Frank Pasqualini

Carl Bodenschatz

STAT 1223

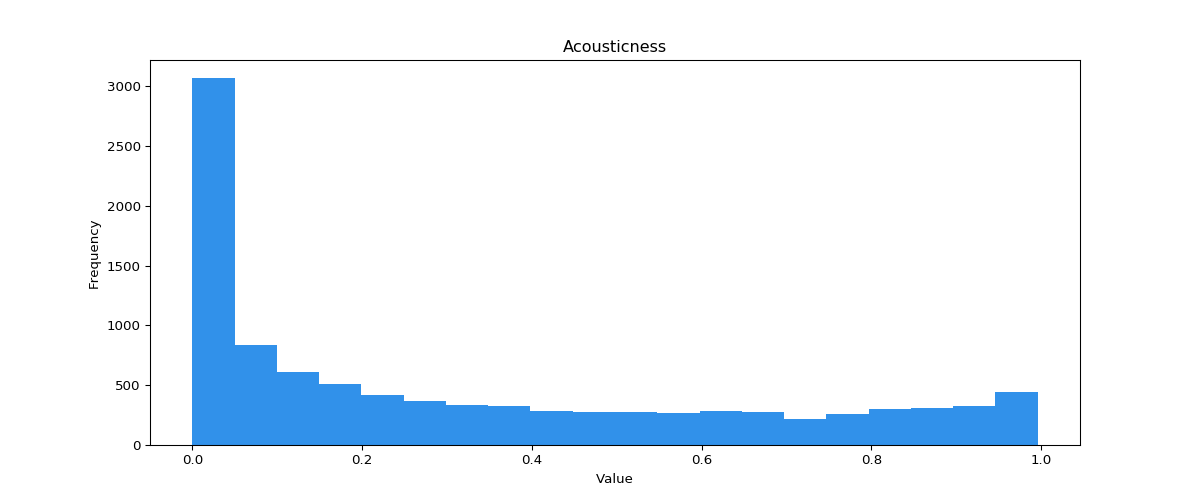
20 Mar 2019

Spotify Music Library Regression Analysis – Progress Report

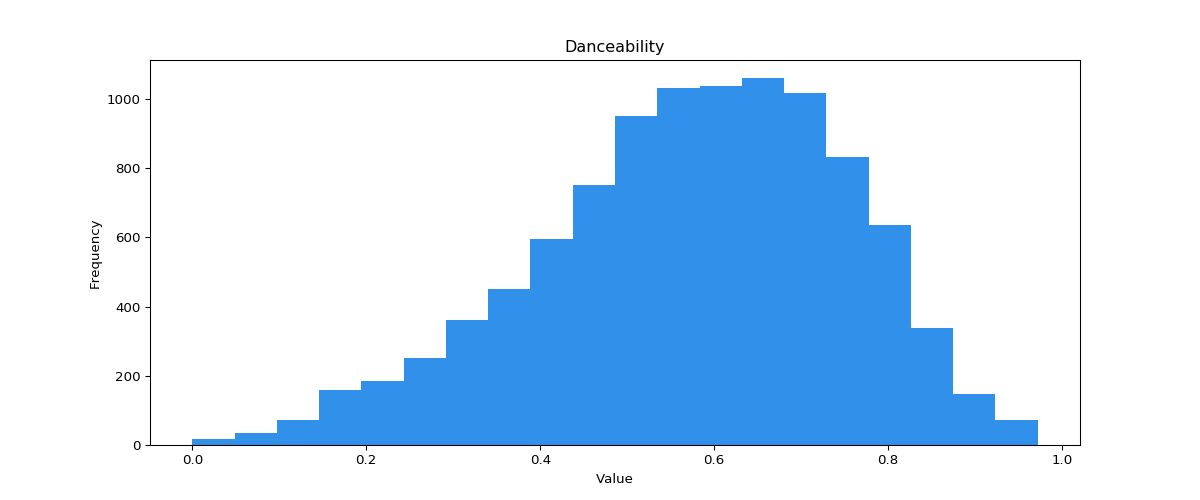
This study is being conducted to build a multivariate regression model relating multiple attributes of a musical track on Spotify to a specific user’s musical preferences. The response variable is a play count, the amount of times that the user has listened to that song. We believe that a play count is a good estimate of how much a user likes a track, because if they like the track more they will listen to it more. It is not a perfect estimate, however, because it is possible for a user to have discovered a track recently that is one of their favorites, but due to not having known about it for a long period of time the play count would be low. The reverse is also true, where a user could have a song that they think is okay but not one of their favorites, but since it has been in their library for years, they have listened to it more times. It is very difficult to quantify how much a user likes a track, and we believe that play count is the best estimate we can currently research. We think these results will be particularly interesting, because if this study is successful, it could be turned into a generalized tool that a user could insert the data from their own music library into. The resulting regression model could be then used on a per-individual basis to find songs that are likely to be enjoyed by the listener, by giving recommendations for songs that would maximize the response variable based on the predictors. This will be a case study based off of the Spotify and Last.fm accounts of Frank Pasqualini.

For this study, we are performing the regression on the music library of Frank Pasqualini, using data from his Spotify and Last.fm accounts. At the time of writing the proposal and gathering the data, Frank had 5,773 tracks saved to his Spotify library and 35,807 scrobbles saved to his Last.fm account from 1,334 artists. These numbers have since increased to 5,774 tracks saved to his Spotify library and 37,503 scrobbles saved to his Last.fm account from 1,345 artists, but these changes will not be represented in the study. A scrobble is a recorded instance of a track being played. Frank has been tracking his scrobbles using Last.fm since December of 2017, and those scrobbles were tracked directly by linking his Last.fm account to his Spotify account, so the data should be mostly valid even though the predictors will be collected from a different source than the response. We are looking into using 15 potential predictor variables, which are as follows: duration\_ms, explicit, popularity, key, mode, time\_signature, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence, and tempo. These were retrieved for each track using Spotify’s Web API.

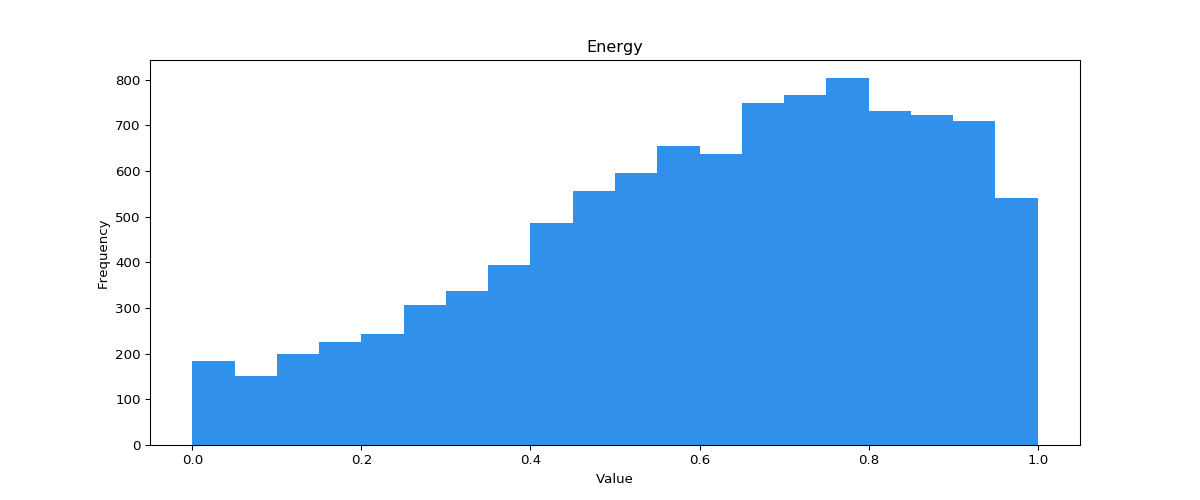
Almost all the variables we used in our model were retrieved from the “Get Audio Features for a Track” Spotify API call. The documentation for this API call gives in depth descriptions about what each of these values represent, as well as providing the distributions that we have included below, but we will provide a general description of each of them. Some of the more basic variables were taken directly from the track object returned from the API call. Duration\_ms is a measurement of the length of the track in milliseconds. Explicit is a categorical variable that tells whether the track has been marked to contain vulgar language. Popularity is an integer value between 0 and 100 that is a calculated by one of Spotify’s algorithms based on total plays and how recent those plays are. Key is a categorical variable representing the musical key the track is estimated to be in for the majority of the track, with 0 being C, 1 being C#/D♭, all the way up to 11 representing B/C♭. Mode is another categorical variable with 0 representing if the track is in a minor key for the majority of the track and 1 if the track is in a major key for the majority of the track. Time\_signature is an estimation of how many beats are in each bar for the majority of the track. Acousticness is a confidence measure of whether the track is acoustic. It has the distribution featured in the following figure:



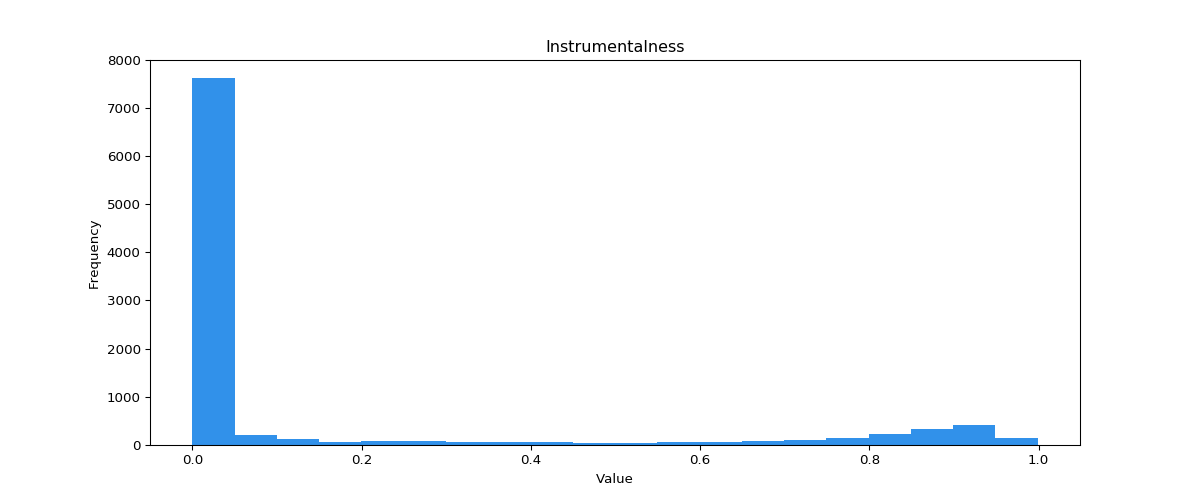
Danceability is an estimate of how well suited a track is for dancing to based on many attributes such as tempo, stability, and regularity, and has the distribution shown in following figure:



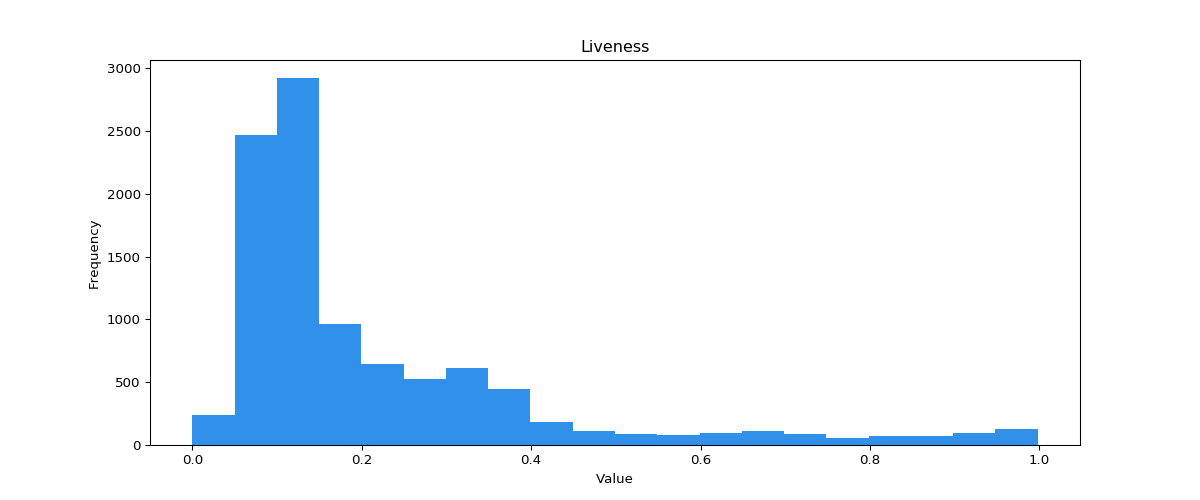
Since danceability is in part based on tempo there will likely be high covariance between these two variables. Energy is an estimate of the intensity of the track based on attributes such as entropy, timbre, and loudness, and has the distribution featured in the following figure:



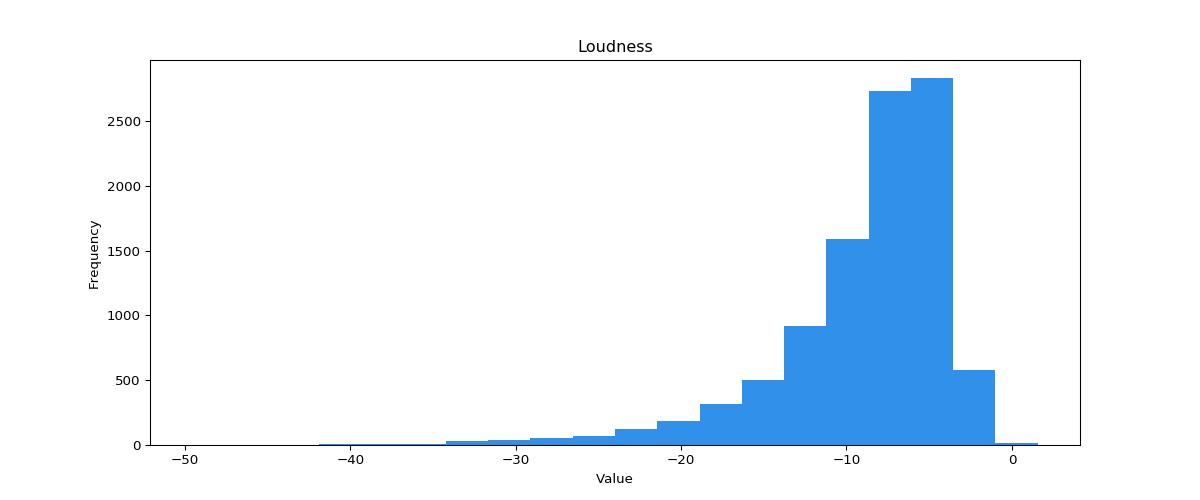
Since energy is in part based on loudness there will likely be high covariance between these two variables. Instrumentalness is a confidence measure of whether the track contains no vocals and has a very skewed distribution represented by the following figure:



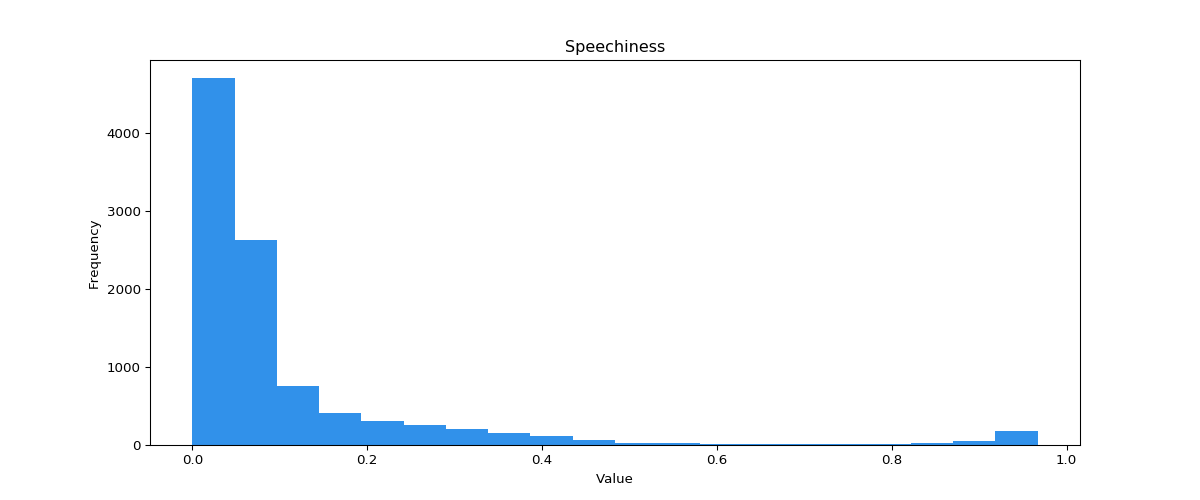
Liveness is a confidence measure of whether there is a live audience in the recording and has the distribution in the following figure:



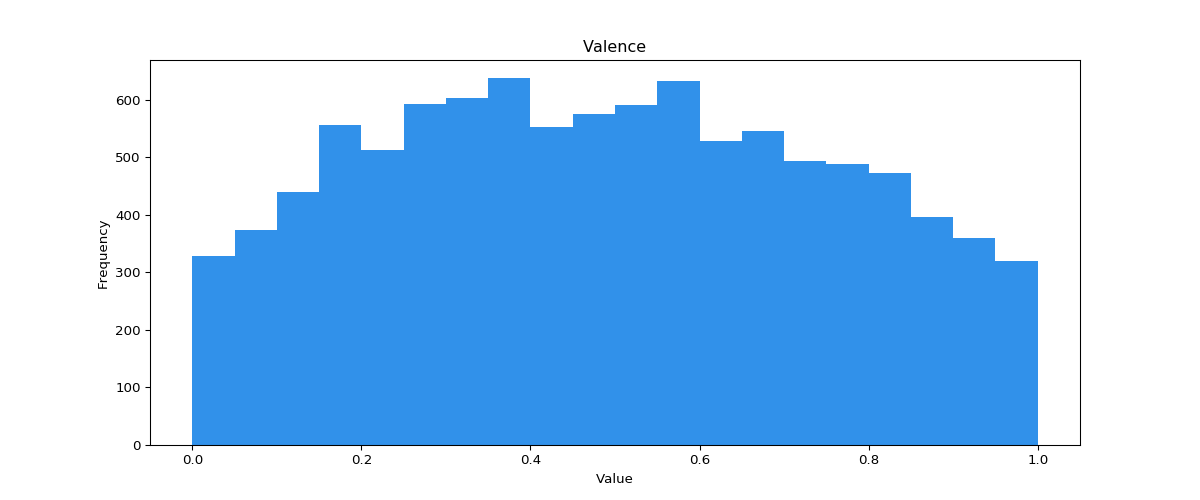
Loudness is the measure of the average decibel value of the track and has the distribution in the following figure:



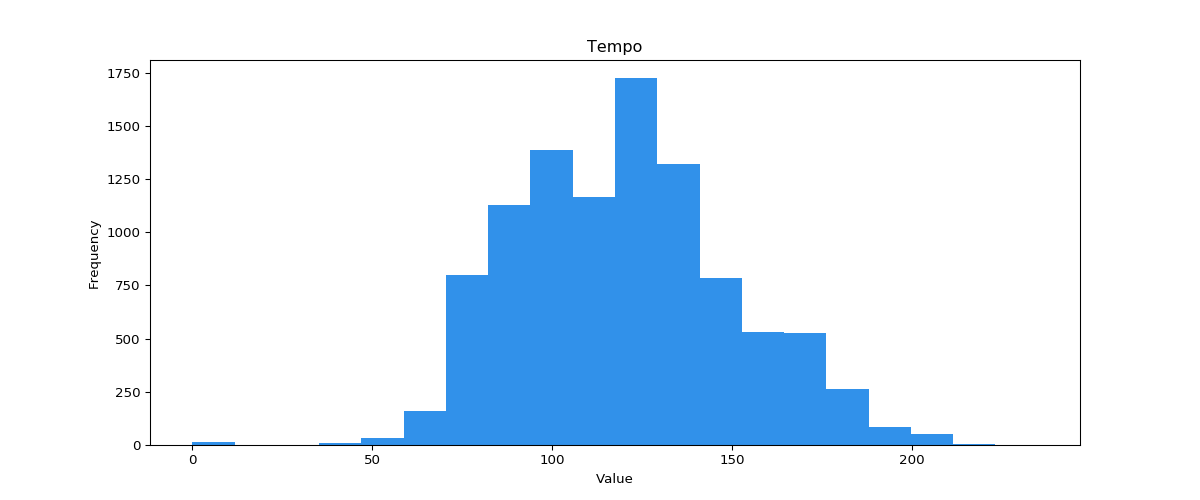
Speechiness represents the exclusivity of speech in the track, and high values are almost entirely speech while low values contain little speech but possibly sung words. It has the distribution featured in the following figure:



Valence is a measure of the “happiness” of a song and has the distribution in the following figure:



Finally, tempo is the estimated average beats per minute of the track and has the distribution in the following figure:



The data for this study was collected with a python script which can be found in the file “Data Scraper.py” on this project’s GitHub page. The script parsed the entire Spotify library of Frank Pasqualini and got the audio features for a random sample of 1000 songs. This was done with the help of the Spotipy python package, which simplifies the Spotify API calls to be easily used in python scripts, but due to some limitations of the package a lot of work had to be done collecting the results in a useable form. The Spotipy package can only gather tracks 50 at a time, so we had to write the script to gather 50 songs, then jump down the list of the songs and gather 50 more in a loop until less than 50 songs were returned, which would be the end of the list. We then had to interpret the results from this data collection into a usable data structure with the variables we needed. The script used Frank’s Last.fm account to get the play count by submitting a GET HTTP request with the title and artist for each track gathered in the sample, and one of the functions in Spotipy to gather the rest of the audio features. Finally, the script took all of the variables we had collected and placed them into an easy to use data structure and exported the entirety of the structure to a file to be analyzed by a different script. We picked 1000 as our sample size because it is a large enough sample size that the results will be significant, but it is not so big that the analysis will take too long to run. Unfortunately, this sample is slightly biased towards songs with higher play counts and popularity, because the user is more likely to have added songs to their Spotify library if they have heard the song multiple times before, which is familiarity bias. This sample also did not return a full 1000 songs, because since the data is gathered from two different sources, it wasn’t possible to correctly link every song. Fortunately, only 2 songs from the 1000 song sample were left out, so the sample still has a strong 998 members.

We will be trying to create the best fit regression model using the 15 variables defined earlier. There are multiple categorical variables, so the regression model will have to feature dummy variables to represent the full spectrum of potential categories, such as “Explicit C minor 4:4” and “Non-explicit D#/Eb Major 5:4”, as well as hundreds of other combinations. However, if enough testing is done to prove that there is not a significant difference between, for example, major and minor songs, then mode could be removed from the equation entirely, simplifying it by a lot. It is very unlikely that the correlations will be linear for most of the remaining variables, so we will likely have to manipulate the variables. For example, in the tempo distribution supplied by Spotify, the value peaks at about 120. Perhaps instead of using the variable [tempo] for this, the variable [(tempo-120)2] would be more appropriate, because it peaks at 120 and decreases the farther you get from 120, which would give more of a linear relationship for regression to analyze. It will take some time and a little bit of trial and error to find the correct transformations, if they exist, but taking that time will help improve the model. We also plan on comparing the distributions from our sample to the distributions provided by Spotify. Because Spotify doesn’t supply the exact numbers, they will have to be estimated by measuring the distributions by pixel height, which could cause some error. We plan on testing the Null Hypotheses that the means and standard deviations of all the samples are equal to the means and standard deviations of their populations given by Spotify.

We hope to conclude whether Frank’s music library’s attributes differ significantly from the population averages and whether a good fitting model can be made to predict how much a user would enjoy a song through play count. We also want to determine if the attributes of the songs can account for a significant percentage of the variability of the data, or if enjoyment of music can’t be accurately measured with numbers and figures. As a team consisting of musicians, statisticians, and computer scientists, it will be very interesting to discover ways to combine these three fields and potentially use statistics to create perfect songs. Right now, the script for scraping the data is already written, and all of the data has been collected. We have most of the analysis script done other than the transformations and regression models. We should be finishing up the regression and transformation parts of the script within the next week or two. The model should be finalized by the end of March. The remaining time will be used to write up the final report using the results we collected.

Works Cited

“Get Audio Features for a Track.” Spotify for Developers, developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/.

Pasqualini, Frank J. “Spotify-Regression-Analysis-Data-Scraper.” GitHub, github.com/Frank-Pasqualini/Spotify-Regression-Analysis-Data-Scraper/.