

more_exploration

February 22, 2022

1 Executive Summary

1. Across all available metrics of success, Spotify-created playlists are much more successful than nearly all user-created playlists.
2. We consider nine metrics of playlist success, including measures of total streams and total users over the past day, week, month and two months. Amongst both the user-created and Spotify-created playlists, these measures of success are very highly correlated with one another, thus suggesting any choice of these success metrics would likely be sufficient.
3. Spotify-created playlists tend to have fewer total tracks while exhibiting fairly similar music diversity per track, as measured by tracks per artist and tracks per album.
4. The distribution of most common genre and mood are fairly similar across Spotify-created and user-created playlists. Some of the most common primary genres are Indie Rock, Rap and Pop, while some of the most common primary moods are ‘defiant’, ‘excited’ and ‘yearning’.
5. For purposes of understanding what predicts playlist success, we consider two outcomes: (1) number of users who had at least stream of greater than 30 seconds today, and (2) the number of users who had at least one stream of greater than 30 seconds in the current month and the previous month. The former outcome provides a measure of short-term success, while the latter provides a better measure of longer-term playlist staying power.
6. Across both outcomes and Spotify-created and user-created playlists, the number of tracks is the most important predictor of success, with longer playlists tending to be more successful.
7. For Spotify-created playlists, having greater musical diversity (as measured by tracks per album and tracks per artist) is associated with increased success. This effect is present for the user-created playlists, but is less pronounced.
8. Greater musical diversity is more important for a playlist’s long-term staying power than for short-term success, with one of the musical diversity metrics being the second most important predictor (behind total tracks) for the long-term success metric.
9. For Spotify-created playlists, the primary moods that are most associated with both measures of success are ‘defiant’, ‘excited’, ‘melancholy’, ‘sensual’, and ‘upbeat’. For user-created playlists, ‘defiant’, ‘excited’ and ‘upbeat’ were also associated with increased success, however ‘melancholy’ is associated with decrease short-term success.
10. For Spotify-created playlists, the primary genres of ‘dance & house’ and ‘pop’ are associated with both short-term and long-term success. “Indie Hood”, ‘Jazz’ and ‘R & B’ increase short-term success, while ‘alternative’ and ‘rap’ increase long-term success.

11. The most success-inducing primary genres for Spotify-created and user-created vary considerably. For user-created playlists, “Children’s”, “Latin” and “Traditional” increase both short- and long-term success.

2 Introduction

The data under consideration for these analyses consists of 403,366 distinct playlists, with 314,899 distinct playlist owners. Of the 314,899 unique playlist owners, 261,040 (83%) have exactly one playlist in the data. Of the owners with more than one playlist, Spotify itself has the most, with 399. The data is composed of only playlists from US owners, and thus extrapolating any of these analyses to other countries is likely unwarranted or should be done with great caution. Each playlist is categorized by its top three genres and top three moods. There are 26 genres and 27 moods under consideration.

There are a number of potential measures of playlist success included in this dataset. Specifically, we have (1) the number of streams of any length from the playlist today, (2) the number of streams greater than 30 seconds today, (3) the number of active users today, where an active user is defined as having a stream > 30 seconds, (4) the number of active users in the past week, (5) the number of active users in the past month, (6) the number of users who had a stream from this playlist for any length of time in the past month, (7) the number of active users in the previous month, (8) the total number of > 30 second streams in the past month, (9) the number of users who were active this month and the previous month. The data also contains a variable signifying the number of > 30 second streams by the playlist owner in the past month, but this analysis will focus on broader measures of success, and thus we will not analyze this variable. Finally, the data also includes the number of users who skipped more than 90% of their total streams today who also used this playlist. This variable does not directly measure the proportion of skips for each playlist, but instead measures simply whether users who skip often listen to this playlist. This variable will thus also not be included in these analyses.

Some of the potential predictors of stream success include: (1) the number of tracks in the playlist, (2) the number of tracks that were added to the playlist today, (3) The number of unique artists in the playlist, (4) the number of unique albums in the playlist, (5-7) the first, second and third most common genre found in the playlist, (8-10) the first, second and third most common mood found in the playlist. While number of tracks, artists and albums are each measures of playlist size, the most direct measure is number of tracks, so we will thus use it directly in the analyses. We will use number of tracks per artist and number of tracks per album as inverse measures of musical diversity. The dataset also contains unstructured tokens associated with each playlist. Due to the time constraints for these analyses and the unstructured nature of the data, these tokens will not be considered in the analyses.

The analyses will be structured as follows.

1. We will first explore the massive differences in playlist success between the Spotify-created and user-created playlists.
2. The substantial correlation between each of the measures of success across both Spotify-created and user-created playlists will next be discussed. A principal components analysis and factor analysis will be performed.
3. We next present the distribution of the predictors of success, including both the numeric

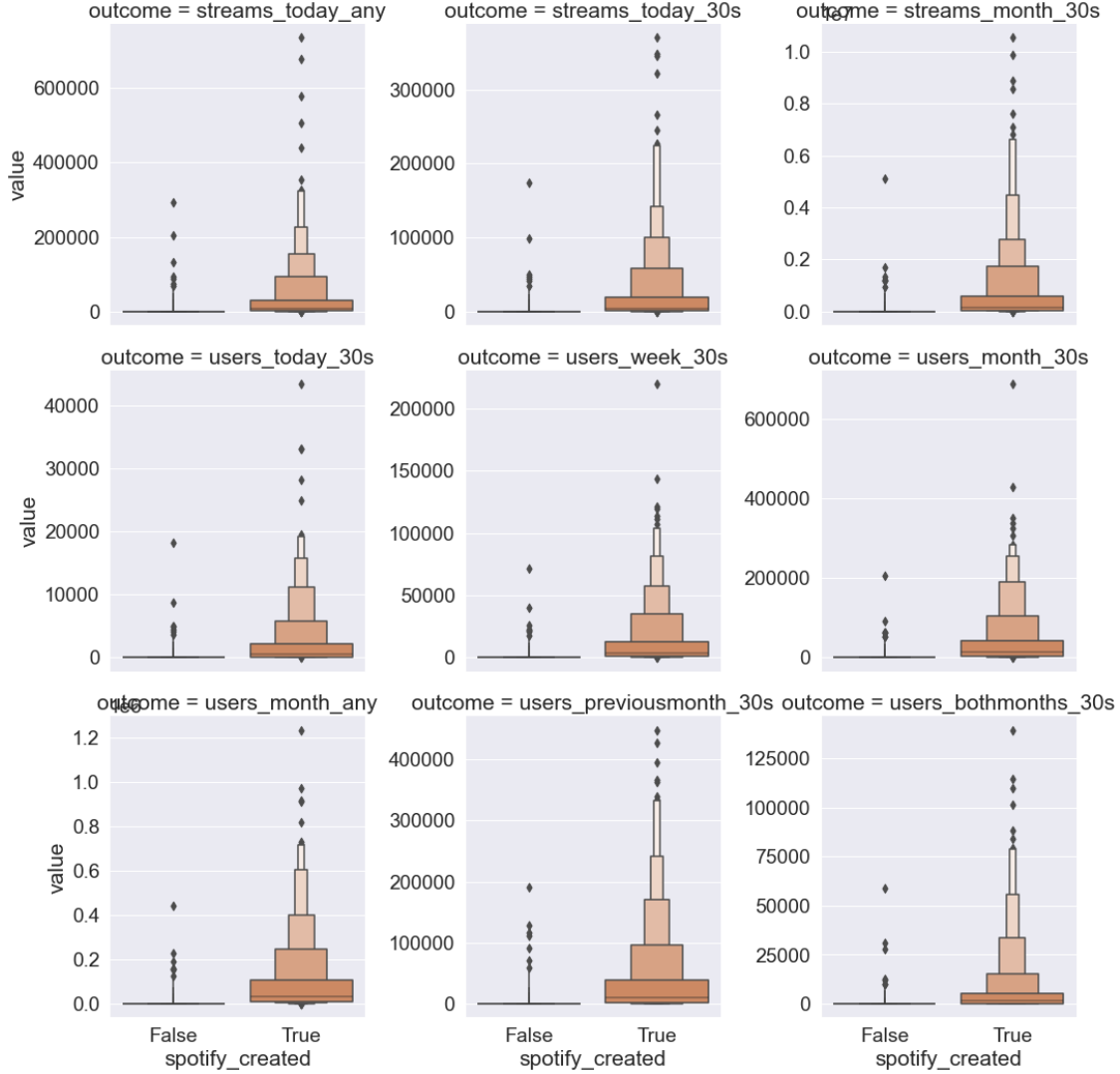
(number of tracks, tracks per artist and tracks per album) as well as the categorical (the mood and genre variables).

4. We finally fit gradient boosting models to predict measures of both short-term and long-term success, stratifying by Spotify-created and user-created playlists. Because we are interested primarily in understanding the factors affecting success, we use SHAP values to measure the most important predictors, as well as whether the predictors are positively or inversely related to success. For the categorical mood and genre predictors, we will present SHAP values for each level of the variable.

3 Comparing Spotify-Created and User-Created Playlists

There are two Spotify-created playlists that constitute extreme outliers across each of the potential outcome variables. The first is a pop, Dance & House, Indie Rock playlist with 100 tracks and has tokens ‘top’, ‘tracks’, ‘currently’, ‘spotify’. The second is a pop, R&B, Dance & House with 51 tracks and has tokens ‘top’, and ‘hits’. These two playlists have more than three times as many streams today as their nearest competitor and more than four times as many > 30 second streams in the past month as the nearest competitor. For the purposes of plotting, these playlists will be removed.

Even after removing the two most successful Spotify-created playlists, there is still a wide gulf between the Spotify-created and user-created playlists. To illustrate this, consider the boxen plot below, which shows the distribution of each of the potential outcome variables stratified by whether the playlist is creator type. We observe that the Spotify-created playlists are much more successful than the vast majority of user-created playlists. However, there are a few user-created playlists that can rival an average Spotify-created playlist.

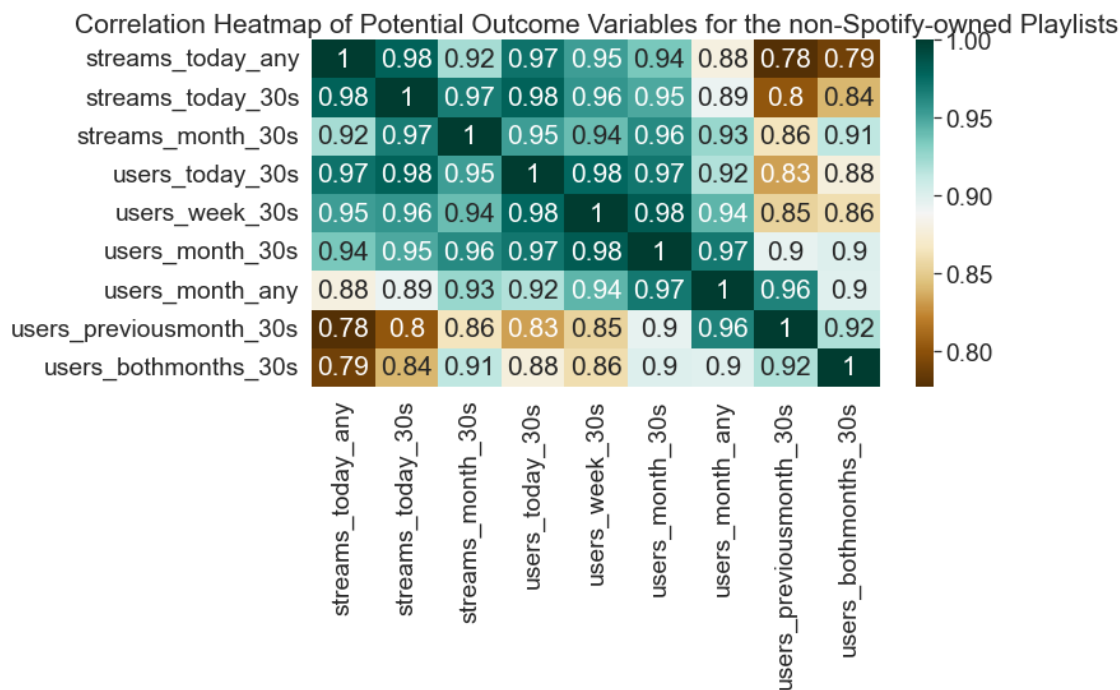


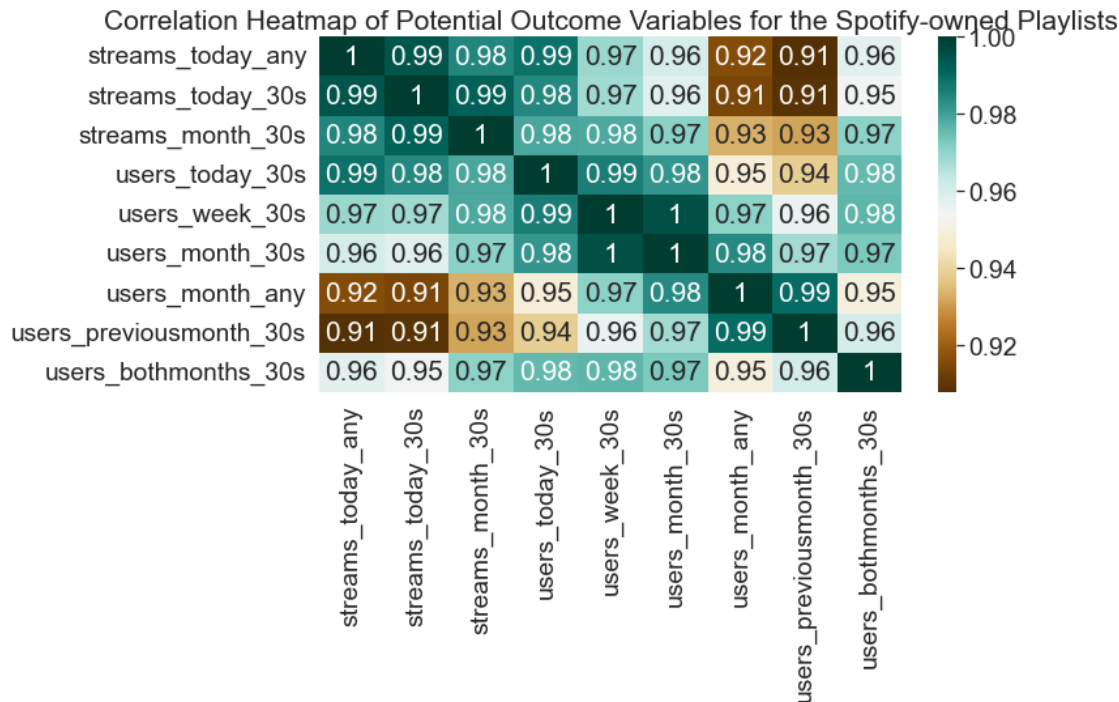
4 Exploration of Potential Outcomes

Because there is such an enormous difference between Spotify-created and user-created playlists, we will continue our data exploration stratifying by whether the playlist is Spotify-created.

To begin understanding the relationship between the potential outcomes of interest, we present a heatmap of the Pearson correlation between each of the outcomes. We create one such heatmap for the user-created playlists and another for the Spotify-created playlists. Amongst the user-created playlists, we see that the minimum correlation between any of the outcomes is 0.78, thus signifying a great deal of correlation between our potential outcomes. While all potential outcomes are highly correlated, the mostly weakly correlated outcomes are the outcomes related to playlists' longer-term success (i.e. the monthly average users in the given month and the previous month) with the more recent measures of success (i.e. the number of total and >30 second streams today and the number of active users today and in the past week). Amongst the Spotify-created playlists, the

potential outcomes are even more highly correlated, with the smallest correlation being 0.91. The Spotify-created playlists exhibit a similar general pattern as the user-created in that the weakest correlation between the outcomes is between the more long-term measures of success and the measures of success in the more recent past. However, the degree of correlation is still immense between monthly active users over the past two months and the number of streams that occurred today, thus indicating that Spotify playlists tend to have considerable ‘staying power’. Of course, if more successful playlists in the past are algorithmically pushed to users, then this could become a self-fulfilling prophecy rather than a true indication of how ‘intrinsically good’ the playlist is.



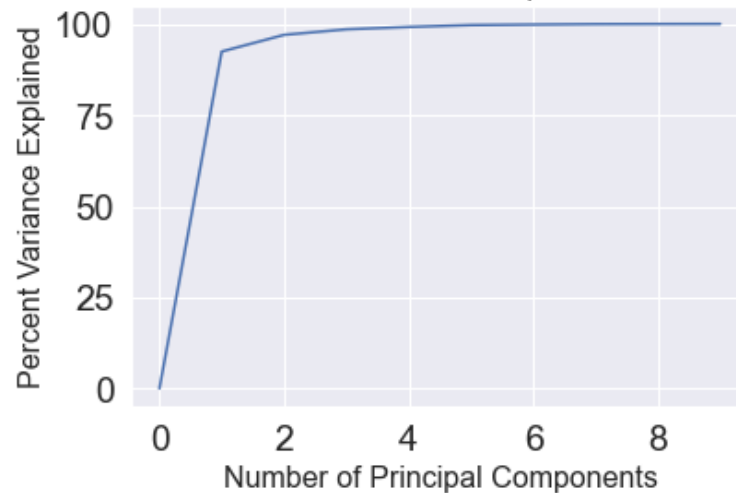


4.1 Principal Components Analysis of Potential Outcomes

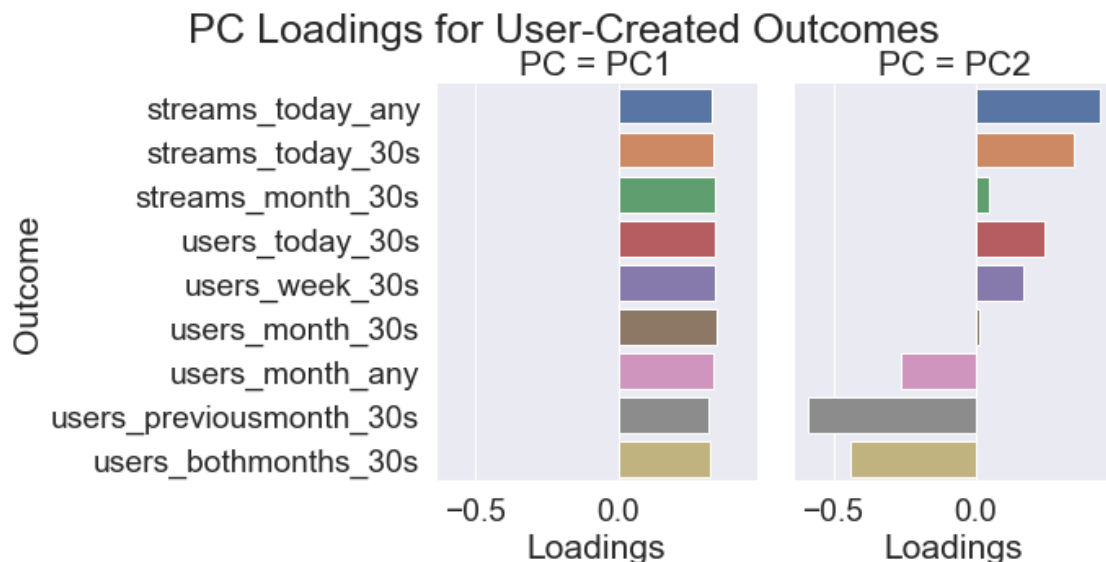
We will continue our exploration of the relationship between the potential outcomes by performing a factor analysis. We observed that the correlation matrix for both the Spotify-created and user-created playlists is quite high. We would thus next like to address the question of to what degree can this information be reduced to one or a few factors, i.e. how much of the information encoded in these 9 potential outcomes is redundant. To help answer this question, we will employ a principal components analysis (PCA) on the user-created outcomes data. The results for the Spotify-created outcomes data are similar and can be found in the Appendix.

From the plot below, we see that the first principal component explains approximately 92% of the variation across the 9 potential outcomes, thus indicating that there is a great deal of redundancy in these measured outcomes. The second principal component explains another approximately 5% of variance. We can thus explain 97% of the variability of the 9 outcomes with 2 numbers (i.e. components).

User-Created Outcomes: Percent of Variance Explained vs. Number of Components



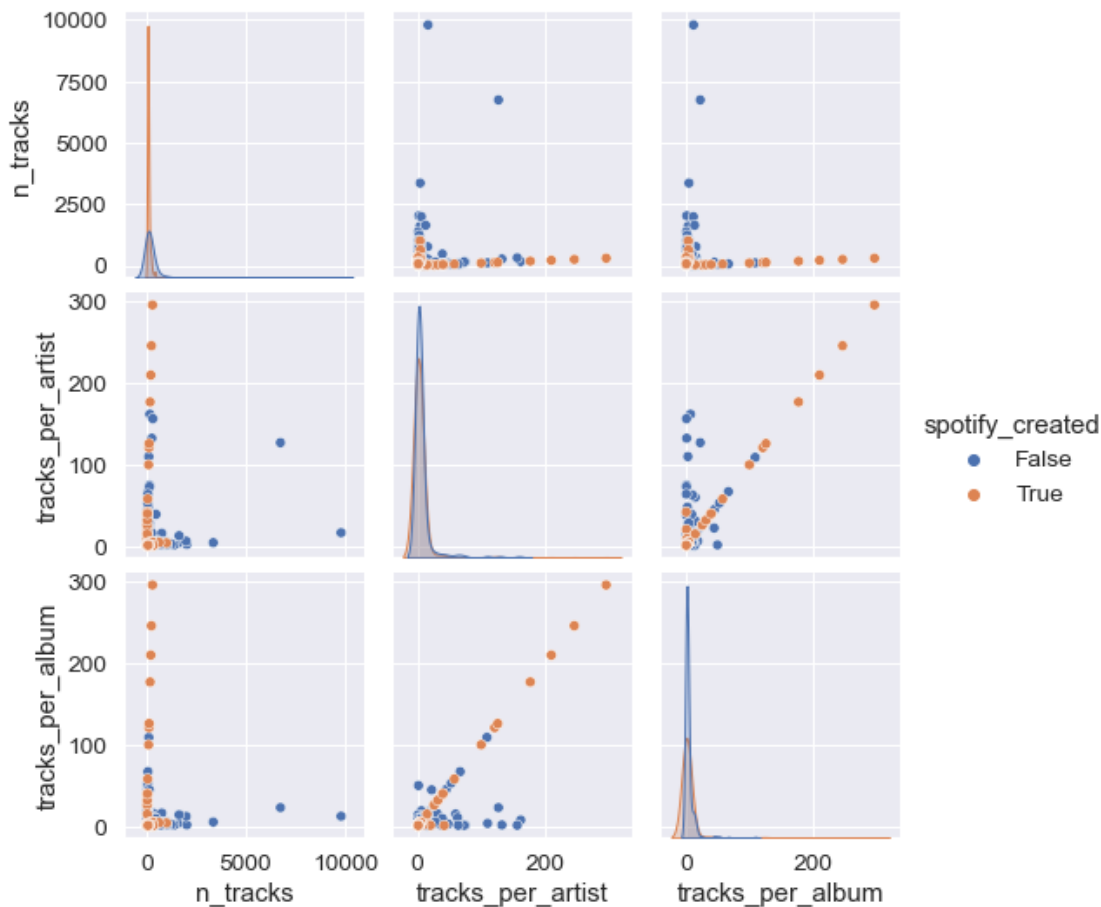
Let's next consider the loadings of each of the first two principal components. In the plot below, we see that each of the 9 outcomes are weighted very equally for the first PC. This is to be expected given the high level of correlation amongst the predictors. Looking at the second PC, we see that the number of users in the previous month and the number of users in past two months have a sizeable negative weighting while the total streams today variables have noticeable positive weighting. This indicates that there are some differences in the behavior of the "total users over the course of months" variables and the "total streams today" variables. We, however, should be careful not to overemphasize these differences, given the high correlation between the variables and the fact that the first PC explains 92% of variability across the outcomes. Nevertheless, when we proceed to modeling playlist success, we will consider two outcomes: total users who had a stream in both months and total users who streamed today.



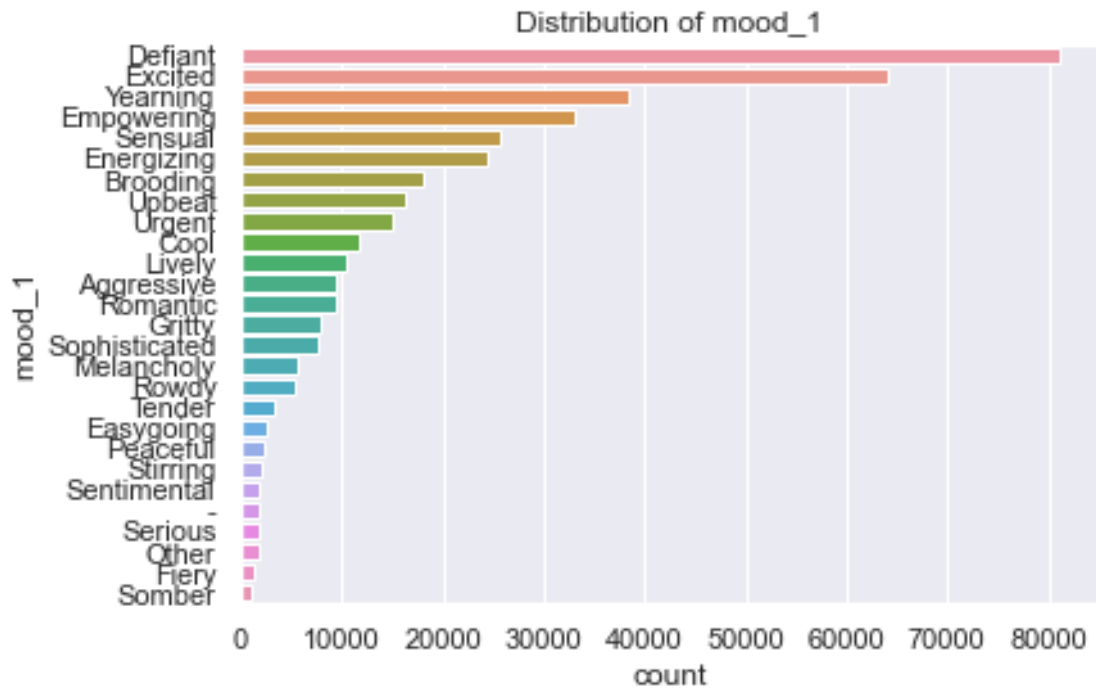
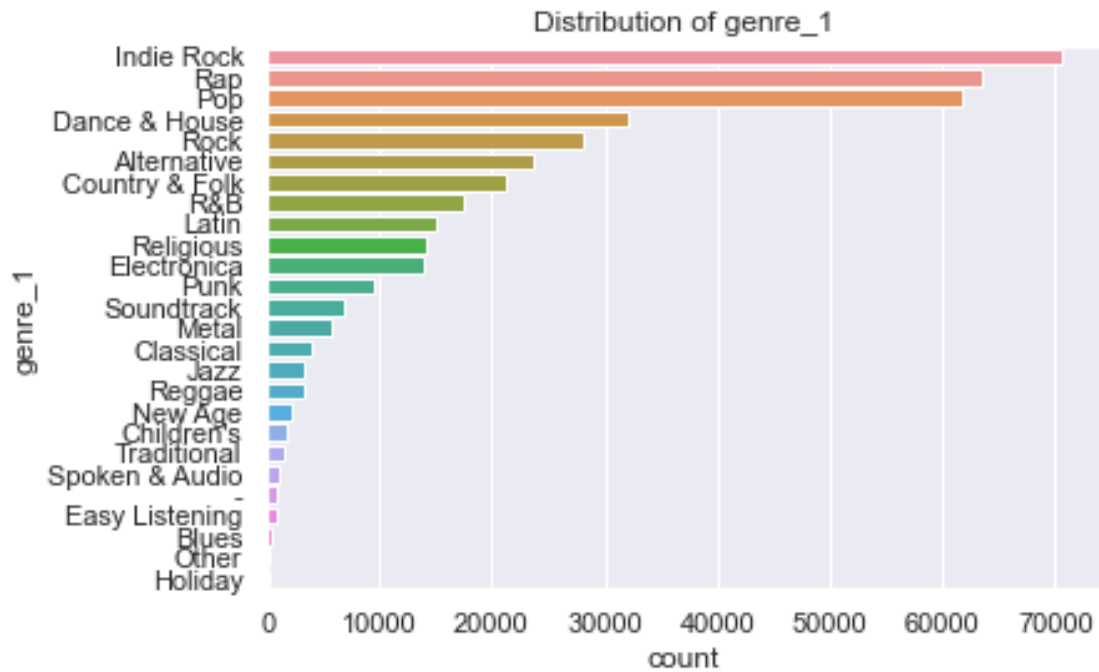
4.2 Exploration of Predictors of Interest

Note that for the plots below, we drew a random sample of user-created playlists for these plots in order for them not to visually swamp the Spotify-created playlists. Because the number of artists and albums in a playlist will tend to increase as the total tracks in a playlist increase, we define tracks per artist and tracks per album as scaled measures of the number of artists and albums that account for the overall size of the playlist. Also, note that the distribution of number of tracks, artists and albums does not vary between Spotify-created and user-created to the degree that the outcome variables do. However, we still observe some difference between user-created and Spotify-created playlists, namely that user-created playlists tend to have more tracks, artists and albums, with some user-created playlists having many more of each.

Let's consider the correlation of the new scaled measures of artists and albums as well as whether we observe differences between the user-created and Spotify-created playlists. Notice that the distribution of tracks per artist and tracks per album appear quite similar for the user-created and Spotify-created playlists. Spotify-created playlist appear to exhibit a little bit more artist and album diversity than the user-created playlists, but the difference is miniscule compared to the outcome variables.



We now turn our attention to describing the categorical predictors: genre and mood.



5 Predicting Playlist Success

We now turn our attention to predicting playlist success as measured by (1) the number of users who had a stream of greater than 30 seconds today (i.e. ‘users_today_30s’) and (2) the number of users who had a stream of greater than 30 seconds during the current month and the previous month (i.e. ‘users_bothmonths_30s’). We choose ‘users_today_30s’ because amongst the user-created playlists, it very highly correlated with streams_today_any, streams_today_30s, streams_month_30s, users_week_30s, and users_month_30s (with the minimum correlation being 0.95). It is also less correlated with users_previousmonth_30s (0.83) and users_bothmonths_30s (0.88). We choose ‘users_bothmonths_30s’ because (1) it is less correlated than many of the other outcomes and (2) it is a good measure of a playlist’s longer-term staying power. We build the models for both the Spotify-created and user-created playlist separately.

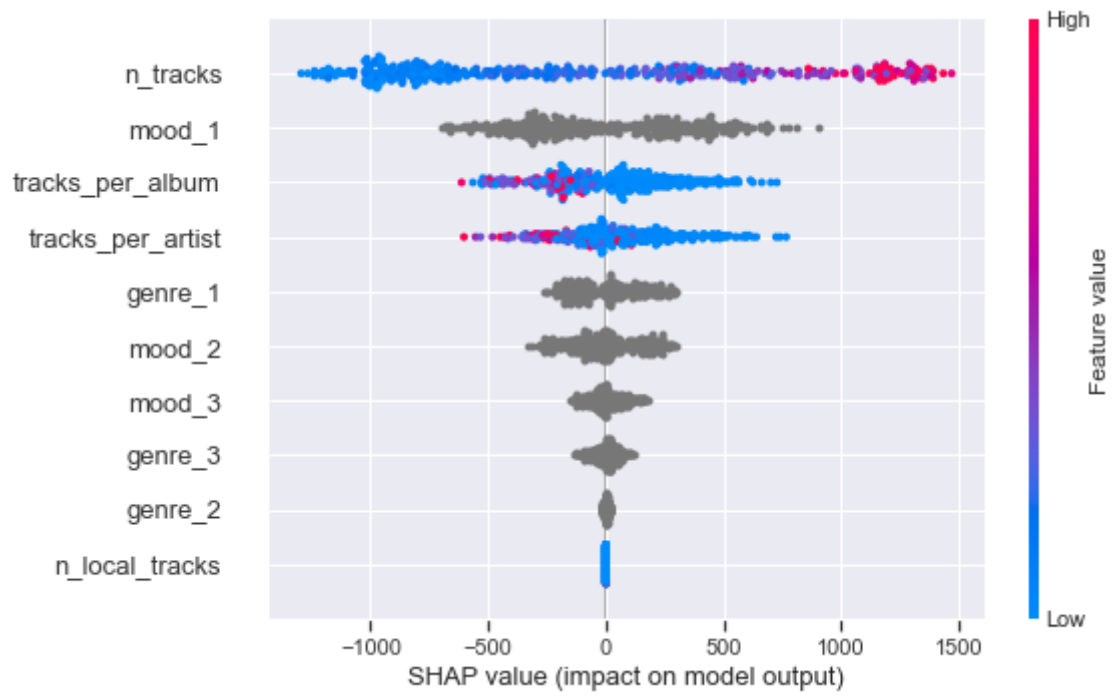
The modeling process that we employ is as follows. We train a lightgbm gradient boosting model using using tenfold cross validation. We employ early stopping so as to avoid overfitting and prevent wasting time and compute resources. The Huber loss function is used in order to reduce the effect that large outliers will have on the model. Based on the best number of trees to build as determined by cross validation, we then refit the model using all the data and fit the given number of trees. Because of the nature in which the lightgbm package handles categorical data, we do not have to one-hot encode the categorical genre or mood data, instead only having to input these as categorical columns in pandas. Note that all hyperparameters are left equal to their default values.

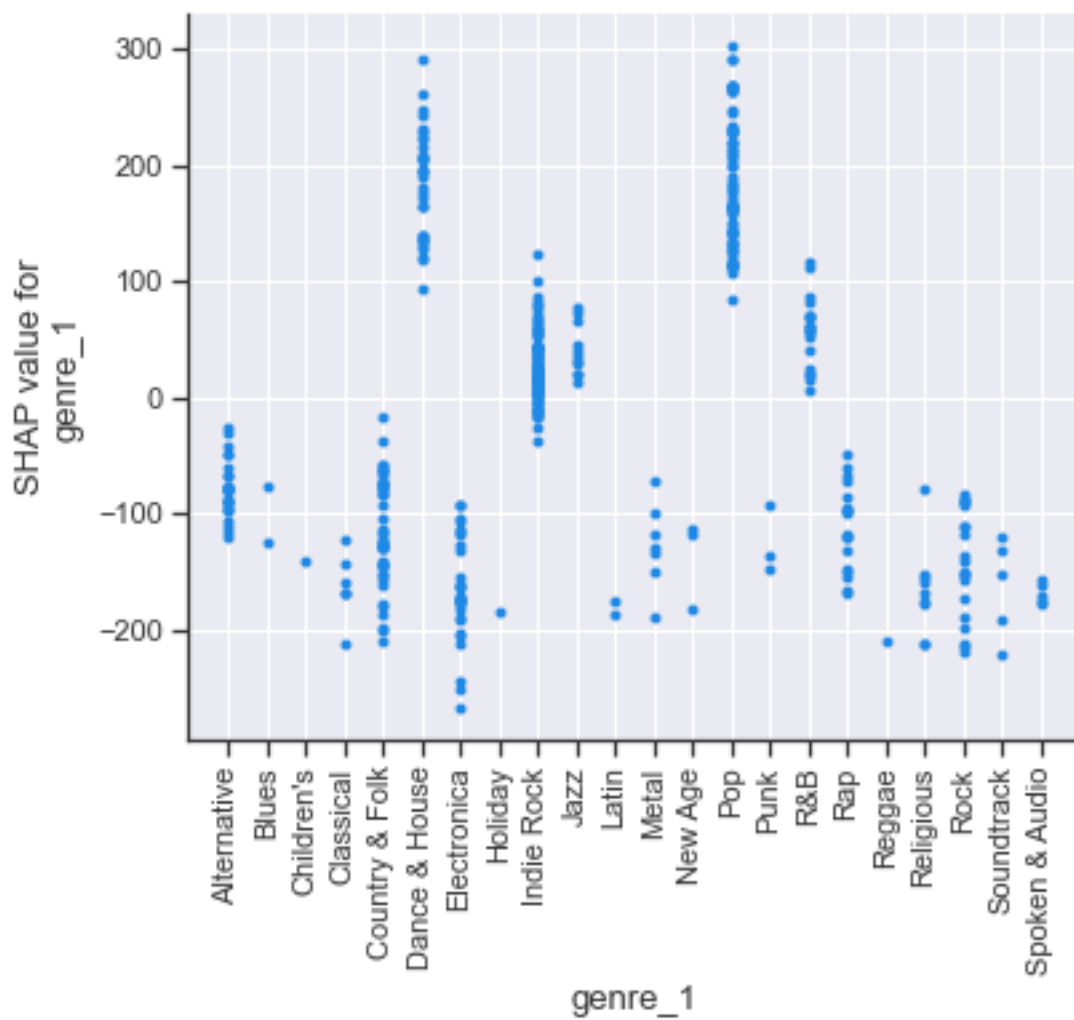
After fitting the models, we will then use SHAP values to help draw inference from the models. In particular, we will show plots of the most important predictors ranked by their mean absolute SHAP value. These plots will also highlight the directional effect of the predictor (i.e. the plots will showcase whether the given variable tends to be positively or negatively related to the outcome). We will also show plots for the categorical genre and mood variables that highlight which genres and moods tend to be related to higher and lower levels of playlist success.

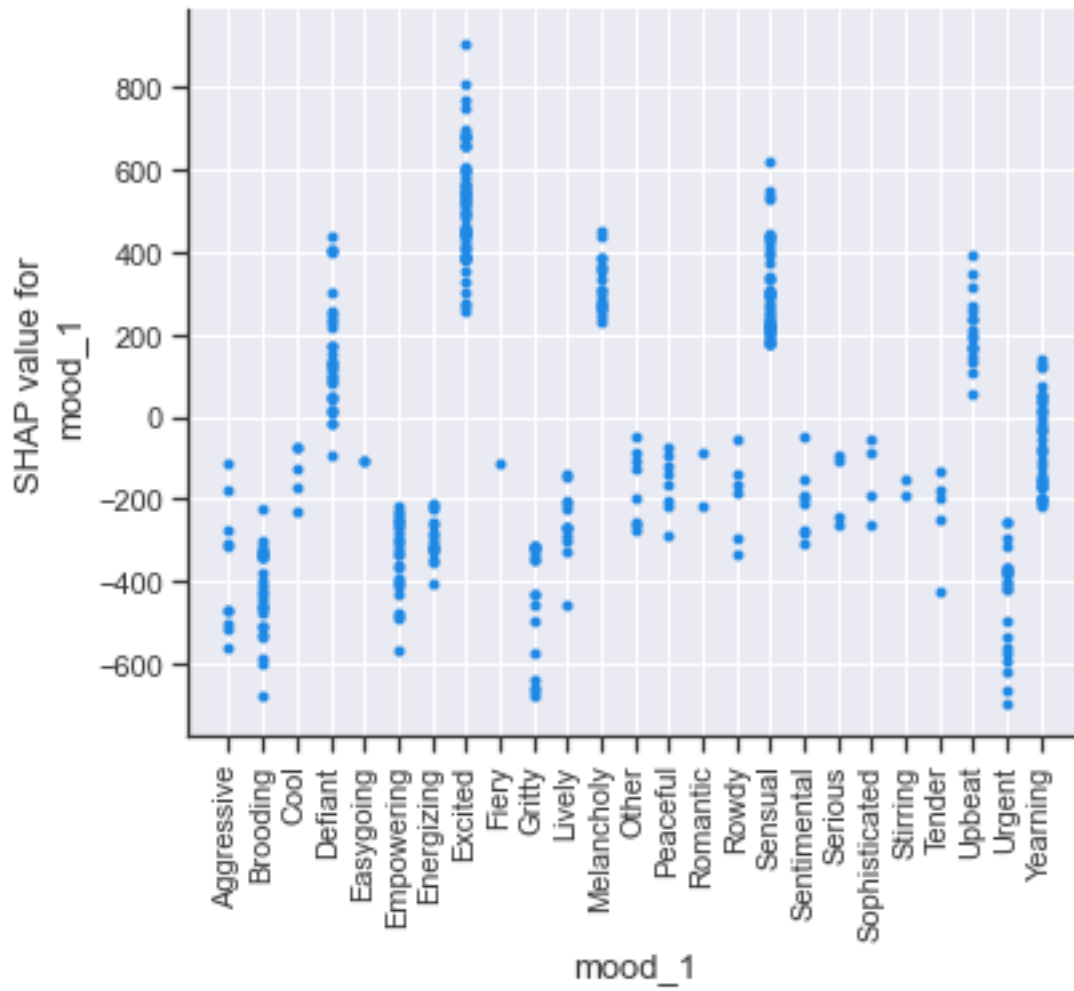
5.1 Modeling Results for Number of Users Today Amongst Spotify-Owned Playlists

From the first plot below, we see that amongst the Spotify-created playlists, the most important variable predicting success is the number of tracks, followed by the primary mood, tracks per album and tracks per artist. Also, notice that large values of tracks (colored in pink) tend to be associated with larger SHAP values. Thus, we find that playlists with more tracks tend to have more users today. On the other hand, notice that for tracks per album and tracks per artist, the large values tend to be associated with negative SHAP values. Thus, large tracks per artist and tracks per album tend to be associated with reduced playlist success. Said another way, increasing the musical diversity of a playlist tends to increase playlist success as measured by number of users today.

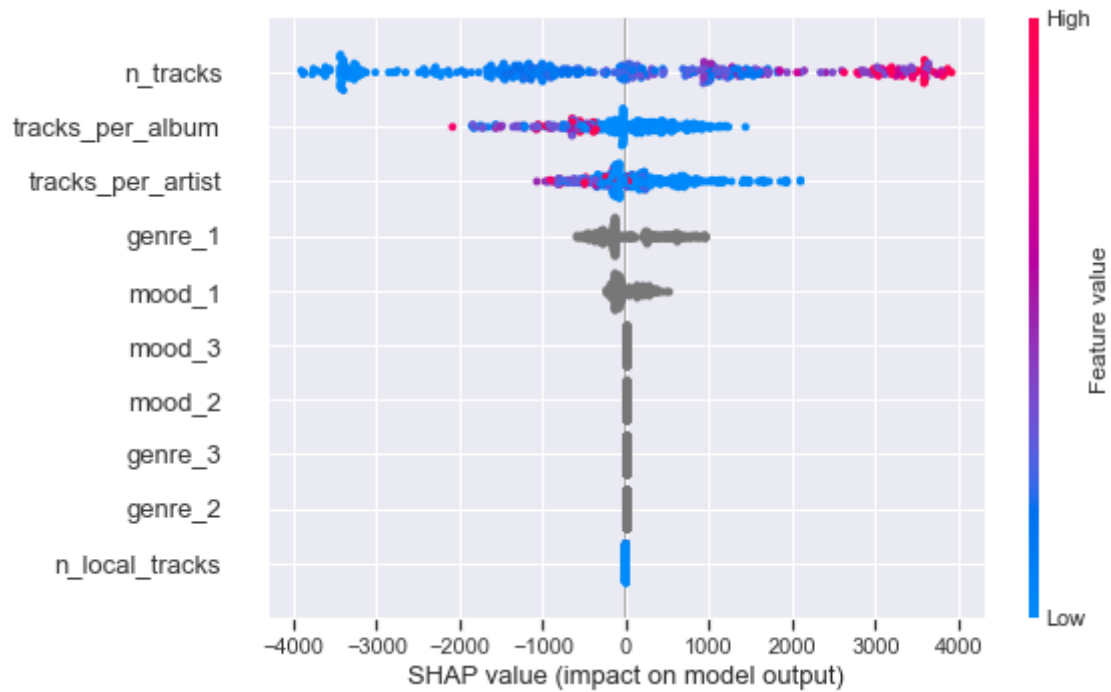
The first plot does not easily allow us to understand individual levels of the categorical predictors of mood and genre. We thus plot SHAP values for each of level of the primary genre and primary mood below. From the plot for primary genre, we can see that ‘Dance and House’ and ‘Pop’ music tends to be positively related to playlist success, while ‘Country and Folk’, ‘Electronica’ and ‘Rock’ tend to negatively related to success (again, as measured by number of users today).

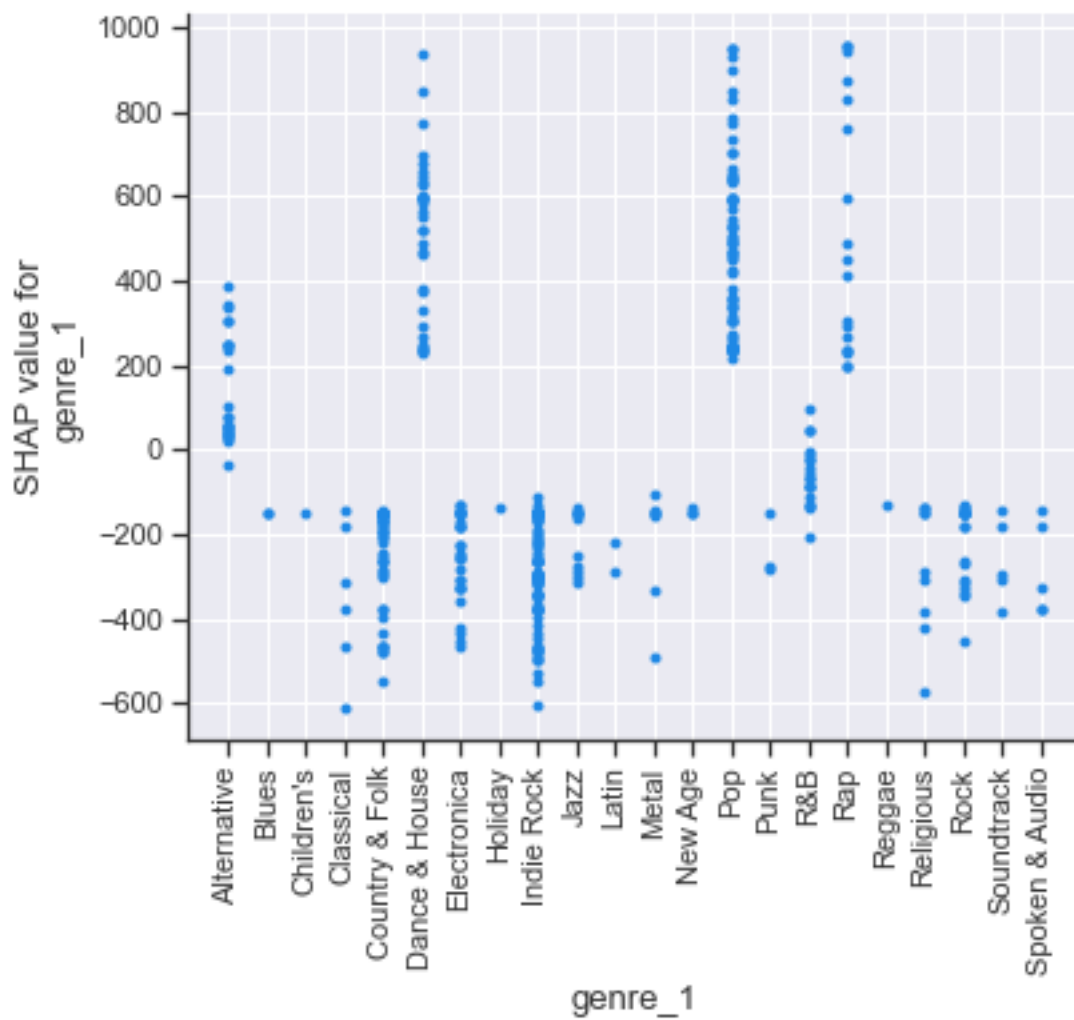


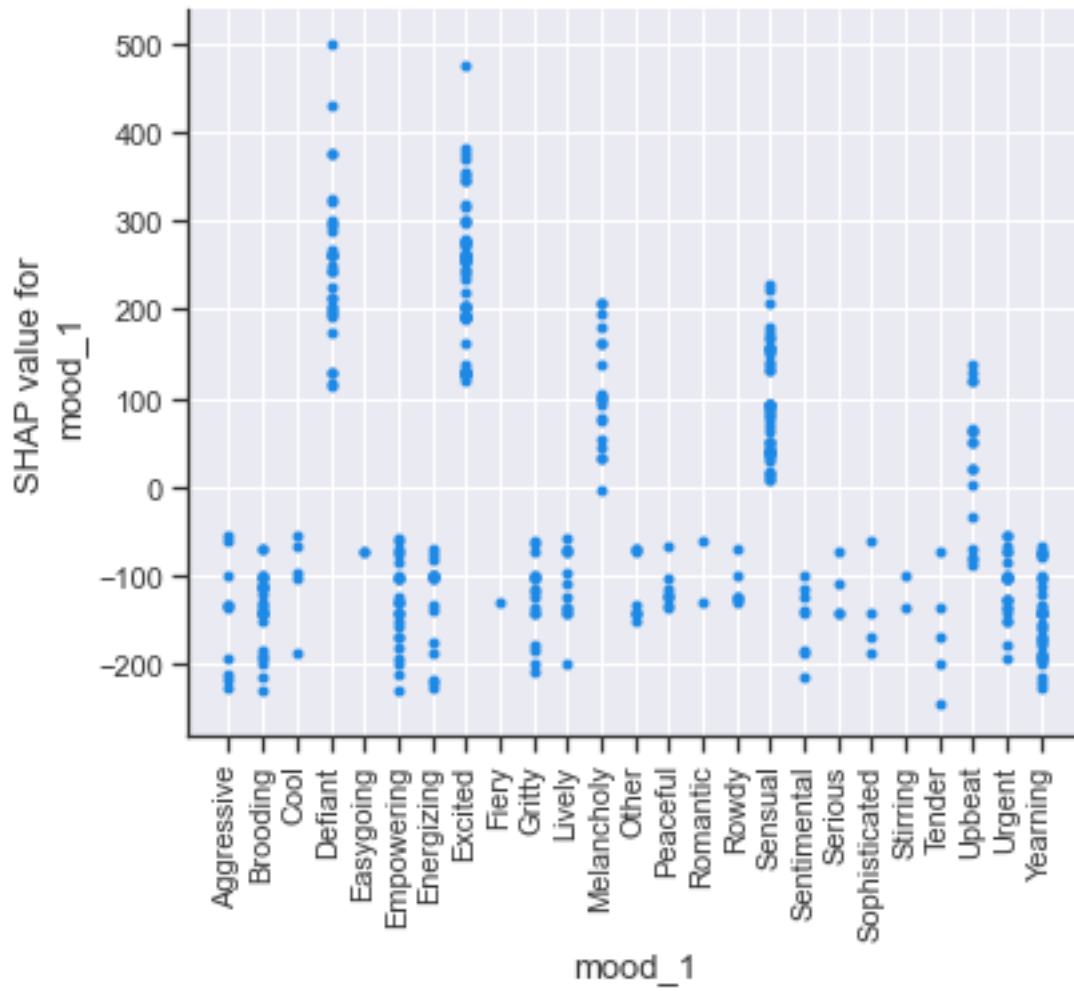




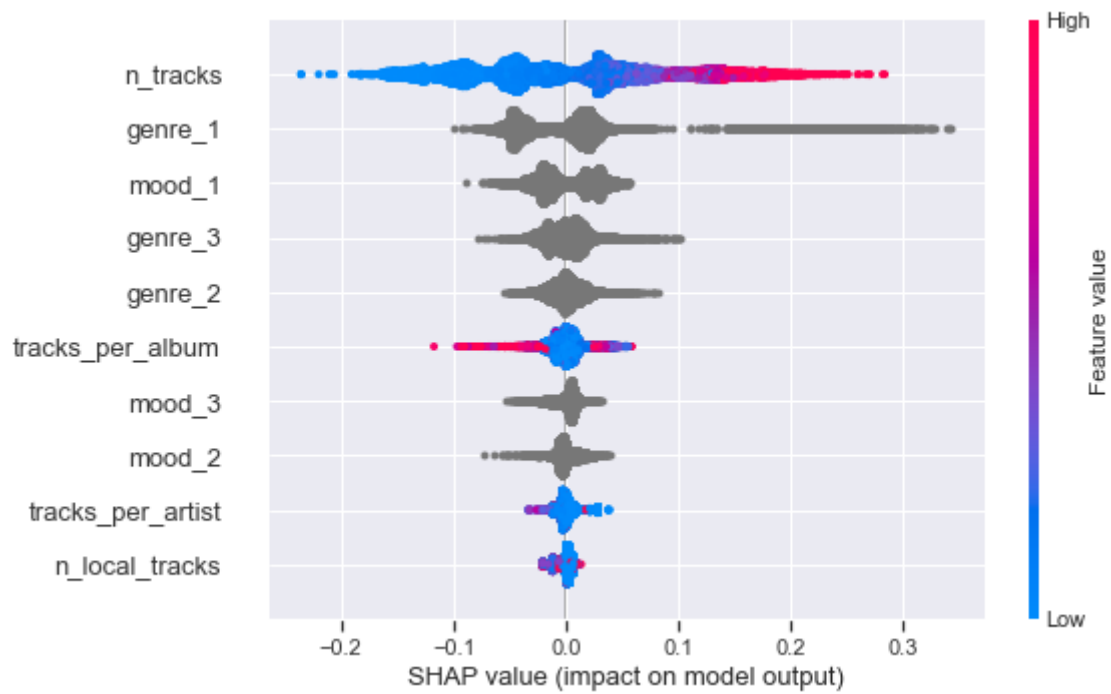
5.2 Modeling Results for Number of Users in Current and Previous Month Amongst Spotify-Owned Playlists

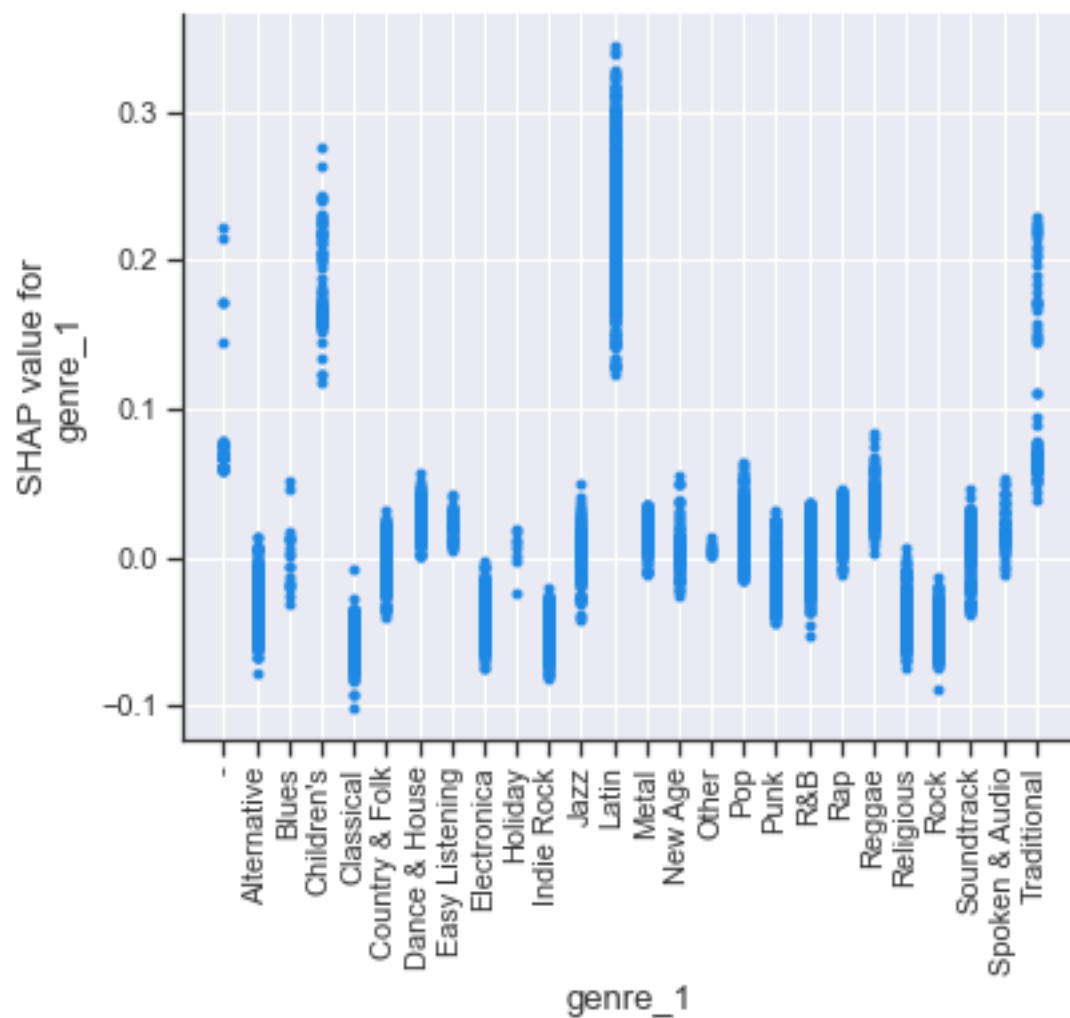


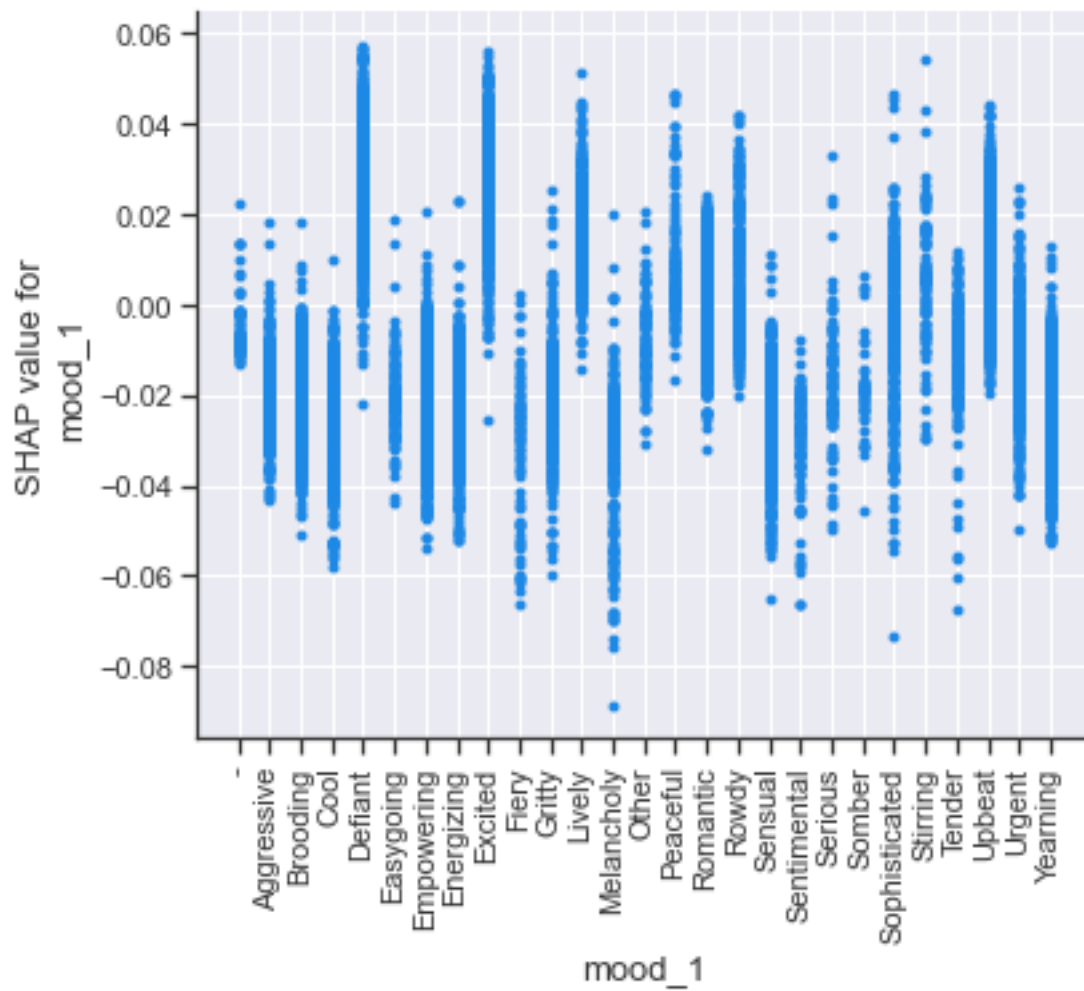




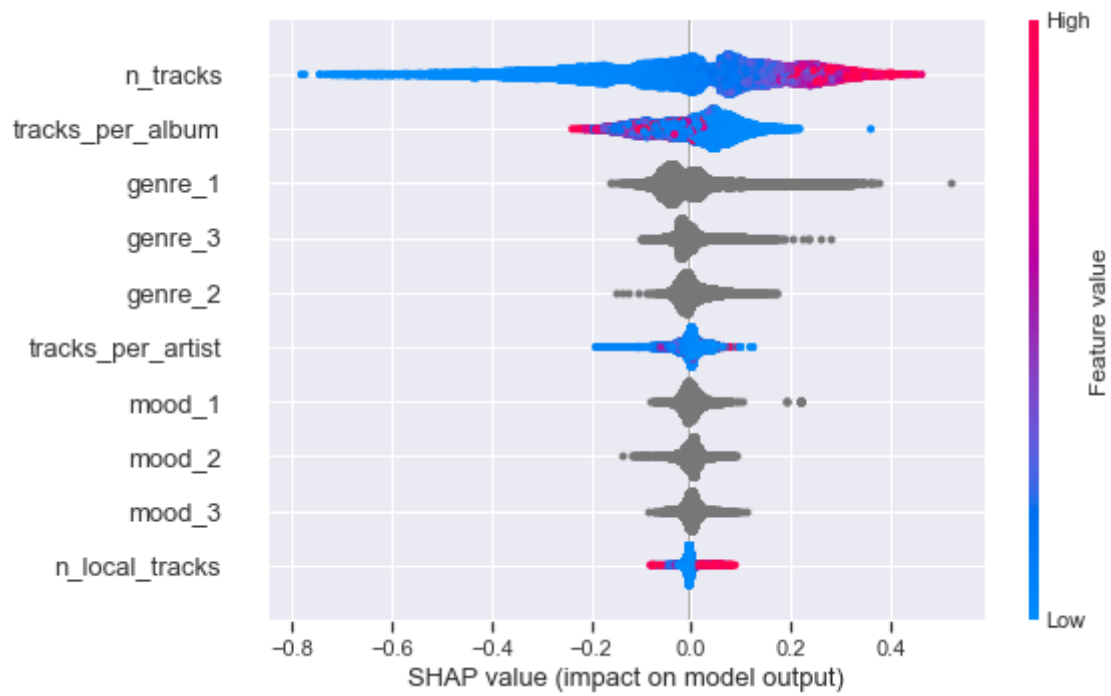
5.3 Modeling Results for Number of Users Today Amongst User-Owned Playlists

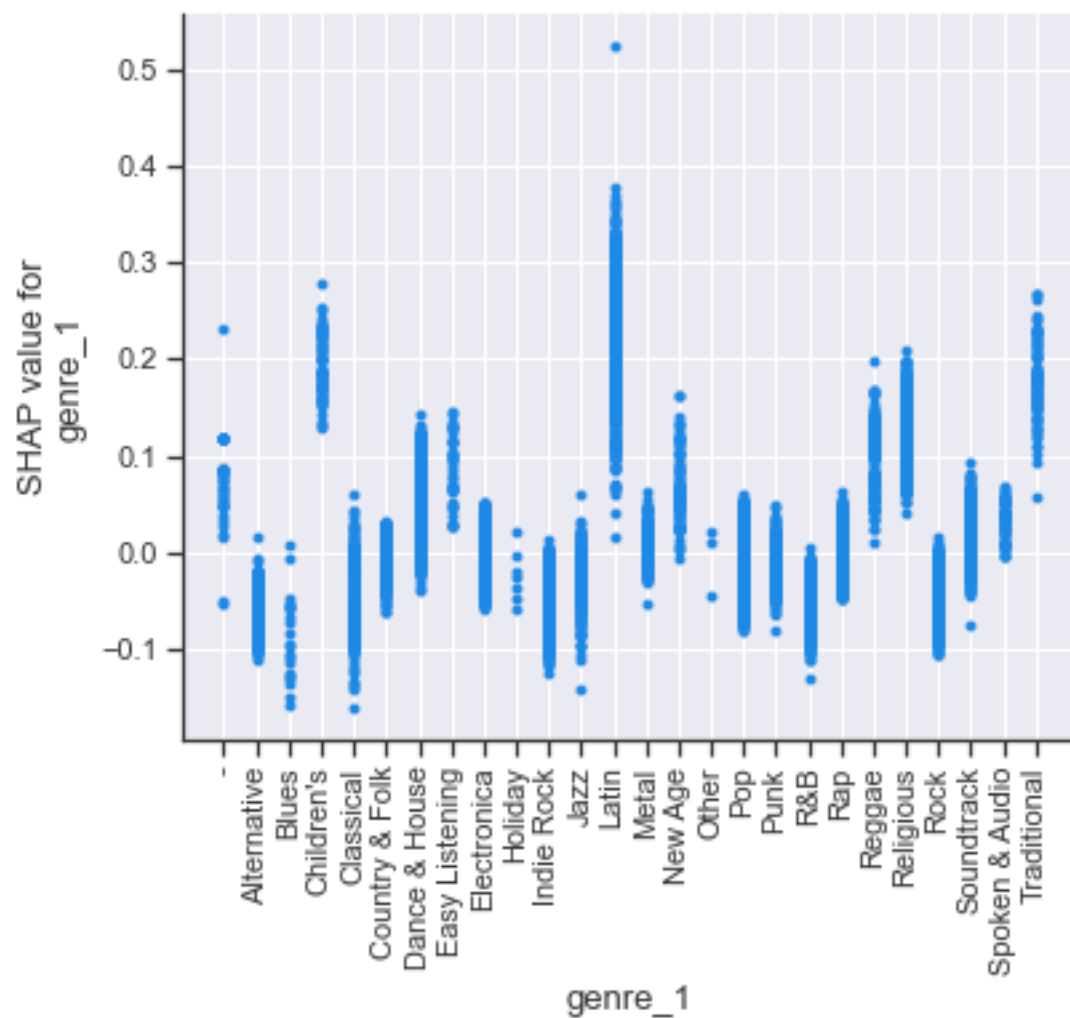


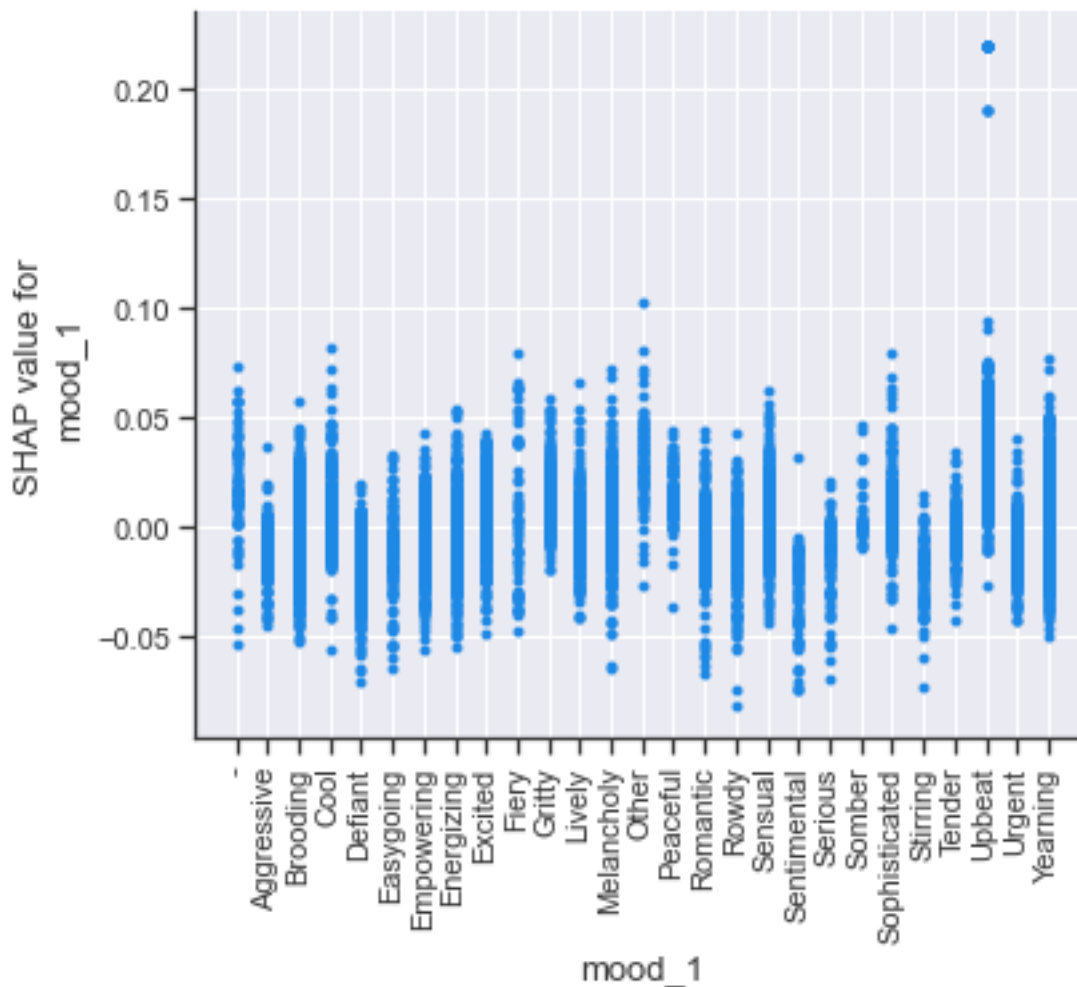




5.4 Modeling Results for Number of Users in Current and Previous Month Amongst User-Owned Playlists







6 Appendix

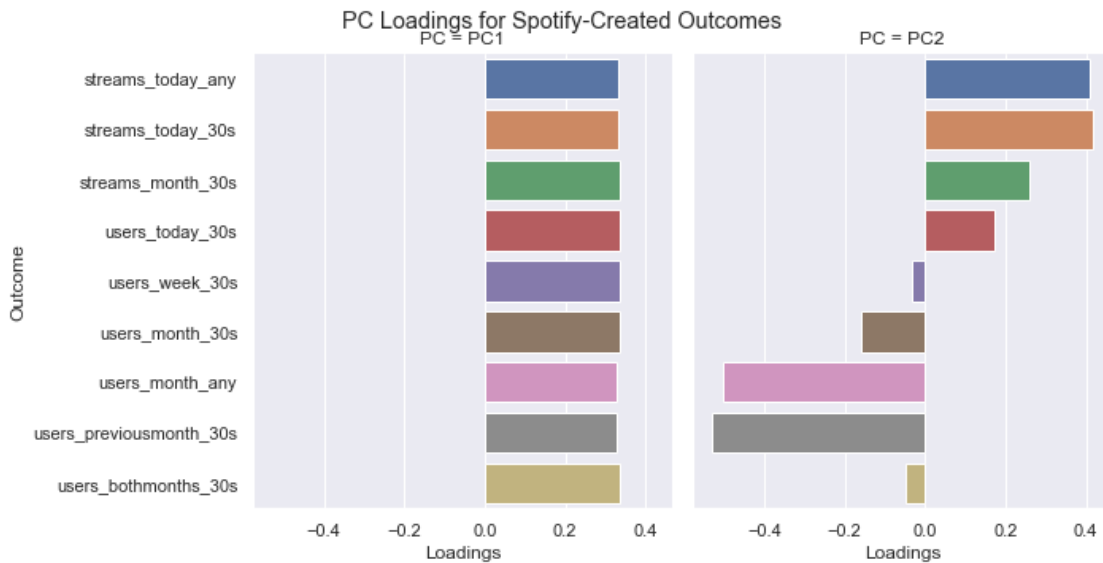
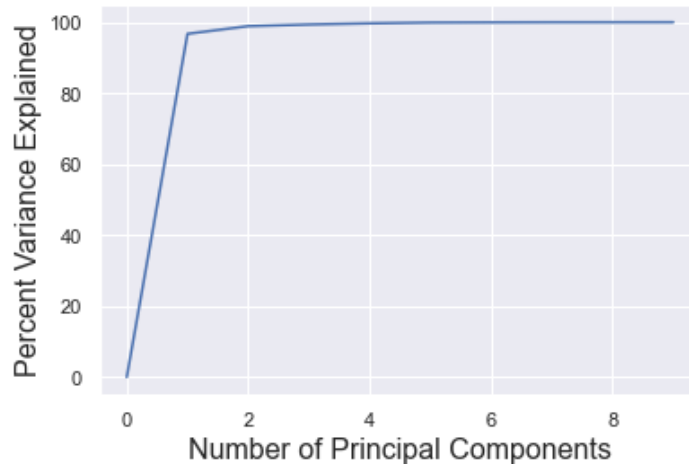
Assumptions: 1. This is a random sample of playlists. If this a biased sample, then any generalizations that we make from the data is likely to be meaningfully inaccurate.

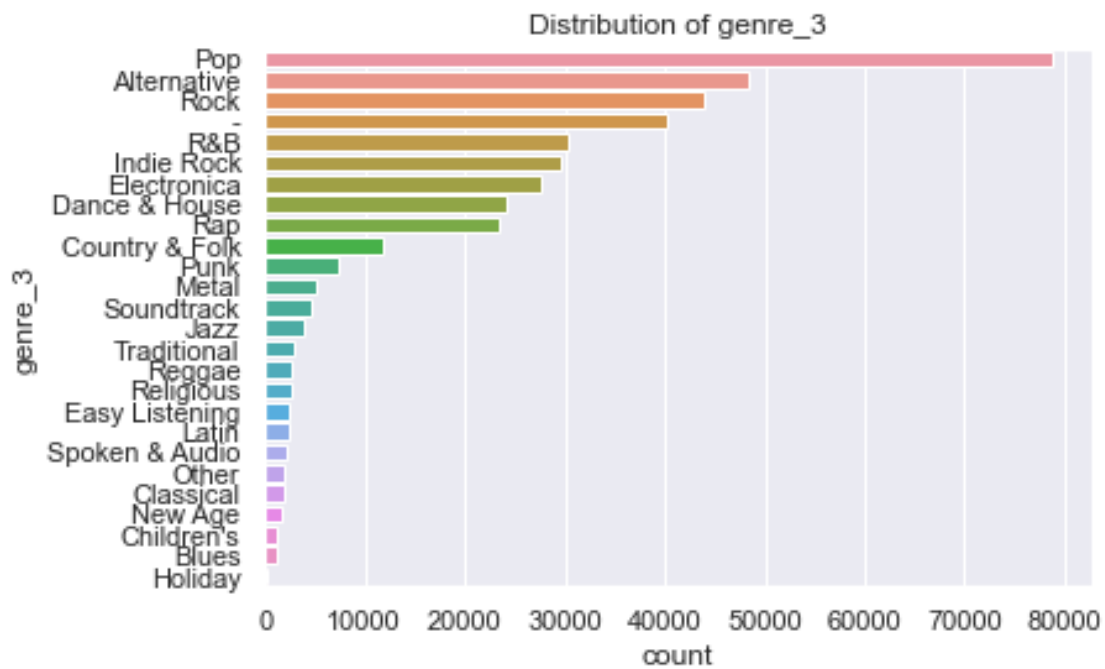
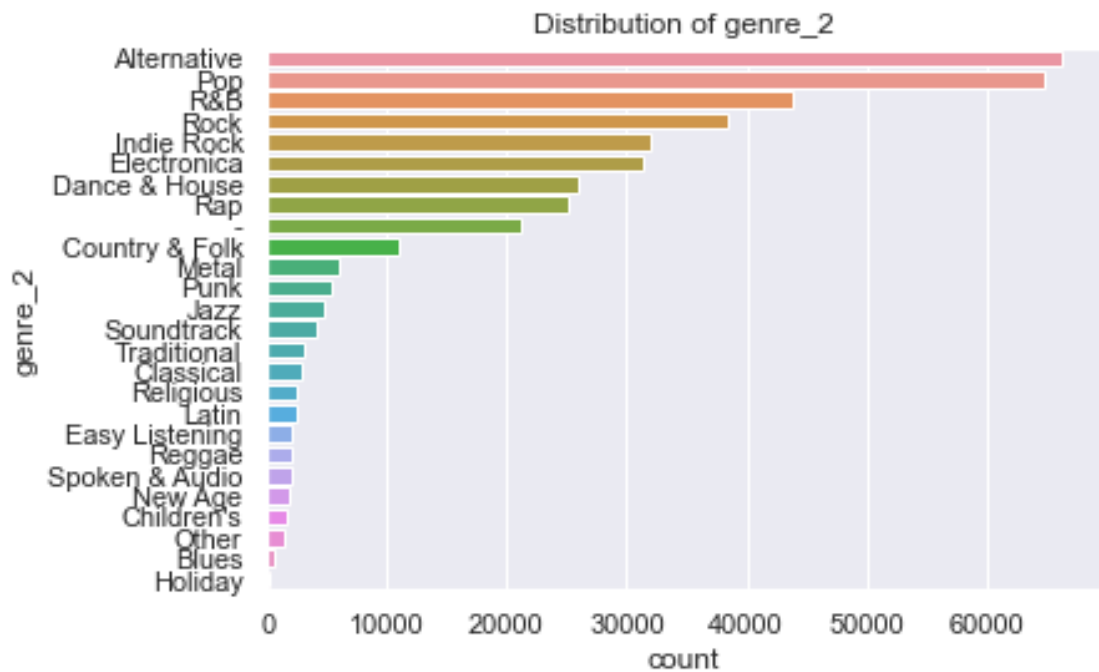
2. Spotify treats each non-Spotify playlist equally in terms of promotion. For examples, if the Spotify algorithms were promoting some genres above others at the time this data was collected, then we are unlikely to get a good read on how genre affects listenership.
3. (a) Spotify treats its own playlists differently than the non-Spotify playlists. If this assumption is correct, then it is likely that Spotify playlists are not particularly comparable to non-Spotify playlists.
- (b) Spotify treats its own playlists equally with each other. Thus, an analysis with only Spotify playlists should be okay.
4. Each playlist included in the dataset has existed for at least two months. This ensures that

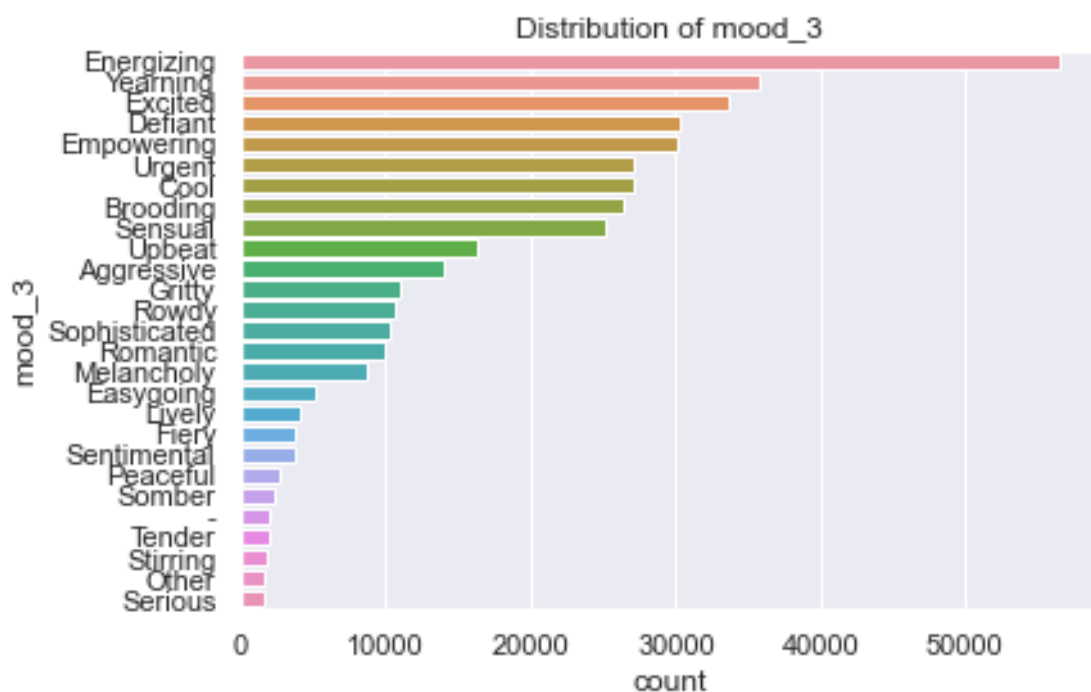
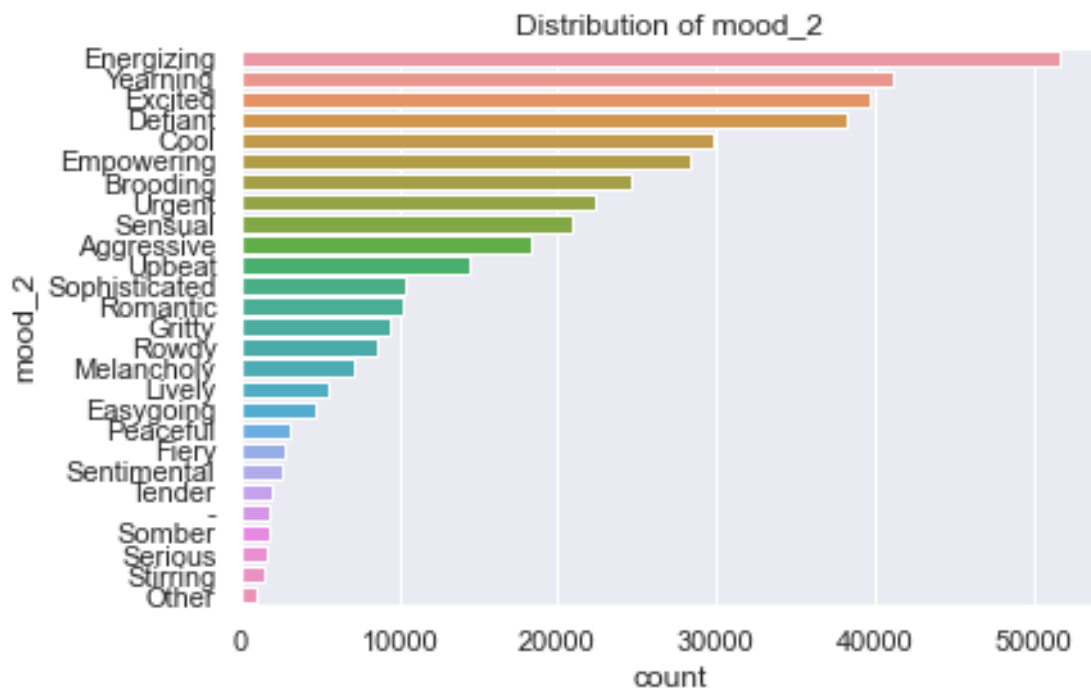
the monthly average users in the previous month variable is not biased by how long the playlist has existed.

5. The Spotify algorithms do not amplify small variations in success. If playlist A was slightly more successful than Playlist B two months ago under ‘fair’ algorithmic treatment, then the algorithms will not amplify playlist A over playlist B, and thus widen the gulf between the success of the two playlists. In other words, there is a fair marketplace for the playlists to compete, where success does not necessarily beget success simply due to the algorithms.
6. For the categorical variables, genre_1-genre_3 and mood_1-mood_3, when the value is ‘-’ this is not a missing value, but is instead imparting the information that the given playlist does not easily fit into the predefined genre and mood types.

Spotify-Created Outcomes: Percent of Variance Explained vs. Number of Components







Signature:

```

shap.dependence_plot(
    ind,
    shap_values=None,
    features=None,
    feature_names=None,
    display_features=None,
    interaction_index='auto',
    color='#1E88E5',
    axis_color='#333333',
    cmap=None,
    dot_size=16,
    x_jitter=0,
    alpha=1,
    title=None,
    xmin=None,
    xmax=None,
    ax=None,
    show=True,
)

```

Docstring:

Create a SHAP dependence plot, colored by an interaction feature.

Plots the value of the feature on the x-axis and the SHAP value of the same
 ↳ feature
 on the y-axis. This shows how the model depends on the given feature, and is
 ↳ like a
 richer extension of the classical partial dependence plots. Vertical dispersion
 ↳ of the
 data points represents interaction effects. Grey ticks along the y-axis are data
 points where the feature's value was NaN.

Parameters

`ind` : int or string

If this is an int it is the index of the feature to plot. If this is a
 ↳ string it is

either the name of the feature to plot, or it can have the form "rank(int)"
↳to specify
the feature with that rank (ordered by mean absolute SHAP value over all the
↳samples).

`shap_values` : `numpy.array`
Matrix of SHAP values (# samples x # features).

`features` : `numpy.array` or `pandas.DataFrame`
Matrix of feature values (# samples x # features).

`feature_names` : `list`
Names of the features (length # features).

`display_features` : `numpy.array` or `pandas.DataFrame`
Matrix of feature values for visual display (such as strings instead of
↳coded values).

`interaction_index` : "auto", None, int, or string
The index of the feature used to color the plot. The name of a feature can
↳also be passed
as a string. If "auto" then `shap.common.approximate_interactions` is used to
↳pick what
seems to be the strongest interaction (note that to find to true stongest
↳interaction you
need to compute the SHAP interaction values).

`x_jitter` : float (0 - 1)
Adds random jitter to feature values. May increase plot readability when
↳feature
is discrete.

`alpha` : float
The transparency of the data points (between 0 and 1). This can be useful to
↳the
show density of the data points when using a large dataset.

`xmin` : float or string
Represents the lower bound of the plot's x-axis. It can be a string of the
↳format
"percentile(float)" to denote that percentile of the feature's value used on
↳the x-axis.

`xmax` : float or string
Represents the upper bound of the plot's x-axis. It can be a string of the
↳format

"percentile(float)" to denote that percentile of the feature's value used on
→the x-axis.

ax : matplotlib Axes object

Optionally specify an existing matplotlib Axes object, into which the plot
→will be placed.

In this case we do not create a Figure, otherwise we do.

File: c:

→\users\administrator\onedrive\anaconda\lib\site-packages\shap\plots_scatter.py

Type: function