Identifying Popularity-Agnostic Factors of Musician Success

Tahmid Ahamed, Shiva Menta
FNCE 237-002
Wednesday, November 30th, 2022

Motivating Questions

Are there any size or popularity-agnostic qualities of an artist that contributes to their overall success in the music industry?

Are certain song features, such as danceability and acousticness, relevant to the overall popularity of an artist?

Does collaborating with other artists make it more likely for an artist to be considered successful?

Is the level of social media engagement a musician has with their community related to their success?

Background and Motivation

As of 2021, the annual revenue of the global recorded music industry was $28.8B.[1] With the advent and growing popularity of streaming services in the past decade, the revenues generated from the music industry as a landscape have largely changed. Many people have argued that the introduction of streaming services as an alternative to consuming music content, other than purchasing records and tracks directly, has ruined the music industry due to affected royalties from artists. Many others have argued that the introduction of these platforms has actually saved the music industry by reducing the usage of illegal streaming sites to consume music, now that music consumption is relatively cheaper. Regardless of the opinions of the masses, one of the facts that remain true is the fact that access to music data is at an all-time high. Streaming platforms such as Spotify and Apple Music store abundant amounts of data on a song- and artist-level basis, such as retention rates, audience diversity, energy levels, collaborations, etc. Considering the music industry is one in which executives and managers aim to maximize profits at every step, access to this music data has revolutionized the space and has changed many aspects of how an artist & label duo approaches the music industry. Artists are now able to get a better grasp of how their releases are performing, who is engaging with their music, what types of music genres are selling better, etc. With all of this information being available to artists and music labels, the following question arises: what are the key drivers that make a musician successful?

As a preface, we need to recognize that many studies have been conducted on what makes an artist ‘successful’, depending on the metric used to measure success. Many times, certain artists are in the right place and at the right time to get picked up by a music label or go viral on social media. Some additional factors that may go into this, which cannot be implicitly measured, are unique approaches to cracking the music industry, connections within the music industry, etc. Many of these musician ‘blow-up’ moments are what catapults them into fame, and as long as they maintain some level of consistency, they are able to stay successful. However, for the purposes of this project, we want to see if there are any common themes across some of the most successful artists that can ultimately be used as a proxy to change strategies for upcoming artists.

[1] https://www.statista.com/statistics/272305/global-revenue-of-the-music-industry/

The first thing we need to consider is what different measurable factors can contribute to an artist's success. Given the variety of genres or types of music styles, the lifestyle of the artists beyond the music, and everything else, there could be several underlying factors that contribute to the success of a given musician. However, some of the more notable factors include previous successes, music labels, collaborations with other artists, levels of experimentation, music genres, time of release, commercial marketability, go-to-market strategy (advertising strategy for songs and how an artist markets themselves), etc. We won't be able to explore all of these factors, however, they represent some of the most influential factors that likely contribute to an artist's success. Although we wanted to explore all of these metrics, many music data sources are paywalled, so we were restricted to limited datasets.

Using the plethora of music/musician data available to determine the leading factors of an artist's success, and how much they affect the success of a given artist could lead us to better understand if a given artist will "blow up" in the future given their current performance. From a profitability standpoint, this question becomes integral to music labels and managers, as these individuals want to maximize their chances of success in selecting the next biggest hit. Looking at what has already been done, most of the analysis on this topic has been done on mostly a qualitative level, rather than a more quantitative level. For instance, how popular an artist is compared to their counterparts, or how "good" their music sounds. These are very subjective metrics, and as such makes it difficult to really judge whether or not a given artist will actually become popular on a wide scale or not. Being able to analyze the available data to see if a few common factors between popular artists could be identified that are not present in significantly less popular artists, would be invaluable to the success of the music industry. This all being said, as avid music enjoyers, we believe that a music industry solely focused on data would be a detriment to the overall music ecosystem and future of music, as much of the artistry and experimentation would be stripped away from this beautiful art form.

Hypotheses

Our null hypothesis, or what we're trying to disprove, will be that there are no popularity-agnostic significant factors that determine an artist's success, measured in average Spotify streams per song. This would imply that there is an unmeasurable mix of factors that

contribute to an artist's success or that the factors we are considering are not the most important factors. If we are unable to identify any statistically significant factors that contribute to and explain the variation in an artist's success, we will be forced to submit to this hypothesis.

Our alternative hypothesis is that there are popularity-agnostic factors that are related to an artist's level of success, as measured above. For the purposes of this project, this hypothesis will predict that the main factors that contribute to an artist's popularity are an artist's genre, level of social media engagement with their community, frequency of collaborations with other artists, level of genre exploration, and aggregate song-level features (e.g. danceability, energy levels). Again, while we could have used the frequency of successes with an artist's previous songs, or any information that is directly related to the artist's fame, we would run into problems of confounding variables, and concerns about causation. This would effectively lead to a model solely driven by these successes.

Datasets

Using beautifulsoup web scraping and Python scripts accessing APIs, we created three datasets from three separate sources, which are outlined below.

Kworb.net Spotify Data

Kworb is a unified, aggregated source for a majority of all daily and weekly charts provided by Spotify. It stores historical data on a top charts basis, artist basis, and song basis. This database will allow us to get a better picture of the distribution of stream amounts over the history of an artist's releases, understand who the most popular artists are, understand what the most popular genres are, etc. The data is stored in an HTML format, which can be easily scraped using beautifulsoup and formatted using various Pandas functions.

We particularly sourced the top 10,000 artists by the number of total streams across their songs. For each of these artists, we used the database to get a list of all of their released songs and the streams associated with each track. This gave us ~140K tracks/data points for ~10,000 artists. Within our models, we used this dataset to calculate our dependent variable metric, the average number of streams per song, and create a list of artists to search for using the Spotify API.

Twitter API + Tweepy

The Twitter API allows programmatic access to Twitter data through a variety of commands. Twitter has two versions of their API available for developer use, v1.1 and v2, which we needed to complete this portion of our dataset. We first used API v1.1 to search Twitter's user database from the list of names present in the Kworb.net dataset mentioned above to get each user's Twitter handle (e.g. @Drake). We then used API v2 to get follower counts, account creation date, and the total number of tweets posted for each handle.

The data was collected and aggregated using a Python script that followed the above steps. The data had to be collected over multiple days due to API call limits imposed by Twitter.

Spotify API

The Spotify API allows programmatic access to Spotify through various commands to get JSON metadata about music artists, albums, and tracks. The Spotify API also provides access to user-related data, like playlists and saved music, which was not necessary for this project. We requested the maximum number of song objects allowed by the API for each artist provided in the Kworb.net database. From each song object, we extracted information about the song's number of artists (including the searched artist), song feature data (e.g. danceability and energy levels), and artist genre data. We'll discuss more about the limitations of the API below.

The data was collected using various Python functions as specified above.

Data Limitations

Sourcing our data was the hardest portion of the project. We looked extensively for music datasets or APIs that would provide all of the information we needed, but unfortunately, most of the publicly available music datasets didn't have artist popularity metrics or a sufficient number of songs for each artist we wanted to search. For this reason, we thought the best approach would be to use APIs and top chart data to create our own data set. This required a good amount of data manipulation, but we believe we were able to source a solid data set. One limitation to recognize here though is that our dependent variable, from Kworb.net, calculated as the average number of

streams per song per artist, was sometimes based on incomplete data; again, this data should still provide a reasonable picture of an artist's level of success.

There were a few factors we wanted to explore that we believed would be significant in our exploration, but couldn't due to data limitations. The first feature we wanted to get data for was song release date data. With song release date data, we would be able to add two additional predictors to our dataset, namely time-based genre relevance of an artist and release frequency data. For the time-based genre relevance, we were initially going to reference genre prominence data over time and cross reference each song's release date with this to see how frequently an artist produced music in the dominant genre, and see how this would impact their performance. We would expect this to be the case given that artists who produce in a more popular genre probably have a higher chance of attracting larger audiences. For the release frequency metric, we hoped to look at how frequently on average an artist releases tracks; this metric would be particularly relevant as consistency is generally reported as a key metric for musician success. This logically makes sense as an artist who puts out new music more frequently is more likely to attract new listeners and retain current audiences. Unfortunately, we were unable to source this data, but we predict that these factors would contribute positively to an artist's success.

The second feature we wanted to get data for was song-specific genre data. Spotify unfortunately only stores artist-specific genre data, so we're only able to get a very high-level analysis of artist genre data. We hoped to create metrics such as the artist's most frequently released genre, or a more accurate version of genre diversity, calculated as the frequency at which an artist does not produce a song in their most common genre, but again were limited by the data.

Additionally, as mentioned above, the Spotify API limited the number of songs we could retrieve for each artist. While this means that we aren't able to aggregate song-level data for every song for a given artist, we are operating under the assumption that the subset of songs that we received are somewhat representative of the artist's true aggregate characteristics. Having a higher quantity of data would make our predictions more accurate. One last thing to mention is that we faced RAM limitations while trying to iterate through the genre objects provided by the Spotify API. For this reason, we could only execute a few searches per data point, so we extracted the

most popular genre present, using string matching, in an artist's array of subgenres. This could have yielded inaccurate results, but still should give a reasonable picture of the data. Additionally, many data points did not fall into one of the categories using our string matching function, so we defaulted to classifying these data points as a miscellaneous genre.

Methodology

Our goal in this project is to gain a better understanding of the relationship between different features that represent a music artist's career, and ultimately determine if there are any common popularity-agnostic themes across artists. With this information, musicians and industry professionals can get a better picture of what makes an artist successful and adjust their strategies for potentially increased profits.

In order to make sure we minimize any confounding variables, we're intentionally excluding data about previous successes, or the streaming data of certain songs. If we were to include this information, our model would be heavily carried by these data points, as artists that have had previous successful songs in the future are likely to either have a similar level of fame to their most popular song, or likely to have their next song be similarly as popular. This trend can be seen across many artists in the Spotify data set we observed, as well as other data sets explored in the preliminary results section.

We had to apply this thought process to our social media data as well. While there is a bit more variation in the social media data, in terms of who actually has official social media accounts, and who is active, we chose not to incorporate any social media follower data, as it could be a proxy for the fame that an artist already has. As an alternative, using Twitter as our way of measuring social media engagement, as it is generally the rawest way an individual can interact and start discussions with their communities, we instead looked at the metric of how frequently a user posts, with the intention of seeing if a higher engagement with the community through social media led to higher amounts of popularity for the artist.

Aggregating this information back to the central song database, our goal is to understand what drives the popularity of an artist based on song-level factors and an artist's engagement with the

outside world. The additional song-level factors we want to look at are the frequency of collaborations an artist has, the song-level features (e.g. danceability), the most popular genre an artist produces in, and the ability to diverge from a genre, etc.

Data Cleaning, Feature Selection, and Feature Engineering

Data Cleaning

Fortunately, a major portion of our data cleaning were done in the data scraping component. While going through song data for each artist, we made sure to not include artists that we don't have data for.

While looking through the Twitter API, our search script always selected the first result according to the keyword which is the artist's name. The Twitter algorithm reliably selects the most relevant and most popular account according to the search term; for musicians, there will rarely be cases in which a fan account or secondary account surpasses the real artist's account in the search results. These cases would also be extremely difficult to identify and would have required a sufficient level of text analysis that we were incapable of doing.

There are quite a few popular musicians who do not have a Twitter account, for example, Ariana Grande and Ed Sheeran, for various reasons. We had two "error" cases to consider in which the artist did not have a Twitter account. The first case we dealt with was when there were search results for artists who didn't have Twitter accounts themselves. For this case, we just picked the first search result, generally a popular fan account, as this could also be a sufficient metric for fan engagement, although not from the artist themselves. As mentioned above, this step made more sense for us than trying to identify the difference between a fan account and a real artist account, as our search results would rarely return a fan account instead of an artist's main account. The second case was when there were no search results for a given search term, or no Twitter accounts matched a certain name. We believe that this could be attributed either to an incorrect search parameter (the artist has a differently named Twitter than their artist name and hasn't set up their relevant keywords), or the artist doesn't have a Twitter account. For all of these data points, we set the number of tweets to zero, as not having a Twitter account is functionally the

same as having a Twitter account but never posting. Looking at the data, this only accounts for a small portion of Twitter accounts.

While looking through the Spotify API, we were unable to secure data on all of the artists in the Kworb.net dataset due to API issues that we couldn't circumvent. For this reason, we had to remove some of the data and proceed.

Feature Selection

As mentioned above, our feature selection is guided by the fact that we want to minimize choosing any features that could share a confounding variable with the artist's total number of streams or monthly listeners, due to the fact that our end goal is figuring out some of the size agnostic features that could lead to a larger following and greater commercial success for a specific artist. The variables that we ended up selecting are included below.

avg_streams (dependent variable)

Description: average number of listens for an artist's songs over available songs present in data
Reason for Inclusion: most reasonable and available metric for popularity and longevity for a specific artist

avg_collab

Description: average number of artists co-credited on track (excluding the main artist)
Reason for Inclusion: collaborations with other artists open up relevant artist to newer communities, which can increase their over popularity over time

genre_dom

Description: most popular genre that an artist produces music in
Reason for Inclusion: certain genres can be more popular than others in certain time periods; we are looking at genre data from the most current time period (2022)

genre_div

Description: number of subgenres an artist produces music in

Reason for Inclusion: an artist's ability to stick to what they're known for or branch out into new audiences can influence their popularity over time

avg_dance, avg_energy, avg_temp, etc.

Description: average level of song features across all measured tracks for an artist (look at Appendix section for descriptions on each variable)

Reason for Inclusion: similar to genre, certain song features such as instrumentation and speed can influence a song's popularity; generally there are normalized bpm ranges between genres

twt_freq

Description: number of tweets made total / number of weeks Twitter account has been active

Reason for Inclusion: Twitter is generally a more discussion based platform which allows for more personal engagement with fans; fans might be more likely to listen to an artist if they have higher levels of engagement with fans through social media

Feature Engineering

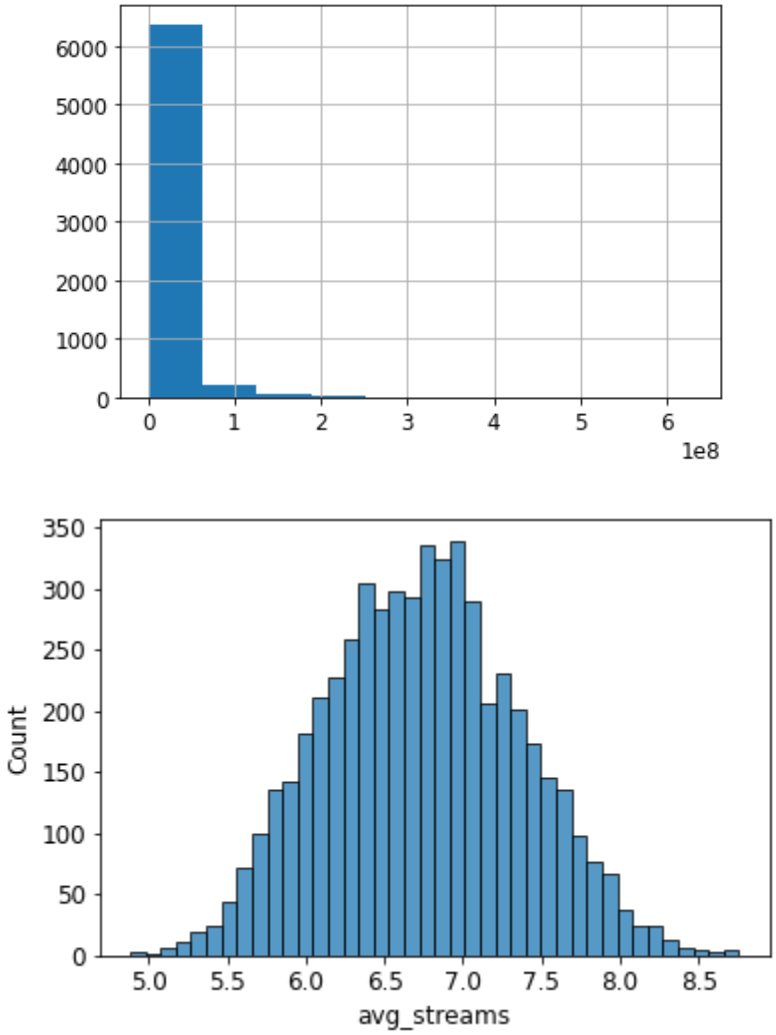
According to the variable descriptions above, we used Python and Pandas to create the relevant features.

Post-Feature Engineering Data Cleaning

With our variables created according to the above descriptions, we wanted to do one more round of data cleaning to ensure that our variables look correct. We did a log transform of our dependent variable to normalize the data a bit, as well as remove some values that became undefined, particularly from twt_freq, in the case that the number of weeks a Twitter account has been active was sourced incorrectly. The music industry is rather centralized in its concentration of resources; the top 0.01% of artists receive a significantly disproportionate amount of streams compared to the millions of artists present in the world.

Exploratory Data Analysis

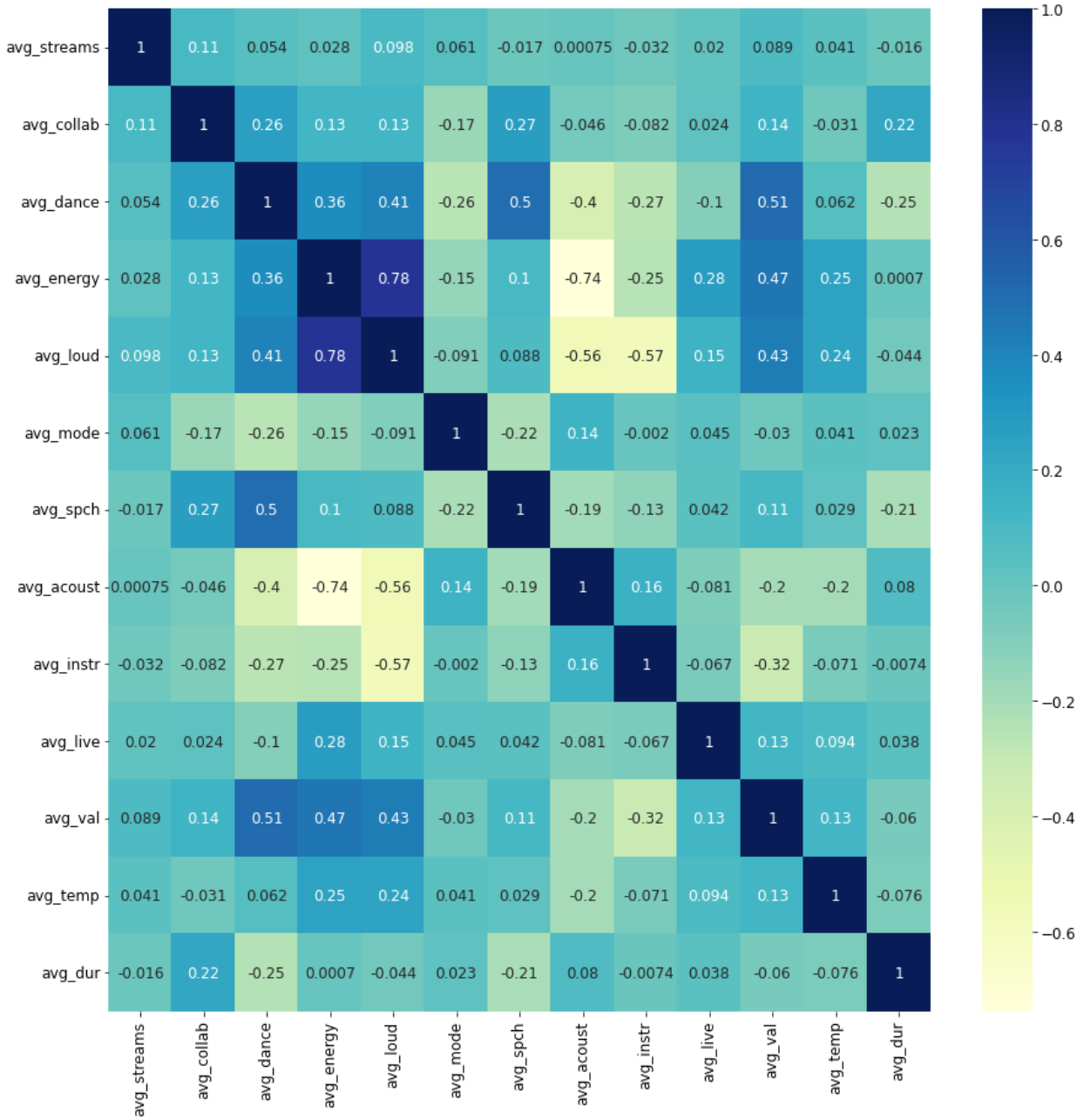
Average Streams per Song



The histogram to the left is for avg_streams. The histogram to the right is log_10(avg_streams).

As previously mentioned, we decided to conduct a log transformation of the data to get more normalized values to predict. As we can see above, the transformation gives a unimodal, more normal-looking distribution of data points.

Spotify Song Characteristics



As a first step for our feature EDA, we wanted to observe the relationship between some of the song-level features sourced through the Spotify API. We believe this will help us understand any underlying trends at a song level.

At first, there don't seem to be many very strong relationships between some of the features. However, we'll include some notable ones below.

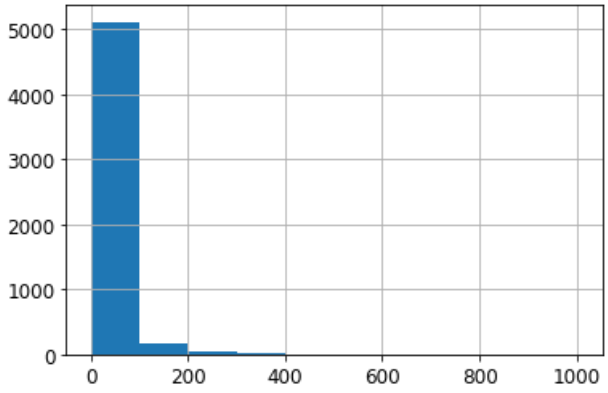
The strongest three positively correlated features are energy levels vs. loudness (0.78), danceability vs. valence (0.51), danceability vs. speechiness (0.51). At a high level, these relationships logically make sense as higher energy songs are generally longer, songs with more danceability are more positive / happy, and songs with more danceability have lyrics to sing along to.

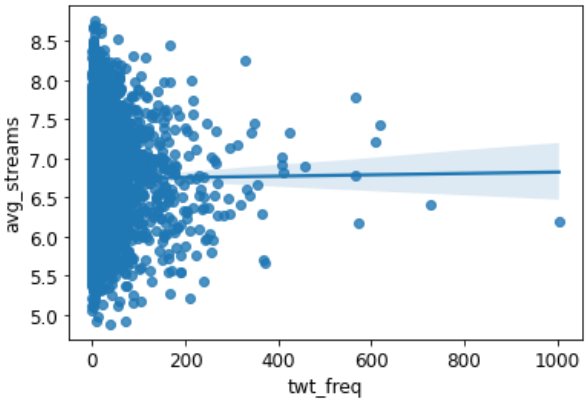
The strongest three negatively correlated features are energy levels and acousticness (-0.78), loudness and instrumentalness (-0.57), loudness and acousticness (-0.56). At a high level, these relationships also logically make sense as acoustic and instrumental songs are less likely to be loud and high energy.

Outside of these relationships, most of the feature relationships are < |0.4|, or not reasonably correlated. Now, moving on to the relationship between our dependent variables and the individual song features, we can see that all of the relationships are < |0.1| correlation, meaning they are also not reasonably correlated. We can see the correlation values raise a bit to ~0.25 for the relationship with the average number of collaborators, for speechiness, song duration, and danceability. These relationships also make sense, given that songs with more than one artist are likely to be more lyrical, longer to accommodate both artists, and have higher energy as the artists can play off each other musically.

To think a bit more deeply about why there could be low correlations between these song-level features and avg_streams, we can think a bit more deeply into how a music consumer thinks. Most consumers, including ourselves, generally tend to follow specific artists first rather than specific genres. Popular artists also generally have a wide variety in the range of song features within their music. For this reason, and considering people generally listen to various different genres, which all have different song-level characteristics. To find more information on the genre-level differences in these song features, check Appendix 2.

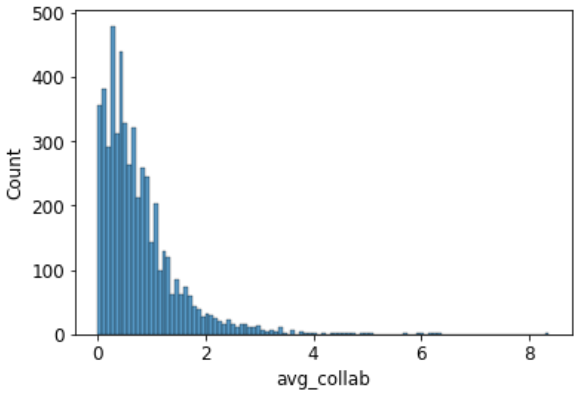
Tweet Frequency / Social Media Activity

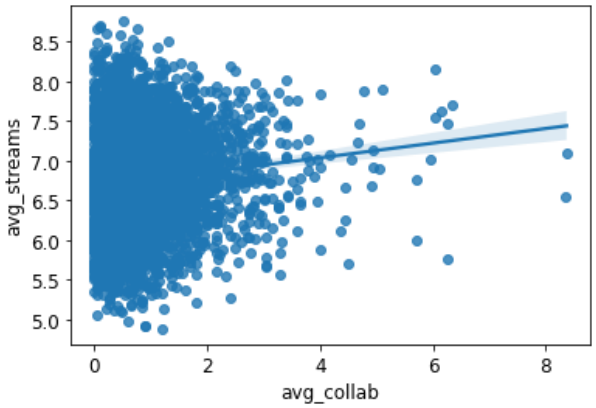




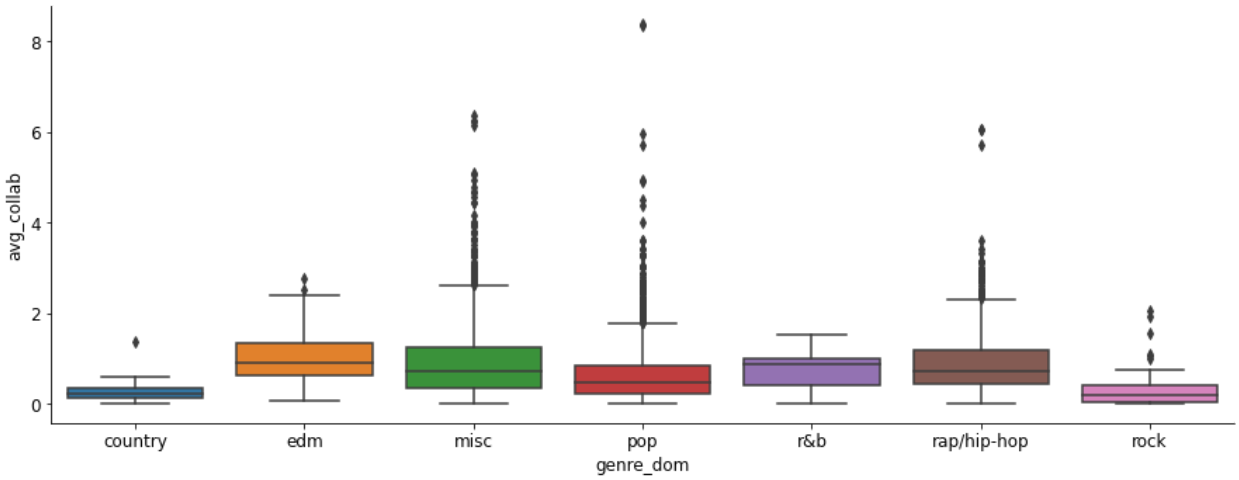
Looking above, we can see that most of the data is contained within the first bucket of data, which in this case is under 100 tweets and retweets per week. However, we have an extremely limited number of outliers in our dataset that skew our visuals. Because many of our data points are 0 tweets per week, we cannot log transform our data easily, so we left it as is. To proceed forward with the analysis, we can see that the slope of the OLS line between twt_freq and avg_streams is almost completely horizontal, meaning that there are no reasonable takeaways from the relationship between social media activity (measured by Twitter activity) and the overall commercial success of an artist. This doesn't necessarily imply that increased activity on social media and engagement with fans cannot build one's fanbase, but it simply implies that within our dataset, of some of the most commercially successful artists, there is no trend within Twitter posting frequency. This can be further seen in our linear regression model.

Collaboration Frequency



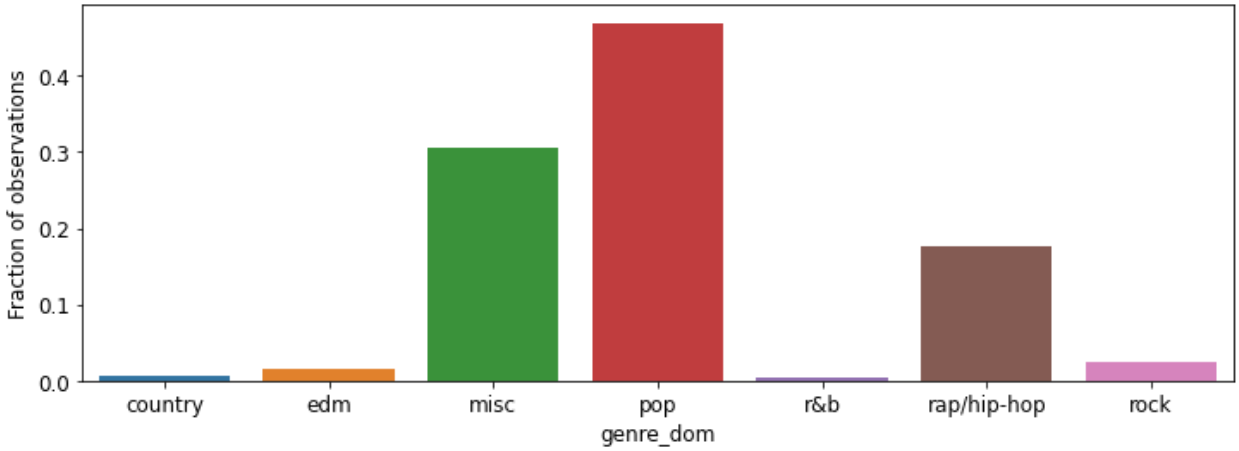


Looking at our models and the trends above, collaboration frequency actually becomes one of our most relevant features. Just based on the OLS line in the scatter plot of avg_streams vs. avg_collab, we can see that there exists some sort of positive trend. What this implies is that a greater number of collaborators on average an artist has on a song may be associated with a greater number of average song streams for an artist. This trend would make sense and is as we predicted, as this is an indicator of an artist's level of experimentation and willingness to reach out to other communities, effectively building a new follower base. Due to the nature of our dataset, there still is a lot of noise, but at the least, we're able to see a mild trend. Looking at the distribution of points in the data set a bit more, we can see that a significant majority of the data points fall in the < 4 collaborators category.



Above, we can see the avg_collab data breakdown by genre. At a quick glance, we can see that the edm (electronic dance music) genre has the highest overall (measured by median) number of average collaborators, with other genres such as r&b and rap/hip-hop, coming in a close second. One interesting trend is that the miscellaneous, pop, and rap/hip-hop categories have the greatest number of outliers, as indicated by the dots above the box-and-whiskers plot, most likely since these categories house the most data points by far.

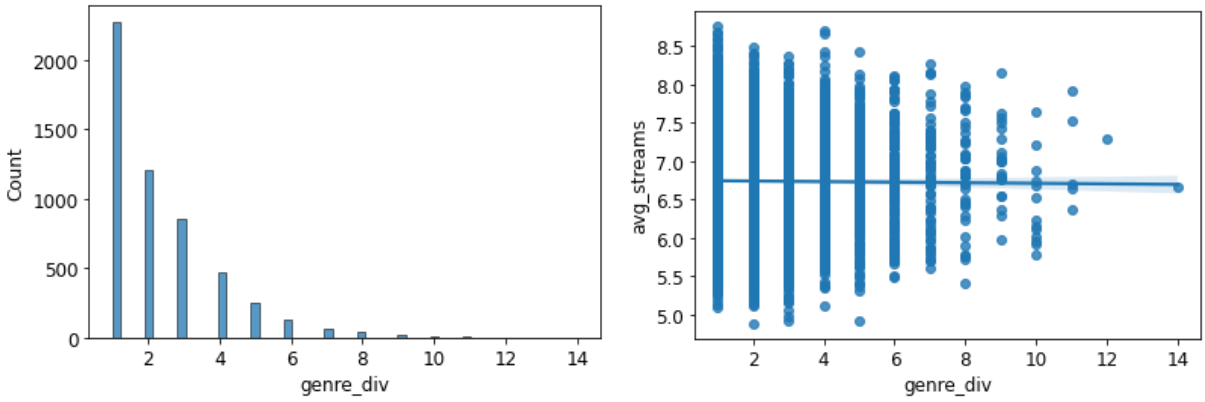
Main Genre



Above, we can find the histogram for the genres available in the training dataset. We can see that pop and rap/hip-hop are by far the two most prominent genres, which also tracks with the datasets we have previously seen that examine global consumption of music in each genre. Unfortunately, due to some problems we faced in data acquisition, we had to filter around 30% of the observations into the miscellaneous category, which includes all genres that couldn't be immediately identified by major genre keywords. By looking at some of the miscellaneous values in the dataset, we predict that most of these data points would be split roughly evenly between all of the other major categories. We included examples of the granularity of the genres provided by the Spotify API.

tony aguirre	['corrido', 'corridos tumbados', 'musica mexicana', 'nueva musica mexicana']
tony bennett	['adult standards', 'easy listening', 'vocal jazz']
tony guerra forró sacode	['forro', 'vaqueiro']
tony igy	['russian dance']
topher ngo	[]
topic42	[]
toquinho	['bossa nova', 'mpb', 'samba', 'violao']
toro y moi	['alternative dance', 'chillwave', 'indie soul', 'indietronica', 'new rave']

Genre Diversity Metric



Looking at the data above, which is the genre diversity metric of each artist, calculated as the number of subgenres assigned to each artist by the Spotify API. Most of our data falls within the 1-3 score categories, but we can see that there are still a decent amount of data points in later categories. If we take a look at the scatterplot, displaying the relationship between avg_streams and genre_div, we can again see that, similar to twt_freq, the slope of the line is rather horizontal, meaning there is no meaningful trend between the genre diversity score of an artist and their average number of streams per song. In our linear regression model down below, we can see that there is a very slight positive coefficient associated with an artist's genre diversity score.

There are two reasons we could be receiving this result from the data. One, which I find more likely, is that our process of calculating the genre diversity score wasn't sufficient enough; Spotify isn't extremely transparent about how it assigns genres to a specific artist, so there could potentially be a lot of minute subgenres that an artist creates music in which represent a larger genre, but doesn't actually imply that they branch out much into completely different genres. Additionally, as a part of our data limitations, the genre data was a bit less reliable than other features considering they don't have genre data for every artist in their database; we spent time looking through Spotify developer forums extensively and found that the Spotify API could be rather unreliable when it comes to sourcing this data specifically. Another reason why our data could be showing this trend is that our method for calculating genre diversity doesn't accurately represent how frequently an artist diverges from their main genre, which is the original way we hoped to define the metric.

Model Development and Analysis

We decided to use a plethora of models in order to determine some relationship between the features we found and the average number of streams. In particular, we decided to employ the use of a standard Linear Regression Model, a Penalized Ridge Regression Model, a Random Forest Model, and a Gradient Boosting Regression model. All would enable us to determine how best to take into account the features we have developed thus far and how we would be able to predict our proxy for popularity.

Linear Regression

For the Linear Regression model, we simply split our training and test data up into the corresponding data frames df_train and df_test, and then determine x_train, y_train, x_test, y_test individually. For the categorical variable of genre_dom, we utilized the built-in pandas get_dummies function in order to create indicator variables for each row in our regression model for every value that existed for that column.

Based on the model summary, there were several key takeaways that we can make with regard to model performance. For example, the R-squared value that was calculated for this data was approximately 0.086. This indicates a very weak correlation and as a result, weaker in terms of predictive capabilities. Moving on to the statistical significance of the various features we see, we have some interesting observations. For instance, tweet frequency, average danceability, average acousticness, and average tempo are features that are not found to be statistically significant. The song features are somewhat surprising to see given the assumptions that an artist's popularity could be reliant on some of the song's characteristic features such as danceability and tempo. In terms of the statistical insignificance of the tweet frequency, given its relatively high p-value at 0.984, there is most likely an external force causing the tweet data we have to not be indicative of any movement in the number of average streams. Below are the results of the ordinary regression analysis.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            avg_streams   R-squared:                       0.086
Model:                            OLS   Adj. R-squared:                  0.082
Method:                 Least Squares   F-statistic:                     24.94
Date:                Wed, 30 Nov 2022   Prob (F-statistic):           1.39e-88
Time:                        03:35:33   Log-Likelihood:                -4777.6
No. Observations:                5346   AIC:                             9597.
Df Residuals:                    5325   BIC:                             9735.
Df Model:                          20
Covariance Type:            nonrobust
==============================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const                      6.7241      0.208     32.259      0.000       6.315       7.133
twt_freq              -3.595e-06      0.000     -0.021      0.984      -0.000       0.000
genre_div                  0.0261      0.005      4.766      0.000       0.015       0.037
avg_collab                 0.0799      0.013      6.187      0.000       0.055       0.105
avg_dance                 -0.0075      0.153     -0.049      0.961      -0.307       0.292
avg_energy                -1.1928      0.182     -6.550      0.000      -1.550      -0.836
avg_loud                   0.0685      0.008      8.795      0.000       0.053       0.084
avg_mode                   0.1873      0.048      3.896      0.000       0.093       0.282
avg_spch                  -0.5174      0.160     -3.226      0.001      -0.832      -0.203
avg_acoust                -0.1202      0.092     -1.310      0.190      -0.300       0.060
avg_instr                  0.3306      0.102      3.255      0.001       0.132       0.530
avg_live                   0.3473      0.175      1.990      0.047       0.005       0.689
avg_val                    0.2800      0.097      2.884      0.004       0.090       0.470
avg_temp                   0.0013      0.001      1.367      0.172      -0.001       0.003
avg_dur                   -0.0352      0.014     -2.537      0.011      -0.062      -0.008
genre_dom_country          1.0768      0.091     11.787      0.000       0.898       1.256
genre_dom_edm              1.0730      0.073     14.742      0.000       0.930       1.216
genre_dom_misc             1.0843      0.042     26.022      0.000       1.003       1.166
genre_dom_pop              0.7706      0.040     19.487      0.000       0.693       0.848
genre_dom_r&b              1.0612      0.130      8.168      0.000       0.806       1.316
genre_dom_rap/hip-hop      0.8426      0.044     19.023      0.000       0.756       0.929
genre_dom_rock             0.8156      0.061     13.378      0.000       0.696       0.935
==============================================================================
Omnibus:                       37.660   Durbin-Watson:                   2.039
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               37.701
Skew:                           0.194   Prob(JB):                     6.51e-09
Kurtosis:                       2.861   Cond. No.                     1.91e+17
==============================================================================
```

We then ran a four-fold cross-validation with the LinearRegression model. Based on the cross-validation, we found that the mean R2 score determined was approximately 0.077, with a minimum score of 0.056 and a maximum R2 score of 0.0872. Based on these cross-validation values, we see that the Linear Regression model is very poor at being a predictive model for the data we are considering. Predicting the test set yields a RMSE of 0.5896.

Lasso Regression

Following our Linear Regression model, we decided to test out a Lasso Regression model. In this case, we utilized a GridSearchCV to determine the optimal parameters that would yield a model with the highest cross-validation score. We executed this GridSearch 30 times each conducting a four-fold cross-validation to determine the cross-validation score. The optimal

alpha value found for the Lasso Regression was approximately 0.000122. The penalty parameter being so low would generally not have a huge effect on the penalization of the predicted values for average streams. As such we would expect that the behavior of the lasso regression under this found optimal value for alpha would be similar to that of the results of the ordinary linear regression.

Once the optimal parameters were determined and the gridsearch concluded, we then created a new Lasso Regression model using the optimal parameters and determined the cross-validation score of the corresponding R2 value. Across a four-fold cross-validation, the average R2 score found was approximately 0.0771, with a minimum R2 value of 0.0572 and a maximum value of 0.0867. Again, this shows that, like the standard linear regression model, the Lasso regression model does not have great predictive capabilities when it comes to predicting average streams from the feature set we are looking at. One particular reason could be that both the ordinary least squares and the lasso regression models cannot capture enough of the nonlinearity that exists between the variables that we are looking at and, as such, fail to adapt to the variability of the data despite the fact there exist statistically significant predictors. When predicting on the test set, the RMSE value for this model is 0.5898, which is slightly worse than the ordinary Linear Regression model.

Random Forest

Next, we decided to go ahead with a RandomForestRegression model. There are several key insights into this model. For starters, we begin by determining the value of the cross-validation score of the standard model trained on the x_train and y_train data. Doing a four-fold cross-validation yields an average R2 score of 0.1104 with a minimum R2 value of 0.0951 and a maximum R2 value of 0.1309. At first glance, it would seem that Random Forest, compared to both linear regression models, is already stronger in predictive capability.

In order to determine the optimal parameters for a RandomForestRegression model for this dataset, we decided to go with a GridSearchCV across several parameters. In particular, we vary across the number of estimators within our Random Forest, the max_features, the min_samples_leaf, and the min_samples_split, where min_samples_split represents the

minimum number of samples needed to split an internal node and min_samples_leaf is the minimum number of samples to be at a leaf node. The number of estimators in this case represents the number of possible trees we would want to build to make a prediction. In particular, we varied the number of estimators to be between 50 and 1000 in increments of 20, the max_features functions to vary between auto, sqrt, and log2, the min_samples_leaf parameter to vary between 1, 2, and 4, and the min_samples_split parameter to vary between 2, 4, and 8. After completing the GridSearchCV, we determined that the optimal parameters include: max_features as log2, min_samples_leaf = 1, min_samples_split = 2, and 600 estimators. Conducting a four-fold cross-validation results in a mean R2 value of 0.1297, with a minimum R2 value of 0.1091 and a maximum R2 value of 0.1438. As we can see here, the predictive capability of this optimal model is much greater than that of the linear regression models. However, despite this, we still see that the R2 is still very low. The RMSE determined for the prediction on the test set using the optimal RandomForestRegressor is 0.5742, slightly better than the lasso and ordinary regression.

Gradient Boosting

The last model that we have considered for this dataset is the Gradient Boosting Regressor. Similar to that of the RandomForestRegressor, we also conducted a GridSearch cross-validation to determine the optimal hyperparameters for the model. In terms of the parameters we varied against, we looked into varying the learning rate, the number of estimators, the max depth, and the min_samples_leaf. The learning rate in this case represents the rate at which the model we are training learns to adapt its gradient, and the max depth in this case is the maximum possible depth of the decision tree that forms for the regression. Based on the outcome of the GridSearch the optimal values for each of these parameters included a max depth of 6, 100 for the number of estimators, 0.05 for the learning rate, and a min_samples_split of 2. Doing a four-fold cross-validation on this model results in a mean R2 value of 0.1131 with a min R2 0.095 and max R2 of 0.1271. At first glance, it is worse than the Random Forest Regression model but marginally better than both Linear Regression Models. The RMSE value for predictions on the test data is approximately 0.5823, worse than the Random Forest Regression model but better than that of the Linear Regression models.

Summary of Models

Model	RMSE	Optimal Mean R2
Linear Regression	0.5896	0.0770
Lasso Regression	0.5898	0.0772
Random Forest Regression	0.5742	0.1297
Gradient Boosting DT	0.5823	0.1131

In summary, after considering the models that we have run on our dataset, there are several key takeaways. For one thing, the Random Forest Model performed better than the others in terms of R2 and RMSE. The pro of using this predictive model to predict average streams for a given artist would be that this model works very well with nonlinear data and is generally robust to outliers. However, some drawbacks associated with this would be a large training time and a large parameter search time for methods such as GridSearch, which in turn may eat into the time it takes to meaningfully use the model. While Linear Regression and Lasso Regression are arguably more straightforward and more interpretable models compared to the others, their lack of performance and inherent assumptions that are being made about the relationship between average streams for an artist and the features we are looking at suggests that they are unreliable for general prediction. Gradient Boosting could be a viable alternative to the Random Forest Regression model if we wanted to focus more on efficiency and a shorter time to train. However, additional methods should be incorporated to ensure that the RMSE/R2 is not sacrificed.

Conclusions and Suggestions

Based on the models that we have tested above, it would seem as though the Random Forest Regression model performs the best out of the four based on the R2 value. However, it should be noted that this is on a relative level. Across the board, the R2 values were very low, and the differences in RMSE between the models were very small. However, as evidenced by the model summary for the linear regression model, we found that there were several statistically significant predictors for the average streams dependent variable; we confirmed this by looking at which coefficients in the linear regression model have a p-value of less than 0.05. This means that there ends up being some variables that, in fact, do predict some of the variability in the

dataset, but again, because our R^2 value is so low, we must not take our predictions too seriously. We'll first proceed by analyzing what this means in the context of the question we're trying to answer and then consider different ways that a model of this type can be improved for future studies.

First, in the context of the question we're trying to answer, to err on the side of caution, we will say that we cannot reject our null hypothesis: that there are no popularity-agnostic significant factors that determine an artist's success, measured in average Spotify streams per song. Or at the least, we cannot reject our null hypothesis with the data we used for this study. Although we found some statistically significant predictors throughout our four models, ultimately, they did not do a great job of explaining the variability in the data. This tells us that there are either other measurable factors that we missed in our feature engineering or some unmeasurable factors that contribute to this growth. Off the top of our heads, separate from the features we mentioned earlier (frequency of releases & time-based genre predominance scoring), looking at the location diversity of an artist's listeners or the music label an artist is signed to could be a meaningful predictor of success; however, this data is probably not extremely easily accessible. Two metrics that would be particularly interesting to look at would be measuring listener/popularity growth trends over time and music critic reviews of the artist's music. For the point on growth trends, which could be measured with percentages to remain popularity-agnostic, this could represent a metric to assess the virality of an artist; this data could even be cross-referenced with more currently relevant social media metrics such as TikTok data to further understand the interaction of a song and the greater world. In terms of music critic reviews, it would be interesting to observe how the increased critical acclaim of artists and their work influences their commercial success. Both of these steps would require significant amounts of web scraping and research, along with the other features mentioned earlier, that would have been much greater than the scope of this course. Coming back to the features we found statistically significant, we would recommend that artists focus on collaborating with other artists more frequently to grow their potential audience base and also push the boundaries of their experimentation while ensuring not to push away any of their current fans.

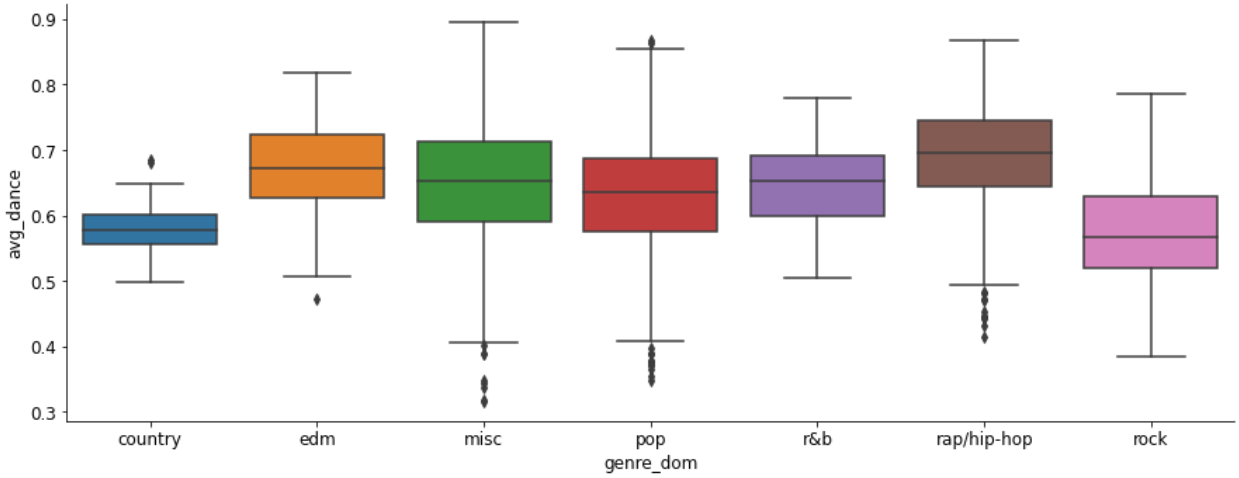
One thing we could do differently in order to improve the predictive efficacy of our models would be to use models that are more readily able to map nonlinearities between different features. One example of this would be neural networks, which are specifically designed to incorporate nonlinearity into their training and predictive processes. Another technique that could be useful would be kernel regression, which would be a nonparametric way of modeling a given distribution of data rather than imposing a specific linear constraint on it such as in linear regression. Both are different from models such as the Random Forest Regressor and the Gradient Boosting Regressor in the sense that they incorporate different, non-tree-based techniques in order to make predictions on their data inputs, and they are specifically more nonlinear in nature.

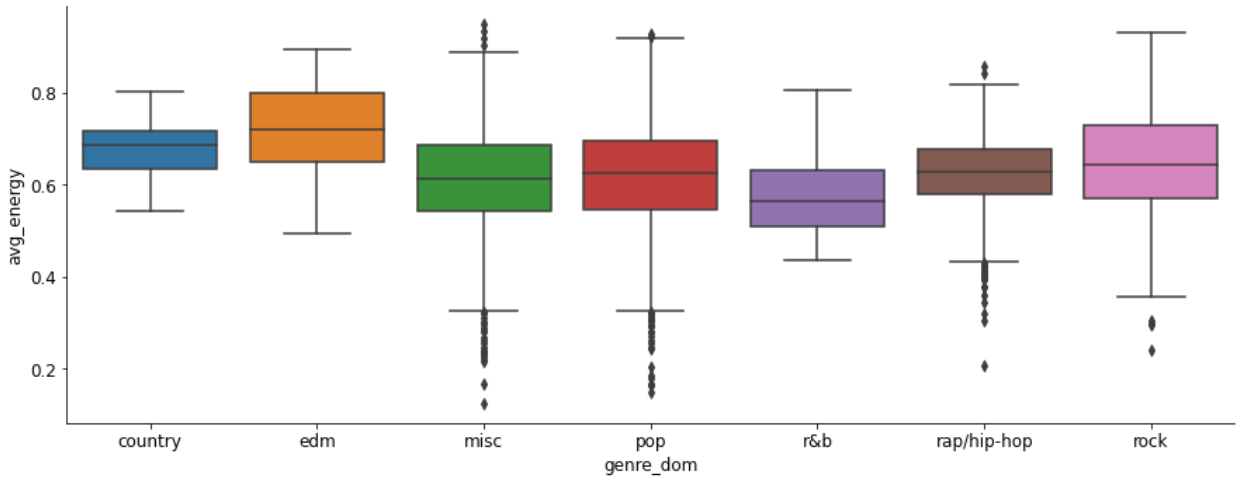
Appendix 1: Spotify Song-Feature Descriptions

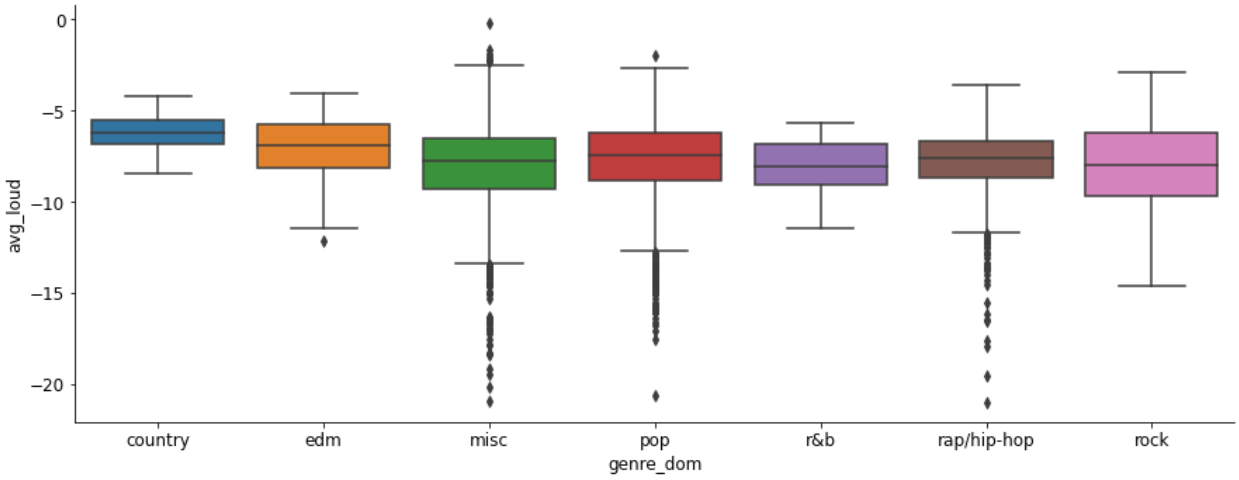
Title	Scale	Description
danceability	0 - 1	Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.
energy	0 - 1	Represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale.
key	various (pitch class notation)	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.
loudness	various (audible human range is 0 - 180 dB)	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks.
mode	0 or 1	Indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
speechiness	0 - 1	This detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.
acousticness	0 - 1	A confidence measure from 0.0 to 1.0 of whether the track is acoustic.
instrumentalness	0 - 1	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”.
liveness	0 - 1	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
valence	0 - 1	Describes 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).
tempo	various	The overall estimated tempo of a track in beats per

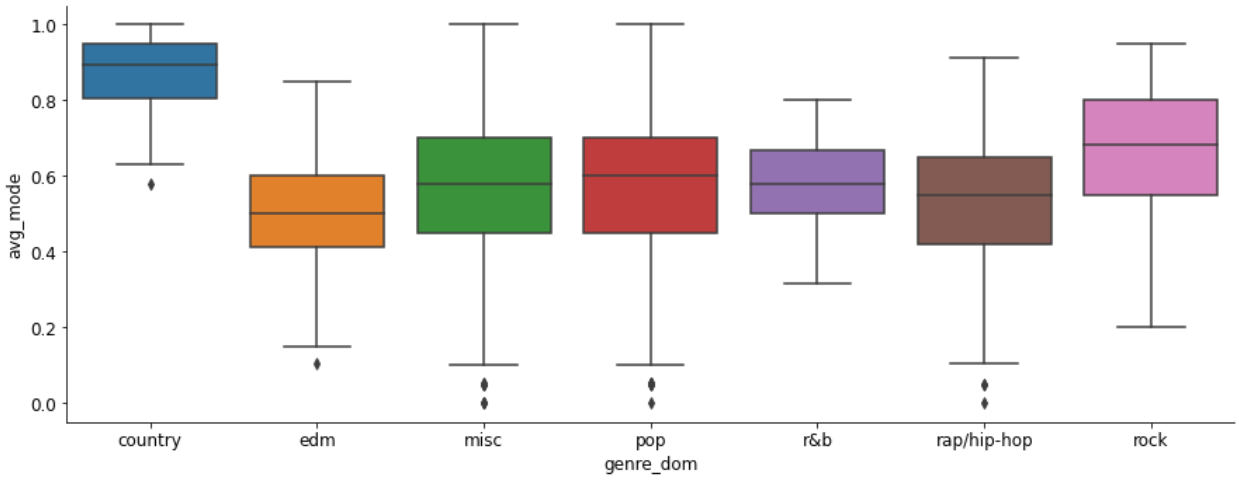
	(generally around 80 - 150 bpm)	minute (BPM). In musical terminology, tempo is the speed or pace of a given piece, and derives directly from the average beat duration.
duration_ms	various	The duration of the track in milliseconds.
time signature	various (most commonly 4/4)	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

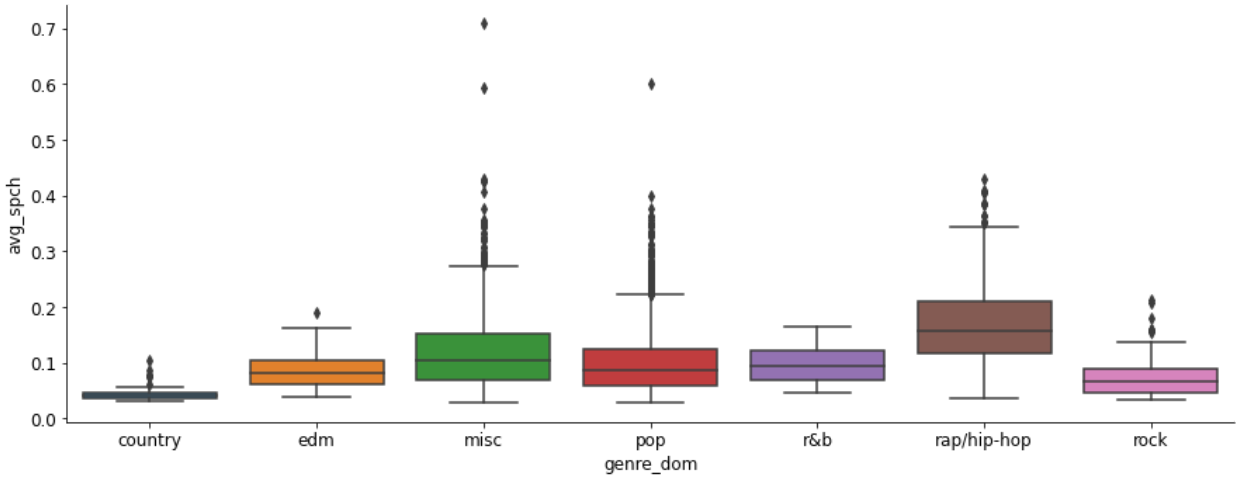
Appendix 2: Song-Level Features vs. Genre

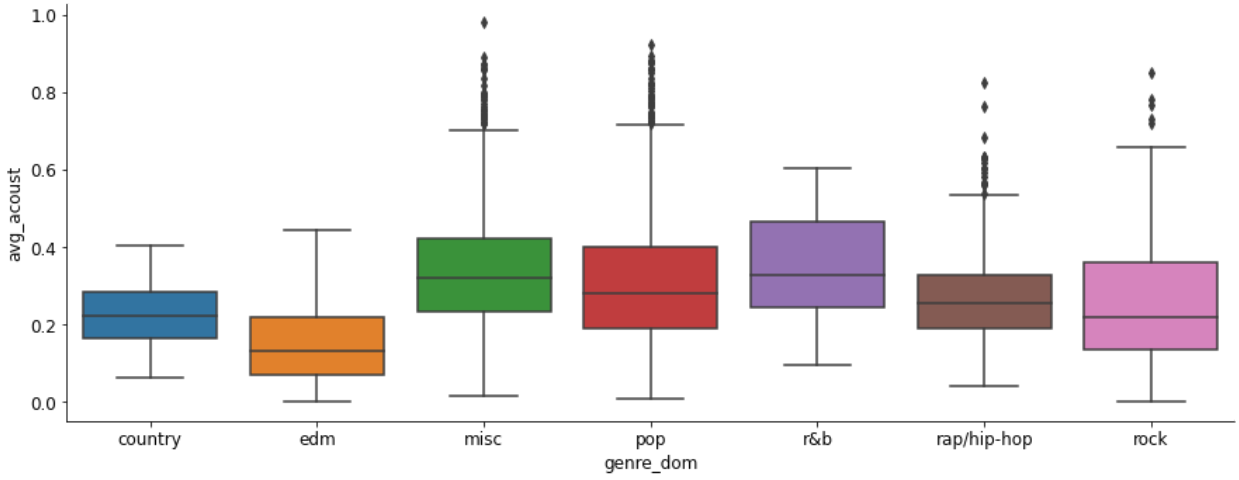


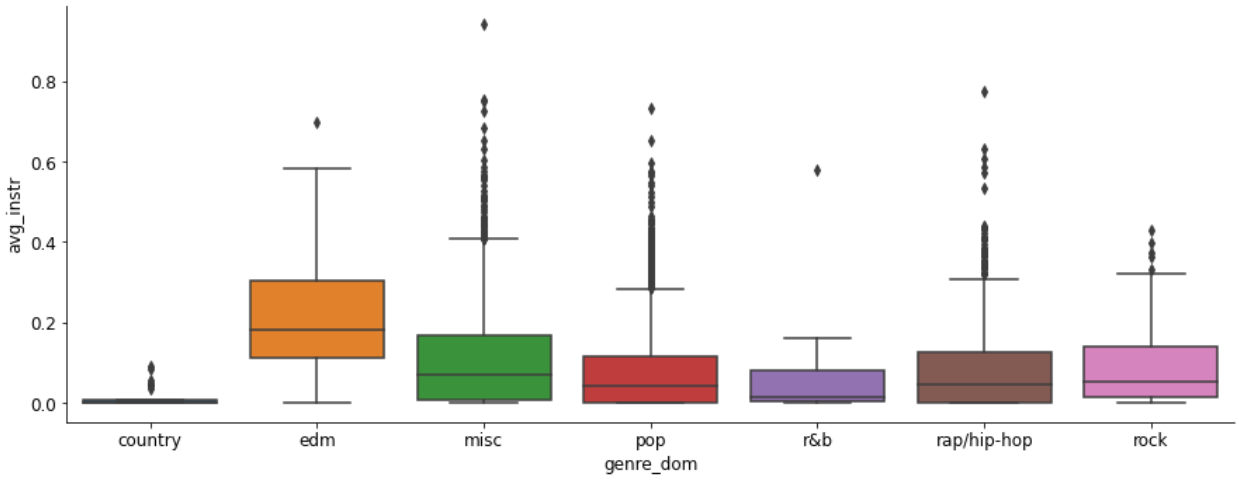


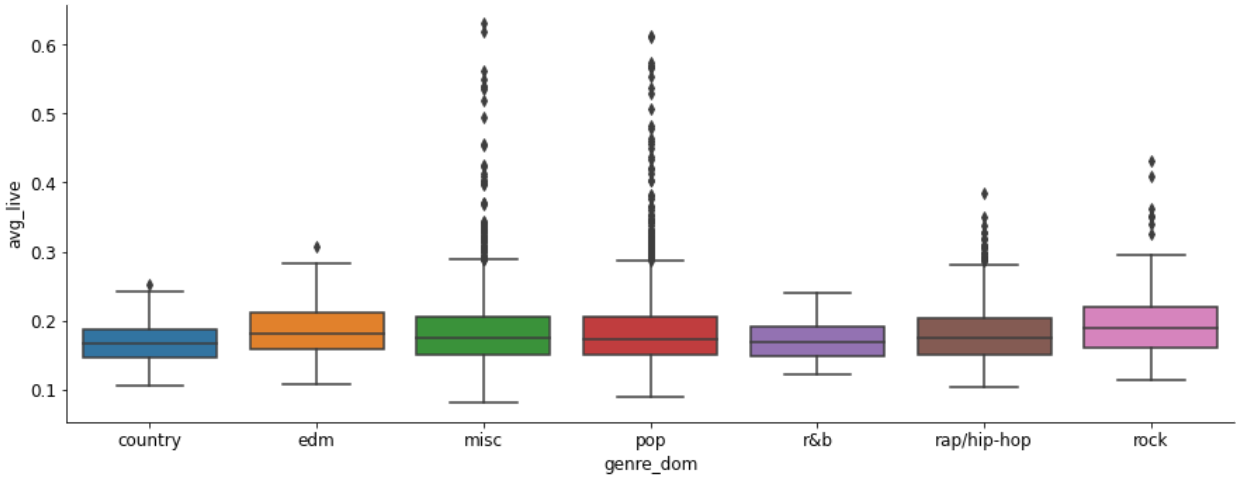


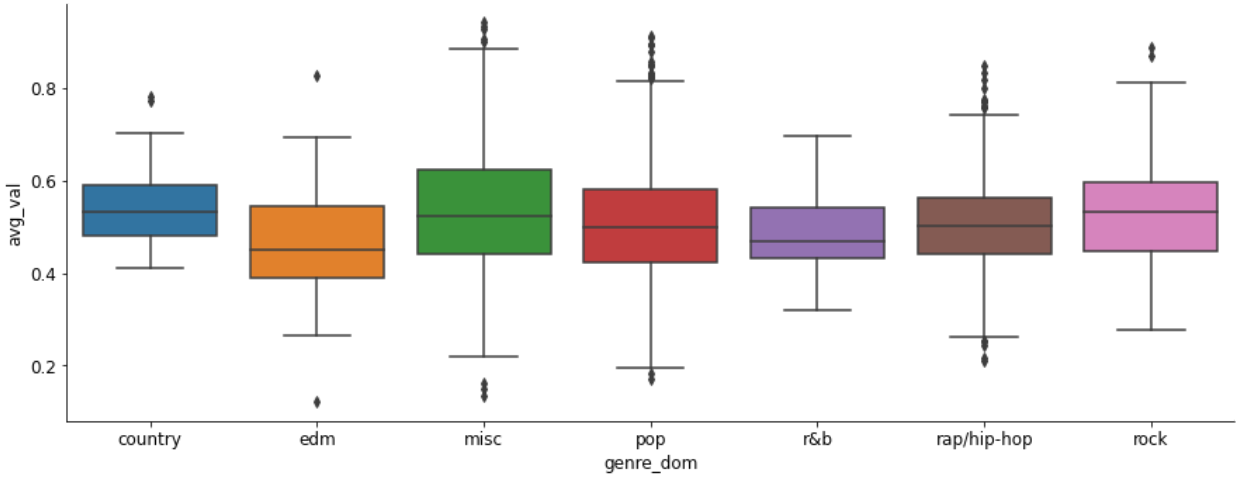

avg_mode
1.0
0.8
0.6
0.4
0.2
0.0
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom

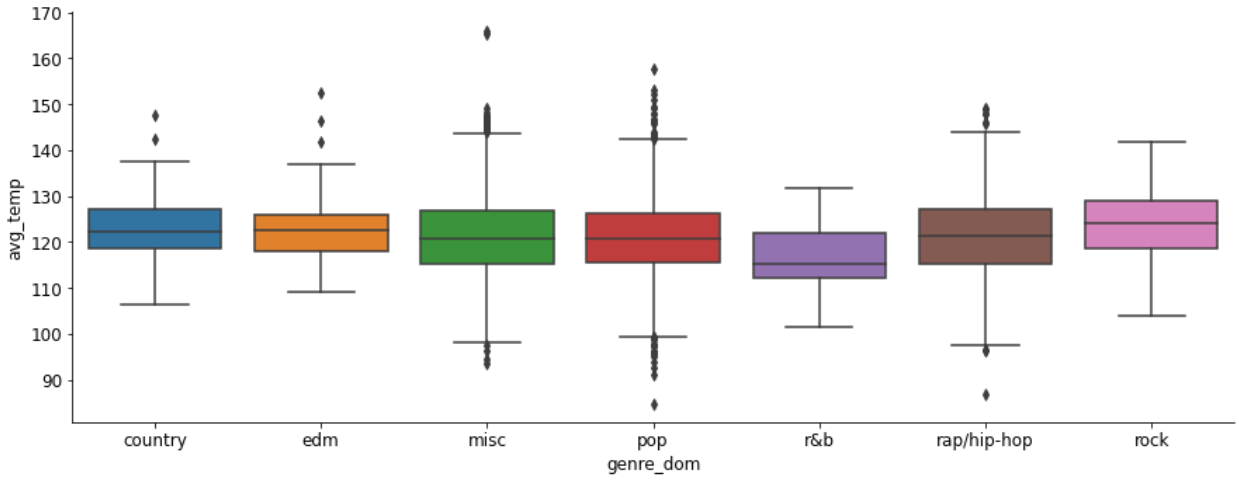

avg_spch
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom

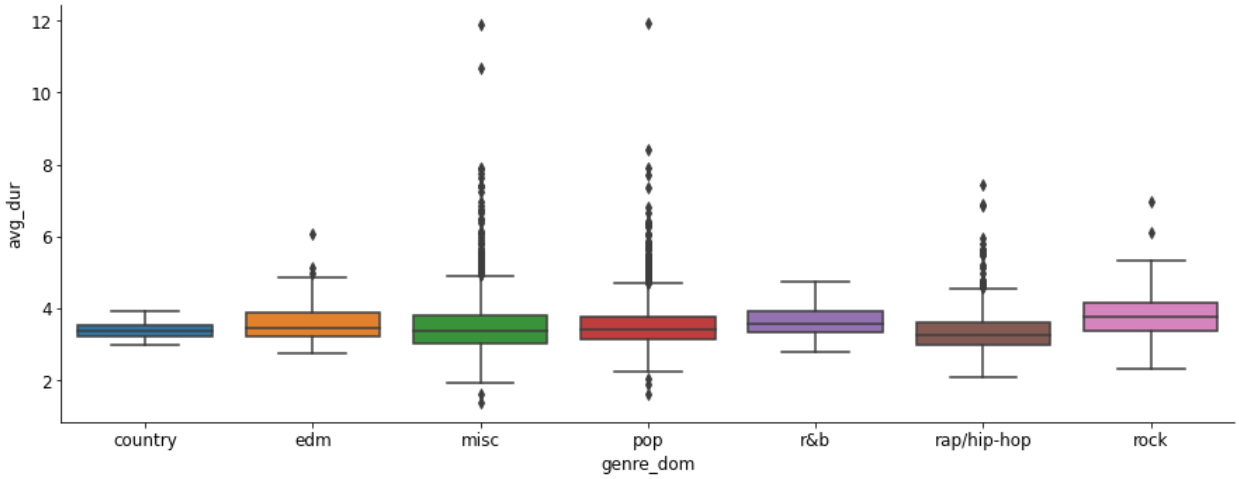

avg_acoust
1.0
0.8
0.6
0.4
0.2
0.0
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom


avg_instr
0.0
0.2
0.4
0.6
0.8
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom


avg_live
0.1
0.2
0.3
0.4
0.5
0.6
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom


avg_val
0.2
0.4
0.6
0.8
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom


avg_temp
170
160
150
140
130
120
110
100
90
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom


avg_dur
12
10
8
6
4
2
country
edm
misc
pop
r&b
rap/hip-hop
rock
genre_dom