Exploring music preferences: an analysis involving music, data, and machine learning

Taste in music is one of the most common ways that people define themselves. Whether by wearing a t-shirt of a favorite band, including favorite music genres on an online dating profile, or posting an Instagram photo from a concert, musical choice is strongly tied to personality and self-expression. For these reasons, people can be quite adamant about their own musical preferences. My girlfriend and I fall into this category. Over the course of our relationship, we have had several debates about the similaritites/differences in our individual music taste. I usually claim that we listen to very different music (specifically that she listens to more "pop" music) while she usually claims that our music taste is fairly similar. About a month ago, I started wondering if there was any quantitative way to assess these claims. I searched the web and found that the popular music streaming service Spotify allows users to access a treasure trove of music data that enables this type of investigation. I quickly made a Spotify developer account and started exploring our Spotify playlists.

In this article, I will share the results of my analysis, which is divided into two parts. The first part focuses on comparing the music on our Spotify playlists by utilizing audio features and genre data that can be extracted from the Spotify API. The second part focuses on using the Spotify data to train a machine-learning model that predicts whether a song is more suitable for her playlist or mine. The model can be used to build individual and shared Spotify playlists.

Tools & Data

The main tool used in this project is the Spotify API service, which can be used to extract the audio features of songs in Spotify. The library Spotipy was used to interface with the Spotify API in Python. All of the data analysis, machine learning, and plotting was done in Python with numpy, pandas, scikit-learn, matplotlib, and seaborn modules.

The music data was obtained using a custom built script that utilizes Spotipy functions to collect all of the songs in a playlist and fetch the audio features of each song. The resulting dataset has 18 columns and 1688 songs, of which 1370 come from my own playlists (Stefan) and 318 come from my girlfriend's (referred to as *Kelsey*, *she* or *her*). The 18 columns contain information about the identity of the song, such as name, artist(s), genre, and release date, as well as the audio features that describe the song. The following audio features were utilized in this analysis:

- Acousticness: a confidence measure of whether the track is acoustic
- Danceability: describes how suitable a track is for dancing
- Duration: the duration of the track in seconds
- Energy: represents a perceptual measure of the intensity and activity of a track

- Instrumentalness: predicts whether a track has no vocals
- Liveness: detects whether a track was played live
- Loudness: the overall loudness of a track in decibels
- Speechiness: detects the presence of spoken words in a track.
- Tempo: the estimated tempo of a track in beats per minute
- Valence: describes the musical positiveness conveyed by a track
- Popularity: describes the popularity of a track

All of the audio features are represented on a scale from 0 to 1, except for loudness (-60 to 0 dB), duration (any number), and tempo (any number).

Is our music taste different?

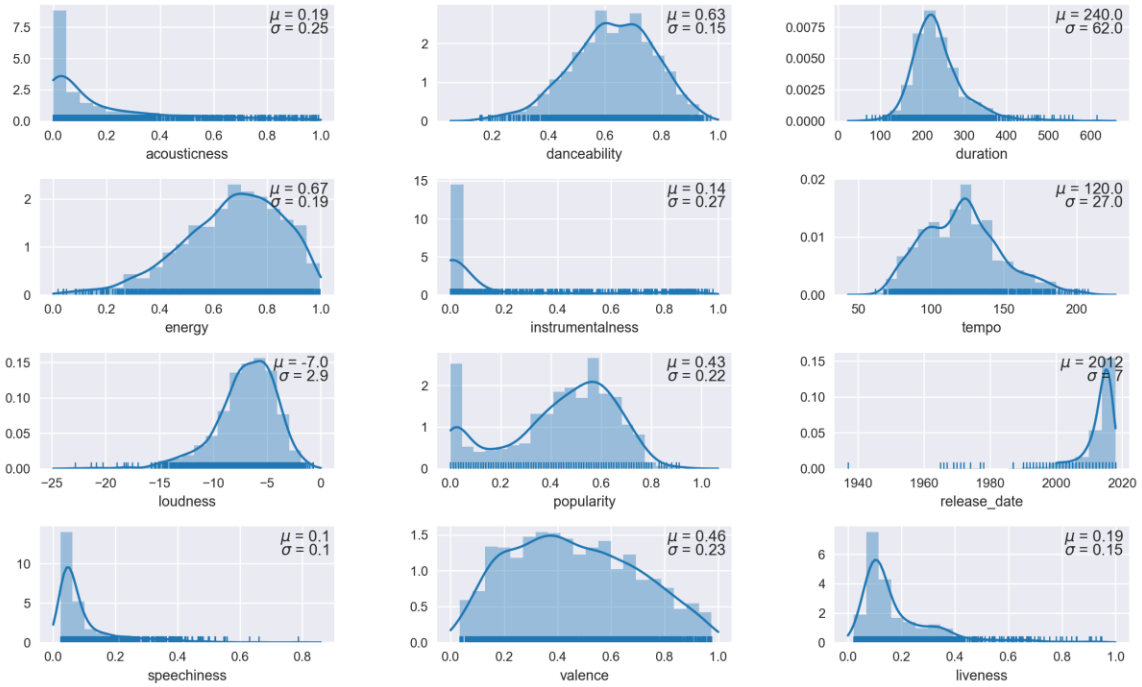
The starting hypothesis of this investigation is that that there are no differences between the music Kelsey and I listen to. In order to assess this hypothesis, I use the Spotify API to scrape the audio features of songs in each person's Spotify playlists and compare the distributions. We'll start by first evaluating the audio features of my own playlist in order to get a sense of what the distributions look like and then compare my distributions with hers. I'll then compare the distribution of music genres in our playlists.

1. Audio Features

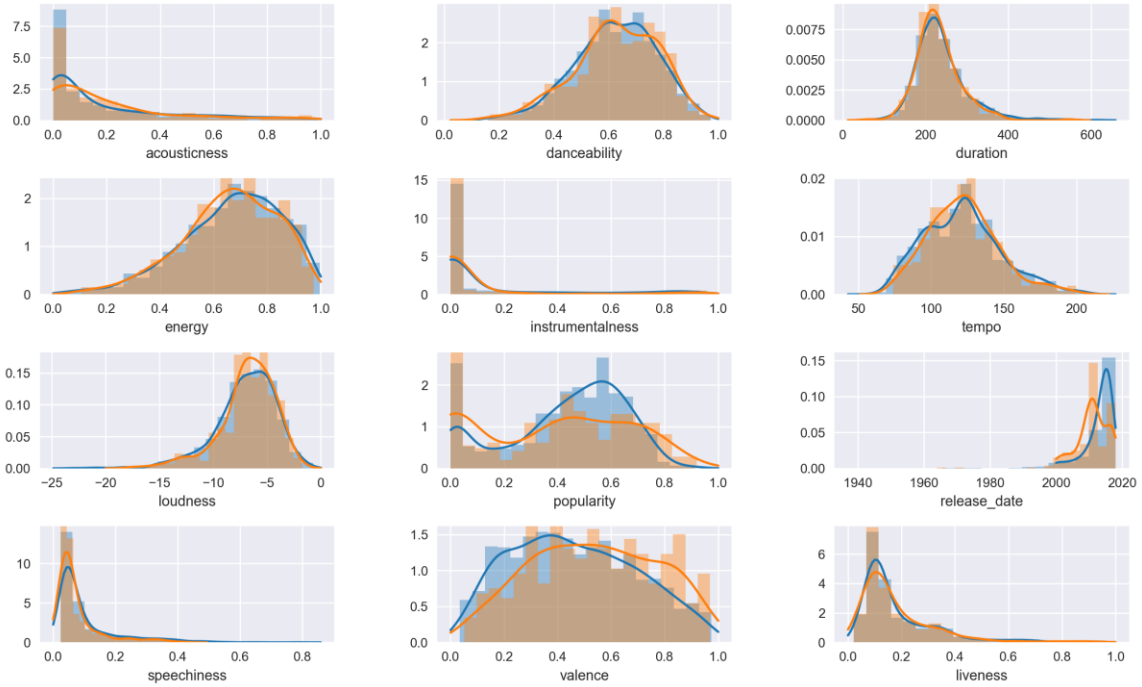
The first figure on the next page illustrates the distribution of audio features in my Spotify playlists and shows the mean values and standard deviations of each feature in the upper right-hand corner. Each subplot depicts the probability distribution function of a specific feature, which gives an indication of the relative fraction of songs within my playlist that have a specific value of that feature. With the range of possible values of each feature in mind, the plots reveal that, in general, **the songs I like tend to be danceable**, **energetic**, and **loud**. Additionally, most of my music is **relatively new** (release date after 2012) and **not incredibly popular**. Lastly, the standard deviations reveal that the distributions are relatively wide, which will become important when we compare distributions.

A similar analysis can be done for my Kelsey's playlist. The second figure on the next page illustrates the distribution of audio features in her playlist (orange) overlaid onto mine (blue). Very quickly we recognize that **the distribution of her audio features are very similar to mine.**

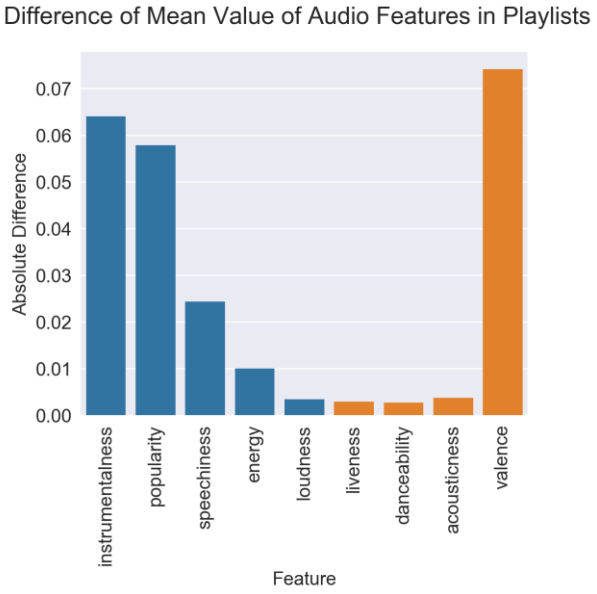
Distribution of Audio Features in Stefan's Playlists



Distribution of Audio Features in Stefan's and Kelsey's Playlists



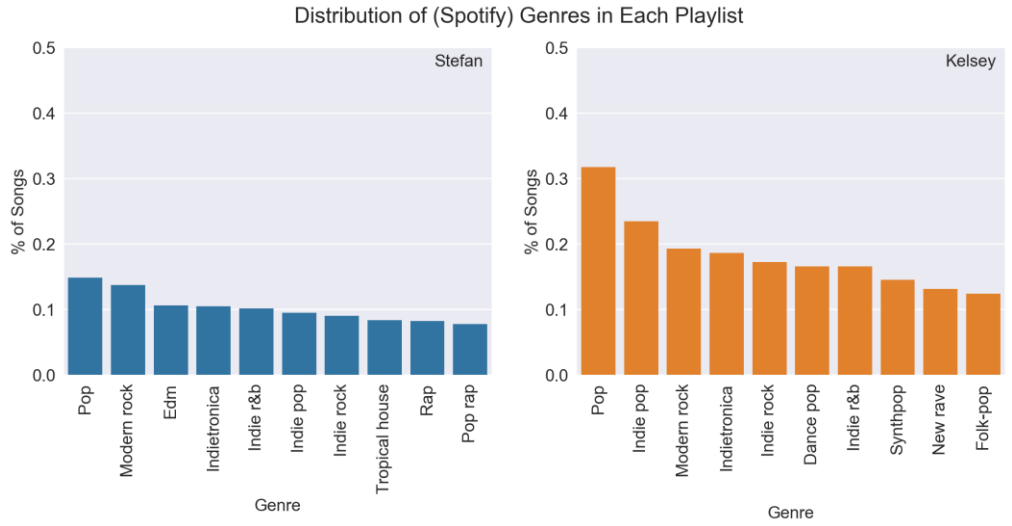
In order to determine the main distinctions, we can look at the difference between the mean values of each feature. The figure below presents this data, where the color blue implies a larger mean value for my own playlist and the color orange implies a larger mean value for her playlist.



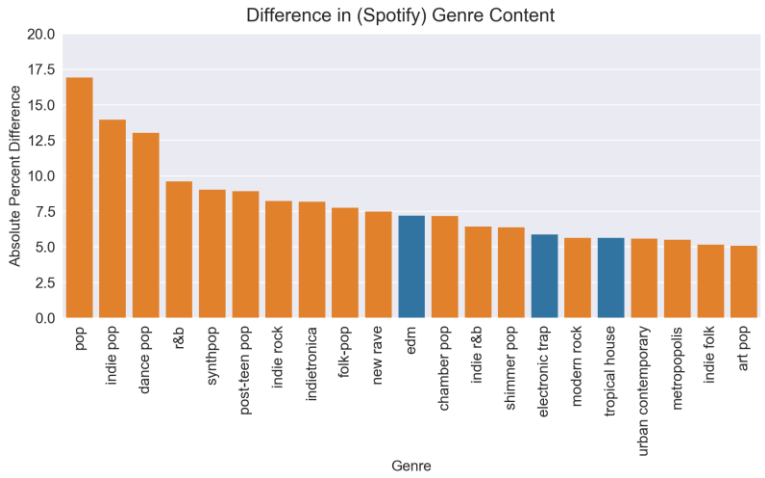
Compared to my music, Kelsey's **playlist is less instrumental**, less **popular**, and **more positive** (higher valence). However, the difference in the mean values is quite small and, as mentioned in the previous paragraphs, the standard deviations are quite large (around 0.2 for features having a range of 0-1), suggesting that there is not really any significant difference. **Any statistical analysis suggests that our distributions are (statistically) the same**. In order to understand how similar our audio feature distributions actually are, we can calculate the amount of overlap between her and my audio feature distributions. To do this, we sum the amount of overlap for each individual feature and then divide by the total number of features, 12. This similarity metric has a minimum value of 0, i.e. no overlap and no similarity, and a maximum value of 1, i.e. complete overlap and a high degree of similarity. For our playlists the similarity is 0.84, suggesting that **in terms of the audio features extracted by Spotify, the music we listen to is quite similar.**

2. Genre

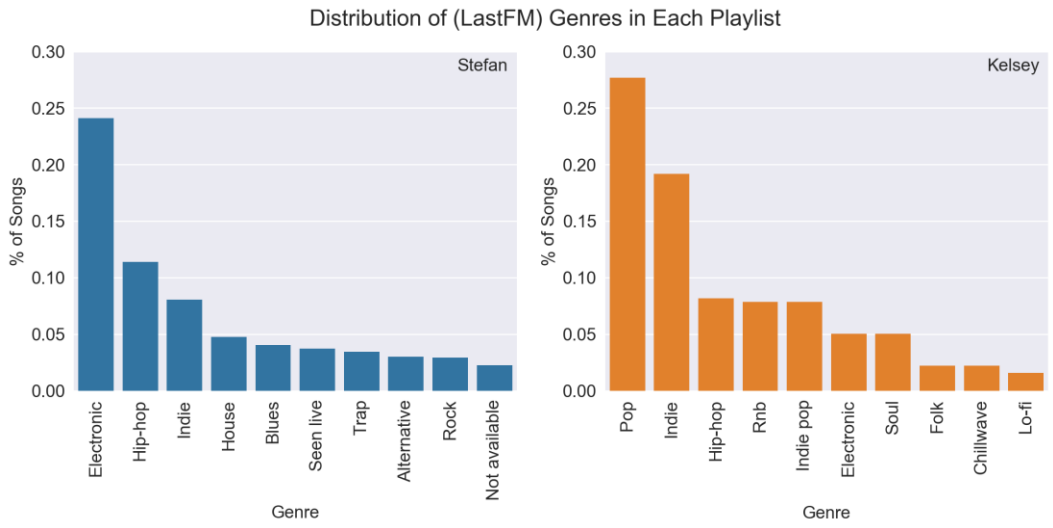
In order to further investigate our playlists, I used the Spotify API to collect the genre(s) of each song in our playlists. To determine if this reveals additional contrast between our music profiles, we can analyze the distribution of genres in each playlist. The figure below illustrates the top 10 genres found in each playlist:



It's important to note here that songs can be tagged with multiple genres and so the sum of the bar heights does not equal 1. Both playlists show a number of the same top genres, namely pop, indie, rock, electronic, r&b, and various combinations thereof (*modern* rock, *indie* pop, *indie* r&b), but the relative fraction of songs tagged with each genre is significantly different. To highlight these dissimilarities, we can calculate the difference between the fraction of songs within her playlist and my playlist that belong to a certain genre. The figure below illustrates the highest differences between our genre content. Blue signifies that my own playlist had a higher fraction of songs within a particular genre and orange signifies that her playlist had a higher fraction.



According to the figures, **Kelsey listens to more pop**, **indie**, and **r&b**, **whereas I listen to more electronic (edm** and **house**). Not shown in these figures is the number of unique genres identified in our playlists: 358 (Stefan) vs. 161 (Kelsey), suggesting that **my playlist spans more genres and is therefore more varied**. In order to corroborate these results, I also scraped genre data from a secondary music database, LastFM. It is important to point out that the way genre data is generated in the LastFM database is different than the way it is generated in Spotify. In the LastFM database, users manually assigning genre "tags" to each artist/song, whereas in the Spotify database a machine-learning algorithm tags each artist/song. The figure below illustrates the results:



The data reveal similar differences between the genre content of my playlist and hers, namely that she listens to more pop, indie, and r&b, whereas I listen to more electronic music. The LastFM genre has the nice feature that many of the small distinctions between genre are not as prevalent as in the Spotify data (e.g. there are no indietronica or indie r&b genres, just indie, electronic, and r&b), which amplifies the differences in genre content. This allows us to gain a better quantitative understanding of our genre differences: she listens to 14 times more pop, 9 times more indie pop, and 7 times more r&b than me, and I listen to 5 times more electronic than her.

To determine if these genre differences are statistically significant, we can model the fraction of songs belonging to a certain genre as a binomial random variable. For example, let's say we are interested in assessing whether or not Kelsey listens to more pop music. We consider that each song analyzed either belongs to the pop genre or it does not. The fraction of songs that Kelsey listens to that are pop, $p_{pop,Kelsey}$, is estimated by her Spotify playlist as 0.27, while $p_{pop,Stefan}$ is estimated by my own playlist as 0.02. It's important to understand that these numbers represent estimates of the true fraction of songs that we each listen to that are pop (i.e. our Spotify playlists represent a small sample of all of the music that we listen to). You

can imagine that if she and I built a second playlist of music we like, the fraction of songs that are pop would be slightly different. A statistical quantity called the *standard error* helps us determine how large this deviation would be playlist-to-playlist and can be calculated using the estimated fraction of songs within a playlist that are pop and the total number of songs in the playlist. Without explicitly detailing the math, the true value of $p_{pop,Kelsey}$ is 0.27 +/- 0.025 within a 95% confidence interval while $p_{pop,Stefan}$ is 0.02 +/- 0.004 within a 95% confidence interval. Therefore, we can be relatively sure that Kelsey listens to significantly more pop. Similar calculations confirm that, statistically, she listens to more indie and r&b while I listen to more electronic music. However, there is no statistical difference between the amounts of hip-hop we listen to. The key takeaway is that although the Spotify audio features did not reveal significant differences between our music taste, **the music we listen to shows substantial difference in terms of genre.**

There are three key outcomes from the preceding analysis in regards to the initial hypothesis that there are no differences between the music Kelsey and I listen to:

1. The distribution of audio features in Kelsey's and my playlist are statistically identical.
2. The distributions of genres in our playlists are statistically different.
3. **Based on the results, we can say there are at least some *qualitative* differences between the music Kelsey and I listen to and hence the original hypothesis is invalidated**. I listen to more electronic music while she listens to more pop, indie, and r&b. In general, my playlists are also more varied, as measured by the number of genres contained within the playlists. However, the audio features suggest that the underlying musical characteristics of the songs we listen to are actually fairly similar.

Should I listen to this song or should she?

The last objectives of this project were to use the Spotify data to:

1. Try to predict whether a song is more suitable for my her playlist or mine
2. Try to predict songs that we both would like (after all, music is meant to bring people together, and up until now I have focused on differences)

Both of these goals can be accomplished by using a classification machine-learning algorithm. For those unfamiliar with these types of algorithms, the basic premise is quite simple: if I feed the algorithm the features of a song (which in our case will be the audio features, e.g. loudness, popularity, danceability, etc., and the genre(s)), the algorithm should spit back the *class* of the song, i.e. whether the song is more likely to belong to my playlist, represented by 0, or more likely to belong to hers, represented by 1. During the learning process, the algorithm is looking for an optimal mathematical function that can be used to make this prediction.

The first task was to store the genre data in a way that the machine-learning algorithm could understand. Instead of a single column in the dataset that specifies which genre(s) a song belongs to (i.e. 'electronic', 'rap'), one column for every unique genre was added to the dataset. If a song belonged the genre 'rap', a value of 1 was placed in the new column with title 'rap.' If a song does not belong to a particular genre, it has a value of 0 for the column of that genre.

The dataset used for this analysis consisted of a subset of my own playlist in addition to Kelsey's full playlist. I did not use my full playlist because the number of songs in my playlist (1370) is much larger than the number of songs in her playlist (318) and would bias the algorithm toward labeling every song as mine. Thus a subset of 318 songs of my own playlist was utilized. The dataset was split into a training set (60%), a cross-validation set (20%), and test set (20%), and the algorithm was trained for various values for the regularization term. For those of you familiar with logistic regression, l2 regularization was utilized and the cost function used log loss. A value of 1/10 was chosen for the regularization term based on evaluating the logistic regression fit on the cross-validation set.

The results were surprisingly good. Out of 100 test sets, the average accuracy of the algorithm was 76%. In other words **8 out of 10 times** the model was able to predict correctly to whose playlist a song belongs. **The average accuracy of the classification rose to 86% when the algorithm was more than 70% sure that the song was classified correctly**. Probabilities higher than 70% were predicted 78% of the time. Unsurprisingly, the top predictors of a class were the features related to genre and not to the Spotify audio features. In fact, a binary classification using just the genre data was equally effective as that using the genre and Spotify audio feature data.

There are two interesting applications of this logistic regression algorithm. Firstly, **the algorithm can be used to create individual playlists for myself and for her.** All that is a required is a pool of potential songs and the associated audio features and genre(s), which can be gathered via the Spotify API. The algorithm can calculate the probability of a song belonging to Kelsey's or my playlist, and if it is above a certain value, say 80-90%, then the song could be added to the playlist. Secondly, **the algorithm can be used to create a joint playlist containing music that both she and I like**. If the probability of a song belonging to a certain playlist is close to 50%, then there are two options: the song could reasonably belong to both playlists and we should both like the song, or the song should not belong to either playlist and we should both dislike the song. It's possible that by comparing the genre(s) of the song in question to our overall genre profile, one could filter out songs that we probably dislike. A shared playlist made using this method is available at the following Spotify URI: "spotify:user:sgheinze:playlist:0YUl9fKcdnGcJOlCnYF8ER."

An idea for Spotify

Spotify uses highly sophisticated machine-learning algorithms to curate popular playlists and make personalized song recommendations. The algorithms include collaborative filtering, content-based recommendation (audio features, probably beyond those accessible in the Spotify API), and deep neural networks. Anyone who uses Spotify and listens to their "Discover Weekly" playlist, which consists of 30 songs chosen by Spotify specifically for the user, can attest to the power of Spotify's algorithms. One of the mini-applications of my (rather humble) logistic regression was to generate a playlist suitable for multiple users based on their Spotify content. The idea was that this could be used when hanging out with multiple friends or family in order to come up with a playlist that is suitable to everyone's liking. This would also likely result in people being exposed to music they were not familiar with but that they would likely enjoy. I'm surprised that Spotify does not have a feature to create these sorts of playlists, given that all Spotify would have to do is find the overlap of recommendations for multiple users. Perhaps this is a feature Spotify should think about implementing.

Conclusion

In this article, I used data to investigate something I've been interested in for many years, music. I showed how I gathered data from the Spotify API to try to answer a rather broad question, "How is the music I listen to different than the music my girlfriend listens to?" By analyzing the distribution of audio features and genre content of our Spotify playlists, I was able to show that there are qualitative differences between our music. To complement this analysis, I built a logistic regression model that could classify with ~80% accuracy whether a song was more likely to belong to one user's playlist or another. Two interesting and fun applications of the logistic regression algorithm were discussed.

All data and code will be available on my github: https://github.com/sgheinze