Genre Classification of Spotify Songs using Lyrics, Audio Previews, and Album Artwork

Kevin Haugh and Tyler Dammann

Introduction

Genre classification is a problem that Spotify relies on humans to resolve, but it is a challenging and highly subjective issue, and in many cases, even humans disagree which genre a song falls into. We aimed to automate the genre classification process using machine learning techniques applied to a song's lyrics, its audio waveform, and its album artwork.

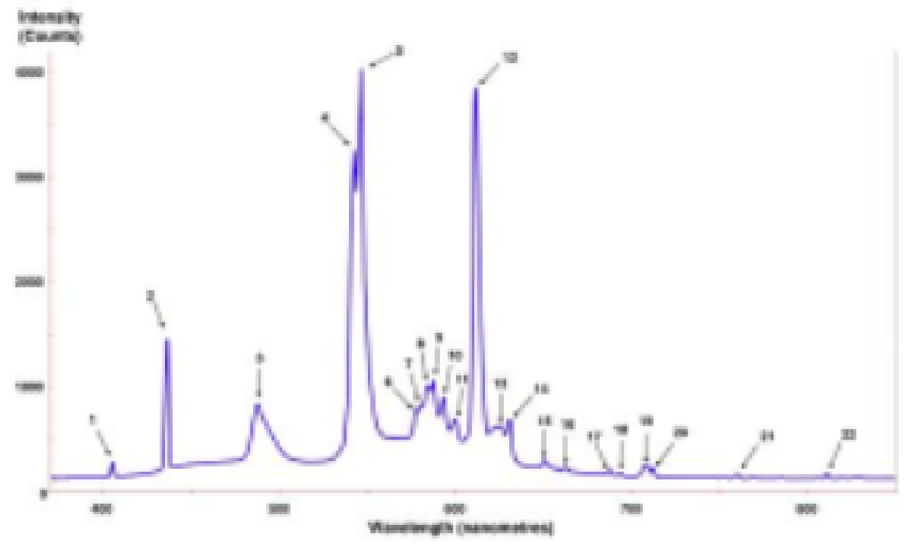
Data

Model 1: Naive Bayes on Lyric Data

Song lyrics, song names, and artist names were taken from a dataset on Kaggle that was originally scraped from LyricsFreak. Lyrics were cleaned by removing delimiters, stopwords, and words not in the English dictionary.

Model 2: Recurrent Neural Network on Sound Data

The Spotify API was used to obtain 30 second previews of each song in the Kaggle dataset. Songs were trimmed to the middle 10 seconds, frames were extracted at a sample rate of 23 milliseconds, and MFCC features were taken at each frame. Mel-Frequency Cepstrums (MFC) represent the intensity of sound at varying frequencies, and Mel-Frequency Cepstrum Coefficients (MFCC) are the amplitudes of the MFCs.



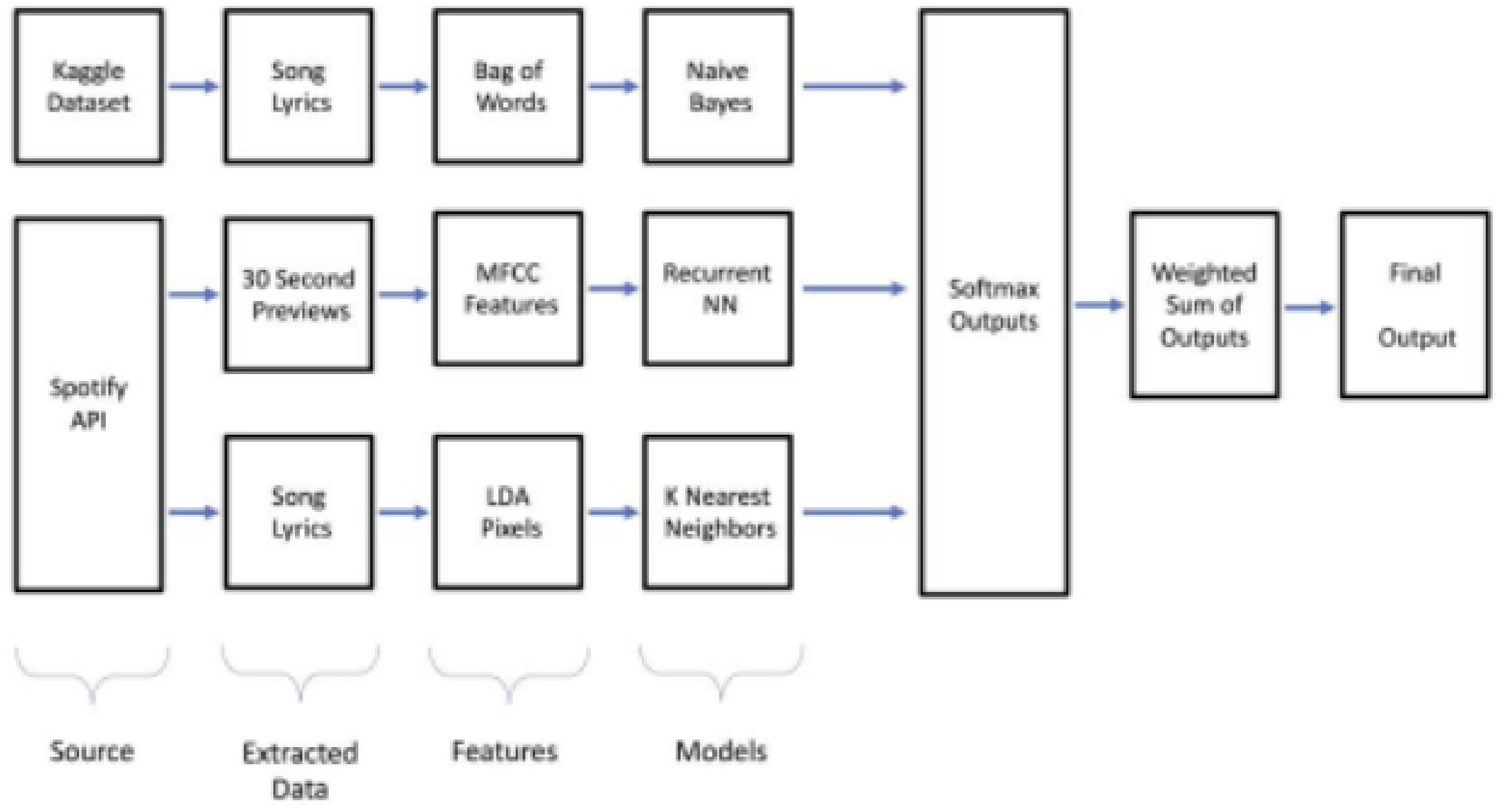
Example MFC and corresponding MFCCs

Model 3: k-Nearest Neighbors on Image Data

The Spotify API was used to obtain album artwork for each song in the Kaggle dataset. Album artwork was converted to RGB matrix representations, and Principal Component Analysis was used to reduce the number of covariates.

Final Dataset

The final dataset contained 4,000 songs in the genres Christian, Country, Jazz, and Metal. These songs were split 80/10/10 into training, development, and test sets.


Kaggle Dataset → Song Lyrics → Bag of Words → Naive Bayes → Softmax Outputs

Spotify API → 30 Second Previews → MFCC Features → Recurrent NN → Softmax Outputs

Spotify API → Song Lyrics → LDA Pixels → K Nearest Neighbors → Softmax Outputs

Softmax Outputs → Weighted Sum of Outputs → Final Output

Source | Extracted Data | Features | Models

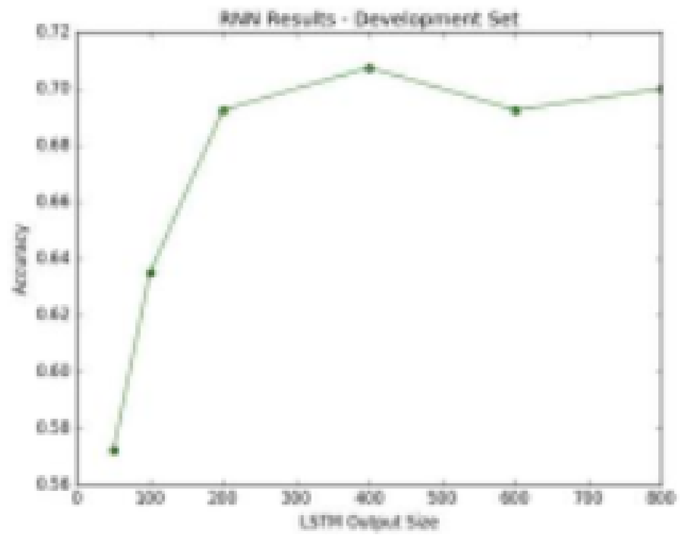
Models

Model 1: Naive Bayes on Lyric Data

Implemented using a multivariate Bernoulli event model with Laplace smoothing.

Model 2: Recurrent Neural Network on Sound Data

The MFCC features were passed into a recurrent neural network consisting of two layers. The first layer is a LSTM layer with 400 output units (see graph to the right). This output was passed to a softmax layer for classification.



Model 3: k-Nearest Neighbors on Image Data

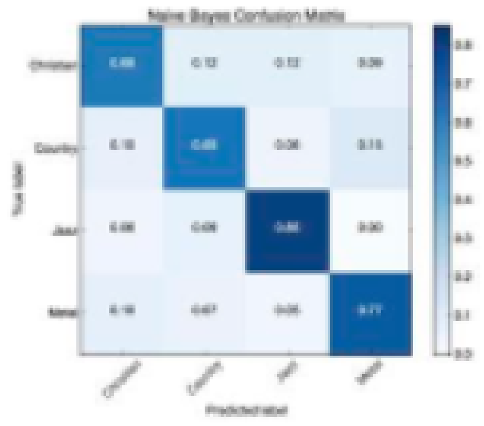
Linear Discriminant Analysis was used to maximize distance between songs of different genres while decreasing the feature space to 20 covariates. kNN was used on the resulting feature set with K = 5.

Final Model: Weighted Sum of Model Output

The outputs of the final layers (probabilities for kNN and Naive Bayes, and the predicted class for kNN, weighted by test set accuracy) are passed to a final model. The final model linearly combines these outputs and creates a final prediction using this combination.

Results

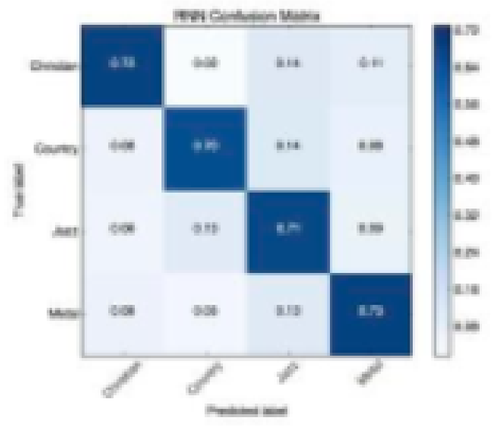
Model 1: Naive Bayes on Lyric Data



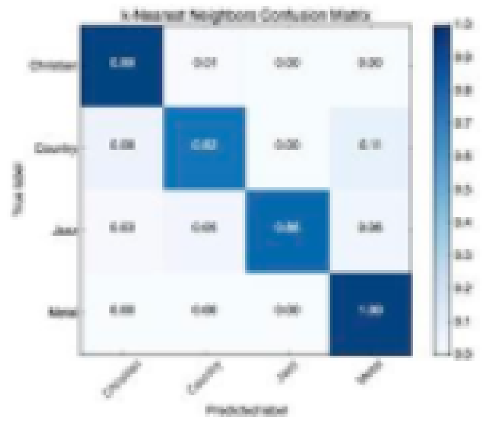
Test accuracy: 0.7425

Model 2: Recurrent Neural Network on Sound Data

Test accuracy: 0.7175



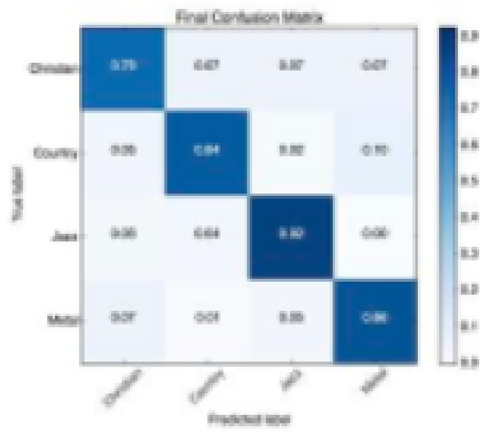
Model 3: k-Nearest Neighbors on Image Data



Test accuracy: 0.9050

Final Model: Weighted Sum of Model Output

Test accuracy: 0.8525



Discussion

The final accuracy of the combined model was relatively high. We were surprised by the performance of the kNN model on album artwork. We didn't expect such a simple algorithm to perform better than more sophisticated models such as the RNN. This speaks to how powerful album artwork can be for genre classification, a feature set that is not often used for this purpose.

Averaging the output of the three models did not achieve a better accuracy than the image data alone. This has to do with high correlation of the outputs between our three models. However, averaging the outputs of the RNN and the Naive Bayes models did improve the accuracy to 0.7625, higher than either of the individual models alone.

Moving forward, we could perform more rigorous cleaning on the lyrics and use more sophisticated NLP models on the lyrics, extend to more genres to train and test on, and potentially use a larger dataset. We could also spend more time optimizing the RNN and look into better ways of combining our models. Finally, we could also look for other features to add, such as information about the song like beats per minute or song title.