Utilizing Machine Learning and Spotify's API for Genre Classification in Music

Ethan David Rayala, Bodhi Harmony, Kevin Dai, Ahmadou Bamba Diouf

December 15, 2023

1 Introduction

This project's objective is to create a machine-learning model capable of identifying a song's genre by analyzing its audio features gathered through the Spotify API. This process involves collecting a range of audio features from various songs, focusing on aspects like rhythm, melody, and timbre, and then utilizing them to train a machine-learning algorithm designed to categorize songs into one of five genres. Additionally, the project assesses the model's effectiveness through an accuracy score. The outcomes of this research have potential implications for enhancing music recommendation systems, streaming services, and the broader field of music analysis.

2 Previous Work

Several past studies have applied machine learning techniques to music genre classification based on sourced audio features from platforms similar to Spotify. For instance, Yang et al. in 2016 processed over 500,000 songs with the Spotify API. The team extracted key audio features and employed a wide range of machine learning methods, including logistic regression and neural networks, to determine the songs' genres. This approach, using a logistic regression model, yielded an accuracy percentage of over 70%. Similarly, Lee et al. in 2018 adopted a comparable methodology but further integrated user listening patterns and demographic data, and in turn, returned an accuracy of over 80% using deep learning techniques. Kumar's 2022 project applied the K-Nearest Neighbors algorithm, achieving around 70% accuracy in music genre classification. Additionally, Ranganath's 2023 study utilized the GTZAN Genre Classification dataset, comprising 1,000 tracks across 10 genres. This project explored various audio features and employed Convolutional Neural Networks, ultimately reaching an accuracy of 92.93% in genre classification.

3 Methodology

3.1 Locations

In our Github, (`https://github.com/erayala15/COMP-562-Final-Project-Spotify`), there is a file called 'fetchspotifyapi.py' which runs the Spotify API to get the set of songs and writes them to a file called 'songdata.json'. The model is in 'model.ipynb' which reads the content of 'songdata.json'.

3.2 Intro

The primary focus of our project was the gathering and collecting of data. We relied exclusively on the Spotify API for this purpose. As detailed in the documentation available on Spotify's website, this API allows for the retrieval of various attributes of a song, based on a set of song IDs. These attributes, which range from 0.0 to 1.0, encompass several aspects of the music.

- Acousticness: A 0.0 to 1.0 confidence measure of how acoustic a track is.
- Danceability: How suitable a track is for dancing-based musical elements including tempo, rhythm stability, beat strength, and overall regularality.
- Energy: A perceptual measure of intensity and activity. Energetic tracks are typically fast, loud, and noisy.
- Instrumentalness: Predicts whether a track does not contain vocals. "Ooh" and "aah" sounds are considered instrumental in this context. Rap or spoken word tracks are clearly "vocal".
- Liveness: Detects the presence of an audience in the track. the higher the liveness of a track, the more likely the track was performed live.
- Mode: Indicates the modality, or key (major or minor), of a track, the type of scale from which its melodic content is derived. A major key is represented by 1 and a minor key is represented by 0.
- Speechiness: Speechiness detects the presence of regular spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audiobook, poetry), the closer to 1.0 the speechiness value is.
- Tempo: The overall estimated tempo of a track measured in beats per minute (BPM).
- Valence: Measure of how positive or negative a track sounds. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

For this project, our goal is to use the above attributes found in a song or track to train a model to accurately predict the genre the song or track belongs to.

3.3 Challenges

Our initial challenge involved identifying a collection of songs belonging to specific genres. Utilizing Spotify's search API, we could specify a genre in the search parameters. However, it's important to note that Spotify's search API does not directly associate genres with individual songs; instead, genres are linked and assigned to artists and albums. Our approach was to use the search API to obtain song IDs and then retrospectively assign genre tags to these songs for our machine-learning model. With this approach, a significant issue we faced was that the songs returned from the API searches were only loosely related to the desired genres and often did not pass the 'ear test' for that genre. Moreover, the number of songs returned for each query was inconsistent. In our preliminary tests utilizing the songs pulled with this method, the model's accuracy fluctuated between 50-70%. It is worth noting, however, that when we broadened our search criteria to include distinct genres like 'classical piano', 'electronica', and 'screamo', our model's accuracy impressively increased to 96%.

3.4 The Playlist Solution

After attempting to locate songs via Spotify's search API, we shifted to using Spotify's official playlists for genre-specific song selection, still applying our method of retrospectively assigning genre tags. This strategy had two drawbacks: potential bias in the song selection and a limited sample size. Nonetheless, given our time limitations, this seemed like the best approach as the songs from these popular playlists appeared to be more representative of their respective genres than compared to those found through the search API, especially when subjected to the 'ear test'.

3.5 Model

For our model, we chose Python and scikit-learn. We compiled the data into a data frame and then divided the data into training and testing sets for model training and performance evaluation, respectively. We then utilized scikit-learn's LabelBinarizer() to transform categorical data into binary format, a requirement for most inputs for machine learning models. Next, we prepared a matrix with the selected attributes for training, and then subsequently utilized the RandomForestClassifier() function from scikit-learn to create 100 decision trees to classify the data for the training process. The model was then fitted to the training data using the fit() function, and predictions were made on the test set with the predict() function.

3.6 Evaluation

To assess our model, we used the 'score' method from sklearn, which calculates the mean accuracy (ranging from 0.0 to 1.0) for the test data against our genre labels. We noticed that accuracy was higher when only two genres were used and decreased as more genres were added. By selectively including specific attributes in the model, such as danceability, speechiness, and tempo, we were able to further improve its accuracy. Interestingly, removing the 'mode' attribute, which represents the major or minor key of a song using only a 0 or a 1, slightly enhanced the accuracy.

4 Conclusion

4.1 Results

In the end, our selection included 50 songs spanning five different genres: Metal, Salsa, Jazz, Country, and Reggaeton. Impressively, our model achieved a .94 accuracy rate. This particular mix of songs revealed that our model was exceptionally proficient in identifying and distinguishing Metal and Jazz, correctly classifying every instance of these two genres. Moreover, it never mistakenly identified any song from the other genres as either Metal or Jazz.

4.2 Constraints

The primary constraint in our study was the limited size of our sample comprising only 250 songs, with 50 reserved for testing. For future improvements, we would have enhanced our method by either better leveraging the Spotify search API or by redesigning our approach to incorporate a broader selection of songs from Spotify's official playlists. A promising strategy to address this issue could involve utilizing Spotify's genre-specific and regularly updated playlists of new music over an extended time frame, considering these playlists are refreshed weekly.

4.3 Future Work / Impact

The significance of our research lies in its empirical approach to identifying the most critical attributes that define a genre based on Spotify's metrics. Future work could expand beyond increasing the sample size to algorithmically fine-tuning the selection of genres and song attributes used in the model. Our model's most unique aspect is its reliance on data pulled not from preexisting song databases. Future exploration could look into the importance of specific attributes in contributing to the distinguishing between genres. For instance, in differentiating classical from rap, attributes like acousticness and speechiness were more accurate than valence and liveness, suggesting the potential of certain attributes as key indicators for classification between certain genres or as universally strong predictors in general.

References

1. C. Ranganath, "Music Genre Classification using CNN," Clairvoyant Blog, 11-Apr-2019. [Online]. Available: https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn/. [Accessed: 08-May-2023].

2. D. Lee, Y. Lee, H. Lee, and J. Lee, "Using Deep Learning and User Data to Predict Music Genres," in Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2018, pp. 447–449.

3. scikit-learn. "RandomForestClassifier - scikit-learn 0.24.2 documentation," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 08-May-2023].

4. S. Kumar, "Music Genre Classification Project using Machine Learning Techniques," Analytics Vidhya, 29-Mar-2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/03/music-genre-class