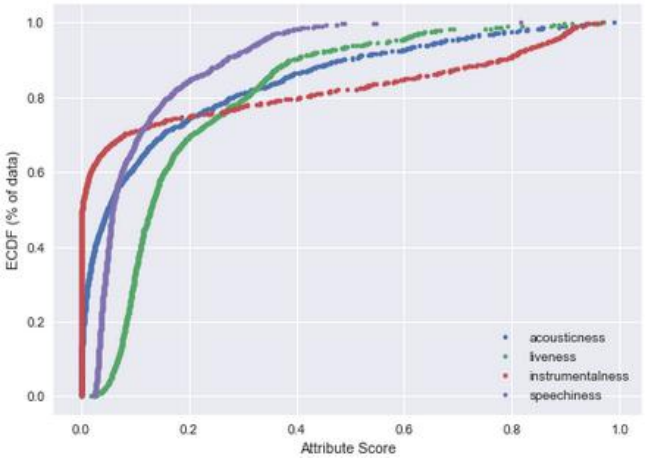
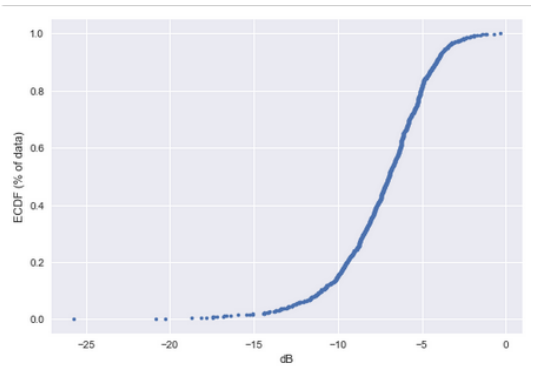
The aim of this report is to extract trends from Spotify user data to better understand their preferences and create a recommendation engine that can accurately predict what songs the user will like. Spotify, like many tech companies today, is constantly working to improve their recommendation systems to ensure user engagement and retention. Through traditional statistical analysis and machine learning techniques, I will create a clear profile of attributes that define the users preferences.

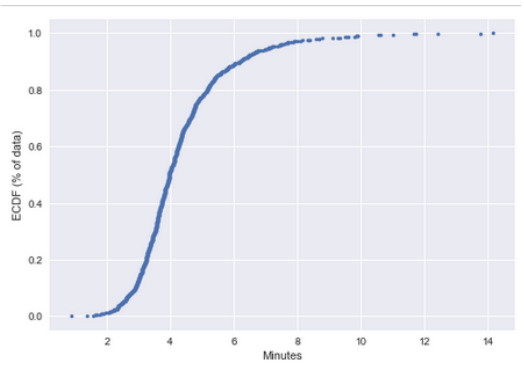
The dataset used here was obtained from Kaggle and contains 2017 songs with their attribute scores and whether or not they were liked by the user. Fortunately, this data was gathered from the Spotify API, therefore the data started off very clean with the columns properly formatted as 'int' and 'float' types. Regardless of this, I checked for missing values anyway to make sure and indeed there were none.

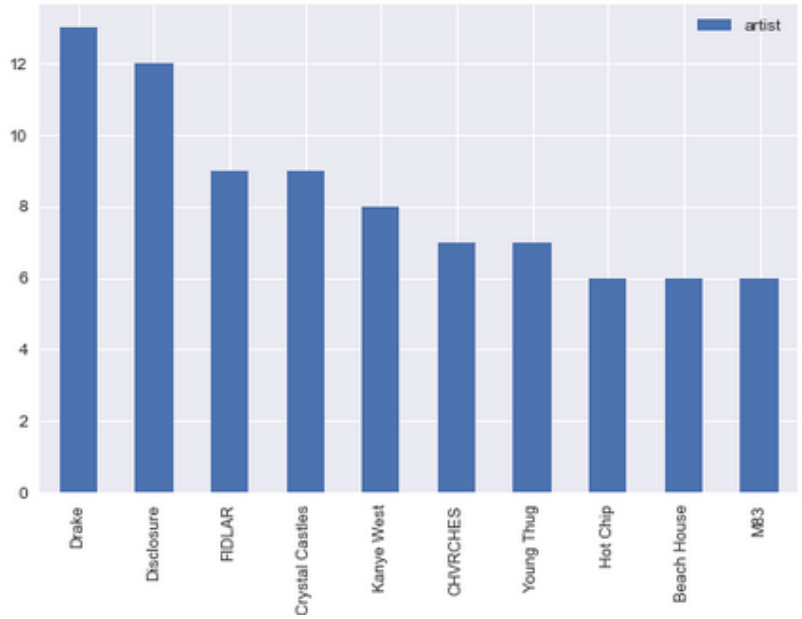
The Spotify API is freely available to generate the attribute scores of a single song or set of songs for analysis and prediction purposes. However, in order to get specific user data with which to train a recommendation engine the user must provide permission via their Spotify app in order for their playlists and liked songs to be accessible. This presents some limitations as to what a third party developer can do as the following recommendation engine may or may not generalize to sets of users.

I began by splitting the liked songs from the rest of the data and inspecting the numerical attributes visually to determine if there were any clear patterns present in the data. In inspecting the distribution graphs of the attribute scores, we see that some are normally distributed and others are heavily skewed (right graph). In context, what this means is that the attributes that are normally distributed are not concentrated at certain values but are instead relatively evenly spread out over the range. By contrast, the skewed attributes provide evidence for preferred values for the given attributes because of the concentration of songs around that value. The variables that exhibited skewness are Acousticness, Liveness, Instramentalness, Speechiness, Duration, and Loudness. The first four are plotted above because they share the same scale; Loudness and Duration are plotted below on their given scales.









Next, I plotted the categorical attributes to check for differences in the frequency of categories. We find that the key denoted by 3 (D#) is the least popular key and the 4 (4/4) time signature is by far the most prevalent value. As it turns out though, these values are known throughout the music industry to be the most common value for these attributes; therefore it is likely that these patterns do not reflect user preference but rather the nature of music itself. Lastly, I generated a plot of the top 10 most liked artist which reveals that the user's favorite artist is likely Drake or Disclosure.

With some variables of interest defined, I proceeded to conduct a statistical examination of the numerical attributes that exhibited skewness as well those that appeared to be different between liked and disliked songs. To determine this

To do this, I split the dataset into the songs that were liked and those not liked to examine the mean differences between the sets. The variables that had considerable mean differences given their scale were Acousticness, Instramentalness, Duration, and Loudness which supports our visual analysis of the liked dataset.

	Z-test results	
Variable	**Statistic**	**P-value**
instrumentalness	-6.93091	4.18e-12
loudness	3.24042	0.001194
duration_min	-6.65947	2.75e-11
liveness	-1.18385	0.236471
speechiness	-6.99662	2.62e-12
acousticness	5.86831	4.40e-09

Z-tests were conducted to determine if these differences were significant and indeed acousticness, instramentalness, loudness, duration, and speechiness all had significantly different means suggesting they are prominent features that the user uses to determine whether or not they like a given song.

In order to create a predictive model for the data, I will use the feature selection algorithm Lasso Least Angle Regression (LARS) as well as regular Lasso to determine if any variables should be dropped from the model. Based on the literature, Lasso LARS and Lasso should yield only slightly different results therefore I will run a model using the recommendations from both methods.

Lasso did not reduce any of the variable coefficients to 0, however LARS reduced energy, key, and time signature to 0 indicating they are not well correlated with the target variable. As expected this is a slight difference but different all the same. Therefore I will run one model with the recommended variables removed and one with all the variables included to compare the initial results before improving the model.

I will be using an Artificial Neural Network (ANN) to predict the target variable from the given song attributes. The architecture of the ANN was determined using the total number of variables and the recommended size given dropped variables. After several iterations that are not noted in the Jupyter notebook, 2 layers with 15 and 11 nodes was chosen; more layers significantly reduced accuracy while reducing variance slightly and more nodes improved accuracy less than the consequential increase in variance.

The initial results reveal that following the LARS recommendation significantly reduces accuracy and increases variance of the ANN, therefore all variables will be included in the final model and which will have its hyper parameters tuned. To tune the final model, I will utilize GridSearchCV from the SKLearn library to test the batch and epoch hyper parameters. I chose not to test the optimizer here since the literature on the optimizers supported by Keras indicates that Adam is very likely the best option given the nature of the data and problem.