

Exploration, Grouping, and Prediction of Genre in Spotify Songs

Team Spotiflies: Aubrey Clark, Kennedy Kang, Aaron Kwan, Joanna Liu, Ethan Mai, Aster Phan

- [GitHub Link](#)

I

In this project, we predict song genre using a [massive dataset](#) of **over 140 thousand songs** from the music streaming platform Spotify. The data set had information about each song's popularity, genre, and other musical qualities. We attempted to classify a song into one of 10 genres by using up to 12 different predictor variables.

- Relevant Predictor Variables: popularity , duration_ms , explicit , danceability , energy , loudness , speechiness , acousticness , instrumentalness , liveness , valence , tempo
- Genres: classical , electronic , folk , hip-hop , jazz , metal , misc , pop , rock , world

II

The main feature we are interested in with this dataset is **song genre**, in particular how it relates to other features. We want to test if the musical qualities of a song are strong predictors of its genre. After cleaning, our dataset contained 114 different genres, ranging from emo to classical to industrial and more. We are interested in investigating things such as:

- The qualities of songs that are correlated to genre.
- If certain genres are more similar to each other than others.
- If we can predict a song's genre through its other qualities.

Understanding these relationships is crucial for several reasons. First, accurate genre prediction can *enhance recommendation systems*, providing users with a more personalized and enjoyable listening experience. For artists and producers, insights from our analysis can *guide the creative process*, helping them understand trends and preferences within different genres. Finally, we contribute to the broader field of musicology by providing *data-driven insights* into the characteristics that are different in musical genres.

III

We tried numerous different methodologies, including **linear regression**, **logistic regression**, **decision trees**, **PCA/clustering**, and **neural networks**. Our key methodologies were **decision trees** and **random forest**, which worked most effectively to predict a song's genre through its other qualities.

With decision trees, we were able to obtain a respectable accuracy compared to our other methods. However, due to the depth of the decision trees, these results were not easily interpretable. Finally, after using an ensemble of trees in a **random forest**, we achieved predictions that were significantly more robust and accurate than the previous methods.

Neural networks also appeared to be a viable alternative, however overall accuracy metrics paled in comparison to decision trees. However, recall scores for classical music and metal were better than they were for **random forests**.

IV

Using a train-test split of 60:20:20 and an ensemble of 200 trees, each with a maximum tree depth of 20 (both parameters learned using brute-force cross validation), we were able to achieve a total **Accuracy** of `0.56` . We generated a **Classification Report** showing the `precision` , `recall` , and `F1 score` for each genre, which helped us understand the performance of our model across different genres. In addition, we generated a **Confusion Matrix**, which provided a visual representation of the true positive, true negative, false positive, and false negative prediction rates, allowing us to see what errors our model was making.

Finally, we used **Feature & Permutation Importance** to identify which predictor variables had the most significant impact on genre classification. In particular, we found that `popularity` , `acousticness` , `instrumentalness` , and `danceability` were the most important to predict genre. This makes sense because some genres are more appealing to wider audiences and these features could logically seem to be able to predict between genres. Interestingly, `mode`, `time signature`, and `key` had little or negative permutation importances, indicating they are not important to predicting track genre.

While our Random Forest model provided robust and accurate predictions, it also had some limitations. One significant limitation was the model's tendency to overfit (the training accuracy was over 90%), especially with a large number of trees and depth. We tried to combat this by limiting the *maximum tree depth*, which resulted in lower accuracy. In addition, the computational resources required for training and tuning the model were substantial (each run took dozens of minutes) which could be a constraint for larger datasets or more extensive hyperparameter optimization.

All in all, our maximum accuracy was still well below `0.7` , which may be a limitation of our dataset itself; this dataset may not capture all the nuances and variations between genres. Additionally, the subjective nature of genre classification means that there can be overlap and ambiguity between genres, making it challenging to reach high accuracy. Future work could involve refining the genre labels, incorporating more diverse and representative data, and experimenting with different model architectures and techniques to improve performance. Despite these limitations, our project demonstrates the potential of machine learning techniques in predicting song genres and provides a foundation for further exploration and improvement.

V

To run the code in **project_code.ipynb**, ensure that Jupyter Notebook or Jupyter Lab is installed, along with the necessary dependencies. First, clone this repository to the local machine and navigate to the project directory. Launch Jupyter Notebook by running `jupyter notebook` in your terminal, open **project_code.ipynb**, and execute the cells sequentially. Alternatively, use the `Jupyter Notebook` extension and use "Run all cells."