Capstone Project 2: Final Report

Springboard Data Science Career Track
Ishmail Grady
July 2019

Problem Statement

Gaining insights from music files has become more important with the growth of technologies that interact with audio. These technologies include phones and smart devices that use features like Siri, Alexa, or Google Home as well as applications that interact with music directly such as Spotify, Tidal, and Shazam. Music classification is a key problem in machine learning that can enhance technology that interacts with music to gain insight or perform various tasks (i.e. music generation, playlist construction, etc.). Music genre classification can be used to provide content-based music recommendations or as the first step in determining what song is being played for technology with the ability to detect audio. This can be especially useful for streaming platforms that allow users to upload music files to their platforms that may have incomplete metadata, such as Soundcloud. The goal of this project is to create a machine learning model that is designed to take audio features of an MP3 file and predict the musical genre of the track. For applications where predicting the exact track is the end goal, this model can be incorporated into a larger model/algorithm to increase its performance and efficiency.

Music information retrieval is a growing field of study and with the advancement of robust machine learning techniques that we can use models to classify music. Among these techniques is the use of deep learning which can provide superior predictive power and flexibility compared to other methods. This can be a large advantage when trying to solve more complex problems like music information retrieval. However, deep learning models often take longer to develop and train, require large datasets, and are prone to overfitting. This project will use deep learning methods and traditional machine learning methods to build models for predicting the genre of a music track. In doing so, this project will explore the advantages and disadvantages of both approaches to determine the best solution.

Approach

To provide a basis for determining if a deep learning model can provide a significant advantage over traditional machine learning models, three models were developed: a deep learning classifier, a baseline ensemble classifier, and a “stacked” ensemble classifier. These three models will be trained on three different datasets:

- The deep learning model will be trained on numerical representations of Mel-spectrograms extracted from raw MP3 audio files.
- The baseline ensemble model will be trained on precomputed audio features from the MP3 files.

- The stacked ensemble classifier will be trained on precomputed audio features from the MP3 files the predicted probabilities from the deep learning classifier which will be added as features.

The performance of the deep learning classifier will be compared to the baseline ensemble classifier to determine if the deep learning model provides significant advantages over traditional machine learning models. The stacked ensemble classifier will determine if using the predictions from the deep learning model can significantly enhance the performance of the baseline ensemble classifier. The performance of each model will be evaluated by their macro-F1 score which is defined as the simple arithmetic mean of the F1 scores for each genre.

Data Source

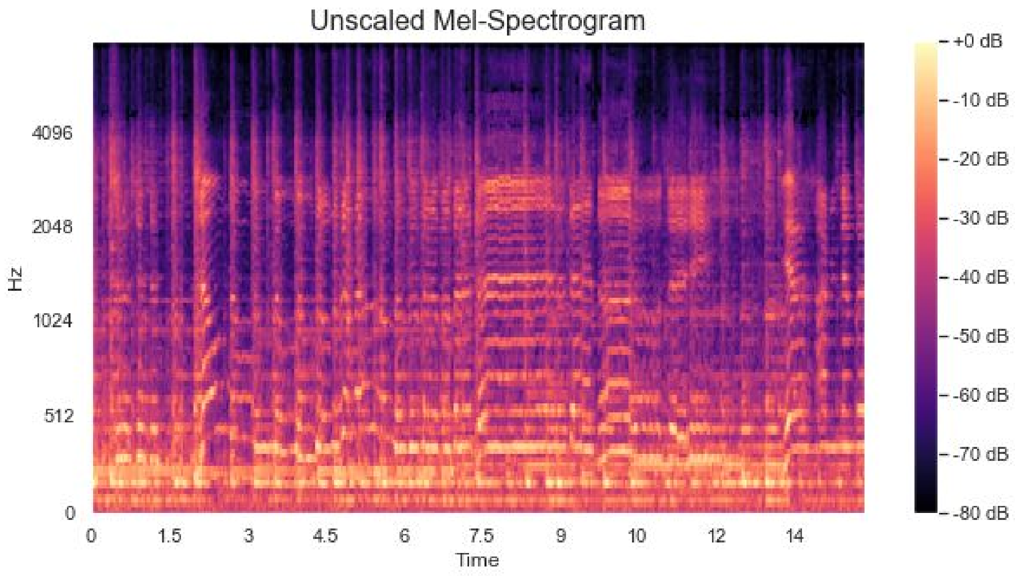
The main data source for this project will come from a compilation of high-quality audio files that are free and legal to use through the Free Music Archive (FMA) which is designed for musical analysis. This dataset contains 8,000 30 second track samples from 8 different genres. The FMA has untrimmed tracks for over 100,000 songs, however, due to memory and computing power constraints, this project will only use a small subset. The dataset contains metadata on each track and pre-computed audio features that are publicly available for download.

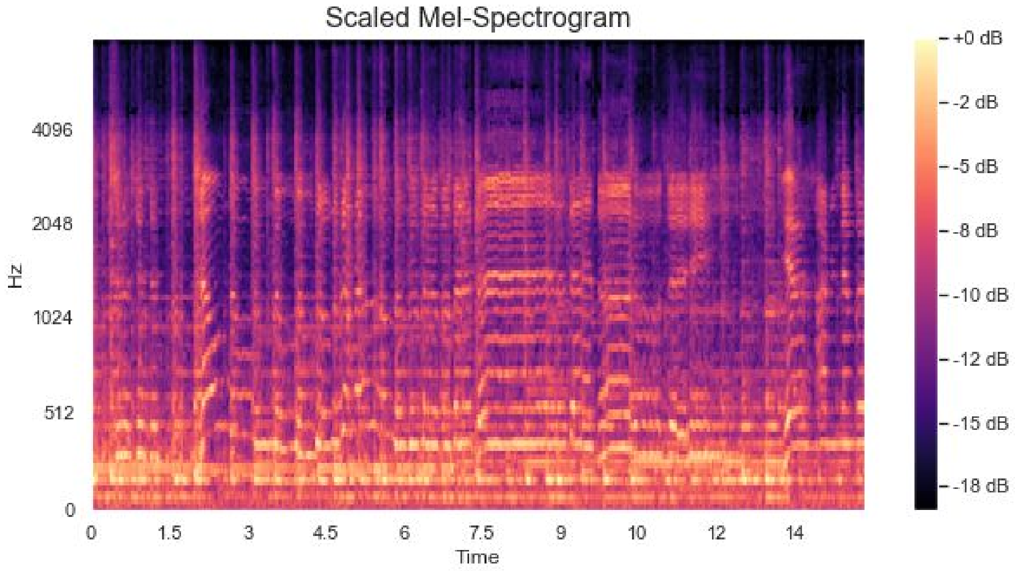
Data Wrangling & Cleaning

This project uses two datasets for the same audio files: Mel-spectrogram arrays of the 30-sec tracks and pre-computed audio features. The Mel-spectrogram arrays were used to train the deep learning models. The precomputed audio features were used to train the traditional machine learning models. The raw MP3 audio files were converted to Mel-spectrograms and were stored as numerical arrays. The conversion procedure takes a significant amount of time for most computer machines and the resulting data is considerably large. 6 of 8,000 tracks were not able to be converted and were discarded from the dataset. The precomputed audio features dataset was constructed from 3 main tables: Tracks (track metadata and response labels), features (pre-computed audio features), and genre (genre labels and hierarchy). The goal of this project will be predicting the top-level genre of a track based on its features derived from the audio alone, therefore metadata such as artist name, number of listens, and year the track was made were removed from the dataset.

Exploratory Data Analysis

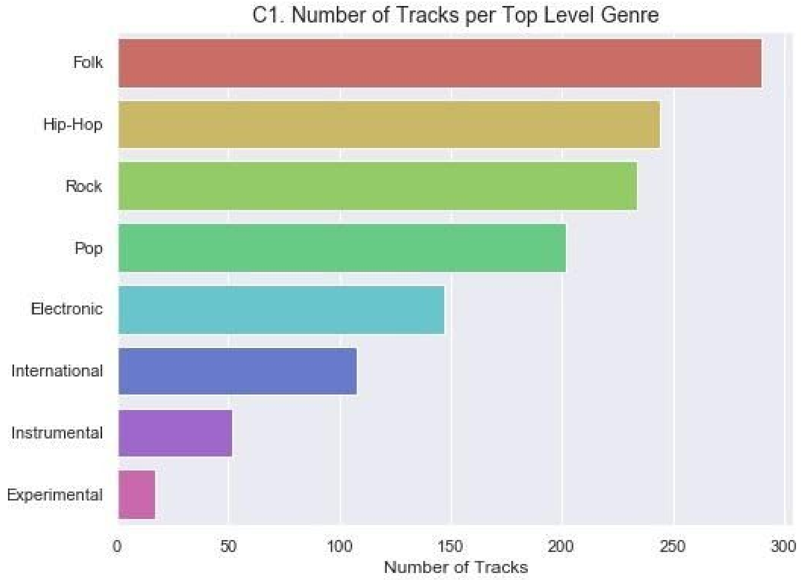
Understanding the proportion of tracks per genre in this dataset will be very important in developing a model. If the dataset is very unbalanced for certain genre’s it will not perform well of a wide range of music. The subset of the FMA data used for this project is balanced across all 8 genres. The dataset was also pre-ported into training validation and test sets in a way that ensured equal distribution of genres and artists. The deep learning models will include convolutional architectures that will be trained exclusively using the Mel-spectrograms of the audio files. An example of these spectrograms is shown below.


Unscaled Mel-Spectrogram

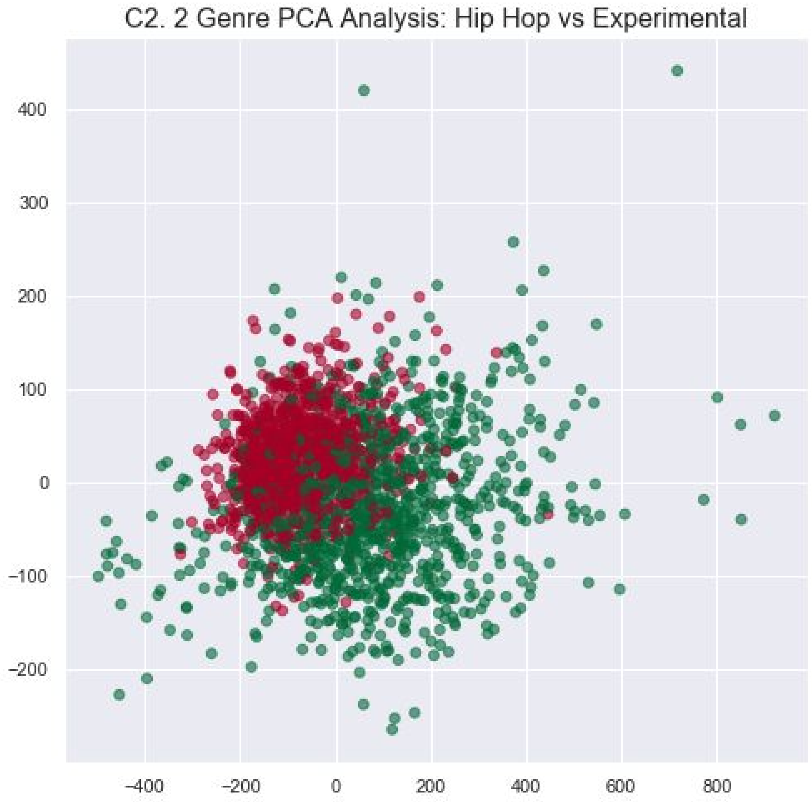

Scaled Mel-Spectrogram

The scaled spectrogram shows the same track, but the features have been logarithmically scaled elementwise. As you can see the spectrogram is identical however the decibel range has shrunk. All of the raw spectrograms were scaled to account for extreme variations in audio features across the dataset.

Discriminatory features will be very important given the amount of crossover between genres that can occur in the music industry. It is not uncommon for an artist to make music that can be classified as Hip-Hop, Pop, and R&B; artist Drake is a perfect example. Artists often collaborate across genres which can also mean the blending of sounds and the creation of new sub-genres. Furthermore, genres such as Rock can contain elements of many other subgenres as seen in Figure C1.



The deep learning models will use the numerical representations of the spectrograms to learn features and classify tracks, other model classes will use precomputed audio features. Among these audio features are Mel-frequency cepstral coefficients (MFCC). These features are derived from the Mel-frequency cepstral representation of audio which takes into account human perception for sensitivity at scaled frequencies which makes them useful for speech recognition and MIR. Principal component analysis (with 2 components) was performed on the MFCC features to explore how well they are able to be used to discriminate between genres. The chart below shows the results for PCA performed on Hip Hop tracks and Experimental tracks. The MFCC features appear to be able to discriminate between these two genres fairly well and the boundary between the two classes appears to be linearly separable.



Deep Learning Models & Results

A convolutional neural network (CNN) and a convolutional recurrent neural network (CRNN) were considered. CNN's are useful for dynamically learning spatial features of an image, and in this application, features of a Mel-spectrogram. Convolutional layers are able to apply filters to images that create a map of the spatial features present in these images. CRNN's combine the advantages of CNNS with the advantages of recurrent neural networks which have the ability to improve predictions by modeling a temporal sequence of extracted features, thus future inputs use information from previous inputs.

CNN

The CNN model is composed of a sequence of 5, 2D convolutional modules. Each module begins convolutional layer. After each convolutional layer, there is a batch normalization and RELU activation layer to improve the speed and stability of the model. Convolutional layers learn the precise locations of features of an image, thus the feature map for one image will look different from the feature map for an image with the same feature but in a slightly different position. To address this issue, after each pair of batch normalization and RELU activation layers, there is a 2D max-pooling layer. These pooling layers make the model more invariant to small spatial translations of features. To help prevent overfitting, a dropout layer which, randomly drops any element of the data with a given probability, was added after each pooling layer.

After the features are extracted by the convolutional modules, the images are flattened, and the data is fed into dense layers to help with classification. The final layer outputs the probabilities of each class.

Parallel CRNN

The Parallel CRNN structure builds on the structure of the CNN. This model contains the same 5 sequential convolutional modules as the CNN described above. However, the CRNN contains a 'sub-model' that is being trained on the same input. This 'sub-model' is a bidirectional recurrent neural network that is useful because it allows the output layer of a recurrent neural network to draw information from past and future states. The spectrograms are passed through the convolutional modules and the recurrent neural networks simultaneously and their outputs are concatenated together. This concatenated output is then fed into the final output layer for classification.

In order to determine the best performing model, the data was split into training, validation, and test sets. The training data were used to learn the parameters of each model. The validation data was used to assess the performance of the model during training parameter tuning. The test set was used to compare the performance of each model to determine the best deep learning model for this project.

The performance of each model is shown below. The Parallel CRNN had the best performance on the test data with a macro-F1 score of .36. This is a significant improvement on the CNN model which achieved a test set macro-F1 of .25 Thus, the Parallel CRNN was used as the final deep learning model and the predictions form this model will be used as features for stacked ensemble classifier.

Deep Learning Classifier Performance Results

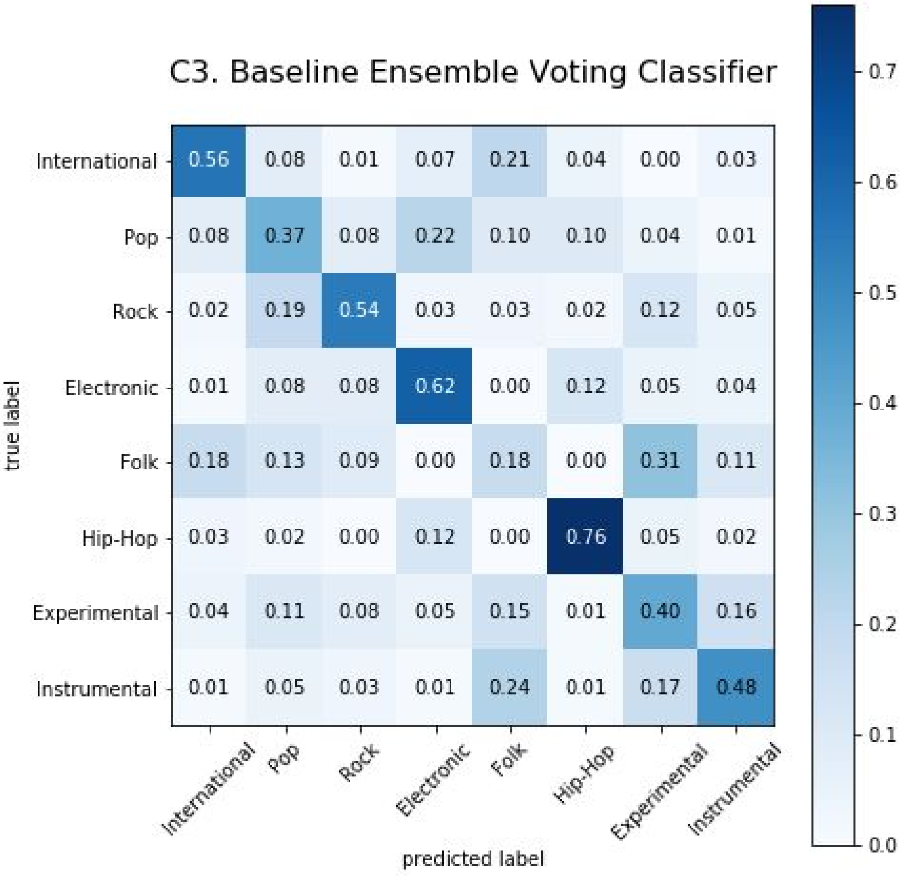
Dataset Split	CNN	Parallel CRNN
Training Macro-F1	.43	.53
Validation Macro-F1	.34	.42
Testing Macro-F1	.25	.36

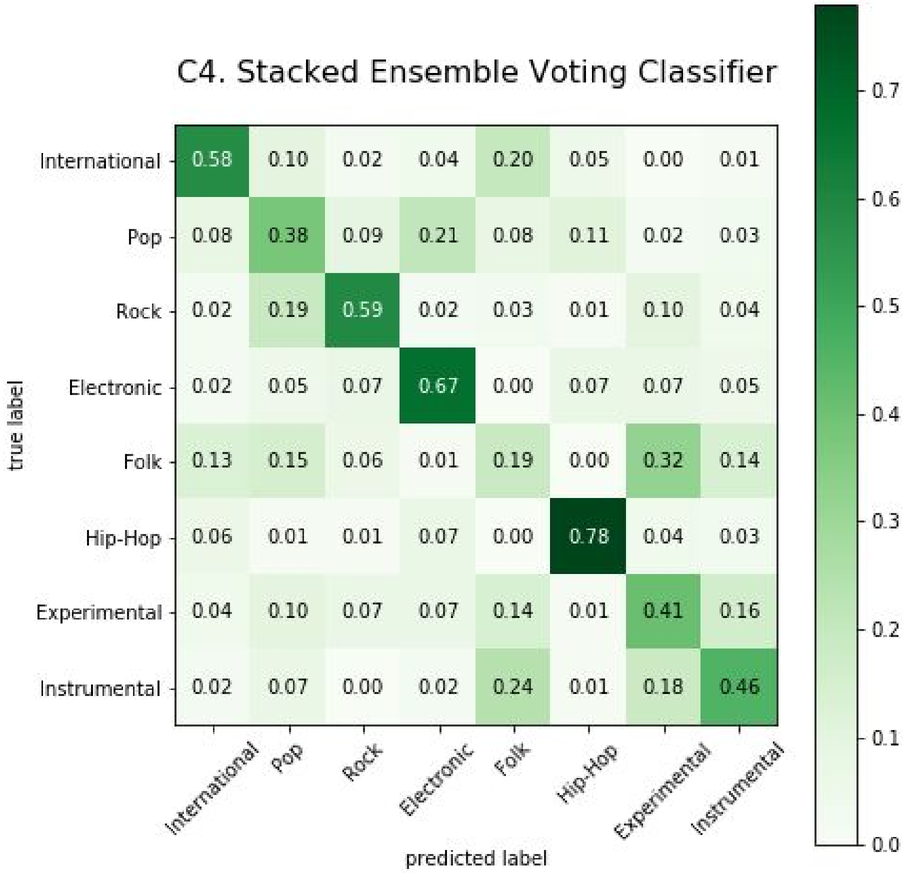
Ensemble Models & Results

Two ensemble voting models were considered: a baseline ensemble voting classifier and a stacked ensemble voting classifier. The baseline ensemble voting classifier was trained on the pre-computed audio features dataset and is composed of a logistic regression classifier, a gradient boosting random forest classifier, and a support-vector machine classifier. The voting method of this model is 'soft' meaning that it predicts class labels based on the argmax of the sums of the predicted probabilities. The stacked ensemble voting classifier has the same composition, however, the predicted class probabilities for each track was added as features to the pre-computed audio features dataset. The data for both models were split into training (90%) and test (10%) sets. The test set macro-F1 score for both models will be compared to determine if using the CRNN predictions as features can significantly improve prediction performance.

The results for both models are shown below. The stacked ensemble voting classifier was has a test set macro-F1 of .51 which is .49 higher than the baseline ensemble voting classifier. Therefore, the use of CRNN predictions as features was able to significantly improve prediction performance.

Figures C3 and C4 shows a normalized confusion matrix for both models. The true positive rate for each class increased for all classes, except Pop and Instrumental, in the predictions of the stacked ensemble classifier. Both models were able to classify Hip Hop tracks relatively well, as they had the best true positive rate and F1 score across all classes. Both models struggled with classifying Folk music the most and had relatively low true positive rates and F1 scores. In fact, when the true genre of a track was Folk music, both models were more likely to predict that the track was Experimental.


C3. Baseline Ensemble Voting Classifier



F1 Scores by Genre

Models	Baseline Ensemble Classifier	Stacked Ensemble Classifier
International	.577	.594
Pop	.373	.354
Rock	.584	.615
Electronic	.588	.632
Folk	.199	.206
Hip Hop	.725	.769
Experimental	.370	.391
Instrumental	.492	.476

The performance of the individual models that compose the ensemble classifiers is shown below. The SVC outperformed the gradient boosting and logistic regression models for both classifiers. The SVC actually outperformed the macro-F1 score of the full baseline ensemble classifier. The full stacked classifier only improved upon the macro-F1 score of the SVC model by 0.05.

F1 Scores by Sub-Model

Models	Baseline Ensemble Classifier	Stacked Ensemble Classifier
Logistic Regression	.44	.46
Gradient Boosting	.47	.49
SVC	.49	.50

Conclusion

The highest performing model for this project with a goal of creating a predictive model for music genre classification was the Stacked Ensemble Classifier with n macro-F1 score of .51. This model would be most useful for predicting Hip Hop, Electronic, and Rock genres which are 3 of the most popular genres in modern society.

By comparing the performance of the final deep learning classifier, the Parallel CRNN, to the baseline ensemble model, it can be concluded that traditional ML methods outperform the deep learning methods for this project. It can also be concluded that the use of the predictions from the deep learning classifier can improve the performance of traditional ML methods as shown by the performance of the stacked ensemble classifier.

In practice, it is recommended to use the SVC model of the baseline ensemble classifier when speed and computing power are important considerations. This is due to the fact that the ensemble voting method did not improve upon the performance of the SVC model, thus the SVC model is sufficient for the task. In addition, even though the stacked ensemble voting classifier is more accurate, this gain of accuracy may not be worth the extra computing resources that it takes to process data and train the model of the deep learning classifier so that its predictions can be used as features depending on the exact application of the model

It is important to note that these conclusions can only be made under the specific constraints and parameters of this project. It is expected that the accuracy of the deep learning models will increase when trained on more data.