Music Genre Classification

Vaibhav Saraf, Saurabh Parkar, Sanjan Vijayakumar

December 9, 2019

1 Introduction

Due to the enormous growth in the streaming music industry, both the users and the streaming services struggle a lot to manage the music libraries manually. Music genre classification is the backbone of various features provided by music streaming applications such as music recommendations, soundtrack recognition, etc. It also streamlines the user experience by allowing users to sort through the music based on their music preference and find similar music to the genres they like. A robust and accurate music classifier using various machine learning models can be used to automate the process of tagging unknown music tracks and improve the overall user experience.

In this project, we plan to build models to classify audio clips into 8 music genres using 3 different deep learning approaches and other regression models. We will be using features extracted from MFCC and Mel spectrogram as they retain both frequency as well as time information. Using these approaches will help us understand how the frequency and time domain features affect the music classification process and also help us compare deep learning models with classical machine learning algorithms.

The source code of this project can be found here:
https://drive.google.com/open?id=117dlxXh-aKZzYK4N9TpTtfGjeder-QB3

2 Dataset

The dataset consists of 8000 MP3 music files with a total of 8 genres, where each MP3 file is of 30 seconds each. The dataset is balanced i.e each genre has 1000 music files respectively. The metadata consists of the following csv files consisting of information for all tracks:

- tracks.csv: per track metadata such as ID, title, artist, genres, tags and play counts, for all tracks.

- genres.csv: all genre IDs with their name and parent (used to infer the genre hierarchy and top-level genres)
- features.csv: common features extracted with LibROSA

3 Literature Review

Previous attempts at music classification were based on some generic features extracted from music like the tempo, pitch or bpm of the music track modeled using classical machine learning techniques like Support Vector Machines and KNN. These methods did not yield very good accuracy. Hidden Markov Models (HMMs), which have been extensively used for speech recognition tasks, have also been explored for music genre classification.
In recent times, due to the popularization of deep neural networks, spectrograms have become a more desired option. Spectrograms have been extensively used for various audio information retrieval applications and have given excellent results for tasks like speech recognition. A spectrogram helps to capture information over 3 dimensions – Time, Frequency and Amplitude. Training on a combination of visual and acoustic features can help to get better results.
Two of the most popular types of spectrograms used for audio information retrieval are Mel Spectrogram and the MFCC. We decided to use the Mel spectrogram for our project as it was found to work better for music genre classification and was also visually more informative.

4 Preprocessing

The input audio files are converted into respective spectrogram images using the Librosa package.

The Fourier transform accepts an audio signal in time domain and decomposes it into individual frequencies. A signal in the frequency domain requires much less computational space for storage.

After applying Fast Fourier Transform on our music file we get the following spectrogram-

Nothing much can be made out of this plot as most of the frequencies and amplitudes that humans can hear are concentrated in a very small frequency range. To solve this problem, we transform the y-axis (frequency) to log scale, the "color" axis (amplitude) to Decibels, which is basically the log scale of amplitudes and Hertz scale to Mel scale.

Mel scale is obtained as a result of non-linear transform on the frequency scale. The Hertz scale in itself is non-uniform meaning the frequencies at 500 and 1000 Hz can be distinguished easily but the difference between the frequencies 7500

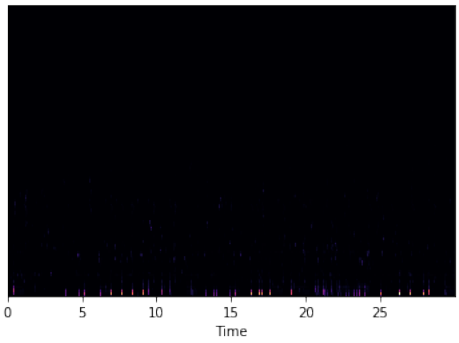

Figure 1: Image obtained after Fourier Transform

and 8000 Hz is barely noticeable.

The transformation converts the entire frequency spectrum and separates it into n_mels=128 evenly spaced frequencies.

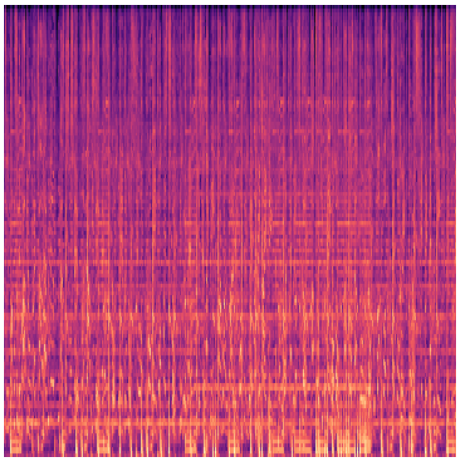

Figure 2: Log Transformed Mel-Spectrogram

The above spectrogram was further converted into RGB colormap to make the colors more distinguishable. The following is the final Mel spectrogram-

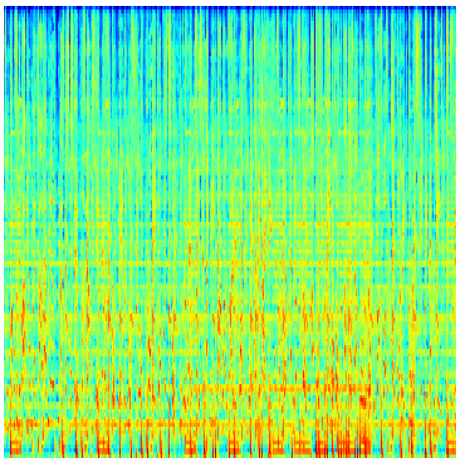

Figure 3: Log Transformed Mel-Spectrogram

The X-axis in the above plot represents the time, Y-axis represents the frequencies and the color represents the amplitude or loudness of the music. The basic idea behind our approach follows that each music genre generates a visually distinct spectrogram and these spectrograms can be treated as images to train over a Convolution Neural Network to make classifications.

5 Models

Our project involves two fold layer model implementation:

- The deep learning-based models which only require the spectrogram images as input
- Traditional machine learning classifiers which make predictions on the features csv files.

For the Basis of comparison, we have a naive random classifier which classifies with the accuracy of $1/8 = 0.125$ accuracy.
We trained 5 classifiers on the training data, and performed testing on the test set. The data set was divided into 75% training data and 25% testing data set. Also, 3 different neural network were also applied on the data set. We evaluated the model performance based on three metrics: Accuracy, Precision and Recall.

5.1 Classification Models

The Classical Models approach makes predictions based on the csv features like MFCC frequency, Tonnetz, Zero crossing rate. Each track is divided into 12 equal parts and aggregates like mean, median, mode, etc of all the features are

calculated for each of those 12 parts.

We have implemented five classical models (LDA, Logistic Regression, SVM, XGBoost and LightGBM). LightGBM and XGBoost are both gradient boosting frameworks that use decision tree-based learning algorithms. The LightGBM model uses XGBoost as a baseline and outperforms it in training speed and the dataset sizes it can handle.

Amongst the model we have employed, LGBM(Light Gradient Boosting Model) gave us the highes accuracy of 64%. Figure 4 below represents the Confusion Matrix for the LGBM model.

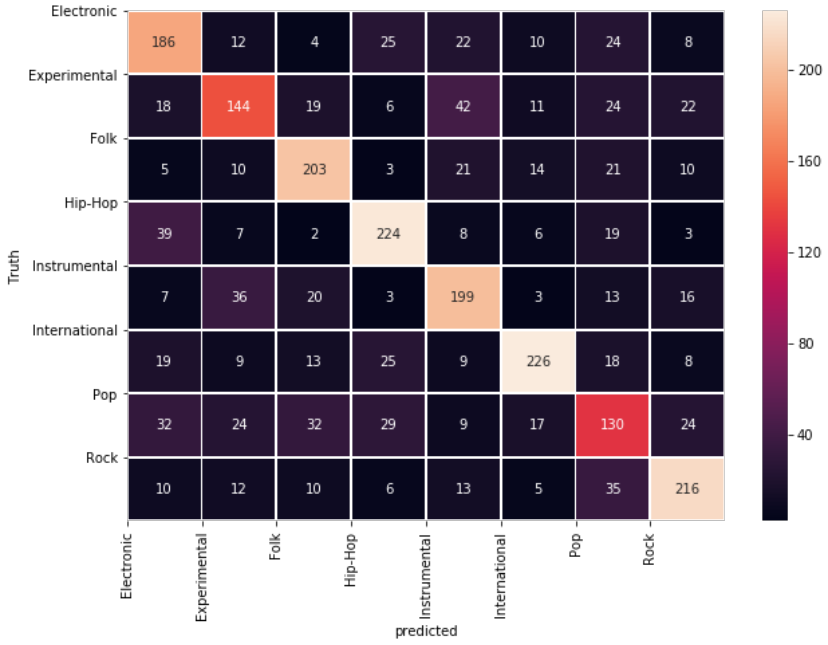

Figure 4: Confusion Matrix for LGBM Model

Plotting the confusion matrix for the LightGBM model, we see that the model predicts all genres with reasonable accuracy. By plotting the matrix for all 8 genres, we can also draw several insights like:

- International music has the most number of correctly predicted tracks whereas Pop music has the least number of correctly predicted tracks
- Most genres are confused with Electronics tracks
- Instrumental music and Experimental music tracks are often confused with each other

5.2 Neural Network Approach

For the Neural Network Implementation, we resized the mel-spectrogram image to the size (128,128,3), where 3 represents 3 color channels(RGB). We used three different neural network approaches:

- CNN(Convolution Neural Network)
- RNN(Recurrent Neural Network)
- C-RNN(Convolutional-Recurrent Neural Network)

5.2.1 Convolution Neural Network

The CNN approach helps to make predictions based on the visual features extracted from the spectrograms.

For this, all the scales and labels were removed from the spectrogram and what we were left with was only the image. The image was then resized to 128 x 128 pixels and retaining the 3 channels (RGB) to prevent any loss of information. Thus the final input shape of the image was 128 x 128 x 3.
The model architecture is given as follows:

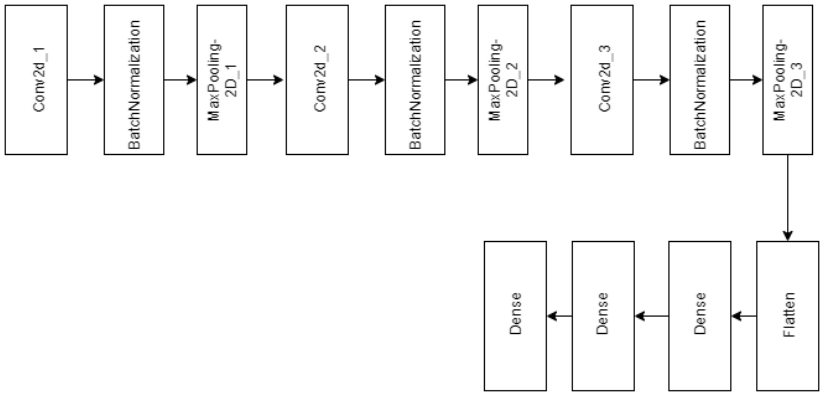

Figure 5: CNN Architecture

The network consists of a 3 layered Convolutional Neural network followed by 3 dense layers with the output layer having a softmax activation with 8 units for the 8 different classes. A padding was added to each of the CNNs to prevent the output of each layer from shrinking. ADAM was used as an optimizer to minimize the losses with a learning rate of 0.001 and a decay rate of 0.001. Since the data was huge we used mini batches of size 128 to make the training more efficient. The model resulted in an accuracy of 52.87%.

5.2.2 Recurrent Neural Network

The RNN approach makes predictions by summarizing the extracted audio features along the temporal component. To generate features, the song was divided into 12 equal parts and mean aggregates of frequency domain features like MFCC, $chroma_stft$ and spectral contrast extracted using the Librosa library were used to build a sequence.

LSTMs (Long Short Term Memory) were used for this architecture which are a type of Recurrent Neural Networks (RNN) that help to prevent the vanishing gradient problem. LSTMs have gate mechanisms that help to regulate information and a cell state which helps to maintain information about previous sequences over long durations of time and hence becomes convenient to model the above features over temporal component. The LSTM Model Architecture is given as follows: Our model consists of 3 LSTM layers (128 units) followed by 2

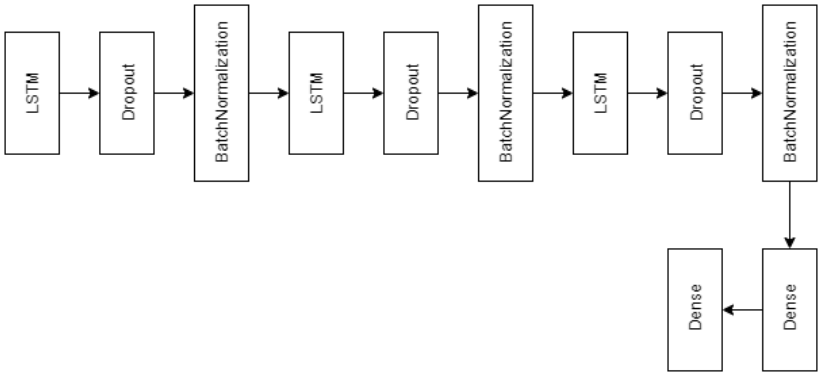

Figure 6: LSTM Architecture

dense layers, and an output layer with softmax activation function having 8 units for each of the 8 different classes. Dropout and Batch normalization layers are included after each LSTM layer to regularize the network. The model resulted in an accuracy of 50.69% which was slightly lower than the CNN approach.

5.2.3 Convolution Recurrent Neural Network

Our final model consists of a 4 layered Convolutional Neural Network followed by 2 layers of Recurrent Neural Networks. The output of the CNN model is a long sequence of values in which every time-step depends on both, the intermediate values and the overall structure of the complete track.

The RNNs were combined with the output from the CNN layer to introduce a time sequential information to the features extracted from the spectrograms. Permute function was used to transform the output from the CNNs into a format appropriate for the RNN. We have used GRUs for this approach which are

similar to LSTMs but have slightly different and simpler internal architecture. A series of batch normalization, max pooling and dropout layers were also added after each CNN layers to regularize the data and preserve only the significant information from the data. The model architecture is given as follows:

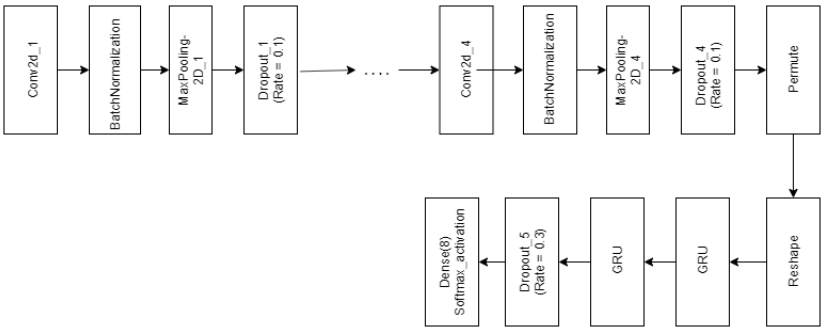

Figure 7: C-RNN Architecture

The C-RNN model performed better than CNN and LSTM models with the accuracy of 55%. Figure 8 below represents the Confusion Matrix for the C-RNN model.

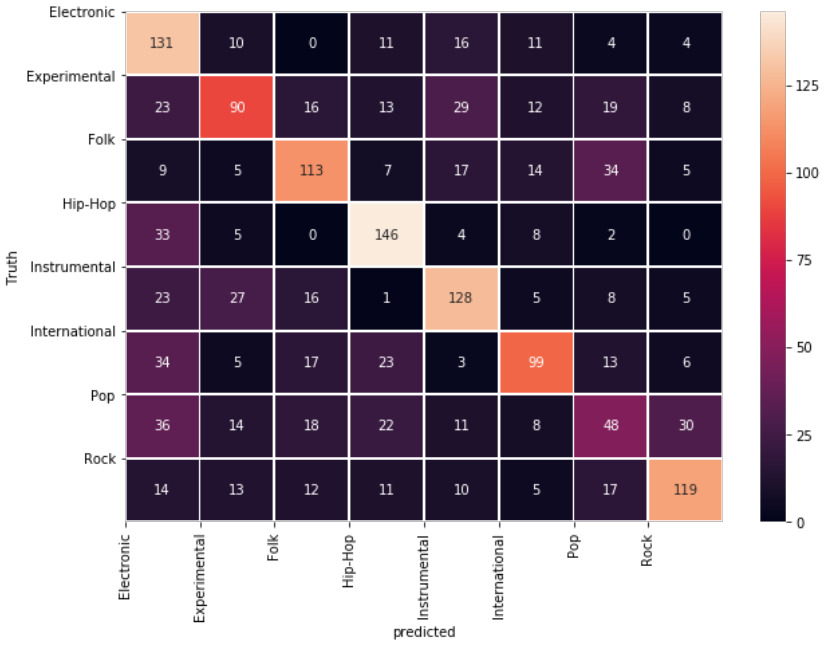

Figure 8: Confusion Matrix for C-RNN

The confusion matrix for the C-RNN model is plotted, we notice all models predict correct genres with considerable accuracy. By plotting the matrix for all 8 genres, we can also draw several insights like:

- The model performs badly in predicting pop music and exhibits an average performance with experimental and international music
- Similar to other models, several genres are being confused with electronic music.

6 Model Performance and Metrics

Sr.No.	Models	Accuracy	Macro-Avg Precision	Macro-Avg Recall
1	LDA	55.75	0.56	0.56
2	XGBoost	57.41	0.57	0.57
3	Logistic Regression(L1 Regularised)	57.45	0.56	0.57
4	SVM	62	0.62	0.62
5	LGBM	64	0.63	0.63

Table 1: Perfomance Metrics of Classification Models

The Classical models have similar accuracies with the LGBM model performing the best with an accuracy of 64%.

Sr.No.	Models	Accuracy	Macro-Avg Precision	Macro-Avg Recall
1	CNN	52.87	0.52	0.53
2	LSTM	50.69	0.51	0.48
3	CRNN	55.48	0.55	0.55

Table 2: Perfomance Metrics of Neural Network Implementations

The Neural Network models have similar accuracies with the C-RNN model performing the best with an accuracy of 55.48%.

The Classification report from the LGBM model is as shown below:

	precision	recall	f1-score	support
Electronic	0.52	0.62	0.56	291
Experimental	0.54	0.45	0.49	286
Folk	0.63	0.65	0.64	287
Hip-Hop	0.64	0.64	0.64	308
Instrumental	0.56	0.61	0.59	297
International	0.69	0.59	0.64	327
Pop	0.41	0.39	0.40	297
Rock	0.64	0.68	0.66	307
accuracy			0.58	2400
macro avg	0.58	0.58	0.58	2400
weighted avg	0.58	0.58	0.58	2400

Figure 9: Classification Report for LGBM

The Classification report from the C-RNN model is given below:

	precision	recall	f1-score	support
0	0.56	0.52	0.54	187
1	0.47	0.49	0.48	210
2	0.62	0.59	0.60	204
3	0.65	0.70	0.67	198
4	0.65	0.49	0.56	213
5	0.52	0.71	0.60	200
6	0.34	0.28	0.30	187
7	0.60	0.62	0.61	201
accuracy			0.55	1600
macro avg	0.55	0.55	0.55	1600
weighted avg	0.55	0.55	0.55	1600

Figure 10: Classification Report for C-RNN

The ROC plot obtained from the neural network C-RNN is shown as below:

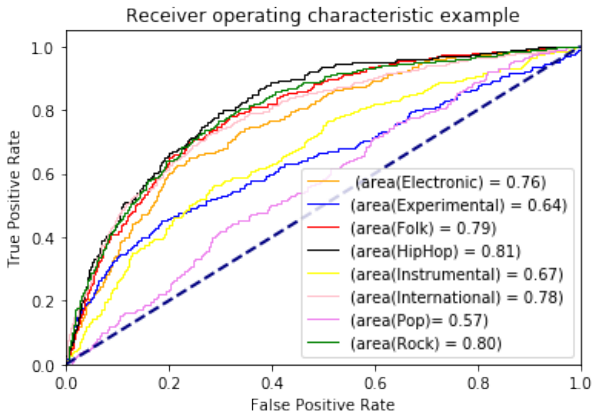

Figure 11: ROC Curve for C-RNN

The ROC plot obtained from the LGBM classifier model is shown as below:

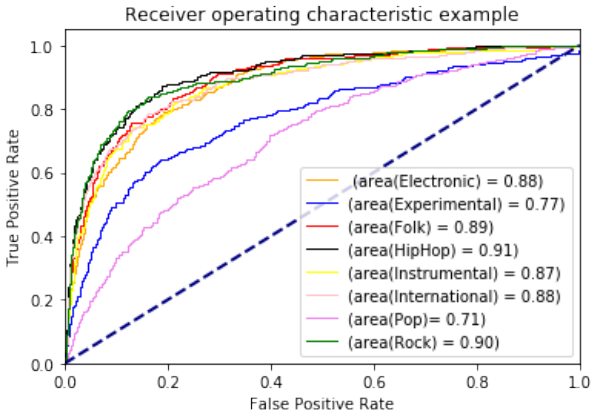

Figure 12: ROC Curve for LGBM Classifier

7 Future Scope

The accuracy obtained on the present dataset can be improved by fine-tuning the model hyper-parameters and including metadata features. Working on MFCC values instead of Mel-Spectrogram values might yield better accuracy as MFCC's are widely used in image segmentation.
Extending the models to make accurate predictions on the FMA Full dataset (containing 100,000+ untrimmed audio clips with 161 unbalanced genres) would make this project a powerful music genre classification tool. Real-time music genre classification can also be implemented.

8 Conclusion

The objective of our project was to accurately classify music audio samples to their correct genre. Accuracy can be increased for genre classification by using more accurate samples of each genre. At the same time, if the number of genres increases, correctly classifying genres for a 30s audio clip becomes a difficult task. Building an optimized model that can classify a large number of genres without losing accuracy should be the end-goal of this project. We achieved over 55% accuracy in several models with 64% being the highest. Being a multi-class classification problem with a random guessing accuracy of 12.5%, our model performs considerably well in predicting various genres. Amongst the Classical models which make predictions based on the features csv, LightGBM and SVM models had accuracies greater than 60%. All other implemented models too had

accuracies in a similar range. Amongst the Neural Network models, considering both CNN and RNN together in a model seemed to be the best method with accuracies of 55%. This accuracy can be improved upon if additional research is put into each hyper-parameter and each layer.

9 References

References

[1] Chi Zang, Yue Zang, Cheng Cheng *SongNet: Real-time Music Classification.* http://cs229.stanford.edu/proj2018/report/53.pdf

[2] Muralidhar Talupur, Suman Nath, Hong Yan *Classification of Music Genre.* http://www.cs.cmu.edu/ yh/files/GCfA.pdf

[3] Robert Adragna, Yuan Hong (Bill) Son *Music Genre Classification.* http://www.eecg.utoronto.ca/ jayar/mie324/musicgenre.pdf

[4] Hareesh Bahuleyan *Music Genre Classification using Machine Learning Techniques.* http://bit.ly/2P6Ejix

[5] http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[6] https://librosa.github.io/librosa/