Advanced Regressions Methods Project Report

Rami Gorle, Judson Murray, Noah McNeill, Nahom Kitefew

May 5, 2023

Executive Summary

Genre is a primary music sorting metric utilized by both music distributors and the general listening public. However, if songs within a given genre have unique, characteristic features, both listeners and curators may be able to discover new music based on these attributes. This project is an attempt to determine whether genre classification can be accurately predicted based on broadly accepted acoustic attributes. Inversely, we wanted to ask if particular genres demonstrate a typical suite of song attributes.

This project utilized Spotify API song metric data to both train and test a multinomial logistic regression model of genre classification. After running correlation tests to avoid multicollinear variables, we cleaned our data by removing songs with missing data, removing multiple genre labels from multi-genre songs, and reclassifying subgenres into accepted broader genre types. From this dataset, we trained a multinomial logistic regression model and ran a backwards stepwise covariate selection process, using BIC as a metric. Our final model contained eight of the acoustic variables, and correctly predicted genre classification for 67 percent of songs. We also constructed a logistic regression danceability model which utilized the same covariate selection process. The final model contained six covariates and demonstrated a 73 percent danceability rating accuracy in post-hoc testing.

We conclude that our modelling approach was fairly successful, and with refinement would show promise for industry applications. With addition of other algorithms and changing our data partition, we expect prediction rates would rise into a more suitable range.

1 Problem Context

Song genre is a commonly used and understood means of differentiating music types. For generations, genre has been the du jour sorting metric utilized by both music distributors and the general listening public. However, precise genre definition is often a subjective and constantly dynamic grassroots process, which causes shifts in classification as well as comprehension by the public. Additionally, as the algorithm-based music selection style of streaming sites has become the overwhelming standard for music delivery method, both the music industry and listening public have become more specialized and curated in their music preferences. This has further split subgenres into even more finely differentiated music scenes, rendering universal recognition of music types even more difficult for listeners across genre. This poses optimal music selection issues for both listeners and playlist curators. Thus, we believe that the concept of automated song linkage and attribute similarity functions may become more desirable.

In order to create a tool that addresses these needs, we first need to ensure that we could create a model that would operate sufficiently. Thus, our first research question was "Can songs be accurately identified to genre using acoustic attribute data?" We will answer this question with the training of a multinomial logistic regression model. Our second question was "How well can particular song attributes be predicted by other acoustic attributes?" We will answer this question with an example model of danceability.

We utilized data from a dataset of songs in Spotify, and we got the data from Kaggle. This dataset contained 42,305 observations and 22 variables. The variables in the data were mostly from the Spotify API, and these variables were classified by Spotify to account for many metrics in one. For example, "danceability" is a variable that describes how suitable a track is for dancing based on many musical elements such as tempo, rhythm, stability, etc. Spotify uses variables like this to quantify metrics for dancing, vocals, and instruments.

There are some issues for our analysis with using this dataset. Primarily, the genre classifications in this dataset incorporate highly specific subgenres that have very small song sample sizes, and even some well known genres had very few songs. These genres would prove difficult to test accurately in a multinomial logistic regression if not reclassified into broader genres. Secondly, some of the songs have multiple genre classifications, which needs to be addressed for the logistic genre classification in our model. Furthermore, some attributes are missing for some songs, rendering equal testability of all songs difficult. Lastly, some of the acoustic attributes of the Spotify API dataset seem subjective. At the very least, there is very little transparency on how the Spotify created metrics were calculated, meaning our methods are tied to the API datasets unless Spotify were to release more information on their attribute creation metrics.

2 Methods

Our Data

Another key variable we looked for our research question is the genres associated with the songs. Each song was classed into specific genres which we wanted to see if we could predict based on the variables in the data. This variable was a major component to our model, and it was used for

classification as it is a categorical variable. For the covarites, the metrics were broadly categorized into acoustic variables, including danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. These variables are thought to be important predictors of genre classification, and were used to train the multinomial logistic regression model.

It is also necessary to split the data into training and testing sets because we need to evaluate the performance of our predictive model on new data that it has not seen before. If we train and evaluate our model on the same data, it is likely to over-fit the data and not generalize well to new data. Splitting the data into a 70/30 ratio for training and testing, respectively, is a common practice in machine learning so we were able to use the 70/30 ratio for training.

Cleaning the Data

Before conducting the analysis, the data was cleaned to remove any songs with missing data, as well as songs with multiple genre labels or subgenres that were not classified as an accepted broader genre type. This cleaning process ensured that the data was appropriate for use in the model, and helped to prevent any bias in the results.

In addition, correlation tests were conducted to avoid multicollinear variables. Multicollinearity occurs when two or more variables are highly correlated with each other, making it difficult to determine their individual impact on the outcome variable. By removing any variables that were highly correlated with each other, the analysis was able to more accurately identify which variables were important predictors of genre classification.

Model Selection

For our first Model, Multinomial Logistic Regression, we used a backward stepwise selection with Bayesian Information Criterion (BIC) as our selection criterion, starting with all predictors in the model and running the stepwise regression to narrow down the number of covariates our model included. However, when analyzing this data set, we used different metrics of assessing correlation in order to determine which combinations of covariates would pose a potential for cases of multicollinearity in our model. This included primarily a correlation plot with all covariates incorporated (shown in figure 1), but also an examination the VIF values for our stepwise model. Using these inferences, we manually removed the song duration and energy predictors from our model as we felt these two could be potential candidates for multicollinearity.

Similarly, for the second model we tested, we also used a backward stepwise regression with BIC as our selection criterion.

Multinomial Logistic Regression

For our project, we wanted to be able to classify different songs by their genres within our data set. Over the course of this class however, the main classification technique that we learned was a Binary Logistic Regression, or a classification algorithm that only classifies observations into one of two groups. In our data set, the genre variable had several levels that songs could be listed as rather than just two, and as a result our group needed to do some research on classification methods

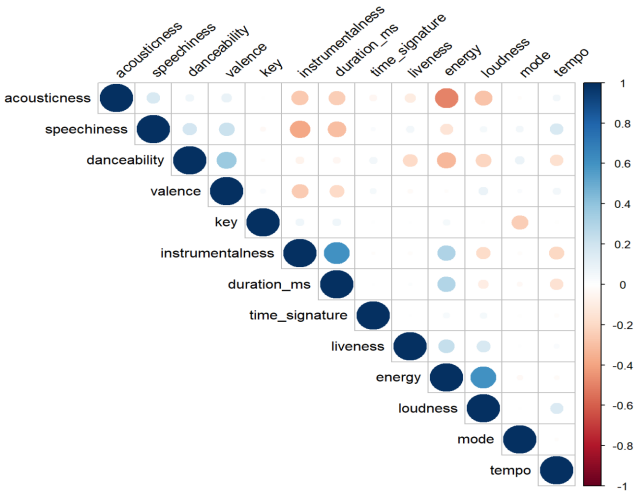

Figure 1: Correlation Plot

we could implement that take into account more than two categories. The Multinomial Logistic Regression is exactly that, an algorithm that allows us to classify observations into multiple levels of a categorical response variable rather than a simple binary classification. Luckily, there is a package in R, "nnet", that allows us to run this model on a selected data set. This implementation utilizes a Softmax Activation Function as the link function for our model that is essentially a generalization of the binary Logit function used in the regular Logistic Regression. Using this function, the model attempts to simulate a linear boundary between response categories by analyzing the data set and assigning probabilities to each level of the response category.

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)}$$

Figure 2: The Softmax Activation Function

Similarly to Binary Logistic Regression, a Multinomial Logistic Regression has several model assumptions that must be met in order to trust inferences we are able to make from it. The first of these is that each observation in the data belongs to a mutually exclusive category of the response variable in reality. This means that when we examine the true genre values for each of our song observations (not the genres predicted by our model), only one genre needs to be listed rather than a combination of genres or multiple genre assignments. This is to avoid correlation within the levels of the response variable. Some of our song observations did indeed have multiple genres listed, so in order to meet this assumption, we removed whichever song genre was alphabetically lower and assigned the alphabetically higher genre as the true genre of a song during the data cleaning process. Ultimately we ended up with the genres shown in figure 3.

Additionally, our model assumes independence of observations and no highly influential points, and

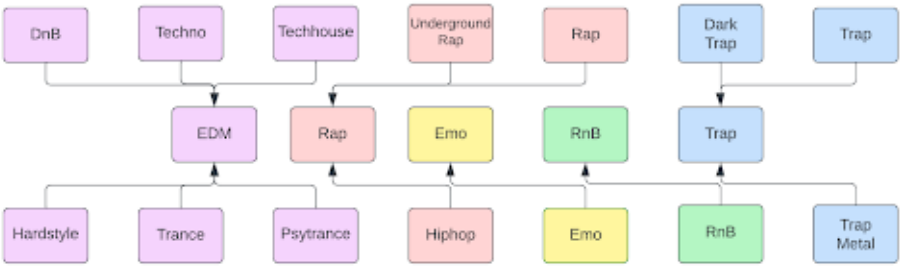

Figure 3: Genre Classification

a measure of Cook's distance as well as standard residual analysis techniques ensured that both of these assumptions were met without having to mutate the data significantly.

In order to test the accuracy and performance of our model, we utilized a standard 70/30 training-testing split for our data set, training our model with the train subset and testing it on the test subset. Our results in this case were decent, but not great, with our model reporting an overall prediction accuracy of 66.85 percent. Ideally, we would like for this number to be higher and more in the 75-80 percent range. Figure 4 highlights the results of our test subset predictions.

Actual vs Predicted Genres According to Model 1

	EDM	**Emo**	**Pop**	**Rap**	**RnB**	**trap**
EDM	3980	84	0	83	2	311
Emo	140	222	0	75	31	46
Pop	21	13	0	82	4	12
Rap	108	56	0	2101	30	347
RnB	15	33	0	398	55	73
trap	898	55	0	637	14	837

Figure 4: Actual vs Predicted Genres

Upon further analysis of our results, we observed that the response level 'Pop' was entirely left out of the consideration of our Model. We suspect this has to do with the way that we partitioned our data. When making our train-test split, we implemented a repeated sample of our data set, and with only 272 total 'Pop' observations, only around .6 percent of all songs being considered were labeled with this genre, thus it was extremely unlikely that they even make it into the training of the model. If we were to continue this research, developing a different way to split our data set would definitely be high priority. Also, another way we could have improved upon our experimental design would be to pick more than one model and do some sort of cross validation in order to compare models, rather than just the one validation split.

When brainstorming classification methods for our project, we came across several different candidate techniques that we could have used instead for our model, and ultimately narrowed our search down to three methods: Decision Tree Model, Multinomial Logistic Model, or a Boosting

Algorithm such as XGBoost. However, there were several key reasons for why we landed on the Multinomial Logistic Regression over the others. First, a Decision Tree model uses covariates in order to create cutoff values for each response category. For example, if we wanted to use a decision tree to predict genre, it might take the "Speechiness" predictor and create a high cutoff for determining Rap and Trap, but this could mean that some of the less verbose Rap or Trap songs could get ignored or misclassified. This model only really informs us on which predictor is most influential for determining each category, but does not provide a classification informed by a large combination of predictors simultaneously, which is what we ultimately wanted. As far as Boosting algorithms, we felt that using Boosting could potentially offer us more accurate (and certainly faster) results, however given our time constraint and their exclusion from overarching scope of this class, we felt that using the Multinomial Logistic Regression would be more in line with the types of techniques we have covered.

Logistic Regression

While doing the multinomial logistic regression model, we stumbled upon another question related to other variables. We saw the potential of how danceability could be used for dj's or people who want to update their playlist with songs with high danceability. We wanted to see how it would predict the danceability of song based on speechiness, acousticness, valence, tempo, duration, and genre. We believed binary logistic regression was the best approach because we can split danceability into ones and zeros based on if they had a higher value than the mean, which was 0.63. This problem was mostly a proof of concept model to see if a person could curate a playlist or set based on a variable, which in this case was danceability.

Logistic regression is a statistical model used to model and predict the probability of a binary outcome, and in this case one represented high danceability and zero represented low danceability. In logistic regression, the dependent variable is modeled using a logistic function, which transforms the linear combination of independent variables into a probability between zero and one. The reason we used logistic regression for danceability is because danceability is a unique variable that Spotify used to classify a song's ability to dance along with. This metric is very valuable as it can help people curate songs based on dance. However, the Spotify API uses many unique variables as metrics which in theory could also be predicted with a logistic regression model.

The first step was to set a reference value to classify if the danceability was higher or lower than the referrence value. We did this by getting the mean of danceability, so we classified each song as one or zero based on if the song's danceability was higher or lower than 0.63. We then created our logistic model in which we checked all the assumptions and made sure none of the variables have high collinearity; we also did not do any transformations on the variabales themselves. After passing all the assumptions, we made prediction probabilities based on the testing data set. We used these prediction probabilities to make a logistic odds plot.

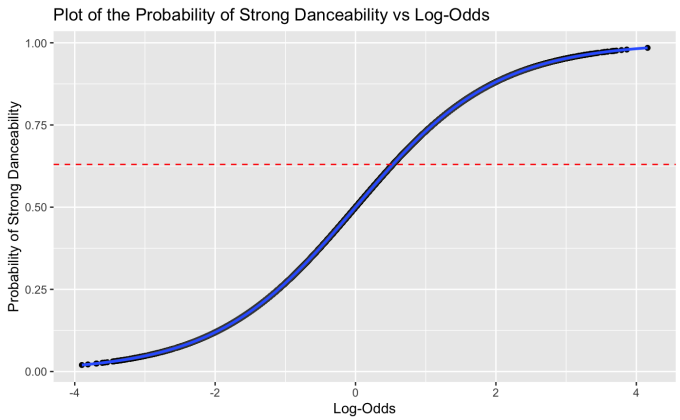

Figure 5: Logistic Regression Plot

From this plot we can see that the model did a good job predicting the probability of danceability based on the log odds. The blue line shows the prediction, the black dots represent the actual points, and the red dashed line represents the reference level which was the mean of danceability. We ca see that the plot has a good overall fit because it shows a tight clustering of the predicted probabilities around the observed outcomes. This plot is also strong because the predicted probabilities showed distinguish between the two outcomes which is represented by zero and one.

Error
Prediction #2

	No	**Yes**
No	319	681
Yes	2262	7580

Figure 6: Error Prediction

As we can see in Figure 6, we had a prediction accuracy of 72.86% which is really good in this case because it accurately predicts if a song has high danceability or not. This model was very interesting to me because we included genre in, and this is very helpful because certain genres have higher danceability scores than others, so it was important to include genre as predictor variable.

Overall, this model is a proof of concept to show that dj's, artists, or music listeners can curate sets or playlist with high danceability. If there were more analysis being done on this model, we would want to understand how danceability is calculated. This could give us a better clue on what should be used to predict it. We may also be able to use other obscure variables that Spotify uses to predict those as well. his logistic model was a good model selection because it showed that finding the probability of a variable reaching higher than the mean is good for producers and dj artists. This model can be applicable to both producers and consumers. As one can curate a playlist based on high danceability, while one can createa set of songs to perform with high danceability.

3 Conclusion

Overall, our first model correctly predicts genre for 67 percent of songs, which demonstrates that song attributes can predict genre with significant accuracy. Optimally, we'd like to see prediction rates near 75 to 80 percent. However, our model does predict certain genres with a high degree of accuracy, as EDM and Rap see 80 to 90 percent prediction rates. Secondly, we show in our second model that one can use Spotify metrics to predict danceability for a song. This presents all sorts of potential usage cases, as one can likely apply this to other acoustic attributes as playlist.