Follow the beat

1. Introduction

You are dining in a restaurant, and once you finish your main course, the attendant appears by your side and suggest a dessert that fits your taste like he is reading your mind. The attendant may not be a clairvoyant, but they could have indicated that dessert by considering the items you selected for dinner or simply offering the most popular dessert.

We have known this kind of recommendation for a while; however, with the rise of e-commerce and streaming services, recommendation systems are part of our daily lives. You finish watching a movie, and a selection of films pops up on your screen, you include an item in the cart of your favorite online store, and related articles are presented to you, and this list keeps growing bigger.

To evaluate different approaches to the recommendation systems, I created a model to suggest new songs based on an existing playlist. Then, using Spotify users' playlists, I took a part of the playlist's songs, recommended the remaining songs, and checked if the recommendations matched the songs on the playlist initially.

Recommendation systems

A recommendation system suggests a new item based on data about the product, services, or the clients themselves. For example, we have collaborative filtering, which models user preference based on past interactions. (https://towardsdatascience.com/neural-collaborative-filtering-96cef1009401)

It was considered the random model where random songs were selected with the same probability and 4 variants of the collaborative filtering model with selection criteria based on:

1. **Track occurrence frequency**: the number of times that a given track appeared in the data set will set the probability of that track being selected
2. **Artists co-occurrence**: artists present in the same playlist; for example, artist A is in playlists 1 and 2. Now we have a playlist that we will provide a list of tracks, and this playlist also has artist A. Then we will select randomly one track from playlists 1 or 2.
3. **Albums co-occurrence**: the same as artists, but considering the albums' co-occurrence
4. **Tracks co-occurrence**: the same for tracks

I assumed the playlist's tracks as the favorite songs of a user, then, if they were suggested, the user would like the suggestion, and it could be considered a success.

On the other hand, I did not include any special treatment for cold starts, e.g., the item's first appearance. In this case, a random song was selected.

Data source

The data for the model was acquired using the API Spotipy (https://spotipy.readthedocs.io/en/2.19.0/ https://developer.spotify.com/documentation/web-api/)

The information is hierarchical based on:

- User ID
    - Playlist ID
        - Tracks
            - Track ID
            - Track name
            - Artists

    - Artists ID
    - Artists names
- Album
    - Album ID
    - Album name

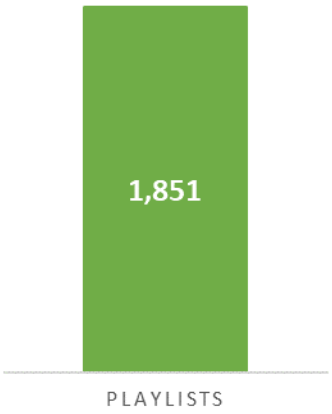
Stakeholders

The subject of this study is relevant to:

- streaming services suggesting the right content is fundamental to keep their users engaged
- e-commerce the model can provide insights on how to improve their suggestions and improve the customer journey
- business any executive that wants to guide their company to success can take advantage of knowing how to provide the exemplary service or product to their clients
- consumers knowing how the companies make use of their data of goods and services usage, and behaviors can instruct them to be aware of these practices and can save them from impulsive behavior of consumption

2. Exploratory Data Analysis

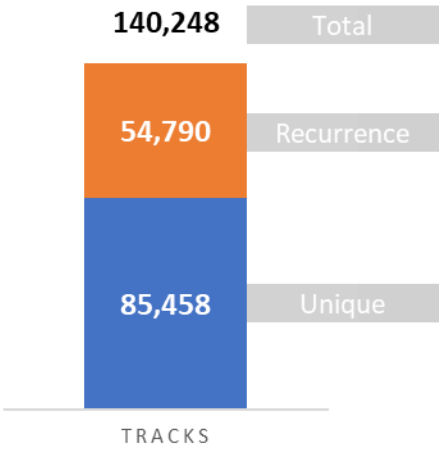
Playlists



- Tracks per playlist:
    - Max: 100 (Totaling: 48% of all playlists)
    - Mean: 75.8
    - Median: 98.0

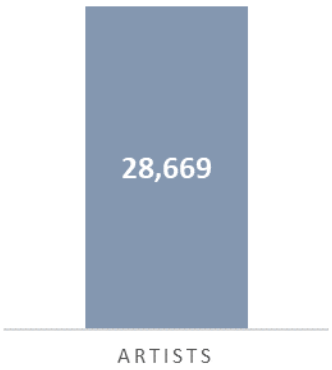
- 4% (75) with 20 or less tracks
- Artists/playlist: 15.5
- Albums/playlist: 33.5

Tracks



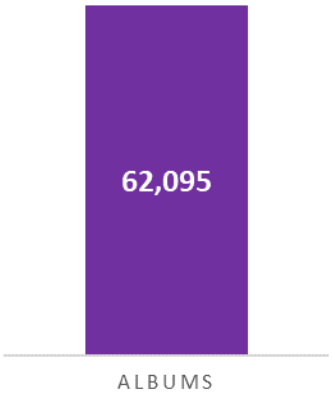
- 67,864 tracks (79.4%) appeared only one time
- Most popular track:
    - "Invisible" from the artists Andra and Lil Eddie - appeared 92 times
- On average, each track appeared 1.6 times

Artist



- 15,609 artists (54.4%) appeared only one time
- "Various Artists" is the most frequent
- Drake is present in 877 playlists
- Albums/Artist: 2.2

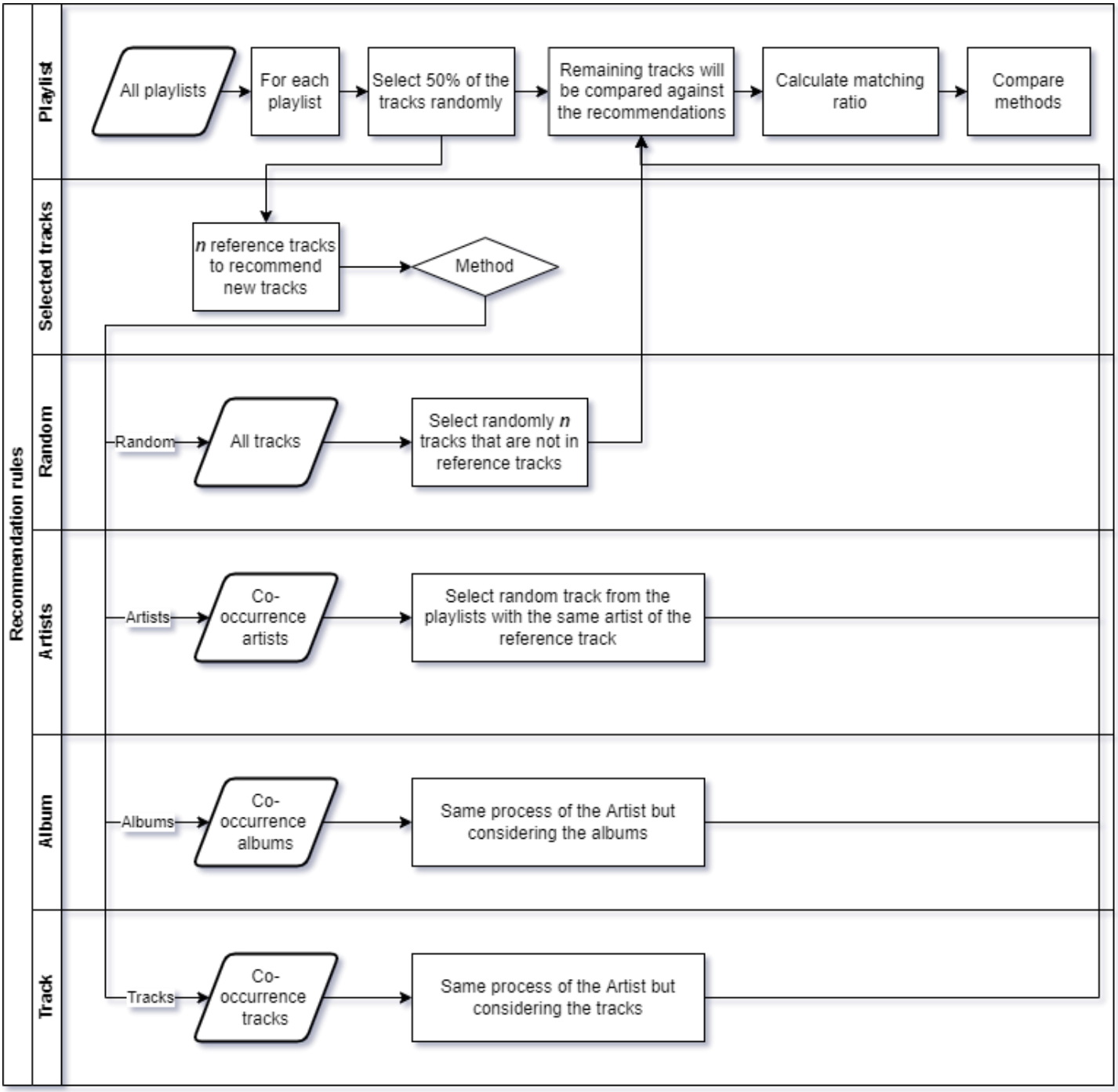
Albums



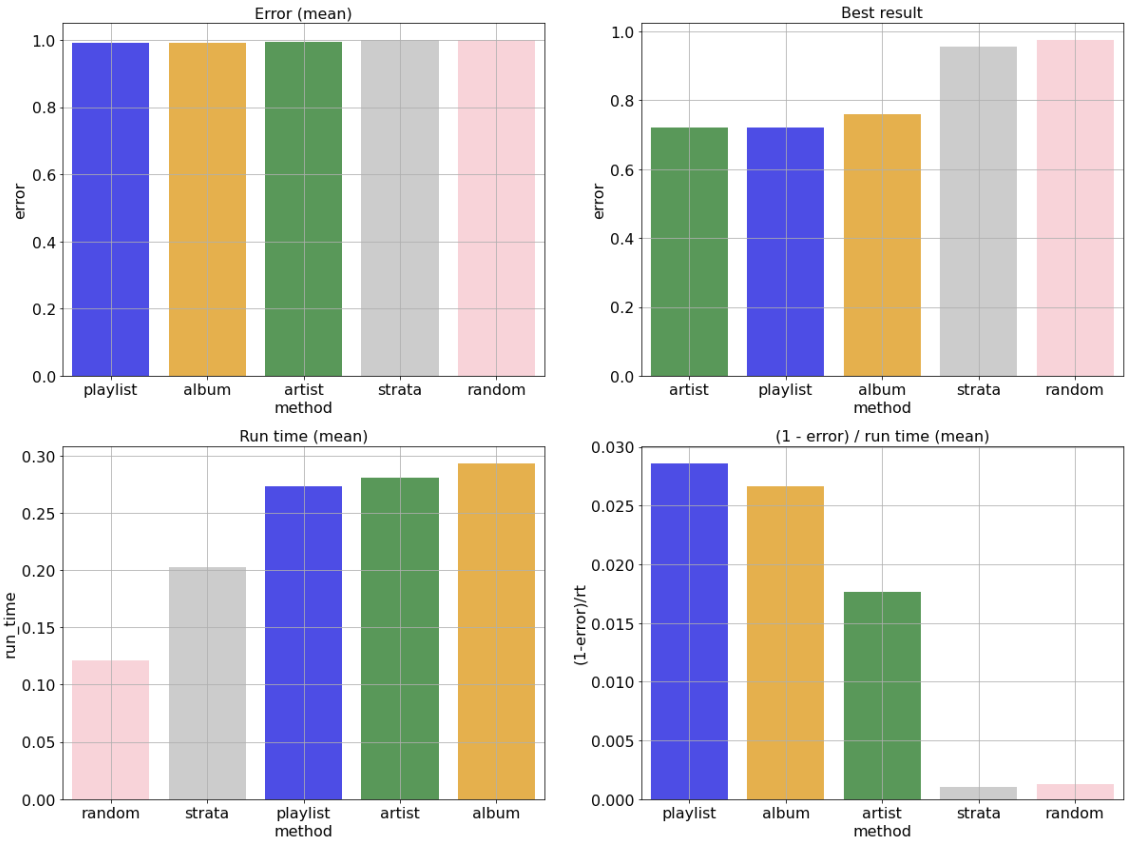
- 43,355 albums (69.8%) appeared only one time
- Album Beerbongs & Bentleys appeared 250 times

3. Modeling

The model considers only one dimension at each time. Then given that dimension, it will select a random song from the co-occurrence of songs for the reference song.


Recommendation rules
Playlist
All playlists
For each playlist
Select 50% of the tracks randomly
Remaining tracks will be compared against the recommendations
Calculate matching ratio
Compare methods
Selected tracks
n reference tracks to recommend new tracks
Method
Random
Random
All tracks
Select randomly n tracks that are not in reference tracks
Artists
Artists
Co-occurrence artists
Select random track from the playlists with the same artist of the reference track
Album
Albums
Co-occurrence albums
Same process of the Artist but considering the albums
Track
Tracks
Co-occurrence tracks
Same process of the Artist but considering the tracks

Model Performance



The recommendation systems using the playlist and the album to suggest a new track tied in the lead with an error of 99.13% and 99.16%, respectively.

The models using the playlist and the artist achieved the best result for a single run with 72% of error.

Run time and performance

The random model is the fastest; however, the other models are not so complex, and the time for each run is short.

Even being slower than the random model, the playlist-based model had the best accuracy per unit of time to run.

More details can be found in the summary table below.

Method	Fraction	Error	Run time	(1-error) / run time
album	0.25	0.98	0.33	0.04
	0.50	0.99	0.27	0.03
	0.75	1.00	0.28	0.01
artist	0.25	0.99	0.32	0.03
	0.50	1.00	0.25	0.02
	0.75	1.00	0.27	0.00
playlist	0.25	0.98	0.32	0.05
	0.50	0.99	0.25	0.03
	0.75	1.00	0.26	0.01
random	0.25	1.00	0.13	0.00
	0.50	1.00	0.12	0.00
	0.75	1.00	0.12	0.00
strata	0.25	1.00	0.21	0.00
	0.50	1.00	0.20	0.00

	0.75	1.00	0.20	0.00

Table 1 – Model performance for the initial fraction of tracks available and method.

4. Conclusion

The performance of the models is far from good, but they showcase that this is a direction that could be explored by using users' knowledge by selecting their playlists and the songs' features.

Modeling

A recommendation system is not a minimal problem. It has multiple dimensions that can be explored and the models proposed here are pretty simple, just considering one feature at each time. The only "ranking" criteria adopted was the recurrence of a given track.

Data

The data extraction is slow, and data acquisition is space-consuming. However, more data could improve the model performance.

Conclusion

Possible approaches to improving the models' performance include ranking and combining multiple features. But we must keep in mind that this will increase the time required to train the model and generate the results.

Then we must tune our decisions based on the requirements of a real-time solution or something else that can be prepared off-line. For example, we could consider solutions with a longer processing time if we create pre-loaded selection lists.