259	34.48%	0.019	32.86%	0.0117	**38.04%**	0.0180	33.86%	0.0107
marital	28.47%	0.0068	32.14%	0.0110	33.14%	0.011	33.00%	0.007	31.87%	0.0138	**33.39%**	0.0107
sport	47.78%	0.0101	45.86%	0.0570	**52.09%**	0.012	48.99%	0.236	50.31%	0.0134	51.49%	0.0145
smoke	45.11%	0.0141	50.23%	0.2722	**53.52%**	0.015	51.91%	0.009	50.53%	0.0196	52.47%	0.0233
drink	49.69%	0.0032	50.42%	0.2561	**53.42%**	0.012	51.47%	0.030	53.28%	0.0146	53.39%	0.0222
country	21.30%	0.0052	25.02%	0.0052	23.74%	0.0052	22.42%	0.0198	23.17%	0.002	**25.44%**	0.002
livewith	41.95%	0.007	45.02%	0.037	48.02%	0.0126	47.30%	0.0168	43.39%	0.009	**48.50%**	0.005
agreeableness	29.91%	0.135	30.23%	0.014	32.77%	0.027	33.50%	0.0038	31.99%	0.247	**33.74%**	0.010
extroversion	31.97%	0.003	29.51%	0.008	**36.22%**	0.010	32.12%	0.0277	34.86%	0.009	35.18%	0.012
consciousness	30.27%	0.009	31.70%	0.023	34.90%	0.010	33.77%	0.0147	33.63%	0.022	**34.77%**	0.0055
neuroticism	32.77%	0.002	30.82%	0.018	35.75%	0.014	35.37%	0.006	35.58%	0.0153	**36.93%**	0.0184
openness	29.35%	0.002	29.69%	0.024	31.89%	0.006	35.14%	0.0317	31.67%	0.0086	**35.86%**	0.0206

Similar to what happened at the playlist level, our models performed in general slightly better than dummy classifiers, in particular for features such as gender, age, drink, and smoke, each of which will be discussed in more detail in the following sections. For the other targets, our approach wasn't successful at all, particularly for personality and economics, the highest scores between our models and dummy classifier are rather similar.

In general, the playlist approach is better than the song approach, since playlists describe in more detail the character of a person or group of people since the playlist includes more songs organized according to user preference. However, even with only a single song, we can also find some information about the group listening to the such song.

7.3.1 Gender

By employing random forest (RF) and using a feature selection strategy, we reached an F1-score of 71.7%, against the 47.60% obtained by the best dummy at playlist level and 63.88%, against the 48.36% obtained by the best dummy at song level. Note that, in particular in the song level, the standard deviation is very small, being around $2 - 3\%$, meaning that models' behaviour is pretty stable. We would like to point out that we report the best result between using all features and the best feature. The number of best-employed features is 15 and other features found are still useful in most of the cases since after comparison, in the majority of cases, the results obtained between approaches between using all features and best features do not differ by so many(around 3-4%)

7.3.2 Smoke and Drink

Random forest turned out to be the model with better performance when predicting *smoke* and *drink* reaching an average score respectively of 60.20% and 62.80% at playlist level; and 53.52% and 53.42% presenting 0.015 and 0.012 as standard deviations by logistic regression model at song level. It is worth mentioning that both cases used the best features as input, more precisely, 15 features were derived using the SelectKbest technique presented at 2, in addition, we include top-correlated features based on Spearman indices.

7.3.3 Age

Although the model for the target Age does not achieve high accuracy, it still deserves mention because the difference between the best model and the dummy one is very noticeable.

The model that reached the best performance for the *age* target feature is the random forest, having a mean F1-score of 43.50%, against the 28.20% obtained by the best dummy at playlist level and 37.72%, against the 26.90% obtained by the best dummy at song level.

In spite of the fact that we used Smote, the samples may not always be of high quality. From what we see in the strong Spearman correlations between age and some features, we believe that having the original balanced dataset would certainly increase these results, however, due to the age composition of the survey respondents, it is very challenging to get responses from more elderly respondents.

8 Discussion

According to our findings, this attack could pose a significant threat to user privacy worldwide. We have conducted our experiments on Spotify, but we now discuss the applicability to other streaming services and provide more reasons about our assumptions, possible countermeasures, and limitation of our work.

8.1 Applicability to Other Music streaming service

To conduct our attack, we need access to public data, which is provided mainly by the streaming service itself. Therefore, this is the first requirement to be able to apply such offensive to other platforms. As reported in Table 3.1, we can perform our attack on all the analyzed music streaming services present in it. The second requirement is to assure to find music preference features that correlate with private information. Among the attributes we used, we found that the music genre, favorite artists, and a custom name for each playlist were useful for the purpose. These kinds of features are rather general and can be easily found in other music streaming services as well. Furthermore, most platforms provide payment options to unlock additional services, such as premium services or unlimited skips for advertisements, etc, and obviously, these kinds of features can be commonly found in reality and are very useful for prediction. To conclude, although we haven't directly demonstrated the attack's applicability on other platforms if these conditions can be met, it seems likely the attack will succeed.

8.2 Reasoning on Assumptions

In our attack, we assumed two key factors:

1. The possibility offered by Streaming music services for the data collection in order to pursue the attack;
2. The platform is trusted ;

Regarding 1) A malicious attacker could collect public information from streaming music services, and ideally, this should not be difficult by using API services, and obviously, this would not take much longer to complete.

Concerning 2) The platform must be trusted otherwise, the streaming music service company could be the attacker of our scenario. It is known that companies use user info to suggest new genres of music every day. Nothing prohibits them to do similar profiling to the one we proposed. Moreover, at registration time, the streaming service companies often ask for information such as the birth date and gender. Even if they are optional fields, there are certainly people giving these informations. In this way, people logging in through it would be found on Social Networks and their data would serve as the ground truth.

8.3 Possible Countermeasures

A first and intuitive countermeasure would involve the music streaming service company producer by setting the users' profiles private by default.

As a second countermeasure, the music platform could allow its users to regulate their accounts' visibility. For example in Spotify, the default option could be that users can only see the playlists and favorite artists with friends and not with all people.

8.4 Limitations

We will now describe and quickly address the limits we found while developing this thesis. First, we have to address a problem connected with data collection: the adequacy of the gathered sample to represent the whole population.

We were able to acquire sufficient information on 750 people by adhering to proper survey procedures in order to exclude inaccurate data and keep our sample as less biased as possible. By

examining the distribution of target characteristics in our dataset, it appears that some classes are significantly imbalanced. (as declared in Section 5.4).

Furthermore, the number of considered users is still relatively small, especially for a such high number of users (422M monthly, as reported in 3.1.4), thus one of our limitations is the size of the sample we deployed. Besides that, we need to acknowledge that data gathering was conducted only through a survey, which required the consensus of users, who also are the very object of the attack. Notice that we had to utilize the survey method for ethical reasons linked to the private data treatment of users, however, a malicious attacker could also resort to social media harvesting. Following that, additional important limitations are related to the feature selection strategy for machine learning classifiers, for example, univariate methods, like *SelectKBest*, were considered in this research, ignoring possible interactions between covariates which are likely to be relevant instead. Another limitation of this work consists on the poor results in terms of predictive performance for the majority of target features, especially at the song level. When commenting on this aspect, one must bear in mind that this is a relatively unexplored field, so a lack of literature negatively impacts the quantity of aspects to investigate and the number of tests to conduct.

Due to the lack of prior knowledge on the issue, the repercussions of this final disadvantage were especially substantial when training machine learning classifiers and tuning parameters for model and feature selection, making such processes exceedingly time-consuming.

9 Conclusions

In conclusion, we proposed a machine learning-based attack in order to infer users' private information starting from the data they generate when listening to music on streaming services.

We used Spotify as a case study to demonstrate the applicability of the attribute inference attack using music data, but under general conditions, we can use any music streaming service that shows publicly users' music tastes. In light of the fact that music is part of us, and it represents what we really are behind masks, if this information is misused, we risk being seen as transparent and being a target for attackers who want to profit by selling our private data to advertisers [28] or even in the worse case, for minors to be tracked as victims of malicious activity like cyberbullying [6] and cyberharassment.

In short, the attack is divided into two stages: the first involves gathering ground truth in order to properly train a classifier, and the second involves employing such a model to derive users' personal data. we collected private data through a survey with good methodology practices and then proceeded to elaborate on such information together. Thereafter, we created two datasets containing users' personal features: one with attributes based on the playlists and the other one based on songs. We then used statistical measures to compute the correlation between music data and the set of survey-collected target attributes, and we examined the results in terms of significance based on the reported p-values. Therefore, we continued with the predictive part, and while some of the obtained results were not very high in comparison to those normally achieved through machine learning nowadays, we are confident in stating that this is normal, given that the research topic is still unexplored and has rather limited literature.

Fortunately, we were able to create models that outperformed dummy classifiers for all of the investigated targets. For some of them, the improvement is not relevant, which is most likely due to the constraints of our study, as we described in 8.4 Future research in this area should attempt to overcome these shortcomings and improve the machine learning part. Furthermore, we may evaluate and profile people based on the results of their playlists and songs. Finally, we can state that exploiting correlations provided by music data is a method worth investigating when it comes to inferring private features, and hopefully, this work will serve as a good ground for motivating future studies in this area.

A

Spotify Survey

Here we attach the survey used in our research study. It was divided into three sections: General Information, Spotify usage Information, and Personality Questions.

General Information This section was used to ask for general information, as well as some private data that, in our opinion, could be inferred through music preference information.
Spotify usage Information In this section we asked for information about the Spotify user experience. We used some of the answers in this section to evaluate the users' attention and coherence.

Personality Questions Ten short personality questions were asked in this section. We used the test studied in [29] in the English version. We asked such questions because we think that personality and music preference are strongly related.

In figure A.1, we report the welcome page of our survey with instructions, and then in the following, we report all the questions.

Figure A.1: Spotify survey

General Information Questions

1. What is your Spotify profile link?
2. What is your Gender?
    - Female
    - Male
3. What is your Age?
4. What is your country?
5. What is your Marital Status?
    - Single
    - In a relationship/Engaged
    - Married
    - Divorced

- Widowed

6. Who do you live with?
    - I live alone
    - I live with others

7. What is your Occupational Status?
    - I am a student
    - I have a job
    - I am a working student
    - I am currently unemployed

8. Do you play any sports/do physical activity?
    - Yes, at amateur level
    - Yes, at competitive level
    - Sometimes

9. How do you consider your Economic Status?
    - 1 (low)
    - 2
    - 3
    - 4
    - 5 (high)

10. Do you smoke ?
    - Yes (almost every day)
    - Sometimes (few cigarettes per week)

- In rare occasion
- No

11. Do you drink alcohol?

    - Very often (5+ times per week)
    - Sometimes (3-4 times per week)
    - Rarely (1-2 times per week)
    - No

Spotify Usage Questions

1. How many hours per day do you spend on Spotify?

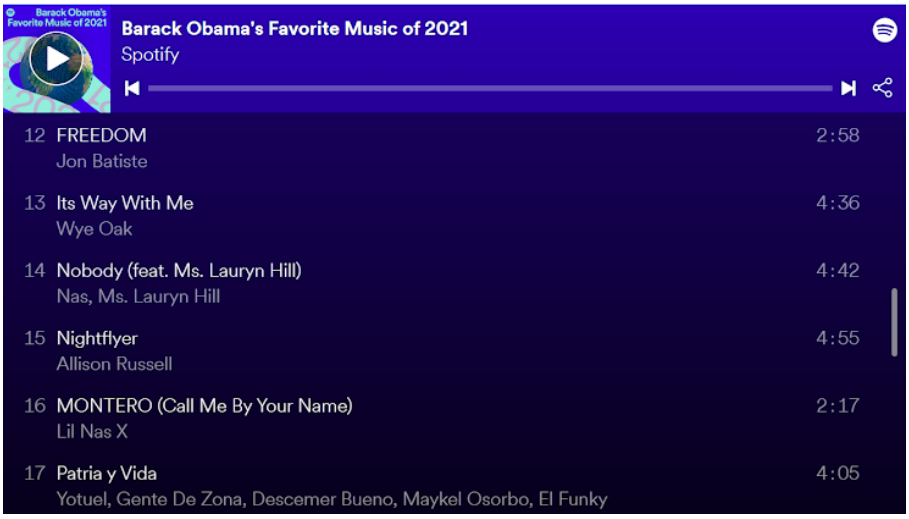
    - Less than 1 hour
    - 2-3 hours
    - 4-5 hours
    - More than 6 hours

2. How long have you been using Spotify?

    - Less than 1 year
    - 2-4 years
    - More than 5 years

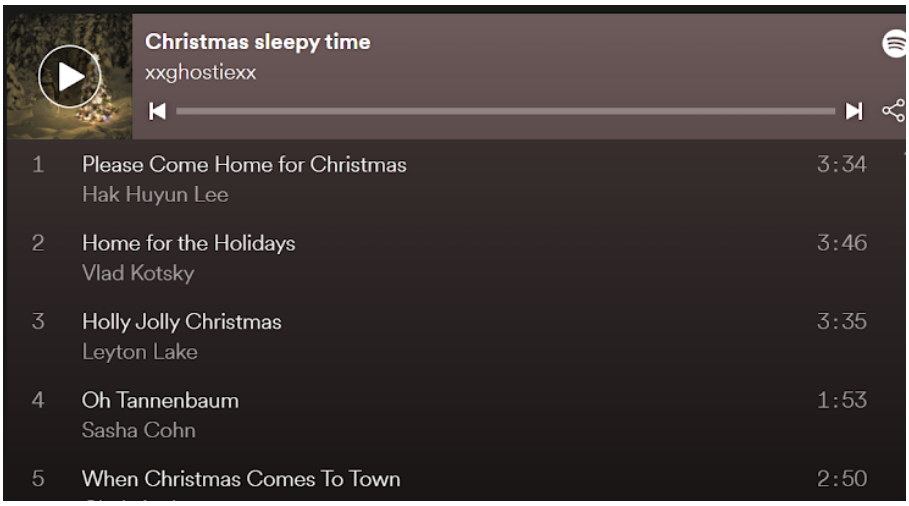
3. Given the image BELOW, which of the following songs is preferred by Barack Obama?

    - Without Me - Eminem
    - Freedom - Jon Batiste
    - My heart will go on - Céline Dion
    - Oskar Schuster - Les Sablons



4. Given the image BELOW, which one is best period to play such playlist?

    - Halloween
    - Christmas
    - Easter
    - Valentine's day



5. What are your favorite music genres? (one or more options)

    - Pop
    - Hip-hop and Rap
    - Rock
    - Dance and Electronic music
    - Latin music
    - Indie and Alternative Rock
    - Classical music
    - K-Pop
    - Country
    - Metal
    - Other

Personality Questions

1. I see myself as someone who..... is reserved:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

2. I see myself as someone who..... is generally trusting:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

3. I see myself as someone who..... tends to be lazy:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

4. I see myself as someone who..... is relaxed, handles stress well:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

5. I see myself as someone who..... has few artistic interests:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

6. I see myself as someone who..... is outgoing, sociable:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

7. I see myself as someone who..... tends to find fault with others:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

8. I see myself as someone who..... does a thorough job:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

9. I see myself as someone who..... gets nervous easily: *

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

10. I see myself as someone who..... has an active imagination:

    - 1 (Disagree strongly)
    - 2
    - 3
    - 4
    - 5 (Agree Strongly)

B

Spotify Api

In this appendix, we give more details about how we retrieved playlist and song information from Spotify and we briefly show some Python scripts used in our research study. For the complete guide, please refer to the official Spotify API page[30].

B.1 Authentication

All requests to Spotify API require authentication. This is achieved by sending a valid OAuth access token in the request header, or by using a temporary client_id and client_secret via an auth_manager. In B.1, we demonstrate how to set up an authentication manager in Python.

Listing B.1: Python example:auth_manager object

```
import sys
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials

sp= spotipy.Spotify(auth_manager=SpotifyClientCredentials
(client_id="8bf396be841a44c4974cb84bed8b6dbf",
client_secret="cf690094f7f84bdf81404959e4175276")
)
```

To obtain Spotify credentials, all we need is a Spotify Developer account, which is free. Once connected to your Spotify account, we can access our credentials from the dashboard.

B.2 Public playlist and song

Tracks, playlists, and users are identified by a base-62 identifier found at the end of their Spotify URI. In general, Spotify ID does not clearly identify the type of resource. In figures B.1, B.2 and B.3, we report respectively the URI of the Spotify playlist, track, and user whose Spotify ID appears at the end of the URI.

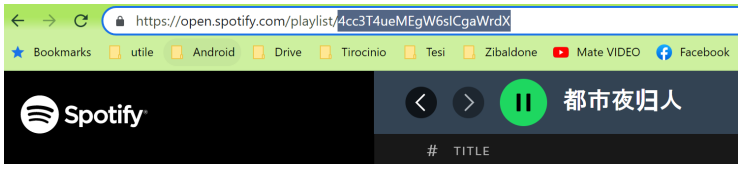


Figure B.1: Overview of Spotify playlist ID in URI .

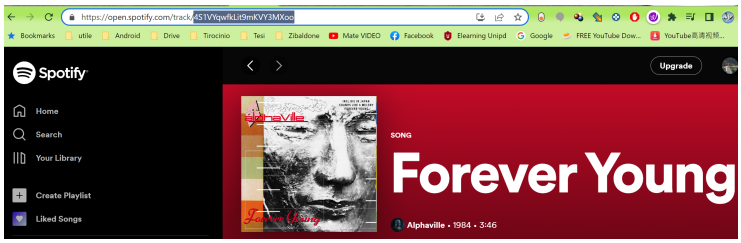


Figure B.2: Overview of Spotify Song ID in URI .

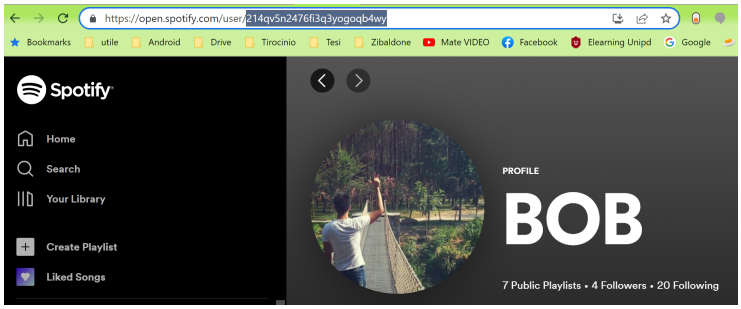


Figure B.3: Overview of Spotify User ID in URI .

Once we have the Spotify Id of some user u, we can use the method in B.2 to retrieve all playlists IDs of u and then use the method in B.3 to retrieve all songs IDs.

Listing B.2: Python example: user$_{p}ublic_{l}ist$

```
def user_public_list(user_in):
#given a user, return all id playlists of such user
    user = user_in
    if len(sys.argv) > 1:
        user = sys.argv[1]
    playlists = sp.user_playlists(user)
    ls_pl=[]
    while playlists:
        for i, playlist in enumerate(playlists['items']):
            campi=playlist['uri'].split(":")
            ls_pl.append(campi[2])
        if playlists['next']:
            playlists = sp.next(playlists)
        else:
            playlists = None
    return ls_pl
```

Listing B.3: Python example: user_public_song

```
def get_song_from_playlist(pl_id):
    #given a playlist,
    return all songs belonging to such playlist

    offset = 0
    while True:
        response = sp.playlist_items(
            pl_id,  offset=offset,
            fields='items.track.id,total',
            additional_types=['track']
        )
```

```
        if len(response['items']) == 0:
            break
        print(response['items'])
        offset = offset + len(response['items'])
        print(offset, "/", response['total'])
```

In general, Spotify API responses include a JSON object. Starting with playlists or song IDs, we can save all information regarding them in JSON format by using the method B.4.

Listing B.4: Python example: JSON_format_save

```
def get_JsonFormat_playlist(pl):
    import json
    ris = sp.playlist(pl_id)
    save(json.dumps(ris, indent=4),str(pl_id))

def save(s,namefile):
    s=str(s)
    text_file = open(path, "w",encoding="utf-8")
    n = text_file.write(s)
    text_file.close()
    print("saved")
```

Finally, once get the JSON file, we can extract features regarding the playlist or song using all the information inside it.

In general, when we download the JSON file of a playlist, it also contains the JSON contents of all the songs that belong to the playlist. This information can be found in the item subsection of the section tracks in the playlist JSON. In the B.5, we report an example of a playlist JSON file

Listing B.5: Json example: Playlist Json file

```
{
  "collaborative": true,
  "description": "string",
  "external_urls": {
    "spotify": "string"
  },
  "followers": {
    "href": "string",
    "total": 0
  },
  "href": "string",
  "id": "string",
  "images": [
    {
      "url": "https://i.scdn.co/image/ab67616d00001e02ff9ca10b55ce82ae
          553c8228\n",
      "height": 300,
      "width": 300
    }
  ],
  "name": "string",
  "owner": {
    "external_urls": {
      "spotify": "string"
    },
    "followers": {
      "href": "string",
      "total": 0
    },
    "href": "string",
    "id": "string",
    "type": "user",
    "uri": "string",
```

```
    "display_name": "string"
  },
  "public": true,
  "snapshot_id": "string",
  "tracks": {
    "href": "https://api.spotify.com/v1/me/shows?offset=0&limit=20\n",
    "items": [
      {}
    ],
    "limit": 20,
    "next": "https://api.spotify.com/v1/me/shows?offset=1&limit=1",
    "offset": 0,
    "previous": "https://api.spotify.com/v1/me/shows?offset=1&limit=1"
       ,
    "total": 4
  },
  "type": "string",
  "uri": "string"
}
```

References

[1] A. Coffey, "The impact that music streaming services such as spotify, tidal and apple music have had on consumers, artists and the music industry itself," *Interactive Digital Media. University of Dublin*, 2016.

[2] "spotify-statistics," https://www.businessofapps.com/data/spotify-statistics/.

[3] Spotify, "Spotify premium," https://www.spotify.com/it/premium/, 2022.

[4] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5802–5805, 2013. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1218772110

[5] F. Commission, *Data brokers: A call for transparency and accountability*, 01 2014, pp. 1–101.

[6] E. Englander and A. Muldowney, "Just turn the darn thing off: Understanding cyberbullying," *Proceedings of Persistently Safe Schools: The 2007 National Conference on Safe Schools and Communities*, 01 2007.

[7] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/Off: Preventing Privacy Leakage From Photos in Social Networks," in *Proceedings of the 2015 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Denver, USA: ACM, October 2015.

[8] J. Morris, S. Newman, K. Palaniappan, J. Fan, and D. Lin, ""do you know you are tracked by photos that you didnt take: Large-scale location-aware multi-party image privacy protection," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.

[9] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings," in *Proceedings of the Sixth ACM Conference*

on Recommender Systems, ser. RecSys '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 195–202. [Online]. Available: https://doi.org/10.1145/2365952.2365989

[10] Y. Zhang, N. Gao, and J. Chen, "A practical defense against attribute inference attacks in session-based recommendations," in *2020 IEEE International Conference on Web Services (ICWS)*, 2020, pp. 355–363.

[11] B. A. Pijani, A. Imine, and M. Rusinowitch, "You are what emojis say about your pictures: Language-independent gender inference attack on facebook," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ser. SAC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1826–1834. [Online]. Available: https://doi.org/10.1145/3341105.3373943

[12] Y. Cheng, J. Park, and R. Sandhu, "Preserving user privacy from third-party applications in online social networks," in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW '13 Companion. New York, NY, USA: Association for Computing Machinery, 2013, p. 723–728. [Online]. Available: https://doi.org/10.1145/2487788.2488032

[13] G. Apruzzese, P. Laskov, E. M. de Oca, W. Mallouli, L. B. Rapa, A. V. Grammatopoulos, and F. D. Franco, "The role of machine learning in cybersecurity," *Digital Threats*, jul 2022, just Accepted. [Online]. Available: https://doi.org/10.1145/3545574

[14] J. Golbeck, C. Robles, and K. Turner, "Predicting personality with social media," in *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 253–262. [Online]. Available: https://doi.org/10.1145/1979742.1979614

[15] D. Jurgens, T. Finethy, J. McCorriston, Y. Xu, and D. Ruths, "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, no. 1, pp. 188–197, Aug. 2021. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14627

[16] N. Z. Gong and B. Liu, "You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors," in *25th USENIX Security*

Symposium (USENIX Security 16). Austin, TX: USENIX Association, Aug. 2016, pp. 979–995. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/gong

[17] C. Abdelberi, G. Ács, and M. A. Kâafar, "You are what you like! information leakage through users' interests," in *NDSS*, 2012.

[18] T. Chen, R. Boreli, D. Kaafar, and A. Friedman, "On the effectiveness of obfuscation techniques in online social networks," vol. 8555, 07 2014, pp. 42–62.

[19] T. Yo and K. Sasahara, "Inference of personal attributes from tweets using machine learning," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3168–3174.

[20] J.-Y. Liu and Y.-H. Yang, "Inferring personal traits from music listening history," in *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, ser. MIRUM '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 31–36. [Online]. Available: https://doi.org/10.1145/2390848.2390856

[21] T. Krismayer, M. Schedl, P. Knees, and R. Rabiser, "Predicting user demographics from music listening information," *Multimedia Tools and Applications*, vol. 78, 02 2019.

[22] M. Schedl, "The lfm-1b dataset for music retrieval and recommendation," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ser. ICMR '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 103–110. [Online]. Available: https://doi.org/10.1145/2911996.2912004

[23] "Apple to pay 3 billion to buy beats," *The new york time*, 2014. [Online]. Available: https://www.nytimes.com/2014/05/29/technology/apple-confirms-its-3-billion-deal-for-beats-electronics.html

[24] "Tidal: 10 things you need to know," 2015. [Online]. Available: http://www.theguardian.com/music/2015/apr/05/tidal-10-things-you-need-to-know-jay-zmadonna-music-streaming

[25] "'reclaiming the music: The power of local and physical music distribution in the age of global online services'," *New Media Society,*.

[26] "Wild patterns: Ten years after the rise of adversarial machine learning," *ARXIV*, 2017. [Online]. Available: https://arxiv.org/abs/1712.03141

[27] M. Fryling, J. L. Cotler, J. Rivituso, L. Mathews, and S. Pratico, "Cyberbullying or normal game play? impact of age, gender, and experience on cyberbullying in multi-player online gaming environments: Perceptions from one gaming forum," *Journal of Information Systems Applied Research*, vol. 8, no. 1, p. 4, 2015.

[28] "Data brokers: a call for transparency and accountability," *Federal Trade Commission*, 2014.

[29] B. Rammstedt and O. P. John, "Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german," *Journal of Research in Personality*, vol. 41, no. 1, pp. 203–212, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0092656606000195

[30] Plamere, "Spotipy," https://github.com/plamere/spotipy, 2019.

[31] J. S. Wiggins, "'the five-factor model of personality: Theoretical perspectives." *Guilford Press.*

[32] T. Chamorro-Premuzic, P. Fagan, and A. Furnham, "Personality and uses of music as predictors of preferences for music consensually classified as happy, sad, complex, and social," *Psychology of Aesthetics, Creativity, and the Arts*, vol. 4, pp. 205–213, 11 2010.

Acknowledgments

I would like to express my deep appreciation to professor Mauro Conti and co-supervisor Pier Paolo Tricomi for their expert guidance and support during this project. It was a great pleasure to gain knowledge and inspiration from them, increasing my interest even more in cybersecurity. I am extremely grateful to my family, who always supported me and made all of this possible. Last but not least, I would like to thank all my friends, in particular to my three colleagues: Alberto, Francesco and Riccardo, who filled my days during this master's degree and made my experience abroad unforgettable.