Drifting Upstream Project Report

Problem Statement

Regression Records (RR) is a burgeoning record label in Austin, Texas. After a couple of hit songs in 2023, executives are exploring strategies for selecting 3 top picks from its roster of hundreds of songs. These top 3 picks will be granted additional promotional budget in Q3 and Q4 of 2024. This extra funding will be allocated toward press releases, playlist curation, song synchronization in films and television shows, and live touring. The RR staff believes that in the business of sound dissemination, it is important to be mindful of trends. The sonic qualities of other popular songs, both recent and historic, set an insightful benchmark into what a popular song might sound like in the future. Thus, RR's Data Science team has added the Sonic Analysis sector to interpret the association between songs' music features and popularities.

As the leading data scientists on the Sonic Analysis team, we've been tasked with building a machine learning model to predict a song's popularity based on its sonic features. Executives at RR will then be able use this model to predict the popularity of songs from its own catalog based on the songs' sonic features. These predictions will help to inform decisions on which 3 songs to choose for the prioritized promotion.

Data Wrangling Pt. 1

To address this regression challenge, we first had to define a metric to measure popularity. Song popularity can manifest as chart success, album / single sales, stream count, social media followers, etc. With the rapid transformation of the record industry, what once constituted success may not be as relevant today. Since streaming revenue accounted for 67% of all recorded music revenue in 2022[1], we initially chose total song streams as the success metric. Furthermore, since Spotify is the world's leading streaming service by subscribers[2], we selected it as the platform from which to measure total streams.

In addition to a popularity score, we had to define relevant musical features that might influence a song's popularity. The Spotify API provides a variety of feature scores for songs. We found several available datasets comprised of numerous songs and their respective feature scores. Common features include: danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence (how positive a song is), all typically measured as a percentage score or on a scale from 1-100 representing the confidence that the song contains the feature, as well as tempo (measured in beats per minute), duration (measured, in minutes, seconds, or milliseconds), and loudness.

[1] https://github.com/david92russell/Drifting-Upstream/blob/main/Reports/References/GMR_2023_State_of_the_Industry.pdf
[2] https://www.businessofapps.com/data/music-streaming-market/

Both the data wrangling and exploratory data analysis happened in two stages. In the first stage, we retrieved a dataset[3] consisting of 953 songs primarily released from 2021-2023, and representing songs that were particularly popular in 2023. Feature columns included track info (such as song name, artist name, and release date), playlist/chart counts (across Spotify, Apple, Deezer, and Shazam), mode (major or minor), key, and all of the aforementioned music features, sans duration and loudness. Our target variable was expressed in number of streams per song.

Initial data cleaning entailed eliminating duplicates. Four songs were identified with two entries per song. We kept the duplicated observation with the higher stream count and dropped the other, sometimes imputing values from a dropped entry into a kept entry to try to consolidate the most accurate representation of features per song. We also had to convert certain columns to the appropriate data type (for example streams was initially an object rather than an integer) and to drop outliers that seemed too far from center to be correct (this included two songs with significantly fewer streams than the rest of the songs in the dataset).

Our next data-cleaning nuance came in the form of an encoding error in which any song and artist titles that had accent marks or apostrophes were encoded with the symbols: 'ï¿½'. Though tedious, we corrected this by identifying all entries with such errors and manually searching for the songs on Spotify and changing the names. This included 65 song names and 48 artist names. Should we address this problem in the future with a larger dataset, we would outsource the cleaning to our AI team to help automate the process.

Finally, with a cleaned dataset of sonic features and stream counts, we considered potential additional features. We found another dataset[4] with different songs that included the same musical features (danceability etc.) as our initial dataset, as well as song duration. Alas, there weren't enough songs present across both datasets to merge them and add the duration feature to our set of popular songs in 2023. Nonetheless, this got us thinking that song duration might be a valuable feature to consider. In fact, we would ultimately utilize a different dataset that included the duration feature (but more on that later). In the meantime, we had valuable insights to explore amongst our set of recently popular songs.

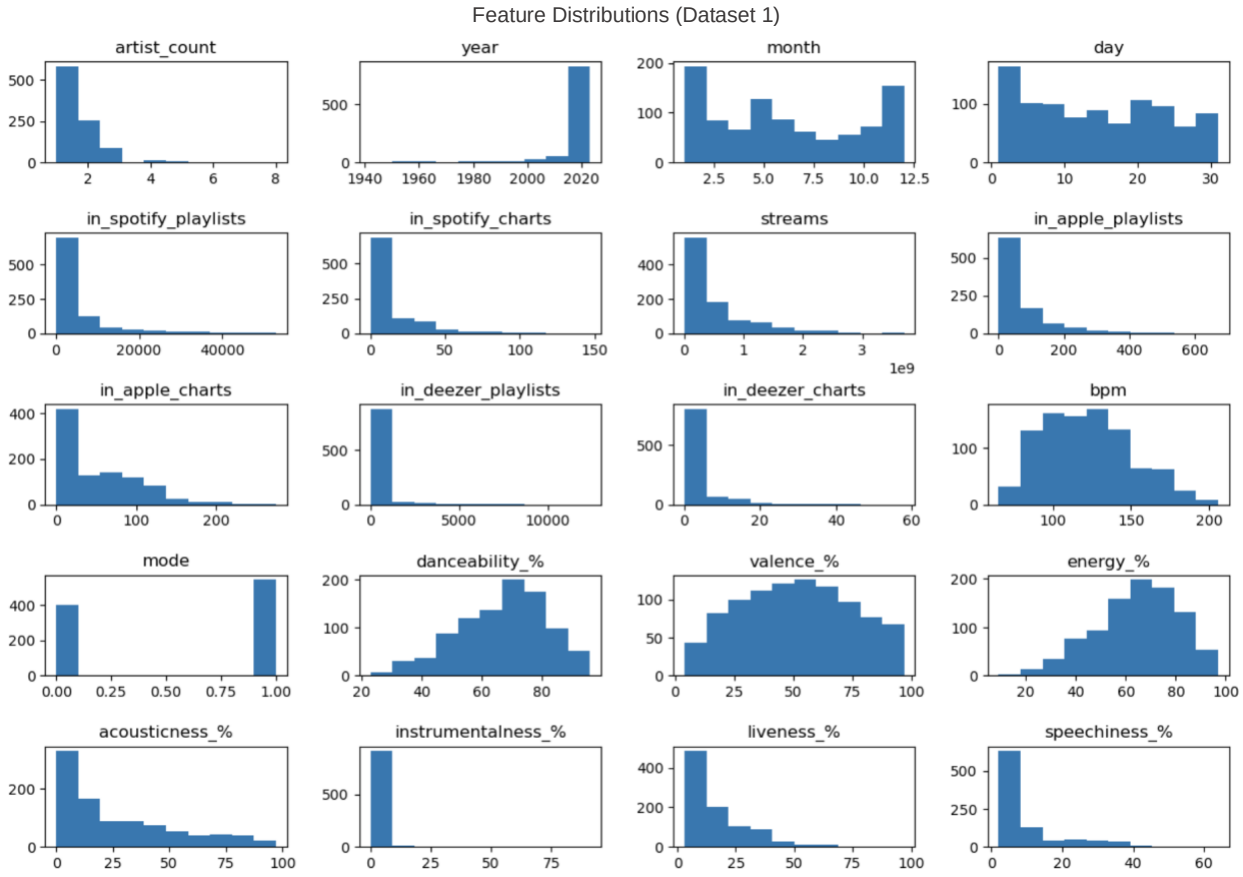
Exploratory Data Analysis Pt. 1

As we commenced the exploratory data analysis, we dropped columns with missing values from our dataset. This included the 'key' and the 'Shazam charts' column which were missing 95 and 50 values respectively. Rather than dropping the rows with the missing values, we chose to prioritize keeping more observations and decided that key and chart information would not be as pertinent to this analysis, as the musical features (danceability, valence, etc.) were our primary focus. With a clean dataset free of null values, it was time to explore some of the underlying distributions and correlations.

[3] https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023

[4] https://www.kaggle.com/datasets/sanjanchaudhari/spotify-dataset

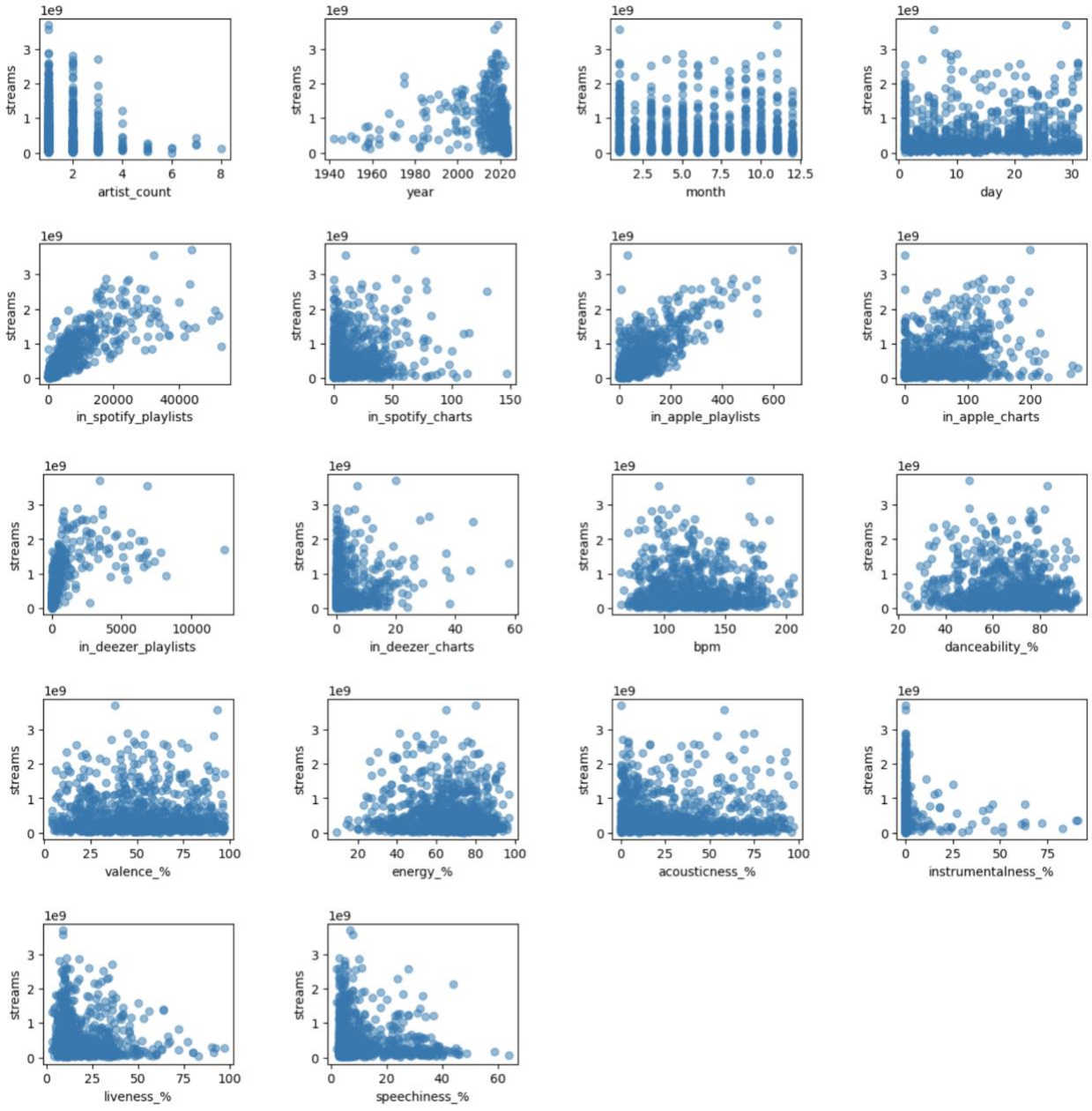
We plotted histograms to understand the distributions of our features. While the tempo (bpm) feature was relatively normal, several features were highly right skewed (acousticness, instrumentalness, liveness, and speechiness), including our target variable: streams. The presence of large outliers indicated that feature transformation would be crucial, and drew attention to the fact that the majority of popular songs in the dataset center around a narrow range of values for a given feature. Another approach we might consider in the future would be to look at the mode feature ratings of songs in this dataset (as these songs were popular recently), and use those scores as benchmarks for what might constitute ideal scores for those features in RR's chosen songs.



Distribution modes even offer takeaways from the month and day variables. We can see that more songs were released at the beginning of a month than any other day, and more songs were released at the beginning or end of a year than any of the months in the middle of the year. Since the songs in this dataset reflect popular songs from 2023, aligning RR's releases with similar release dates might help to bolster attention for these songs, as early or late year and early month releases seem to be most common for recent popular songs.

We also reviewed scatterplots of each numeric feature against streams.

Streams / Feature Correlations (Dataset 1)



We see that the music features are not linearly correlated with streams. This informs us that we should consider decision tree models for predicting streams, rather than a linear regression. While we can't ascertain much from the music feature scatterplots, the playlist scatterplots provide some clear illuminations. We see definitive positive correlations between streams and the number of Spotify, Apple, and Deezer playlists that the song is included in. We further examined the relative representation of playlist counts in our dataset across these platforms and learned that Spotify represents 92% of the playlist inclusion among total playlist inclusion in the dataset. Herein lies further support for Spotify as our primary platform of focus.

Though we saw the strongest correlations with streams amongst the playlist features, these features aren't pertinent to this specific analysis as we seek to understand which music features correlate with a song's popularity. The exploratory analysis of playlists' relationship with stream count highlights the importance of getting our songs playlisted, but as we are trying to pick songs to promote playlisting based on their musical features, a song's playlist count will be apparent after we have selected the appropriate songs to promote for playlisting. Thus, we reduced our features to only include tempo, mode, and all of the music quality features, as these features represent the factors RR will have to assess its own catalog for song selection.

Modeling Pt. 1

With an overview of distributions and feature / target correlations in our working dataset, we proceeded to model training. We split the data into 70 / 30 % train and test sets and identified the mean absolute error (MAE) as our score metric. Because the data include quite a few outliers, we opted to use the MAE instead of the root mean squared error, so as not to penalize larger discrepancies that might result from these outliers. Additionally, since the stream count is in the tens and hundreds of millions, we trained the model on the the log function of the score, and then converted the predictions back by raising them to the exponent function when comparing to the actual stream counts.

Our first model was a Random Forest Regressor. The untuned model yielded a train MAE of ~186M streams and a test MAE of ~342M streams. Upon tuning the model (we tried both a Randomized Search CV, and a custom tuning of individual features using a for loop – the custom tuning yielded the better result), we achieved train and test MAEs of ~336M and ~332M respectively. We also compared this to an Extra Trees Regressor which produced untuned train / test scores of 0 / ~340M and tuned train / test scores of ~342M / ~335M. Finally, we compared a Gradient Boosting Regressor that produced untuned and tuned train / test scores of ~310M / ~348M and ~189M / ~375M.

As the Random Forest Regressor was our strongest working model (since MAE represents how far off our predictions are, we want for this score to be lower), we compared its results after scaling the data using Standard Scaler, Min Max Scaler, Robust Scaler, and Power Transformer. The Power Transformer yielded the best resultant train / test scores of ~335M / ~332M.

These scores were much higher than we hoped for and won't provide reliable insight into which songs to select from the catalog, as a MEA of ~332M means that in theory a song predicted to have 332M streams could actually have 0. The difficulty of predicting an accurate stream count with a tuned regression model trained on scaled numeric music features, indicates that these features might not provide enough information to generate a reliable prediction.

In order to evaluate options for generating stronger predictions with this dataset, we examined the results of training and testing on different split sizes. Indeed, a larger train set yielded better predictive ability for our tuned Random Forest Regressor with Power

Transformed feature values, with train / test MAE's of ~335M and ~313M for an 80 / 20 split, and ~332M and ~268M for a 90 / 10 split. We also tested our model on a refined dataset in which we removed all songs in the top 5% of stream count, as stream count was heavily right skewed and had some songs with significantly more streams than the majority of the pack. This further improved performance, yielding train / test MAE's of ~247M / ~270M on an 80 / 20 split. Though this indicated how strongly outliers in this dataset were stifling our model's predictive ability, we still weren't satisfied with the results. We sought to discover how additional features might help predict streams.

By adding the features regarding playlist and chart counts per song, as well as release day (encoded to binary columns with Pandas' Get Dummies function), we were able to achieve train / test MAE's of ~178M / ~67M on an 80 / 20 split with an untuned Random Forest Regressor. Moreover, the 'Spotify playlists' feature made up for 73% of feature importance. Again, as this feature won't be available to RR when choosing songs from its catalog, a lesson here is that in order to generate more streams, one of the primary promotional goals should be to get the songs in more playlists. Considering playlist count's strong correlation with streams, we also considered it as a target variable instead of streams, but we quickly dropped this possibility when our Random Forest Regressor yielded 80 / 20 train / test split MAE's of 4498 / 4549 playlists (over half a standard deviation; not much better than our stream predictions).

Our final attempt at improving predictive ability on this dataset ultimately led us to the next stage of this project in which we repeated the process with a new dataset entirely. Due to the variability of stream counts, we considered smoothing out our target variable by converting it into deciles. This would take care of the outliers and allow us to predict a range of stream counts that a song might fall into. Our Random Forest Regressor was able to predict streams within 2 deciles of both the train and test set. Perhaps predicting stream ranges would be a more realistic goal than predicting actual stream count. The decile stream ranges are indicated below, with each row representing a decile and its minimum and maximum stream count.

Decile Stream Count Ranges

```
d_1 1365184 71423324
d_2 71573339 121077868
d_3 121189256 167076418
d_4 168448603 221752937
d_5 222410722 290833204
d_6 291709698 383835984
d_7 387080183 554875730
d_8 556585270 822239726
d_9 822633917 1302184087
d_10 1304313953 3703895074
```

Data Wrangling & Exploratory Data Analysis Pt. 2

Thinking of the target variable in terms of deciles led us to further explore other datasets. With stream count being a highly variable quantity, often in the hundreds of millions, we were open to alternatives. Also, with limited features in our initial dataset, we sought a new set with additional features that might help strengthen predictions. Indeed, we found such a dataset[5] that included the same musical features of danceability, energy, speechiness, acousticness, instrumentalness, liveness, valence, and tempo, as well as the new features of loudness and duration. This new dataset also had categorical features for the song's key (without missing values), as well as genre (EDM, rap, pop, R&B, Latin, or rock) and subgenre (24 unique categories).

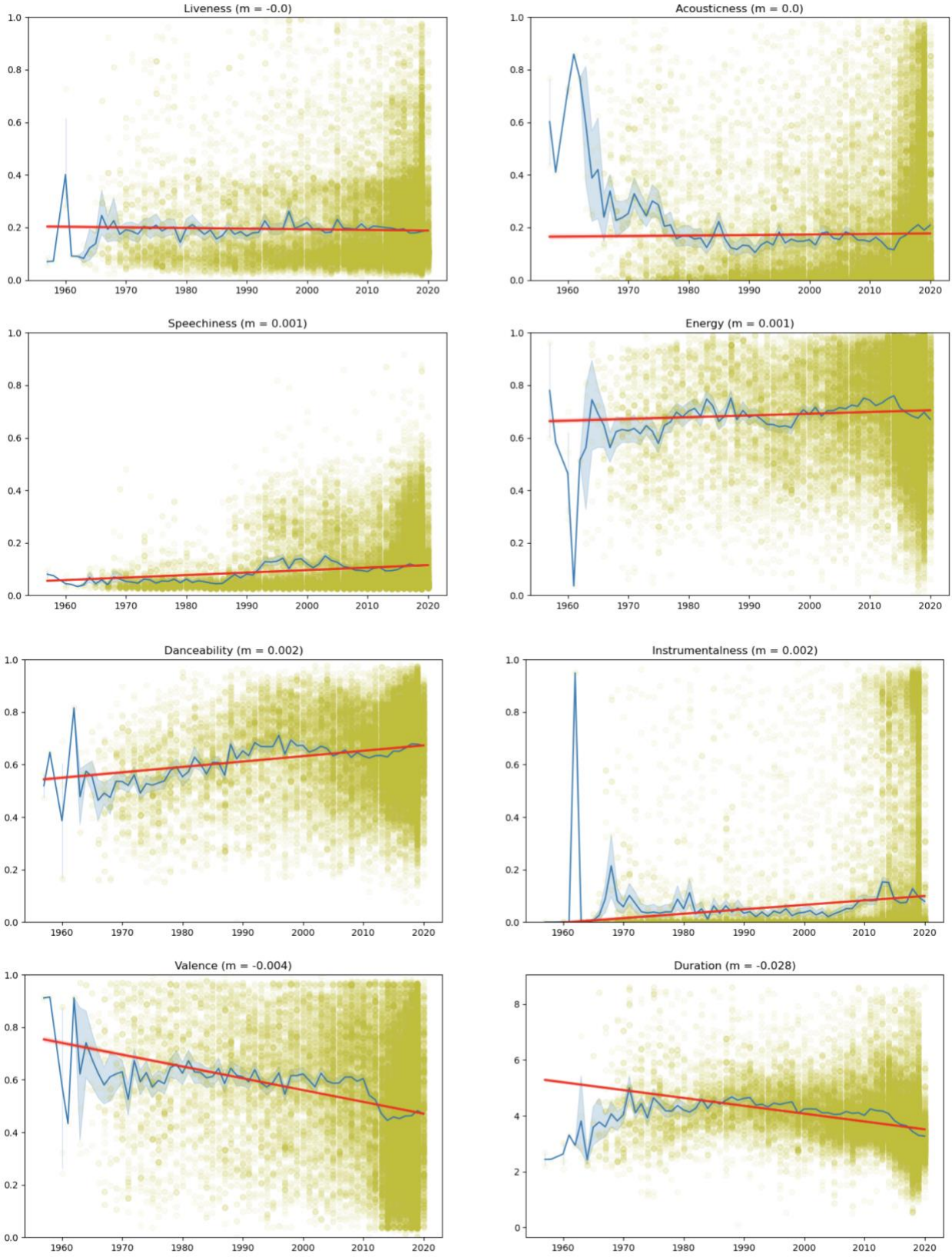
With a broader variety of features (and significantly more observations: 32,833 total songs), we were confident that this dataset might be a better candidate for training a model to predict popularity. In fact, the target variable was just that: a popularity score from 1-100. Similar to the stream count deciles target (though spread out over 10 times the range), the popularity score might prove an easier metric to work with than total stream count. Alas, later attempts to translate the popularity score into a real-world metric (such as the equivalent number of streams) posed difficulty that we will discuss later. Nevertheless, this dataset granted a wealth of additional insights.

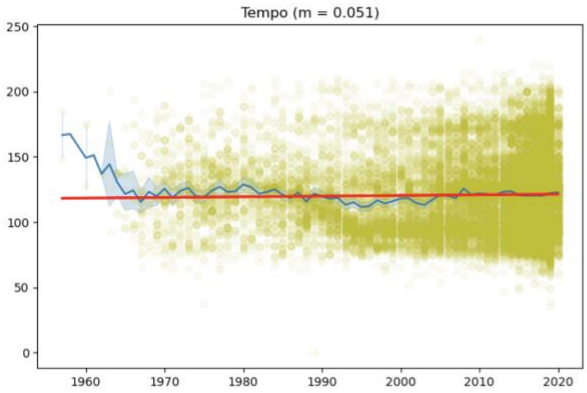
The new dataset was much cleaner off the bat and required little maintenance to get into workable shape. After converting release dates into datetime format, we performed a time series analysis. As the dataset spans songs released from 1957 – 2020, we explored how each of the music features trended through the years, at least as represented by the sample of songs comprising the dataset. While this dataset provides greater song representation over the years, as compared to our initial dataset that mostly consists of songs from the past 3 years, the past 3 years are in fact missing from the new dataset, and thus it is more representative of long-term trends than immediate ones. By reviewing both datasets, we can glimpse trends both recent and over the longer term. Regarding song popularity predictions, we determined that given this new dataset's inclusion of additional features and more observations, it would be our preferred choice to proceed with for predictive modeling. While neither dataset is perfect, in future work we might seek out a dataset that achieves the best of both worlds: ample features and observations, and recency.

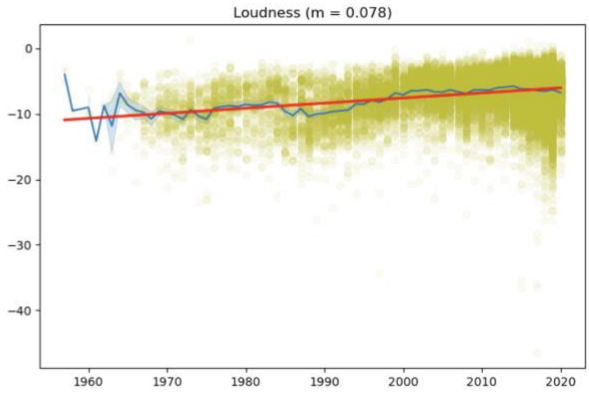
The time series analysis revealed that over the total span of sample representation from 1957 – 2020, songs have become louder, faster, shorter, and of a lower valence (less positive).

[5] https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs

Music Feature Changes Over Time (1957 – 2020)

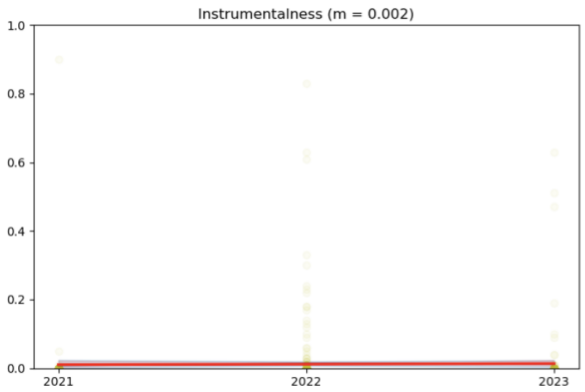


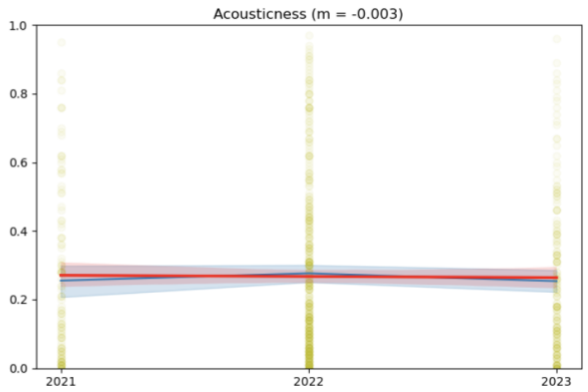


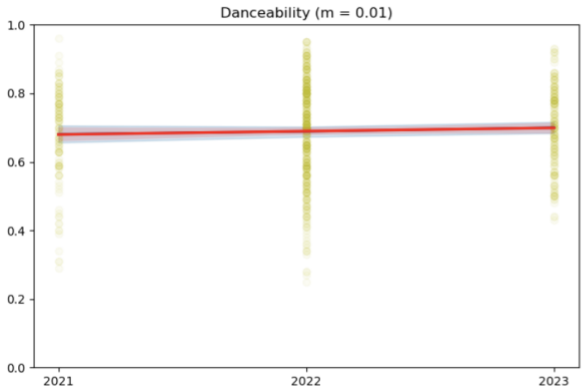


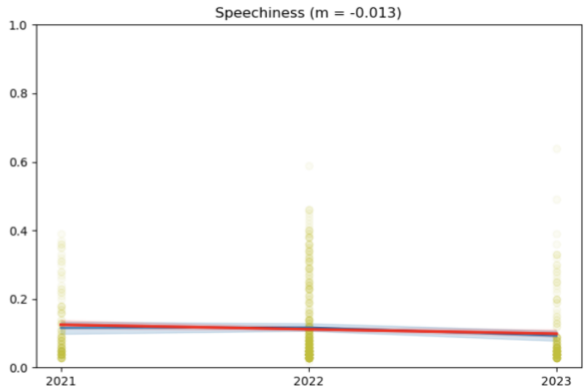
To account for the past few years, we also performed the time series analysis on the songs from 2021 – 2023 from our initial dataset. Since these didn't include loudness or duration, we instead viewed the latest 3-year change in loudness and duration in the songs from our new dataset (2018 – 2020). Interestingly, our initial dataset of more recent songs actually indicates a slight decrease in tempo in the past 3 years, contrary to the longer-term trend of the past half-century+. This of course, might also be due to a difference in the sample in each of the respective sets.

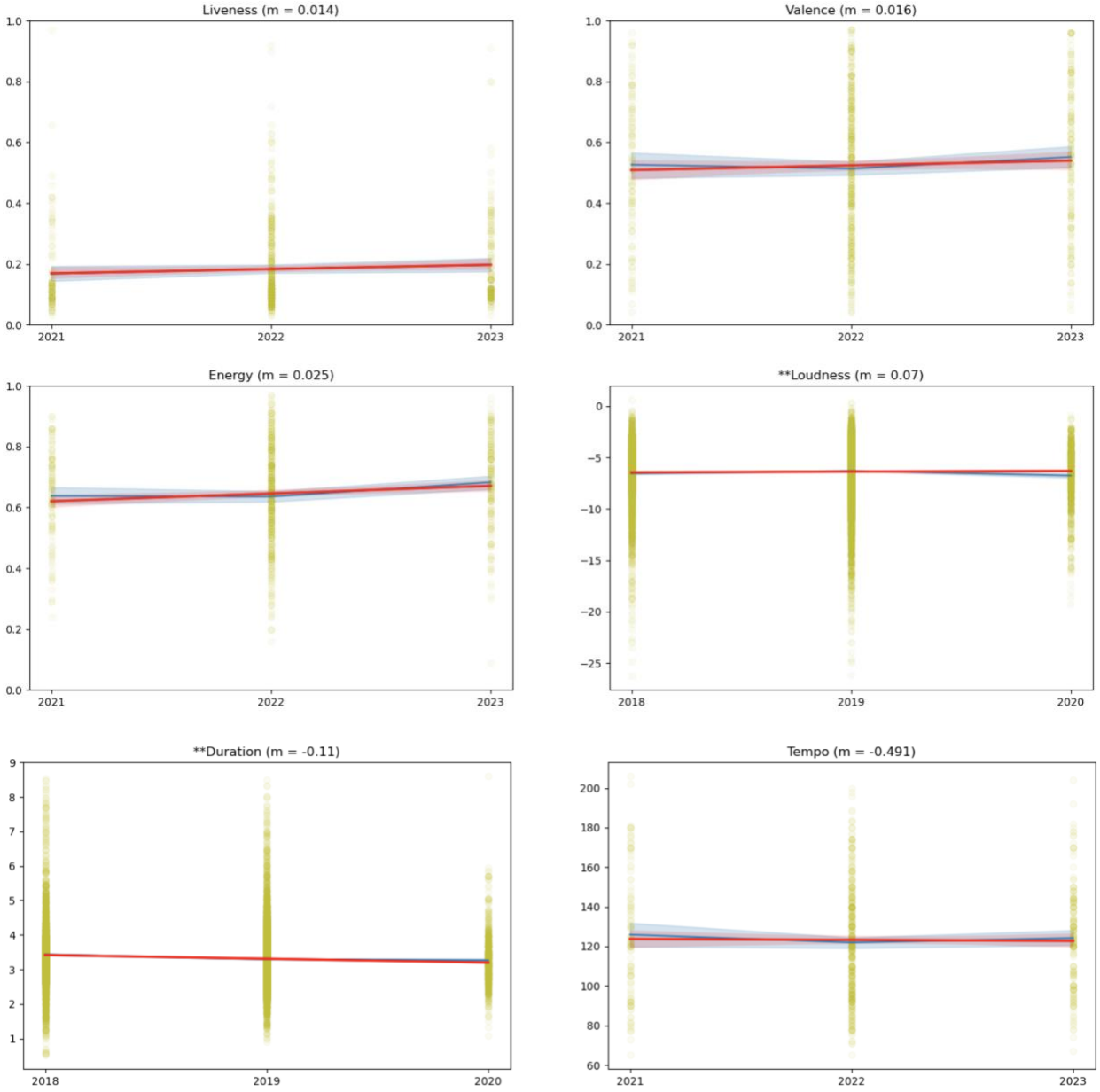
Music Feature Changes Over Time (2021 – 2023)





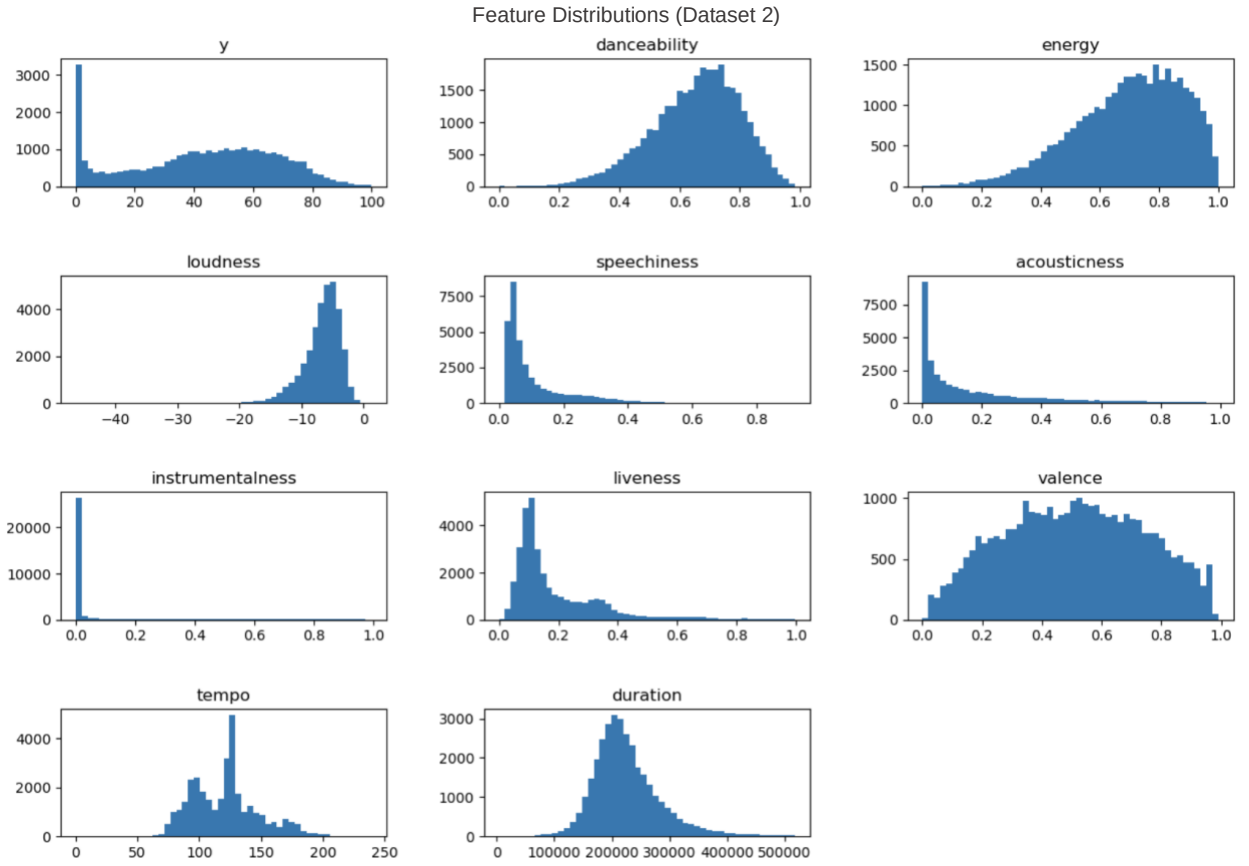






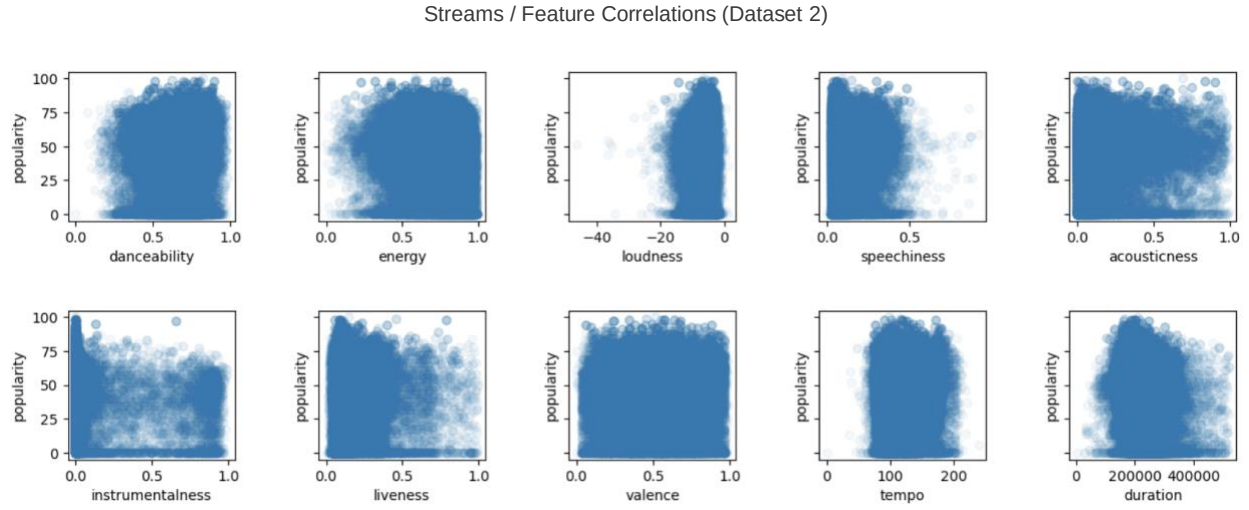
Naturally, we observe that the changes over a broader span of time show greater variation than changes over only the past few years. These insights might help RR further consider which direction music features are trending.

Returning to our new dataset, we one-hot encoded all of the categorical variables (genre, subgenre, key, and mode). We then examined distributions of the numeric music features. The ‘y’ variable refers to the popularity score.

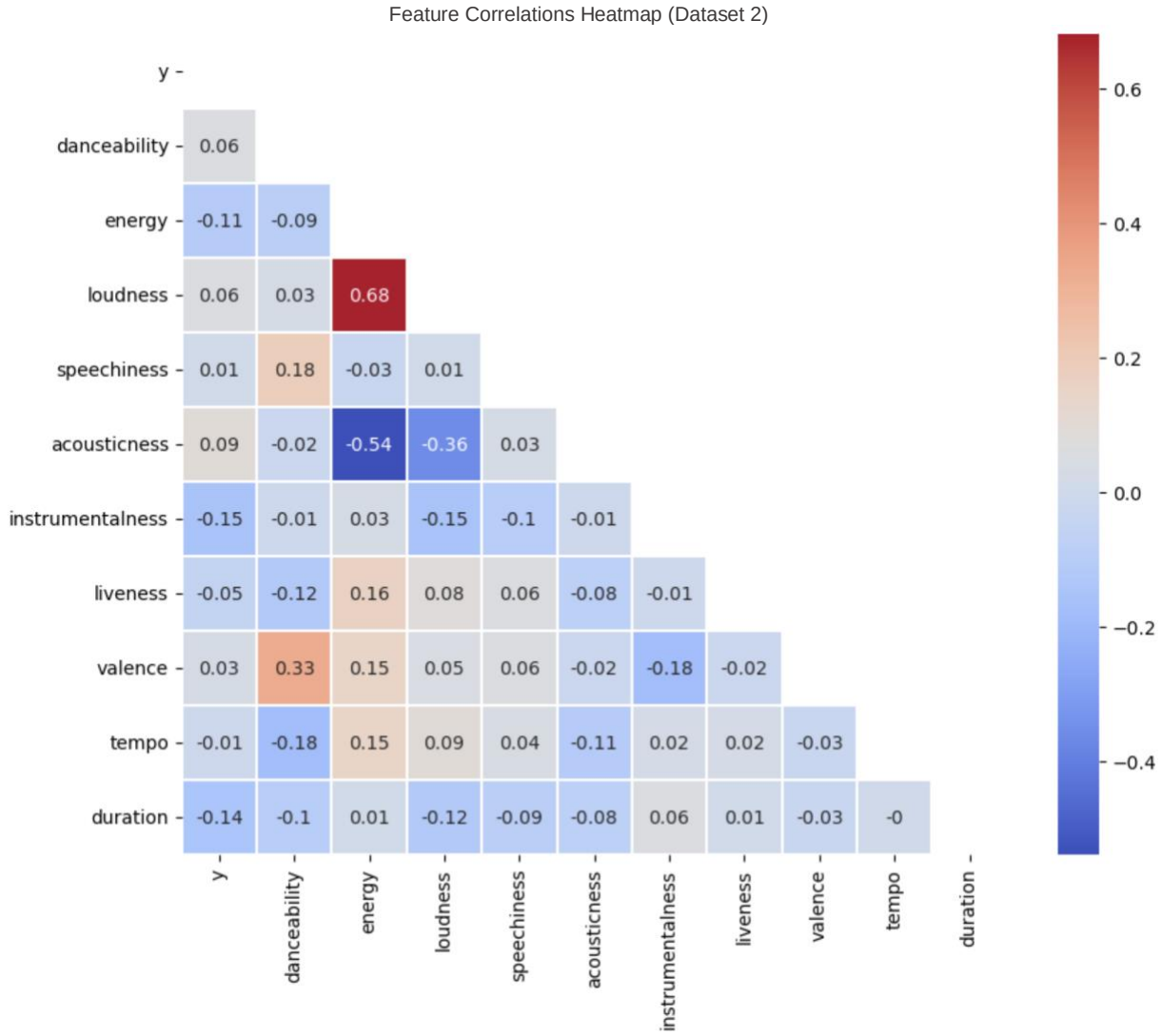


Similar to our initial dataset, acousticness, instrumentalness, liveness, and speechiness are all highly right skewed while valence looks fairly normal. We can see that about 125 beats per minute is, by far, the most common tempo. Though 0 appears to be the most common popularity score, popularity scores within the interquartile range seem to follow a normal distribution. The high presence of 0's might be a flaw in the reporting and something to consider in future analyses.

The scatterplots of music features and popularity scores resembles a matrix of blobs.



As the abundance of observations crowd across nonlinear spans, we will again build models based on decision trees for these data. Though the correlations between popularity score and numeric music variables were difficult to glean via scatterplots, a heatmap clearly illustrated the minute correlations between the features and the target, as well as between the features amongst one another.



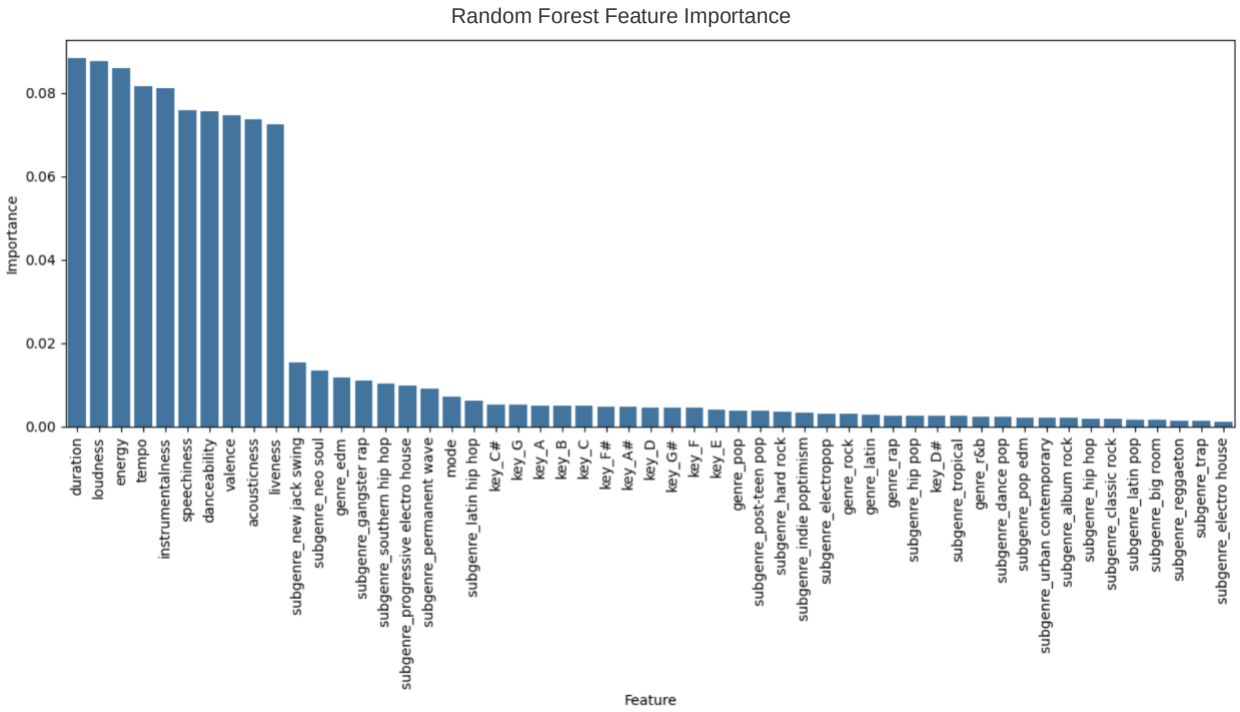
Most of the features don't strongly correlate with one another, but the strongest linear correlations are positive between loudness / energy (0.68) and valence / danceability (0.33), and negative between acousticness / energy (-0.54) and acousticness / loudness (-0.36). The strongest (though not very strong at all) correlations between popularity scores and features are negative with instrumentalness (-0.15) and duration (-0.14).

With a clean and properly encoded dataset of 10 numeric features, 4 one-hot-encoded categorical features spanning 43 columns, and 1 target variable of popularity score, we were ready to return to modeling.

Modeling

We partitioned the dataset into an 80 / 20 train / test split, entailing 26,266 observations in the train set and 6,567 observations in the test set. To prevent overfitting, we tuned models across a 5-fold cross-validation (CV) on the train set before introducing the test set. To measure best performance, we used either the mean or median of the cross validated MAE score – whichever was lower (we used both mean and median in case the mean was thrown off by outliers in a given fold).

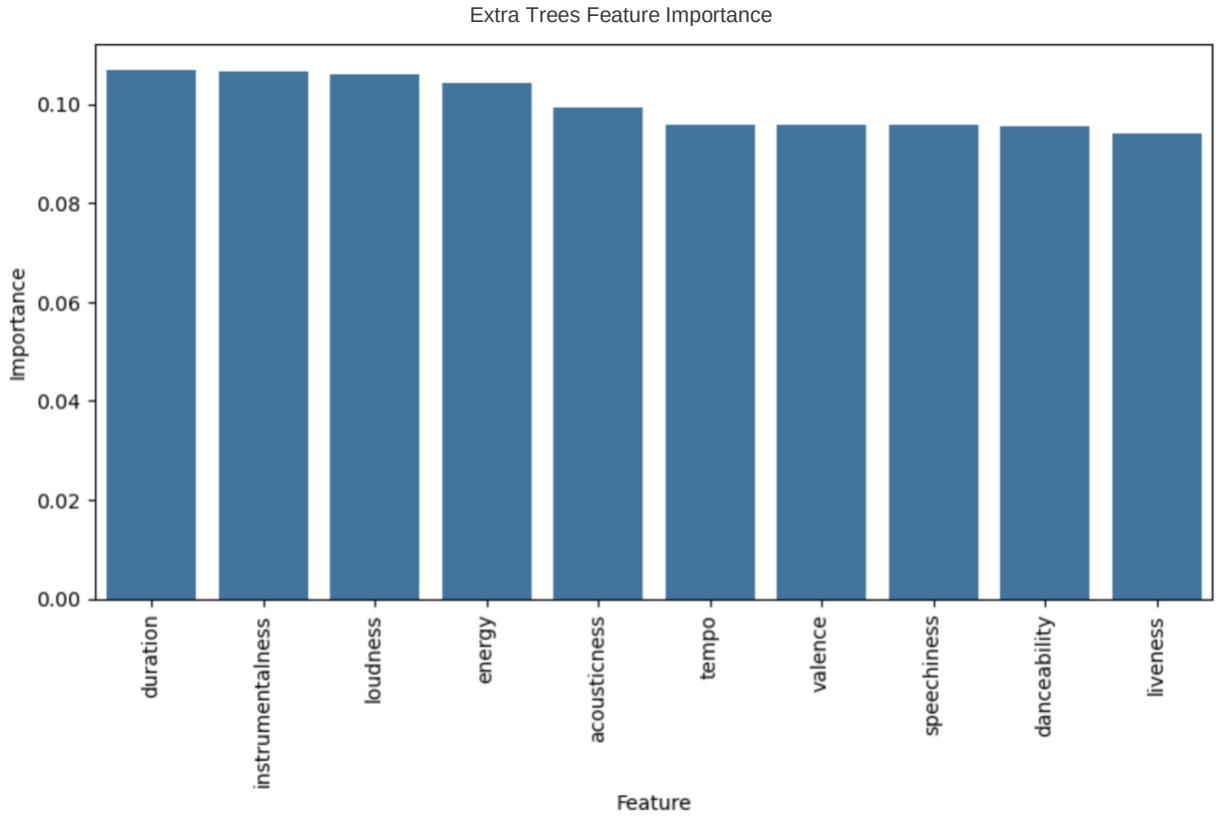
We tested 5 models: Random Forest, Extra Trees, Gradient Boosting, Hist Gradient Boosting, and XGBoost Regressors. We tuned all models with Optuna, which enabled us to explores combinations of values for chosen hyperparameters within a specified range. Our initial untuned Random Forest Regressor produced a MAE of 17.24, with the tuned version improving to 17.21. Upon ranking feature importances, we saw a steep drop off from the numeric features to the categorical ones.



Upon training an untuned Random Forest Regressor on only the numeric features (duration, loudness, energy, tempo, instrumentalness, speechiness, danceability, valence, acousticness, and liveness), the lowest (mean or median) MAE of the CV prediction score improved to 16.94. The categorical features were in fact hindering the predictive ability of the model. With more time, we could individually check the inclusion of numeric features plus a combo of each of the categorical features, to discern the combination that yields the best performance, but given the time and computation that this would require, we opted for expediency by only keeping the numeric features. Moreover, when we apply this feature selection to RR's song catalog, exclusively using numeric music features will allow for songs of all keys and genres to be considered.

After tuning a new Random Forest Regressor to only the numeric music features, the model yielded a CV mean score of 16.86 with a standard deviation of 0.099 and a test prediction MAE of 16.31. While the better performance on the test score again highlights the likelihood that larger outliers are present in the training set, it also implies that our model isn't overfitting to the training data. The top 4 most important features for this model were duration (0.113663 – the proportion of importance that this feature represents out 1.0), loudness (0.110351), tempo (0.105200), and energy (0.101374).

The Extra Trees Regressor turned out to be our best model. The untuned version produced a CV lowest (mean or median) of 15.98 and the tuned version produced a CV Mean score of 15.94 with a standard deviation of 0.088 and a test prediction MAE score of 15.30. Parameter specifications of the tuned model include a max depth of 46 and using 1458 estimators. The top 4 most important features for this model were duration (0.106840), instrumentalness (0.106628), loudness (0.105952), and energy (0.104153).

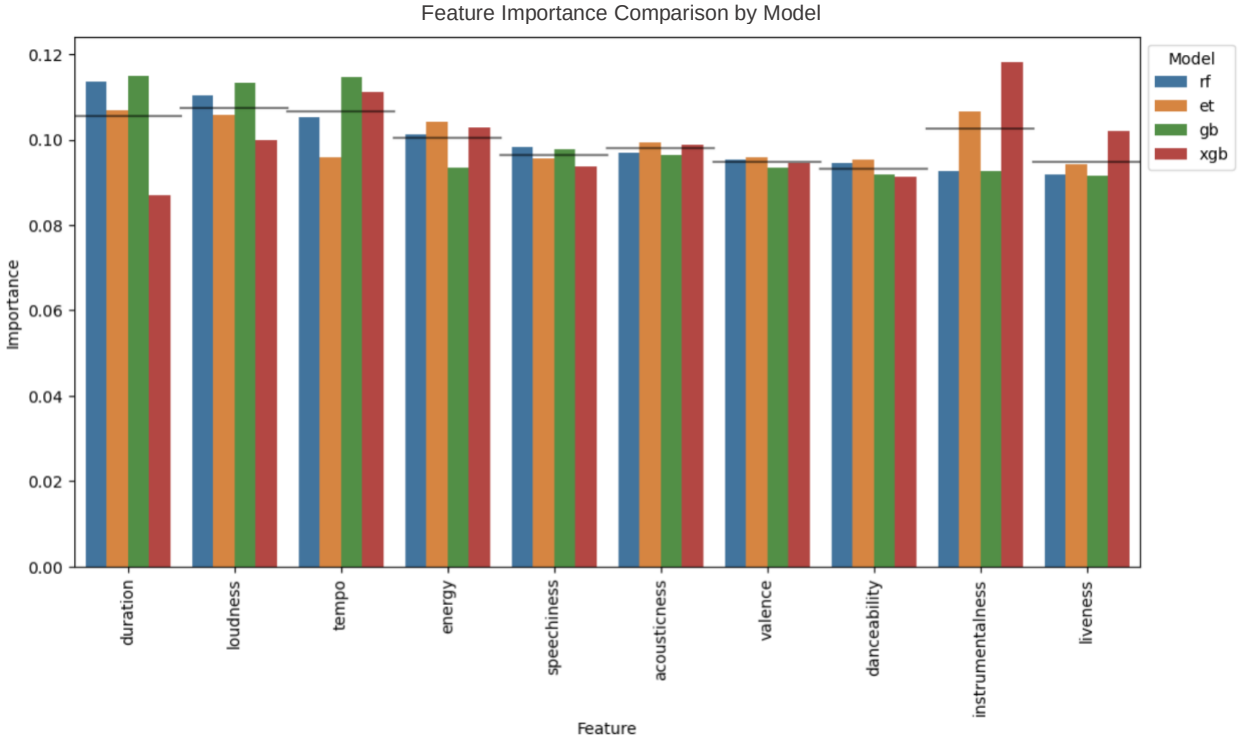


The untuned Gradient Boosting Regressor returned a CV lowest (mean or median) of 19.59. The tuned version significantly improved to a CV mean of 16.83 with a standard deviation of 0.119 and a test prediction MAE of 16.00. The model's top 3 most important features were duration (0.115055), tempo (0.114567), and loudness (0.113379). There was a steeper drop after these to the fourth most important feature of speechiness (0.097661).

The untuned Hist Gradient Boosting Regressor had a CV lowest (mean or median) of 19.15. The tuned model produced a CV mean of 18.24 with a standard deviation of 0.198 and a test prediction MAE of 17.68. Hist Gradient Booster doesn't include a feature importance attribute.

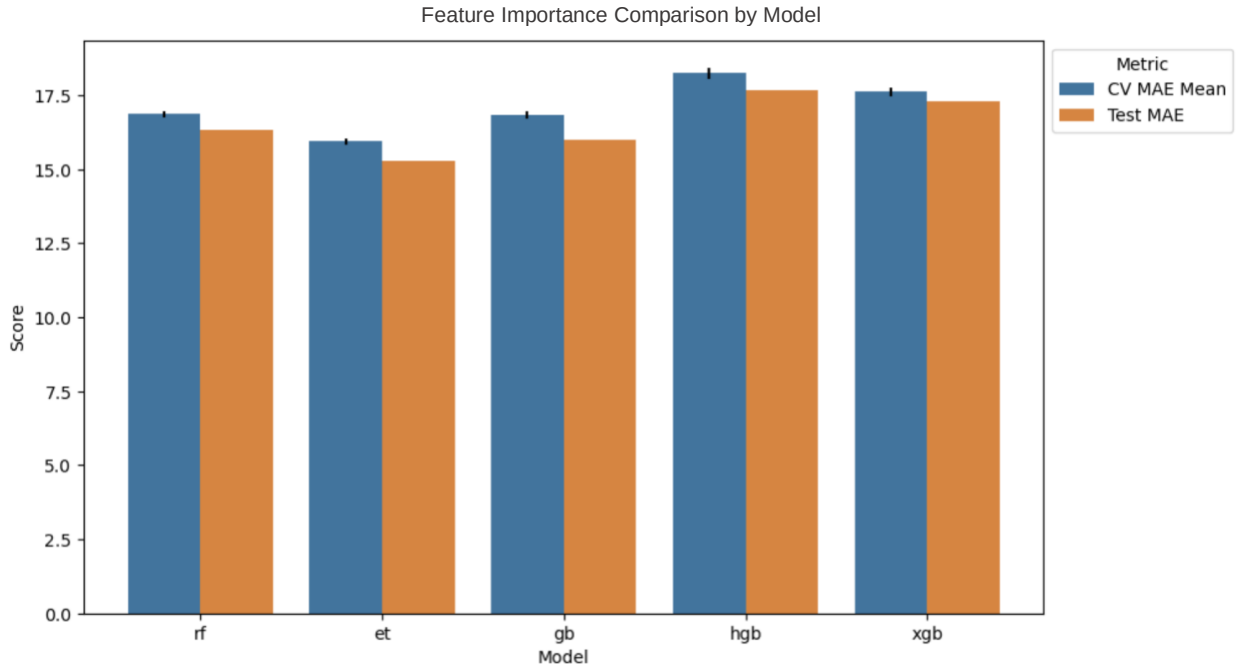
The untuned XGBoost Regressor produced a CV lowest (mean or median) of 18.87. The tuned model had a CV mean of 17.62 with a standard deviation of 0.140 and a test prediction MAE of 17.29. The most important feature for this model was instrumentalness (0.118265), with a drop before tempo at second (0.111191). There was another drop in relative importance before the other features, with energy in third (0.102951) and liveness in fourth (0.102194).

In looking at collective feature importance across the 4 of 5 tested models with a feature importance attribute, we see that on average, duration, loudness, tempo, and instrumentalness are the most import features.



Three of these—duration, instrumentalness, and loudness—are indeed the most important features for our best model, the Extra Trees Regressor. We can also see that the XGBoost Regressor shows some of the more extreme feature importance ratings compared to other models, with a higher emphasis placed on instrumentalness and liveness, and a lower importance placed on duration. The models' relative importance rankings of these features shows more variation, as they do with loudness, tempo, and energy, whereas speechiness, acousticness, valence, and danceability demonstrate more uniform importance rankings across models.

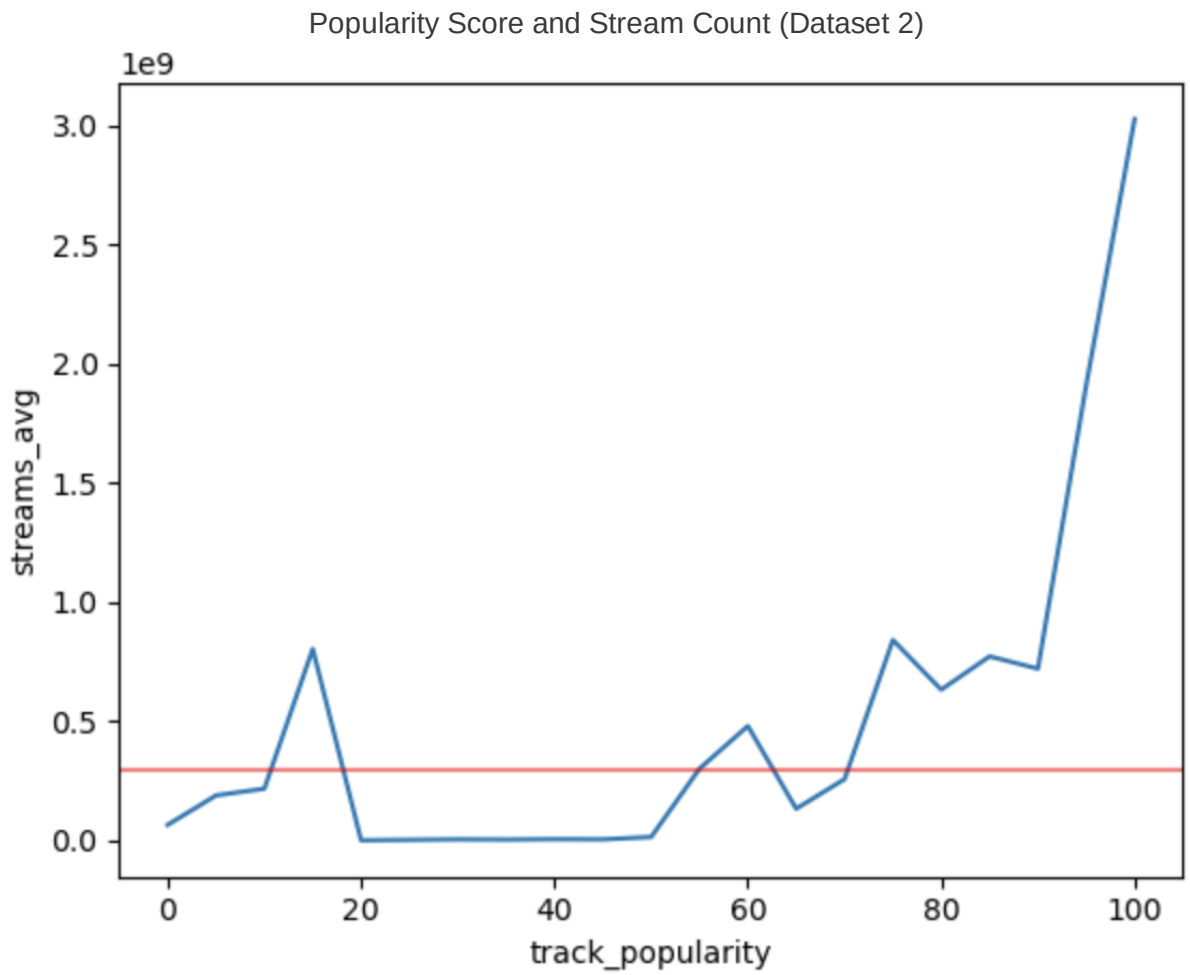
When we compare model performance, we confirm that the Extra Trees Regressor has the lowest MAE on both the training set CV mean and the test set prediction, as well as the lowest standard deviation among the CV scores. The other two best models are the Random Forest Regressor and the Gradient Boosting Regressor, with the XGBoost Regressor and Hist Gradient Boosting Regressor not performing quite as well.



In summary, after trying and tuning 5 models, we have selected an Extra Trees Regressor that is capable of predicting a song's popularity score within approximately 15.30 points based on the song's duration, loudness, energy, tempo, instrumentalness, speechiness, danceability, valence, acousticness, and liveness. Of these, duration, instrumentalness, and loudness are especially important considerations. Great! So where do we go from here?

Future Implications

Now that we are capable of predicting a popularity score within 15.30 points, it is important that we understand what these scores indicate. Are they based on number of streams, chart success, revenue accrued, social media followers, or something else? Since our initial target variable was streams, we sought to generate a ballpark estimate of a popularity score's associated stream count. To do so, we randomly sampled 63 songs from the dataset. For each popularity score in increments of 5, from 0 to 100 (21 increments total), we selected 3 songs and looked up the number of Spotify streams that each of these songs had accrued. The results were a bit discouraging, as we discovered a broad range of stream counts across popularity scores. Some songs with low popularity scores had significantly more streams than other songs with high popularity scores, and vice versa. This doesn't bode well for the veracity of these popularity scores as genuine indicators of a song's success in terms of stream count.



The red line in the graphic indicates 300M streams, which is a little above the median stream count of our first dataset (by about 10M streams). We see that songs with a popularity score over ~70 consistently have more than 300M streams. This indicates that based on this small random sample, in terms of stream count, the upper 30% of our second dataset has about the same lower threshold as the upper half of our first dataset. With our MAE of 15.30, we would want to select songs with a predicted popularity score of 85.30 or higher, in order for them to land in the upper half of stream counts in the first dataset, e.g. recent songs.

Given the extremely limited sample size, however, we can't take these numbers to heart. Should we continue to work with this dataset and popularity score, we would need a much larger sample to convert this score into a more accurate stream count measure. A key lesson in this process is that we should fully understand our target variable before we build a model to predict it, as in this case, the popularity score might not be as ideal an indicator of a song's success as we had originally hoped. Moreover, by addressing the difference in average total streams between each of the popularity score increments from our sample, a MAE of 15 = ~444M streams from the total 63-song sample, or ~429M when excluding the lower and upper 5% of outliers, and ~183M when excluding the lower and upper 25% of outliers.

This reinforces the notion that measuring stream count across many songs entails massive variability, making it difficult to pick a given number of streams as an ‘ideal’ success metric. Proceeding with this project, we need to further evaluate what constitutes a song’s success. If revenue is the goal, how do / can, we measure potential revenue in terms of stream count? When a viable threshold is established, we can predict whether a song will fall above or below this threshold, so as to select songs for marketing that are most prone to success.

So far in this analysis, we’ve examined two datasets. The first includes songs that were primarily released from 2021 – 2023, with the features: danceability, energy, speechiness, acousticness, instrumentalness, liveness, valence, and tempo, and the target variable: streams. With this set we built a Random Forest Regressor that can predict a song’s stream count within ~332M streams, or within 2 deciles of stream count as divided into 10 groups. The second dataset consists of songs released from 1957 – 2020 and includes the same musical features as the first set plus the features of duration and loudness, which proved to be important to our models. Instead of stream count, this set’s target variable is a popularity score from 1-100. We built an Extra Trees Regressor that can predict a song’s popularity score within 15.30 points.

In future modeling, we will dive deeper into quantifying an appropriate stream count threshold to constitute success. We learned that the features of loudness, duration, tempo, and instrumentalness were generally important across models, with our Extra Trees Regressor also highlighting energy as an important feature. Looking ahead, we can explore the Spotify API for a wider selection of songs with all of these features. Additionally, we can more thoroughly explore how genre influences stream counts, and perform segmented analyses that look at music features’ effects on stream counts for songs within given genres. Given the broad variability of stream counts, it might help focus our aim if we choose a genre category within which to focus. This will also give us more insight into which playlists we might want to pursue inclusion in.

If we continue to model using popularity score as our target variable, it will be important that we develop a system for quantifying this score in terms of actionable metrics, such as stream count or total playlist inclusion. We might consider adding members to our team tasked with documenting stream counts per popularity score points, or leverage the automation of AI to complete this tedious process.

Finally, as we continue to better define our metrics and explore available data, we should surely try out additional models and hyperparameter tuning. It is only fitting that Regression Records gave us a regression problem. We have plenty of work ahead, but this analysis should provide a good starting point for understanding how music features relate to stream counts and popularity scores, the relative importance of these features, how these features have trended over time, and the comparative performance of several regression models. With our current working models, we will be able to generate some baseline predictions for stream counts and popularity scores of the songs in RR’s catalog. May this report be another portage in our journey upstream.