

Is my Spotify music boring? An analysis involving music, data, and machine learning



Juan De Dios Santos
May 28, 2017 · 10 min read

A couple of days ago I was chatting with a friend, while listening to my Spotify saved songs. After a couple of songs she interrupted the conversation to tell me: “Your music taste is interesting...your playlist has a lot of **variety**, **instrumental** songs, and some of them are **boring**”.

I laughed at that comment because it is not the first time I have heard that. I’ll admit that my music taste might be a bit *weird*— for example, I could start the day listening to *Kendrick Lamar*, then switch to the *Inception* soundtrack, followed by some Spanish salsa.

However, her comment gave me a nice idea: let’s check what the data says about this.

So, I ran an experiment.

In this article I will share the findings of this experiment, in which I analyzed my Spotify songs to see if they are indeed varied, instrumental, and boring. Moreover, to make the problem more interesting, I compared each of these three characteristics against my friend’s songs. Lastly, I trained a machine learning model with the purpose of predicting if a song would be more suitable for my playlist or hers.

The tools

The principal tool used in this project is the *audio features* component of the Spotify API service. These audio features represent characteristics about a song, such as how *acoustic* and *loud* it is. I will give a more detailed explanation of the features later.

Python was used to obtain the data using the library Spotipy, and to train the machine learning model using scikit-learn. The analysis of the data was done in R.

playlists of a user, and all the songs of a particular playlist. Once I had the basic information of the songs, including their Spotify ID, I was able to get the audio features of them using the same script.

The resulting dataset is made of 15 columns and 1074 songs, of which 563 come from my playlist, and 511 from hers (from now on I will refer to my friend as *she* or *her*).

Of all the 15 columns of the dataset, only those related to the audio features were used. In the following list I'll introduce them, and explain what they mean (in some cases I'll just copy/paste the description from Spotify). Note: the value of all the features is in the range 0.0 -1.0.

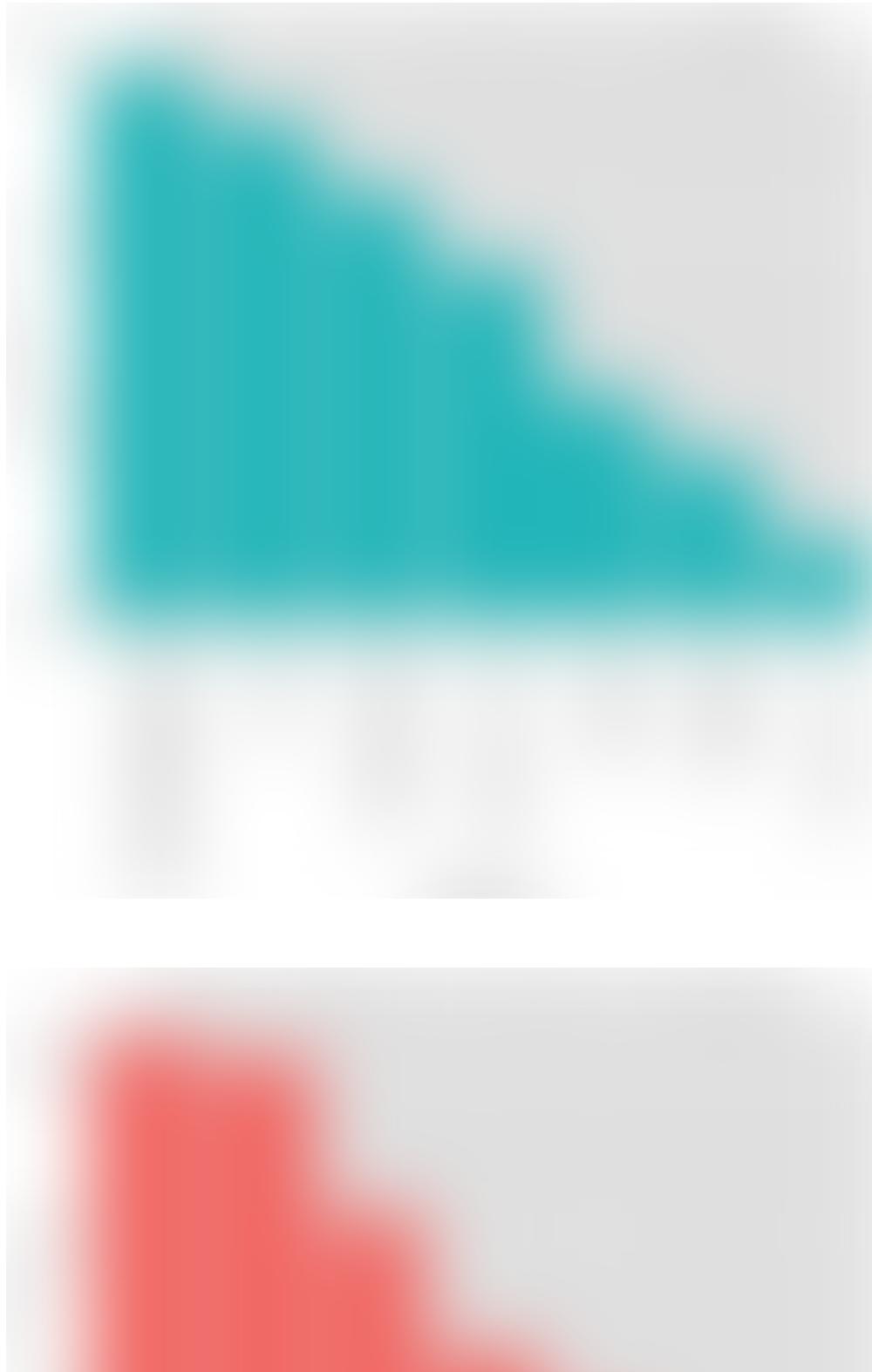
- Instrumentalness: This value represents the amount of vocals in the song. The closer it is to 1.0, the more instrumental the song is.
- Acousticness: This value describes how acoustic a song is. A score of 1.0 means the song is most likely to be an acoustic one.
- Liveness: This value describes the probability that the song was recorded with a live audience. According to the official documentation “*a value above 0.8 provides strong likelihood that the track is live*”.
- Speechiness: “*Speechiness detects the presence of spoken words in a track*”. If the speechiness of a song is above 0.66, it is probably made of spoken words, a score between 0.33 and 0.66 is a song that may contain both music and words, and a score below 0.33 means the song does not have any speech.
- Energy: “*(energy) represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy*”.
- Danceability: “*Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable*”.
- Valence: “*A measure from 0.0 to 1.0 describing 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)*”.

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including
cookie policy.

X

are varied, instrumental, and boring. So, I investigated each of these three attributes, to test if she is right or not. In this section I will describe my findings.

I'll commence by presenting two plots of the mean value of all the audio features of both playlists, so we can have an idea of which are the predominant features of each dataset and to familiarize with it.



To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

X

The first plot shows that the prevalent feature of my playlist is **instrumentalness**, and the second plot, which represents her playlist, shows that **danceability** is hers. However, how big is the difference between these values? The chart below is the result of what happens when the mean of each of her features are subtracted from mine.



Clearly we can see that **instrumentalness** and **acousticness** (the blue bars) are the distinctness features of my playlist, with a difference of **0.53** and **0.1**. On her side,

This outcome can be interpreted as this.

- My songs seem to be less vocal and more instrumental.
- Her songs are livelier.

So, after seeing these results, we can all agree that the answer to the question “*is my playlist instrumental?*” is a definitive yes.

Variety

The next question I’ll answer is: *how varied is my playlist?*

To answer it, I investigated how similar or dissimilar the songs of each playlist are. For example, a very varied playlist means that the user has many songs from different genres. The opposite of this, a low varied playlist, is one where mostly all the songs belong to the same genre.

The technique I used to check how varied our playlists are, was a simple look at the **standard deviation** of the audio features. Here are the plots.





By looking at the plots is a bit hard to decide which of the playlist is more varied, however if all the values are summed up, and the means calculated, we find out that the mean value of all the standard deviations of my playlist is **0.244**, while the sum of all the individual values is **1.713**. Regarding her playlist, the respective values are **0.174**, and **1.218**.

What are the implications of this? A high standard deviation says that the scores of the audio features of my songs are not that similar, meaning that for example, I could have many songs where the instrumentalness value is really high, while also having songs where the same value is really low.

Conclusion: my playlist is at least more varied than hers.

Boring

Finding the answer to the question “*how boring is my playlist?*” was one of the most fun parts of this work because, what exactly is a boring song. What is boring for me could

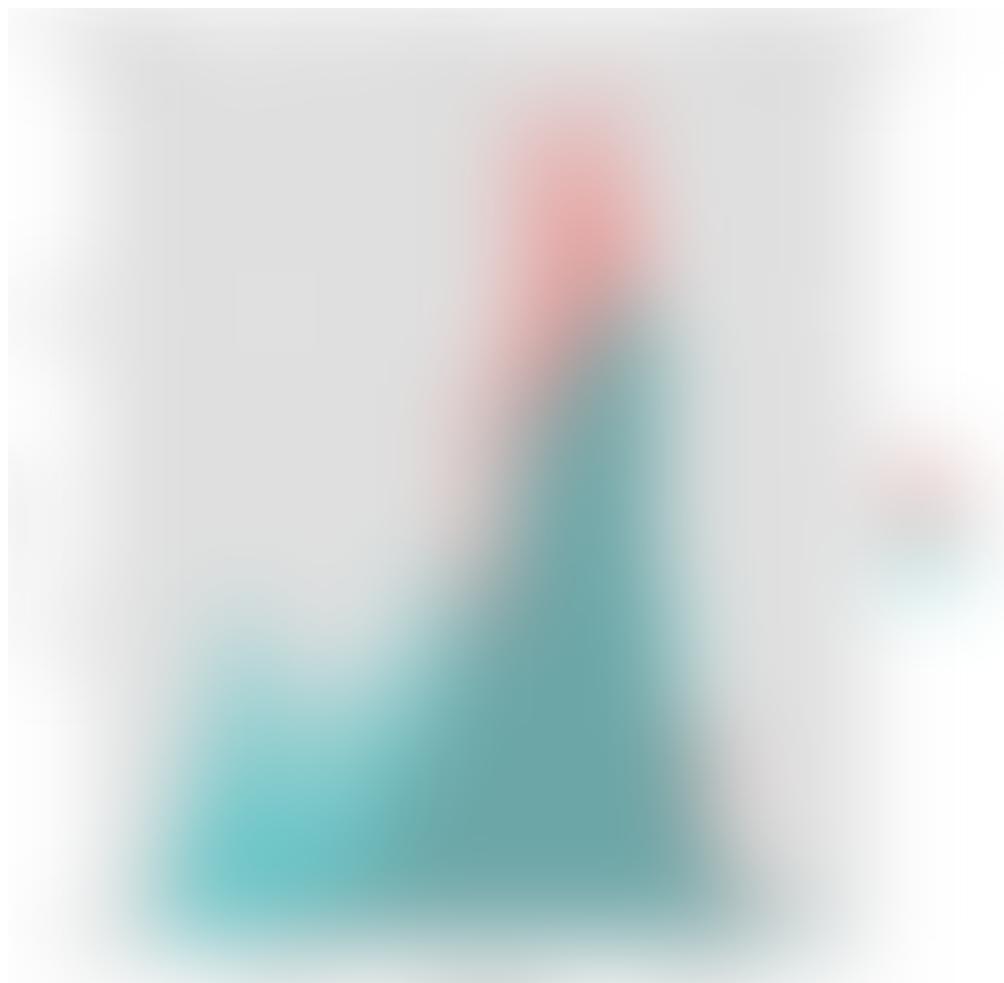
said party. So I came up with a simple equation that involves the **energy**, and **danceability** feature, plus two features I haven't introduced yet: **tempo**, and **loudness**. The reason why I chose these is because at a party I would like to have loud music, with a nice tempo that brings out the energy in me (and the others), to get in the mood for dancing.

This is the equation:

$$\text{boringness} = \text{loudness} + \text{tempo} + (\text{energy}*100) + (\text{danceability}*100)$$

And according to it, the lowest the score is, the more boring the song is, and the highest the score, the more fun it is. Also, note that energy and danceability are multiplied by 100 because the loudness and tempo values are not in the range 0.0–1.0, and I wanted to keep everything more or less in the same range.

These are the results:



To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy. X

The image above is a histogram, and it shows how the boringness score is distributed among each dataset. The first thing one might notice is that the peak of the pink area, which represents her playlist, is higher than the blue one, implying there are many values in that range. Also, the pink region is denser on the right side (this is called **left skewed**), in comparison to the blue region, meaning that most of the boringness values of her playlist are greater than mine.

To complement this histogram, the mean value of the boringness was computed, and the values are **201.735** for me, and **233.284** for her. The following plot illustrate this.



Thus we can conclude that according to my own equation, **my playlist is more boring than hers**, and that I would hate to hear my own music at a party.

Is this song hers or mine? A machine learning approach

The last objective of this experiment was to investigate if it is possible to predict if a song belongs to my playlist or hers using machine learning.

I like to define machine learning, in particular the subfield of supervised learning (which is the one I'll apply here), as the task of using a system to learn about the patterns of a data set. During this learning process the algorithm is looking for an optimal mathematical function, or a *way*, that is able to explain the relationship between the features of the data (i.e. the audio features), and something called the *label* or *class* of the data (i.e. the owner of the playlist, me or her). Thus, when the system has learned about the data, it should be able to infer or predict the class of a new set of features using the knowledge it learned during the learning step.

In the case of this work, this means that we will have a machine learning system trained with the dataset used in the previous part. Said system should be able to determine if a new array of audio features is for a song that would be more likely to appear in my playlist or hers.

Of all the existing machine learning models, I used one called *logistic regression*. For the sake of keeping this article simple and friendly, I won't explain how logistic regression works. Instead, I'll just say that it is a mathematical equation in which the target variable, called the *dependent variable*, or *the thing we want to predict* (in this case that is the owner of the playlist), *depends* on one or several independent variables (the audio features), plus some magic. Something like this:

```
dependent_variable = magic(independent_variable_1 +  
independent_variable_2 + ...)
```

The value of the dependent variable is a number between 0 and 1, and in this case, that value represents the probability a song has to belong to my playlist, or hers — if the value is less than some threshold X, the song belongs to me otherwise it belongs to her.

Before showing the results, I would like to describe the training settings for the readers who have knowledge about the topic. The model parameters *alpha*, and *number of iterations*, were obtained using *grid search* and *cross-validation*. The best configuration found is the following:

- Penalty: *elasticnet*,

- **n_iter**: 50

The following snippet (Python) is the code used for training the model.

```
parameters = {  
    'alpha': (0.001, 0.0001, 0.00001, 0.000001),  
    'penalty': ('l2', 'elasticnet', ),  
    'n_iter': (10, 50, 100),  
    'loss': ('log', )  
  
}  
  
# Perform a grid search with cross validation to search for the best  
parameters.  
  
grid_search = GridSearchCV(SGDClassifier(), parameters, n_jobs=-1,  
                           verbose=1, cv=5, scoring='accuracy')  
  
grid_search.fit(data, labels)
```

The results were surprisingly good! Out of several training and testing sections, the average accuracy was **82%**, in other words **8** out of **10** times the model was able to predict to whose playlist the song belongs.

Once the model was trained, it was tested using two songs it didn't encounter during training. The prediction of the first song, called *A Better Beginning* from the videogame Mass Effect Andromeda, was "me", while the second song, *Love On The Brain*, by Rihanna was "her". Pretty accurate.

Conclusion

In this article I showed how a silly remark or opinion regarding my music, turned into an experiment. By using the audio features API component of Spotify, I was able to find out that, just like my friend said, my playlist is varied, full of instrumental music, and somehow boring. To complement this analysis, a machine learning model, logistic regression, was trained with the purpose of predicting if a song is more suitable to my

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

X

All the code used is available at <https://github.com/juandes/spotify-audio-features-data-experiment>. This includes the Python script used to obtain the data, and to train the machine learning model, and the R script used for the analysis.

For more information about Spotify's audio features, check out the official documentation at <https://developer.spotify.com/web-api/get-audio-features/>, and for an introduction to logistic regression, I recommend the following article: Logistic Regression for Machine Learning

If anyone spot any typo, inconsistency, or would like to ask or say something, please do comment :)

Thanks for reading.

[Machine Learning](#) [Data Science](#) [Spotify](#) [Data](#) [Towards Data Science](#)

[About](#) [Help](#) [Legal](#)