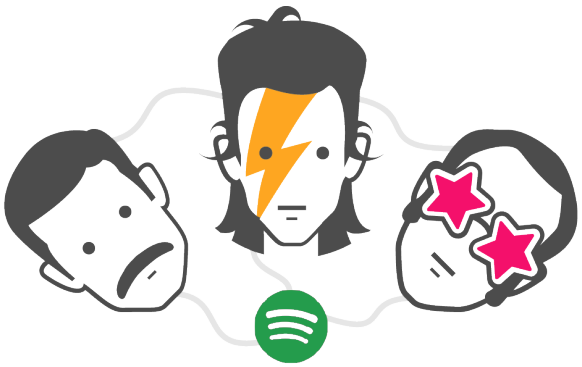
EE-558: A Network Tour of Data Science

Semester Project

January 18, 2019



FRIENDS WILL BE FRIENDS

A NETWORK TOUR OF MUSICAL FRIENDSHIP

Lecturers:

Frossard Pascal, Vandergheynst Pierre

Main Teaching Assistant:

Michaël Defferrard

Team 33:

Tobias Barblan
tobias.barblan@epfl.ch
Laura Bujouves
laura.bujouves@epfl.ch
Liana Mehrabyan
liana.mehrabyan@epfl.ch
Jeremy Wanner
jeremy.wanner@epfl.ch

1 Introduction

Music is universally created and enjoyed and has been connecting people through the ages. Today diversity in music industry is at its peak. It seems that music tastes across the globe vary to greater and greater extents. In this project, we have endeavored to analyze whether or not music still truly connects people and how one's musical taste is reflected in their friendships. Using network analysis tools, we want to understand if real friends have common music taste, and how this taste spreads through a network of friendships.
In a second part of the project, we used classification techniques to in an effort to predict how popular a music track will be, in order to explore the differences between different classification algorithms.

2 The Data

For the first part of the project, we used the Spotify API and Spotify python packages to scrape the *You Top Songs of 2018* playlist [1] information of 25 volunteer Spotify users, many of which are friends in real life. The set comprises a total of 1274 artists and 2207 songs and the data frame contains features describing the artist, genre, track duration and musical features such as acousticness, tempo, liveliness, danceability, key, loudness etc.

For the second part of the project, track popularity prediction, we used an audio downloads library called Free Music Archive (FMA). Along the usual meta data found in musical tracks (title, artist, genre, playback count, ...), a host of other features—provided by Echonest (now Spotify)—were available. The number of tracks in this data set is 13,129.

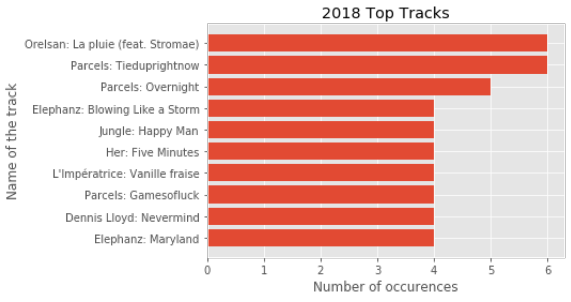


Figure 1: **Top Tracks:** Summary of the most frequently found titles in the playlists from the 25 users.

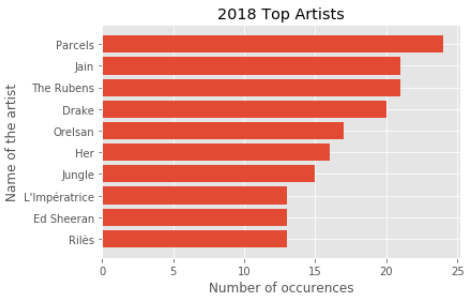


Figure 2: **Top Artists:** Summary of the most frequent common artists among the 25 user playlists.

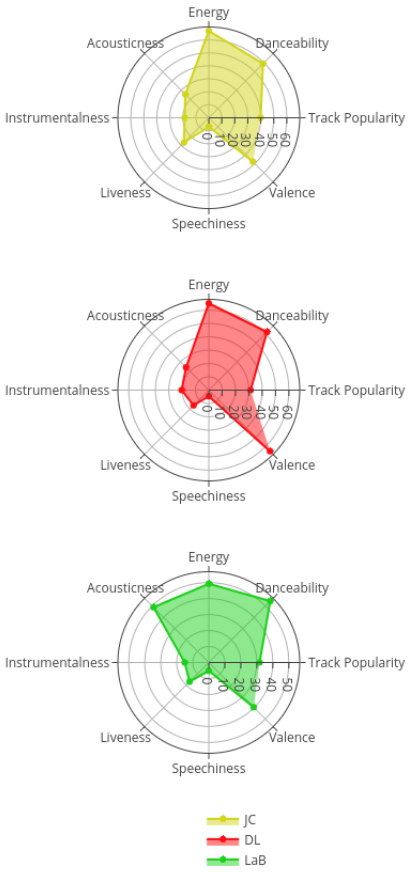


Figure 3: **Median percentage of songs features per user:** This type of visualization highlights the differences between the different users.

3 Exploratory Analysis

To investigate whether we had an interesting network to analyze from the Spotify data of our friends, we looked into similarities and divergences in specific features. In particular, we hoped to see recurrence of individual tracks and artists in different playlists. Figures 1 and 2 summarize most frequent tracks and artists when looking at the whole data of 25 user playlists. The chart of top tracks is lead by a duo of popular contemporary artists and followed by two tracks by the Australian group Parcels. This is followed by artists mainly from

independent pop-rock genres. As for the artists, Parcels is leading the ranking as expected. Different genres are present with rap and pop-rock being the two main genres which characterises the given population of users. This facts indeed demonstrate that there are both artists and tracks that are common to individuals in our data. Moreover, our friends frequently listen to several songs from each artist.

Next we visually compared the characteristics of several individuals overall listening styles 3.The characteristics we chose had been pre-calculated by Spotify to describe interesting aspect of the music such as acousticness, liveness etc.

We identified that we could see significant differences in the style of music that people listened to based on these characteristics. Summary of several users' preferences can be found in Figure 3. As an example, JC seems to be listening to rather energetic and dance music, DL prefers generally less popular tracks, valence in which are more important. LB, on the other hand, listens to tracks with more acoustic characteristics. This indicates that we had a music network that had enough diversity to analyze.

4 Network Analysis

4.1 The Songs Network

By loading all songs from the available playlists, keeping only one occurrence of each song, dropping all non-number-like meta data and then standardizing it[1], we were able to generate the songs network shown in figure 4.

Figure 4: Songs from the Spotify playlists, only the 13 most significant edges per node are shown (spring layout)

We can see from visual inspection that this network, although complicated-looking, is not random. We therefore ought to be able to find some interesting relationships in the data...

4.2 The Social Network

A first network was created based on the social connections between our friends. Friendships created an unweighted graph with 25 nodes and 51 edges, making up and average degree of 4.008. The distribution however is varied, with several nodes of very high degrees. As is seen in Figure 5, our network consists of a central cluster, surrounded by smaller clusters. The node JC (who initiated the data collection from a group of acquaintances who, in turn, conveyed the message) can be considered as a central node, Similarly, LB is an important node in data collection. The collection process explains why some nodes have a very low degree (JB) and others, more active, have a higher degree.

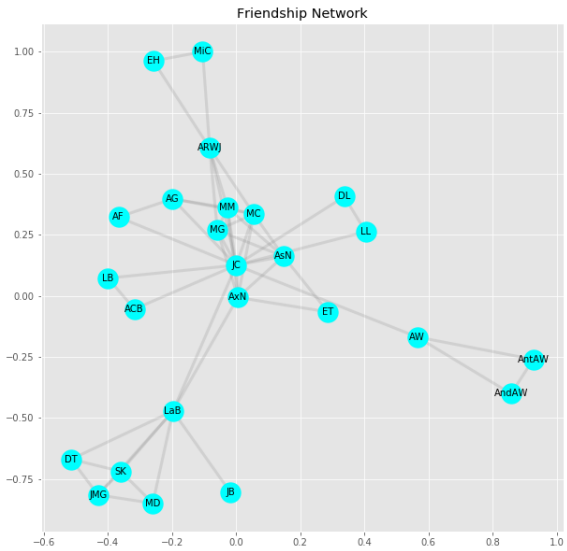


Figure 5: Network visualization of different relationships between users whose playlists have been recovered.

4.3 Defining a Network of Songs

In order to analyze the relationships between playlists, an adjacency matrix of music characteristics was constructed. These characteristics included:

Feature	Importance Score
Popularity	Danceability
Energy	Key
Loudness	Mode
Speechiness	Acousticness
Instrumentalness	Liveness
Valence	Tempo

Table 1: Characteristics included in adjacency matrix construction

[1]By standardize it is meant that we subtracted the mean and divided by the standard deviation on each feature.

These characteristics were then weighted in order to create a similarity matrix, which was subsequently translated into an adjacency matrix using euclidean distance.

4.4 The Musical Network

The first network of music that we visualize connects friends to common artists in their playlists. In this graph the black nodes represent artists, whereas the orange nodes represent the Spotify users (Fig. 6). As seen by clusters on the periphery of the graph, many of the artists are unique to an individual and reflect one's personal taste in a deeper way. The central nodes however are shared by several users which indicates that many artists are listened to by several users, as indicated during the descriptive analysis. This will be the basis for the work that will follow.

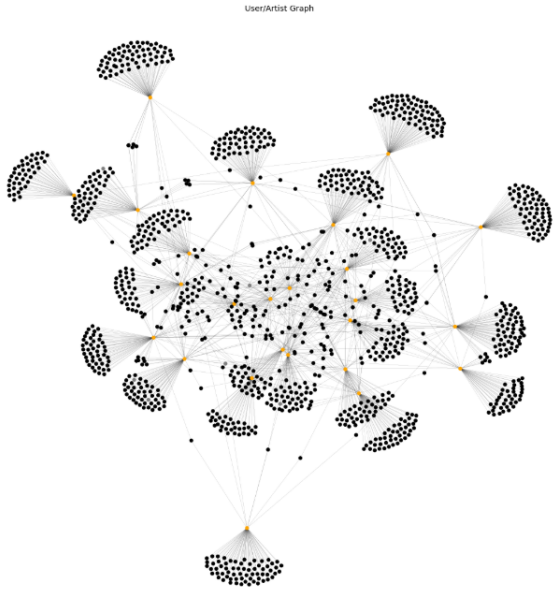


Figure 6: **The User / Artists graph:** Visual representation of the network of users (in *orange*) and artists who compose their playlists (in *grey*).

Figure 7 represents the relationship of users based on the number of common artists they listen to and displays interesting insights. As it can be seen, the strongest relationship exists between MM and MC, who share 25 artists on both their playlists. These two users are a couple, which certainly explains the musical compromises that emerge here. Some users like DL have few links with others. This may be due to the geographical and generational differences between a Swiss student and an thirty-year-old Australian.

Subsequently, we transformed the user/artist graph into a weighted graph that represents affinities between users based on the number of common (Fig. 7). This resulted in a graph with 25 nodes, 221 edges and an average of 17.6 degrees (Fig. 8). This average degree is much higher than the average degree of 4.0 found in the social network, which is logical given each user has a playlist of 100 songs.

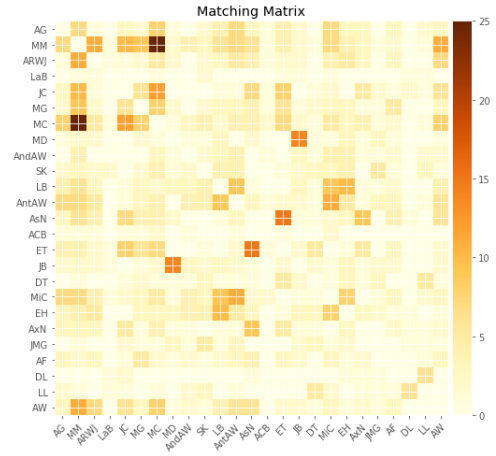


Figure 7: Visual representation of the adjacency matrix between the different users based on the shared artists. The darker the color, the more strongly the two users are connected.

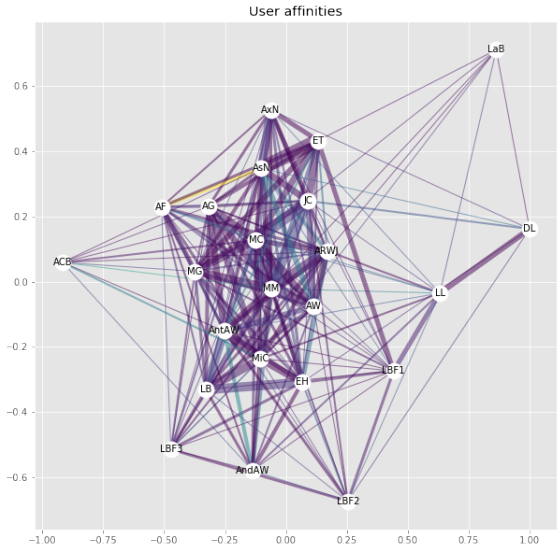


Figure 8: This network is the visualization of the relationships presented in the previous figure. The width of the lines connecting two users is proportional to the weight connecting them in the matrix. Here, each playlist has a maximum of a hundred artists, which explains why many users are connected by at least one of them. To make the graph more readable, a filter on the adjacency matrix can be applied to reduce the number of edges to the most consistants.

In order to compare the social and music networks, we represent the similarities using the NetworkX algorithm of betweenness, which is calculated based on the sum of fractions of all the shortest paths that pass through each node.In practical terms, a high betweenness would identify the person that you would go to if you wanted to be introduced to the people that you do not already know. The algorithm used is as follows:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest $(s,t) - paths$, and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s, t. If s = t, $\sigma(s,t) = 1$, and if $v \in s,t$, $\sigma(s,t|v) = 0$. The betweenness is represented below:

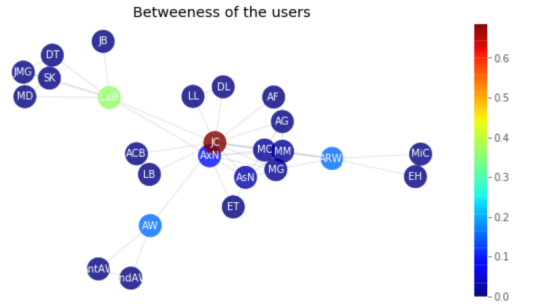


Figure 9: User's betweeness is an indication of its centrality in the network. In this case, it is the real network of friendship between users. The centrality of the nodes here emphasizes the importance of the user in the data collection process. The graph answers the question "By which user did I have access to this data? "

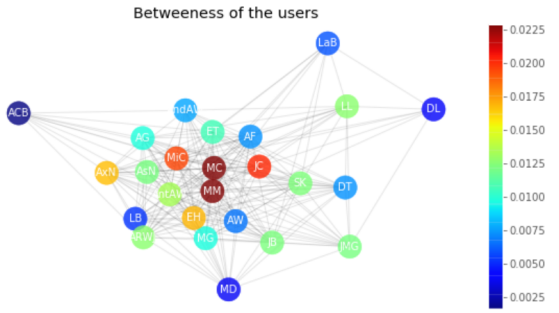


Figure 10: Visualization of betweeness in musical friendship graph.

It is clear that in the social network (Fig. 9), JC and LaB are the points of data collection, demonstrated by their higher betweenness than the rest of the graph. The relationships in the music friendship graph (Fig. 10) however transcend the social connections and are much more varied.

Next we want to determine if our Spotify users can be clustered into distinct groups. For this we do a community analysis to partition users. We have use the Louvain method with optimized based on modularity; a measure of the density of links inside communities as opposed to exterior to communities. The equation for partitioning is as follows:

$$Q = \frac{1}{2m} \sum ij \left[Aij - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where:

- A_{ij} represents the edge weight between nodes i and j
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively
- $2m$ is the sum of all of the edge weights in the graph
- c_i and c_j are the communities of the nodes
- δ is a simple delta function.

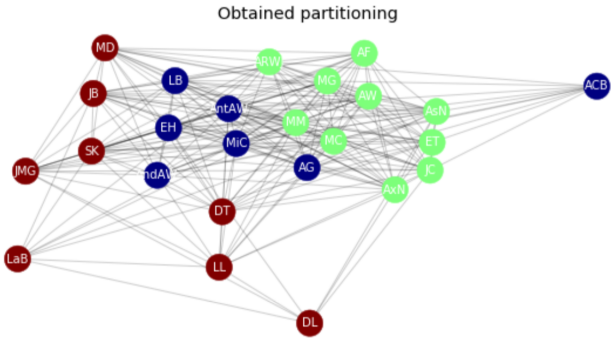


Figure 11: Partitioning based on the Louvain method: blue(non-EPFL users), green(EPFL users), red(users living outside Switzerland)

As seen in figure 11, the Louvain method resulted in three distinct clusters. We can see that partitioning based on music listening has well reflected social groups or other listener characteristics such as nationality. The blue cluster is composed mainly of non-EPFL users, with a few links with the main connected friends from the real friendship adjacency. The green cluster contains a core group of EPFL students and their close friends. The third cluster is composed of foreign students or friends that live or come from far away from Switzerland. Differences in music culture, ages or simply the fact that these users have only few social connections to the central cluster explain this. It is really interesting to see that without knowing anything about real-life relations, music tastes can lead to clusters that represent real social connections.

5 Song Popularity Prediction

Popularity of a track seems to be something unpredictable, as one can never know what song is going to be the next big hit. But with the right tools and the right data, a data scientist can predict whatever is necessary. Being able to estimate the success of a track based on its audio features can be a useful tool for music producers and artists. That is why, we focus on developing predictive models as well as exploring what features play an important role in a track's play counts.

After preprocessing the data to get it in a right format and removing some extreme outliers using visualization tools as well as the IQR rule, we proceeded to analyzing feature relationships between our variable of interest—play counts, and other variables. Observing Zipf's law in the distribution

of play counts (Fig. 12), it was decided to transform the target variable into a categorical one. [2] To that end, 'popularity' scores on a scale of 1 to 5 were assigned to each track based on the quantile interval that a track's number of plays falls into. As a result, a classification task was created. Feature interpretation was as important as performance in this task, that is why Random Forest classifier was used to asses which features have the most importance in prediction. [4] Table 2 presents the five most important features deduced by the Random Forest classifier.

Feature	Importance Score
Interest	0.368397
Acousticness	0.048640
Artist hotness	0.047152
Track duration	0.046834
Artist familiarity	0.046609

Table 2: Feature Importance for Play Counts Score

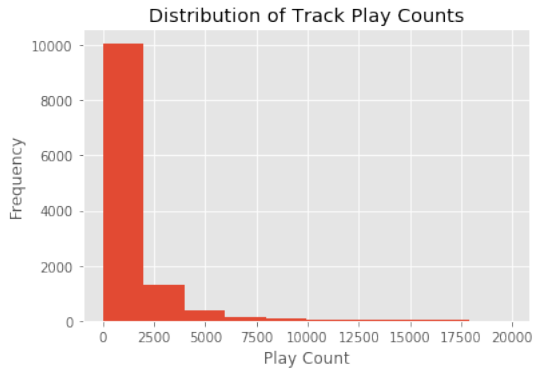


Figure 12: Distribution of Track play counts which resembles Zipf's law- a common phenomenon in social sciences.

A classifier with top important features was then trained on the training data which resulted in 80% accuracy while tested on the 30% reserved testing data set.

It was also interesting to assess the performance of KNN algorithm to see whether predicting a track's play counts based on 'similar' tracks will have good performance. To that end, a ten fold cross validation was performed to identify optimal number of neighbors to look at. $k = 6$ was chosen as an optimal parameter and the algorithm was tested on the testing data set. An accuracy of 76.4% was observed which did not manage to out stand the performance of Random Forest classifier. Finally, we ran a convolutional neural network (CNN) on the graph using the methodology and functions from "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering"[4].This did not improve upon the Random Forest classifer (accuracy of 51.9%).

6 Further Work and Improvements

One limit that we often attained in our work was the computational power that we had available. The convolutional neural net took about half an hour to run on just 0.1% of the reduced-feature FMA data set; by having more powerful computers we wouldn't need to sub-sample by that amount and we would most likely get better results.

Another important improvement vector would be to make use of the massive amount of data—about 500 number-like features—provided by the *librosa* music and audio processing library.

Further potential analysis ideas are, for instance:

- Start with one track and see how many artist hops are needed to reach another friend on Spotify (Stanley Milgram's small-world experiment)
- Reduce the number of favourite tracks per user from 100 to 10 and see how the new results compare
- Generate new meta data using Natural Language Processing on genre tags and incorporate it in the learning algorithms described above

7 References

[1] Spotify 'Top 100 Songs of 2018' playlist taken from spotifywrapped.com (2018)

[2] Newman, Mark EJ, Power laws, Pareto distributions and Zipf's law. Journal of Contemporary physics vol. 46 no. 5, pp. 323–351 (2005)

[3] Klusowski Jason M., Complete Analysis of a Random Forest Model, arXiv:1805.02587v5 (2018)

[4] Defferrard M., Bresson X., Vandergheynst P., Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, arXiv:1606.09375 (2017)