Omar Aboubakr
Hussein Faissal Farid Faissal
Juliette Clark
Rebecca Mirvish
Deepta Jasthi

Song Genre Prediction Using Spotify Features and Genius Lyrics

1. Motivation

The use of machine learning techniques in the field of music has gained significant attention in recent years. In this project, we sought to explore the potential of using Spotify data to predict the genre of a song. With the vast amount of music available on the platform and the rich metadata provided by Spotify, there is a wealth of information that can be used to train a machine learning model to make such predictions. By doing so, we can potentially improve the organization and recommendation of music on the platform, as well as gain insight into the characteristics that define different music genres. We also sought to take our analysis a step further and understand the predictive power of lyrics – sourced from Genius.com – in comparison to song attributes from Spotify when it comes to determining music genre. This project serves as an introduction to the application of machine learning in the field of music and the potential benefits it can bring.

2. Dataset

2a. Spotify

Our Spotify dataset was gathered from Kaggle. The dataset is composed of 21 columns, and with non-numerical columns including artists, album name, and track name. The unique identifier for each song was the track_id. However, after some initial examination we realized there were a few duplicate songs. Therefore, to fix this issue we grouped the dataset by `track_id` to get rid of any duplicates. This left us with a data set of 21 columns and 89,741 rows, 1 for each song.

The next issue we had was making it so that the non-numerical columns could be used in a categorical machine learning model, which involved taking a bag of words approach with artist, album name, and track name. After some basic cleanup involving removing punctuation, numbers, and stop words, we discovered an issue when attempting to use stemmer to create stem words for all the words in our dataset. That issue was that a lot of the songs were not in English. Therefore, we used a dictionary of English words from `nltk` to remove any words that were not in english, before applying our stemmer. Finally, we combined all the words from artists, album name, and track name into one column that got vectorized to be used in our machine learning process.

One other issue we faced was that due to the size of the dataset, and low RAM, our runtime for some of the models was too long. To fix this we made a smaller version of the cleaned dataset called smaller, which only included songs from the top 30 most common genres. This reduced the number of songs from 89,741 to 29,817. Then examining this dataset, we realized that after removing non-english words, half the songs had empty values for the words column, so we dropped any song which had no english words whatsoever. This left us with 15,760 songs, but when comparing the accuracy of our model on the 29,817 songs to the 15,760 songs, the accuracy was of course much better with the 15,760 songs. Our final dataset was then 15,760 rows (songs) by 91 columns, where the additional columns on the 21 were from the vectorized words column (1 column per word).

2b. Genius.com

After predicting track genres using song attributes as features, we wanted to see if we would have even more success with an NLP approach, predicting genres using song lyrics. We decided to only include songs with lyrics that are in English. We used the `langdetect` library on all the song titles in our original Spotify dataset from Kaggle to filter out any songs with non-English titles, this can be found in the `getEnglishTitles.py` module.

In the `scrapeSongId.py` we took a random sample of 6000 songs from the dataset and attempted to scrape their Genius.com Song IDs using Genius API. The scraping algorithm was slow because of the speed of API calls, each Song ID took about 1.1 seconds to scrape, therefore we did a random sample of 6000 instead of doing the whole dataset. This process took about 1 hour 20 min to run. We filtered for non-English songs again at this stage, tagging songs with a language tag other than 'en'. Of those 6000 songs, about 4500 were successfully found and their IDs were exported into a csv to be used in the next model.

In `scrapeLyrics.py`, we used Genius API again to scrape the URL of each song and then used `BeautifulSoup` to extract the lyrics and used RegEx and string manipulations to clean up the lyrics. Each song took about 1.3 seconds average to extract and clean, which took about 1 hour 20 min total. About 3,700 songs were successfully scraped.

After we obtained our lyrics dataset, we loaded it into a notebook for preprocessing and modeling. To clean the table, we ran `langdetect` once gain on the lyrics to remove non-English songs. We also removed songs without lyrics, and we filtered the dataset to only include songs from the 20 most frequent genres. We wanted to reduce the number of possible labels to compensate for the lower sample size and prevent overfitting to build more accurate models. We then cleaned the song lyrics by converting them all to lowercase, removing punctuation and stop-words, and lemmatizing the words.

3. Models and Analytics

3a. Song Attributes Approach

The baseline accuracy for predicting genre based on song features (predicting `show-tunes` as the majority class) was 5.39%.

When using the logistic regression, the accuracy was 16.6%. This is a slight increase in accuracy.

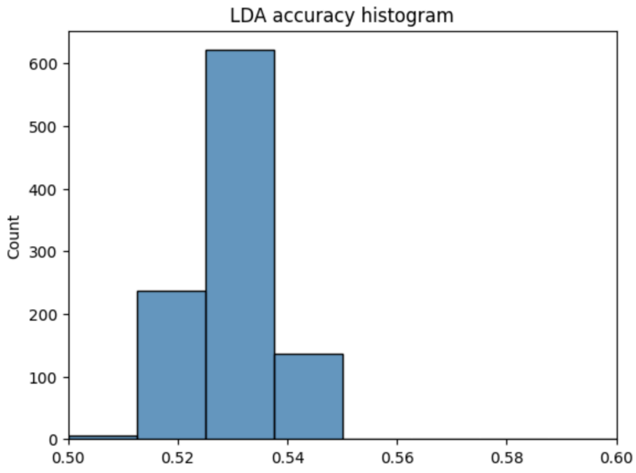
For the CART model for predicting genre based on song features, the accuracy was 61.44% with the default Decision Tree Classifier hyperparameters. After using 5-fold cross validation and implementing Grid Search on the ccp_alpha, max_depth, min_samples_leaf, and min_samples_split hyperparameters the accuracy increased slightly to 61.74%.

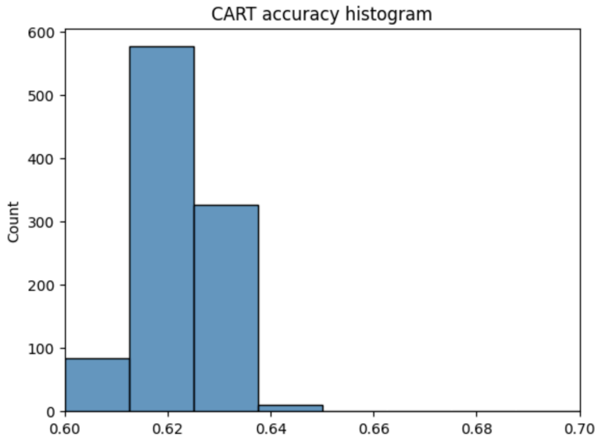
When using a random forest classifier and setting `max_features` parameter to `'sqrt'`, and `random_state` to 80, the random forest accuracy was 78.6%. The 5 genres that were most commonly misclassified were Spanish, synth-pop, guitar, trip-hop, and idm. This is a substantial increase in accuracy.

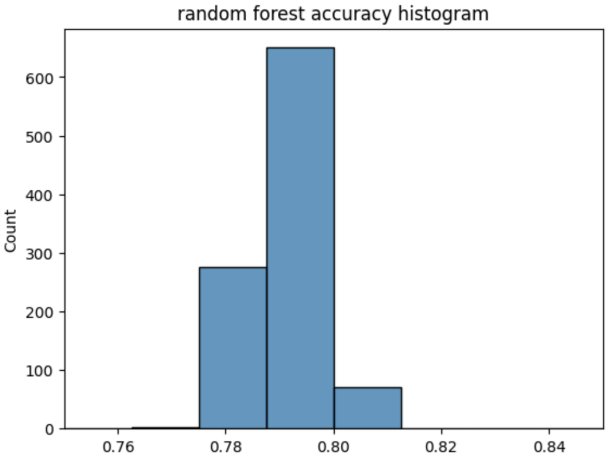
For the LDA model for predicting genre based on song features, the accuracy was 54.25%, which was better than logistic regression, but much worse than random forests. The 5 genres that were most commonly misclassified were show-tunes, children, tango, salsa, and reggaeton.

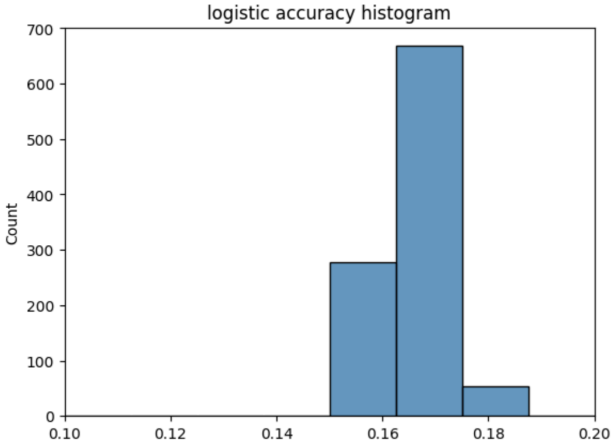
Based on the success of CART and Random Forest models, we decided to try Boosting. Our initial Gradient Boosting model was nearly as successful as our Random Forest model with an accuracy of about 78.3%. When using XGBoost, our accuracy was 81.1%, which makes it our highest performing model.

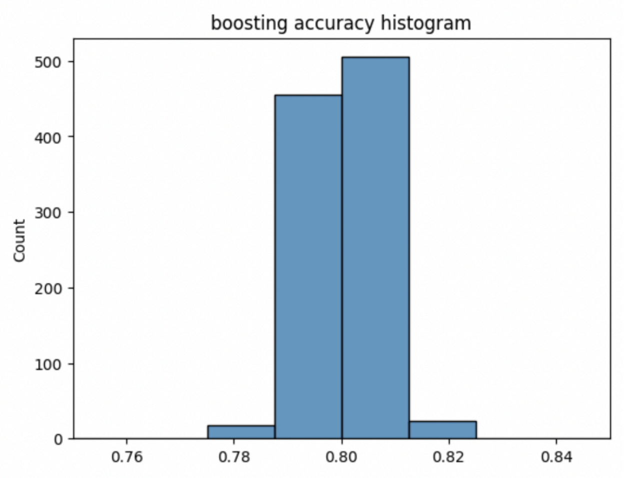
After running all these models once on our original train-test split, we wanted to bootstrap our data set to get a better understanding of the average accuracies of these models. Using the limited RAM of Deepnote (similar to Colab) we were only able to bootstrap 1000 samples which took around 20 mins to run. Looking at the histograms below, we can see the accuracies of each model across the 1000 bootstrapped samples.



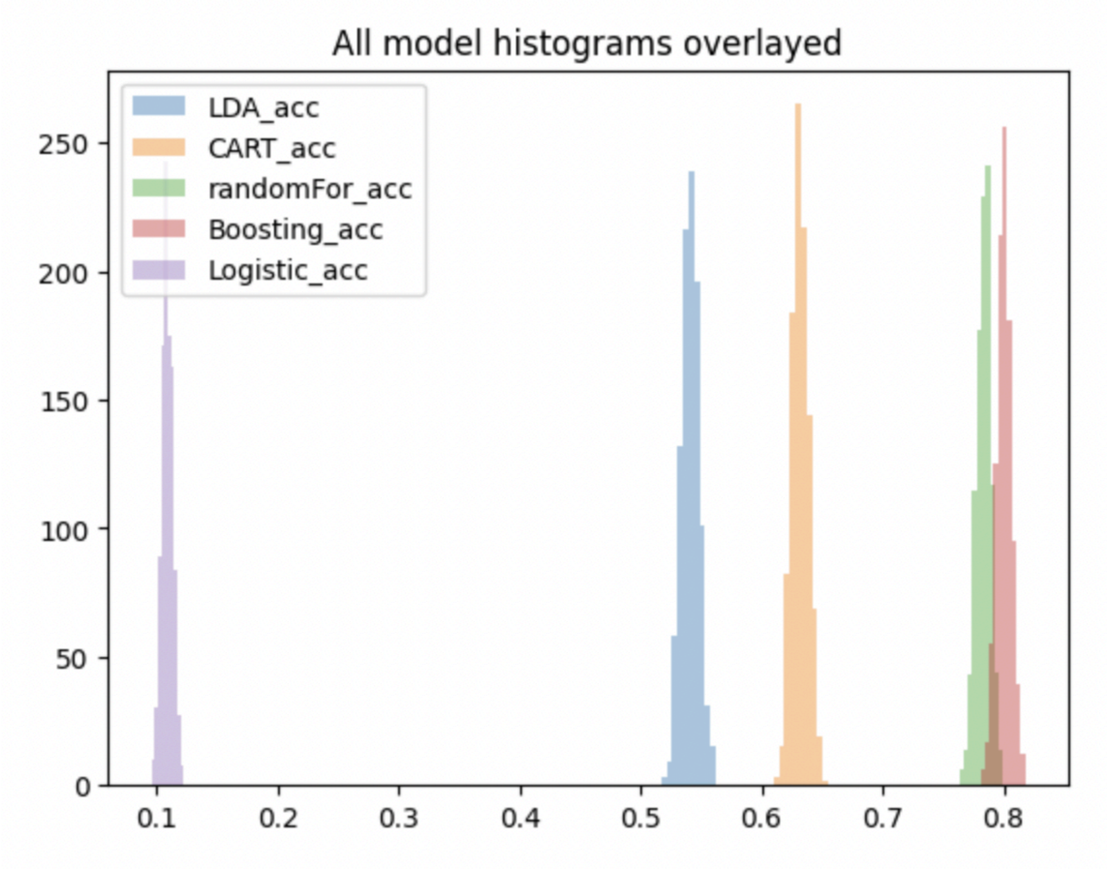








Looking at each individual histogram, it is difficult to compare the models because the x-axis is different for each mode, therefore we graphed all the models on one chart to get a better picture. From this we can clearly see that XGBoost was the best model in terms of classification of genre, with an average accuracy near 0.805. We can also see that logistic regression was clearly the least accurate model with an average accuracy of 0.166.



3b. Song Lyrics Approach

For the NLP approach of modeling using song lyrics, our baseline accuracy (predicting `garage` as the majority class) was 10%. We tried 3 CART-based models.

The first was a simple Decision Tree Classifier with Cross-Validation. The model gave us 17.8% accuracy which was higher than our baseline but still incredibly low. Our highest performing model was a Random Forest that gave 25.8% accuracy, which still is not high. Gradient Boosting performed similarly to Decision Tree, with 18.4% accuracy. Cross-Validation took an abnormally long amount of time in our Decision Tree, and we still had a low accuracy, so we opted out of using it in our ensemble models.

3c. Comparison of Modeling Approaches

Our original goal was to find the best approach and model to predict song genre. The song attribute approach proved to be more effective than the lyrics approach (for this dataset). We cannot say for certain if this conclusion is generalizable. It is likely that some of the difference in model performances can be attributed to our practical constraints, not necessarily a fundamental difference in the predictive power of both approaches. Mainly, our sample size was low due to the difficult web-scraping process. We can infer that a musical attribute-based approach works better than a lyrical approach for more granular genre labeling. If someone were to create a similar-sized Spotify dataset with fewer genres that each contain more songs, and if we had the machine power to extract lyrics for every song and run predictions on them, it would be interesting to compare both approaches again and see if the results are different.

4. Impact

While our NLP approach to predict genres using song lyrics didn't have high enough accuracies for us to say that this approach works, in the future we could try again with a higher sample size of songs to better train our data and reach a conclusion. However, the web scraping pipeline we built to provide the lyrics can be very powerful. Its ability to take in a song title and artist name and return the lyrics can be useful for anyone doing modeling or analysis work in a similar area.

However, using song attributes to predict genre turned out to be fairly successful with XGBoost having the highest average accuracy of about 0.805. This shows us that genres are predictable based on song characteristics, and with a higher sample size of data and potentially more specific attributes, genres could be predicted with even higher accuracy. This provides many benefits for a consumer ranging from better song recommendations to more accurate genre-related sorting mechanisms within the app.

To expand our analysis, we could modify our machine learning models to include songs in languages other than English. Considering the fact that Spotify is a global platform this would be a logical next step to increase the scope of our work and allow us to work with more data.

5. Project Submission

Here is the code and data files link and here is the data source link.