Music Genre Prediction

By: Katy Bohanan

Abstract

The objective of this project was to develop an accurate music prediction model using preprocessed music data. The model predicts the genre based on a set of attributes, including danceability, energy, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration. The Random Forest algorithm was chosen due to its ability to handle categorical data in large datasets. The results show that the model was able to accurately predict dance/electronic genres based on the attributes provided, however, it lacked accuracy with other genres like Rap and R&B.

Introduction

The question being addressed is can we accurately predict a song's genre based on attributes like energy, tempo, key, danceability, etc.? This project aims to answer that question through data collection, visualization, and machine learning.

Related Work and Data Collection

The idea for this project was found on DataCamp.com, project title: Classify Song Genres from Audio Data. In this project, music data is used to classify songs as either rock or rap. Wanting to do something similar, I found a dataset on Kaggle. This dataset includes preprocessed data from around 42,000 songs. These data include Song Title, Danceability, Energy, Speechiness, Loudness, Key, Acousticness, Instrumentalness, Liveness, Valence, Tempo, Mode, Duration, and Genre. Each song in the dataset has a unique set of attributes which can be used to determine the genre. The genres in this dataset are rap, r&b, trap, trap metal, emo, pop, dark trap, techno, tech house, drum & bass, trance, psytrance, hardstyle, underground rap, and hip-hop.

Methodology

The first steps of the project were cleaning up and understanding the data. Some columns, like the song's title and Spotify URL, in the dataset were completely unnecessary moving forward and were removed from the dataframe. I also combined genres "rap", "hiphop", "underground rap" and "dark trap" and "trap" in order to reduce redundancy. The pop genre was also removed because it was such a small part of the dataset with only 146 rows. This left us with 41,844 songs and 11 genres. After this, the song counts per genre are as follows: 10,751 rap, 7,565 trap, 2,999 trance, 2,975 tech house, 2,966 drum & bass, 2,961 psytrance, 2,956 techno, 2,936 hardstyle, 2,099 r & b, 1,956 trap metal, and 1680 emo. I then created a correlation matrix to see which values were more or less correlated with genre. From this information, I opted to drop the attributes that were less correlated, which were mode, loudness, and time signature.
In order to understand the data better, I created a few scatterplots to see the relationship between attributes which I believed would have the strongest correlation with each other. Danceability and valence had the strongest correlation, which can be seen from the scatterplot. It would be safe to assume songs that are high in valence are also high in danceability. There are not strong correlations among the other attributes, however, we can see in the scatterplots that songs in each genre of music tend to have similar attributes. The exception to this is the rap genre – in the scatterplots, we see data points for rap all over the plot, meaning that there is high variance within the genre.

After cleaning up the data, it was sorted into labels (genre) and features. The data was then split into training and testing categories. Eighty percent of the data would be used to train the model and the other twenty percent used to test the model. I also define random_state for the sake of reproducibility.

Based on the needs of this project, Random Forest seemed like the best algorithm to use. Random Forest works well with categorical data, reduces, overfitting, handles large datasets, and includes feature importance. All of these reasons lead me to believe I would get the best outcome using Random Forest Classifier. In order to find the best parameters for the model, I used RandomizedSearchCV. This would search random values for parameters n_estimators, max_depth, and min_samples_split. I used a 5 fold cross validation scheme and weighted f1 as the scoring method. I chose to use the weighted f1 score over accuracy because there is uneven distribution of classes. RandomizedSearchCV returned 135 trees, max depth of 11, min_samples_split of 6 are the best parameters to use for the model. These parameters return a weighted f1 score of 0.74. Creating a graph for feature importances, we can conclude that 'key' is not a valuable feature and can be removed from the dataframe. Tempo by far has the greatest importance, followed by duration and instrumentalness.

Results

The results show that the Random Forest model was able to very accurately predict drum & bass, psytrance, and tech house. Techno, hardstyle, and trance were slightly less precise, but still fairly accurate. Rap, trap, and emo all had an f1 score between .65 and .75, making them less precise. The model could not accurately predict r&b or trap metal. While many dance genres have very specific guidelines, the other genres may have greater variances in their attributes. The model might have a lot of false positives or negatives for something like rap because it rarely falls into a specific tempo, for example. There were also quite a few similar genres in this dataset, which could have led to less accuracy. I believe I could have achieved greater accuracy with a dataset that had less genres, but more distinguishable genres, such as jazz, rock, rap, and techno. Overall, I am happy with the fact that the model is able to accurately predict the dance and electronic genres, but moving forward, I would like to see how I can further optimize it.