CSCI S-109a Introduction to Data Science
Pavlos Protopapas and Kevin Rader

Scope of Work for Data Science Final Project

Prepared by Group #50
Nils Fredrik Karlsson Peraldi, ninfendo@gmail.com
John Daciuk, johndaciuk@gmail.com
Ralph Aurel Tigoumo Ngoudjou, ralphwantek@gmail.com

Project Statement and Background

Spotify is a music streaming service which has over 40 million songs. Each of the songs can be combined into a Playlist by either Editorial staff or Spotify Users; algorithms; or a combination of both algorithms and Editorial (a term coined Algotorial). Currently, the service has more than 2 Billion Playlists, and their Playlist functionality has become extremely popular among users, as it's an effective way of discovering songs they like.

The company is therefore striving to improve its Recommendation System to predict songs which can be good candidates for playlists, which can improve the scalability of their current Recommendation System. In order to achieve this goal, the company created a challenge in which it released the Million Playlist Dataset (MPD), a Dataset containing 1,000,000 playlists created by users on the Spotify platform. The aim of the participants in the challenge will be to develop a system for the task of automatic playlist continuation. Details are below.

Goals: The main goal of this project is to develop a system for the task of automatic playlist continuation. The MPD dataset will provide a set of playlist features, and our recommendation system shall generate a list of recommended tracks that can be added to that playlist, thereby 'continuing' the playlist. The task is defined formally as follows:

Input

A user-created playlist, represented by:

- *Playlist metadata* which includes attributes such as the name, description, last date modified, etc.
- *K* seed tracks: a list of the *K* tracks in the playlist, where K can equal 0, 1, 5, 10, 25, or 100

Output

- A list of 500 recommended candidate tracks, ordered by relevance in decreasing order.

Given the input specifications above, our system should be able to deal with playlists which has no song, a problem known as "Cold Start".

We'll use the ground truths provided in the dataset in order to accurately assess the performance of our system. We'll then need to calculate metrics to see how well our models are doing. To be in line with the challenge Spotify designed, we'll use the metrics they suggested, which are **R-Precision, Normalized Discounted Cumulative Gain (NDCG), Recommended Songs Clicks,** and **Rank Aggregation** using the Borda Count election strategy. We included Literature Review for the metrics related to such similar tasks below in the Literature Review section, together with some potential models we believe might work on this task.

Questions

Like most high quality Data Science Projects, we intend to ask and answer some questions using the Spotify data provided. Some of these are listed below, but we're not restricting ourselves to these questions, and we'll try to come up with more as we move forward with the project:

1. Does the MPD contain enough relational information alone to render our other datasets superfluous?

2. To what degree does the popularity of a song predict one's enjoyment or level of engagement? Are there as many playlists composed of 50% popular songs as there are playlists composed of 90% popular songs?

3. Would people prefer to listen to what is familiar or experience something new? What is the right balance? When people do listen to new music, what must it have in common with what they already know?

4. Is genre or time period a better predictor in matching songs to a playlist?

Literature Review

Evaluation in Information Retrieval and A Review on Evaluation Metrics for Data Classification Evaluations.

We're including the above two papers for review because we believe that this task is very similar to an information retrieval task, where given a query (features extract from a Playlist), we retrieve documents (audio songs). Reading through the above book chapter will enable us to understand more profoundly how such a system is evaluated, which will shed light on how our metrics should be computed.

Natural Language Processing in Information Retrieval and Neural Methods for Information Retrieval

These papers will assist us in choosing a suitable model for the problem at hand. Our primary plan is to vectorize the metadata word tokens of the songs, and even other information such as the time the song was produced, and combine these together, then classify these playlist-song pairs, in order to come up with a score of whether the playlist-song pair is a good one (1) or it's a bad one (0). We intend to use `sklearn` for this task, but in case we use a more complex model, we might use `keras` or `xgboost` libraries.

Available resources/data

There will be two sources of data:

- **Million Playlist Dataset** from Spotify (http://recsys-challenge.spotify.com/). It contains 5.4 GB of data and was created in 2018. It contains the following information for each playlist:
    - Playlist name
    - Indicator if the playlist is a collaboration
    - Date when it was last modified
    - Number of albums the songs are from
    - Number of songs in the playlist
    - Number of followers the playlist has
    - Number of edits
    - Playlist duration
    - Number of artists
    - Song information:
        - Artist
        - Track Uri
        - Artist Uri
        - Track Name
        - Album Uri
        - Duration
        - Album Name

  The Million Playlist Dataset will be the primary dataset we use in this problem. However, should we deem fit to do so, we'll include the Million Song Dataset into our analysis. The Million Song Dataset is described below.

- **Million Song Dataset** The Dataset contain over a million songs. We generated the dataset from the Last.fm API (https://www.last.fm/api). It has the following information for each song:

- Artist
- Title
- Timestamp
- Similars
- Tags

We'll try to explore the dataset in two routes:

1. We'll calculate some basic statistics and frequency counts in the dataset like the number of followers, number of tracks in a playlist, etc, and then we'll try to come up with basic visualizations of these. This will help us to answer intuitive questions about the dataset. Some of these questions have been listed out in the Questions section of this Scope of Work.
2. We'll vectorize all the features, and then come up with a high dimensional dataset. We'll reduce the dimensions using techniques like Principal Components Analysis, to enable visualization. These visualizations will help inform or guide us as to algorithm tuning and model selection.

Preliminary EDA

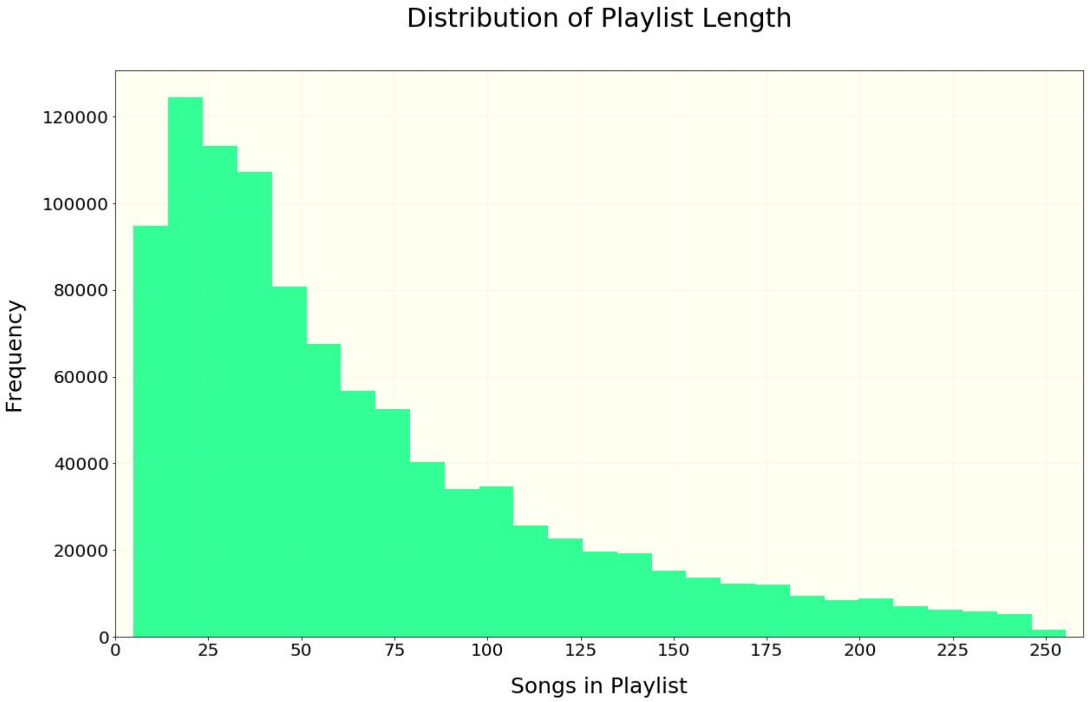
The data that we have is a trove of information. The MPD is over 5 gigs of json files of (mostly) random playlists that Spotify has grabbed from unnamed users. One of the principle challenges of this project will be training on such a quantity of categorical data with limited computing resources. We intend to take advantage of 'sklearn' to train models efficiently. Should 'sklearn' give us not so satisfactory metric performance, we'll try our hands on a Deep Learning library like 'keras'.

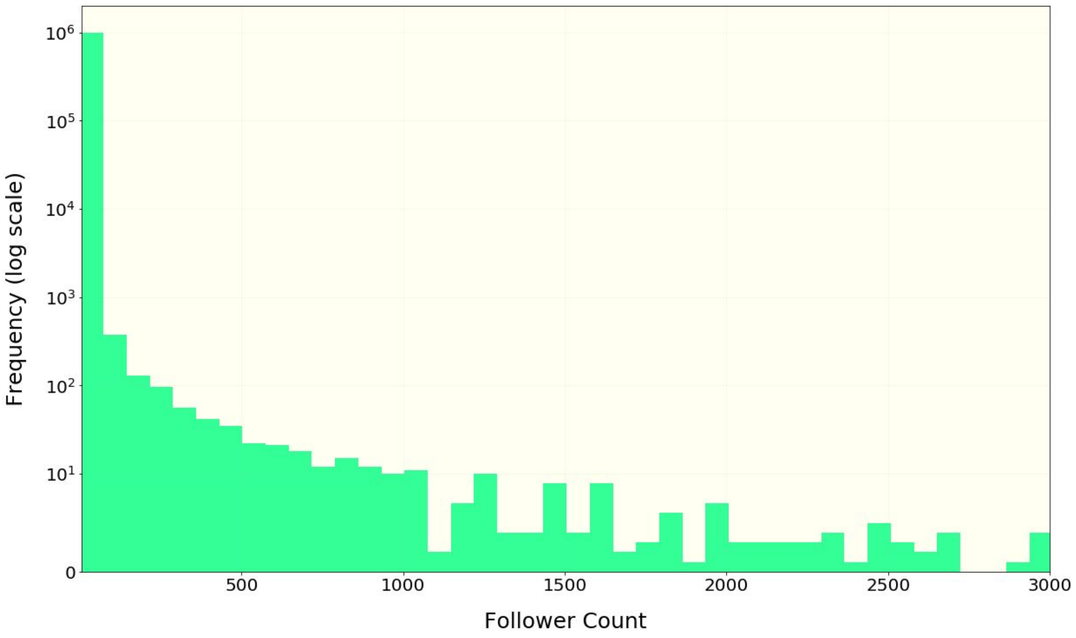
From the MPD, both the playlist length and playlist follower count histograms are approximately power law distributions. There are a handful of playlists with over 50,000 followers, but the large majority of playlists have only a few followers. The few playlists with many followers may be disproportionately useful to our recommender system. The playlists in the data set were selected by Spotify to have between 5 and 250 tracks. Many playlists naturally have about 25-50 tracks with 66 being the average track count.

Only 1.9% of the playlists have descriptions, but that still leaves almost 19,000 playlists with descriptions that will likely be helpful. On average, each song appears in about 30 playlists, and, across all playlists, each unique artists has about 8 unique songs.

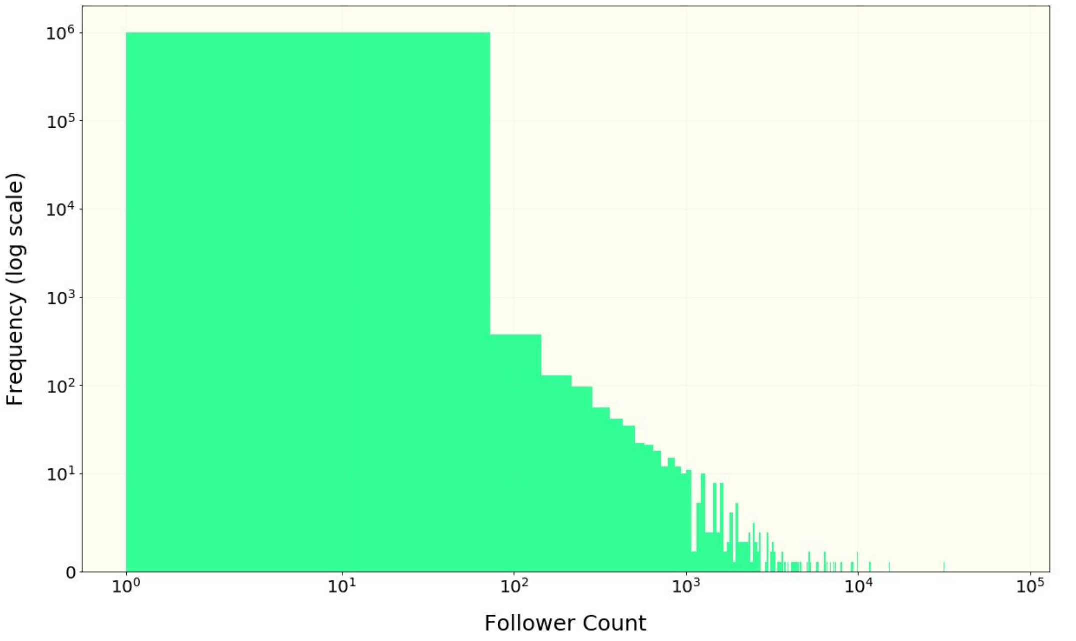
The most popular playlist titles are country, chill and rap; the most popular artist, by far, is Drake. Our preliminary EDA has produced the plots and chart heads below. We have produced pandas dataframes with the 250 most popular playlist titles, artists and song titles so that any algorithm we employ will have a convenient way to access various measurements of popularity.

We have found redundancy in the playlist dataset and song dataset since each song/ track in the playlist dataset also has meta-data. We've decided that the 'tags' in the Million Song Dataset may be useful; however, we plan to first create a sklearn logistic regression model using only the Million Playlist Dataset. We hypothesize that the MPD may be so rich in song relation information that a robust prediction model may be trained on it alone. Having access to song lyrics, we may be tempted to play with information, although we will try to not overcomplicate.

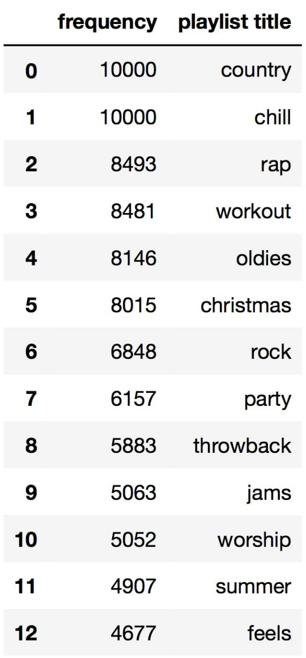




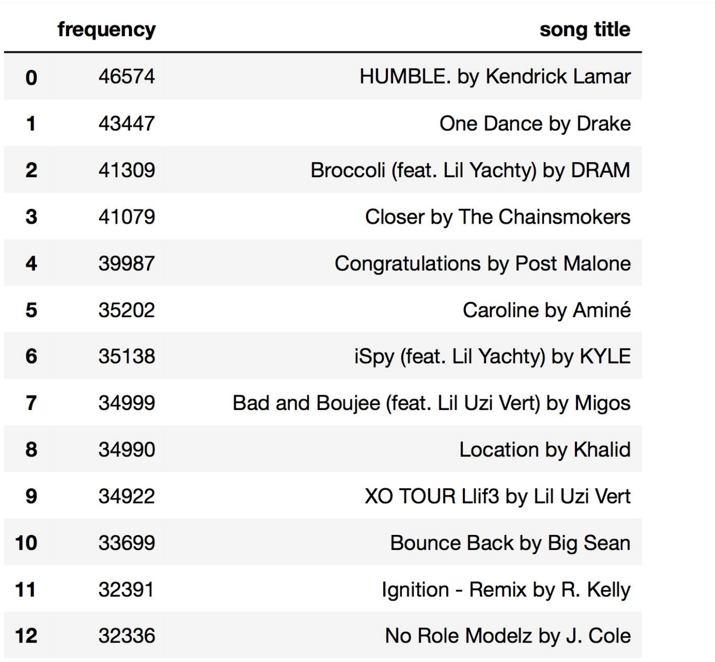
Playlist Follower Count Distribution (Linear X Axis)



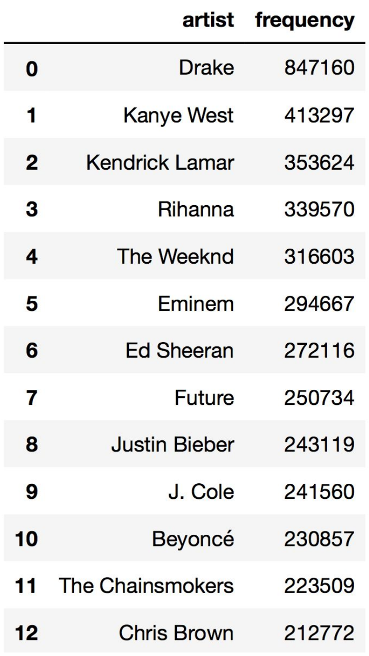
Playlist Follower Count Distribution (Logorithmic X Axis)



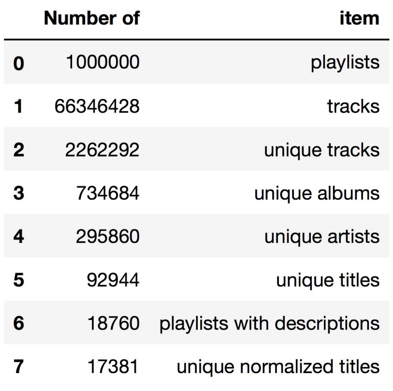
	frequency	playlist title
0	10000	country
1	10000	chill
2	8493	rap
3	8481	workout
4	8146	oldies
5	8015	christmas
6	6848	rock
7	6157	party
8	5883	throwback
9	5063	jams
10	5052	worship
11	4907	summer
12	4677	feels



	frequency	song title
0	46574	HUMBLE. by Kendrick Lamar
1	43447	One Dance by Drake
2	41309	Broccoli (feat. Lil Yachty) by DRAM
3	41079	Closer by The Chainsmokers
4	39987	Congratulations by Post Malone
5	35202	Caroline by Aminé
6	35138	iSpy (feat. Lil Yachty) by KYLE
7	34999	Bad and Boujee (feat. Lil Uzi Vert) by Migos
8	34990	Location by Khalid
9	34922	XO TOUR Llif3 by Lil Uzi Vert
10	33699	Bounce Back by Big Sean
11	32391	Ignition - Remix by R. Kelly
12	32336	No Role Modelz by J. Cole



	artist	frequency
0	Drake	847160
1	Kanye West	413297
2	Kendrick Lamar	353624
3	Rihanna	339570
4	The Weeknd	316603
5	Eminem	294667
6	Ed Sheeran	272116
7	Future	250734
8	Justin Bieber	243119
9	J. Cole	241560
10	Beyoncé	230857
11	The Chainsmokers	223509
12	Chris Brown	212772



	Number of	item
0	1000000	playlists
1	66346428	tracks
2	2262292	unique tracks
3	734684	unique albums
4	295860	unique artists
5	92944	unique titles
6	18760	playlists with descriptions
7	17381	unique normalized titles

Sources:

- https://recsys-challenge.spotify.com/
- https://recsys-challenge.spotify.com/rules