Indian Music Genre Classification

Siva Aditya Gooty
431000650

Sriram Chakravarthy
629009746

December 12, 2021

Acronyms-MFCC - Mel-Frequency Cepstral Coefficients, STFT - Short Time Fourier Transform, SVM - Support Vector Machine, kNN - k Nearest Neighbors, CV - Cross Validation, PCA - Principal Component Analysis, FFT - Fast Fourier Transform

1 MAIN GOAL

Indian music encompasses multiple genres in numerous varieties and forms. In this project, four widely popular genres are considered, namely-Bollywood Pop, Carnatic, Ghazal and Sufi. SVM, kNN, Random Forest and Neural Networks are employed for the genre classification and the performance of each model is evaluated.

2 MAJOR ACTIVITIES AND CHALLENGES

2.1 Audio file sampling

The dataset consists of four distinct genres each containing 100 files of .mp3 format. Each .mp3 audio file is of 45 seconds duration. Prior to the sampling, each .mp3 file is converted to .wav format using FFmpeg and pydub. The.wav files are sampled at the rate of 22050 Hz using librosa.

2.2 Feature Engineering

Audio files when sampled using librosa.load() carry little to no information for the purpose of classification. However, certain acoustic and temporal characteristics of each audio sample are utilized in frequency and time domain for the model development. In this project, features are extracted and

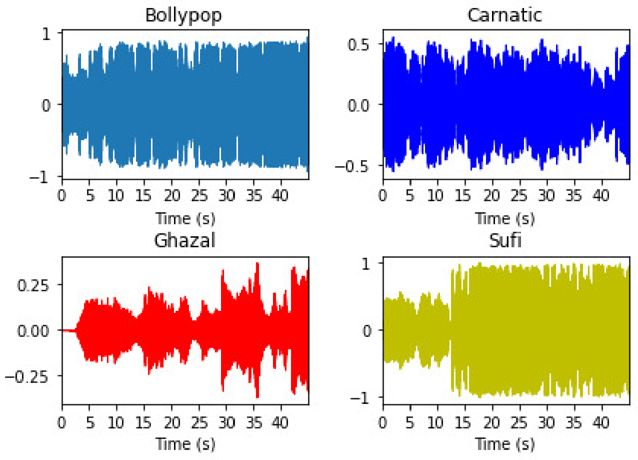

Figure 1: Time domain samples of each genre.

analysed with the help of librosa.features().

Owing to the fact of multiple feature availability, it was a major challenge to choose the best set of features for the task of classification. The dataset for this project is not been explored extensively in any previous works unlike the GTZAN dataset. In the next sections, based on the feature selection, two approaches that gave the best results are described.

2.3 Approach 1

In this approach, the feature set is extracted in both time and frequency domain. A total of 8 features along with it's significance are outlined below:

1. **Feature extraction**

   - **Spectral bandwidth**

     stft() of an audio file time series when converted into magnitudes are treated as input to `librosa.feature.spectral_bandwidth()` resulting in P'th order spectral-bandwidth. For every audio file, the mean of the 1st order spectral bandwidth magnitude is treated as a single feature. Following plots illustrate the spectral bandwidth of a single audio file from each genre:

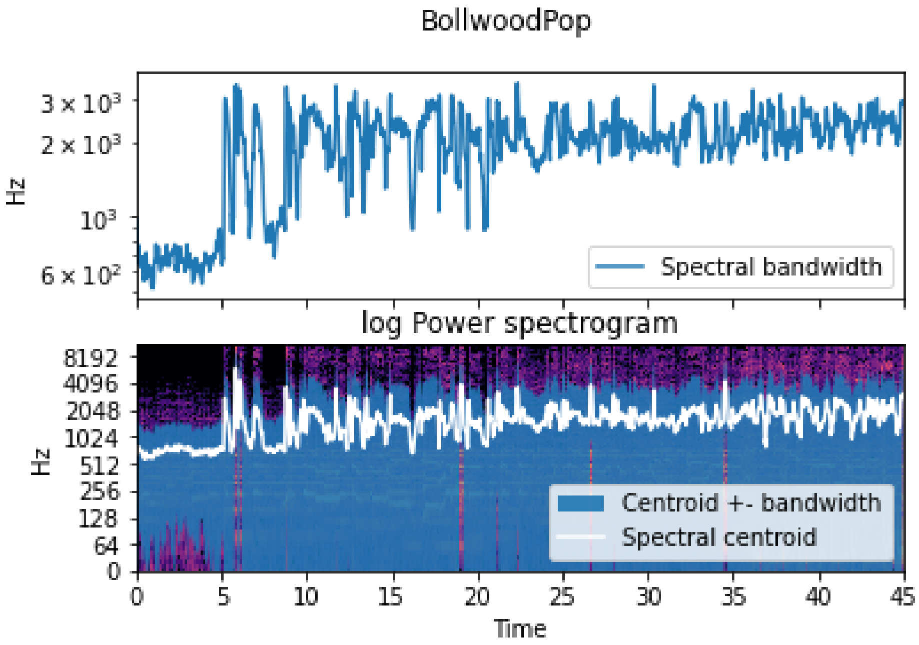

Figure 2: Spectral bandwidth of BollywoodPop song

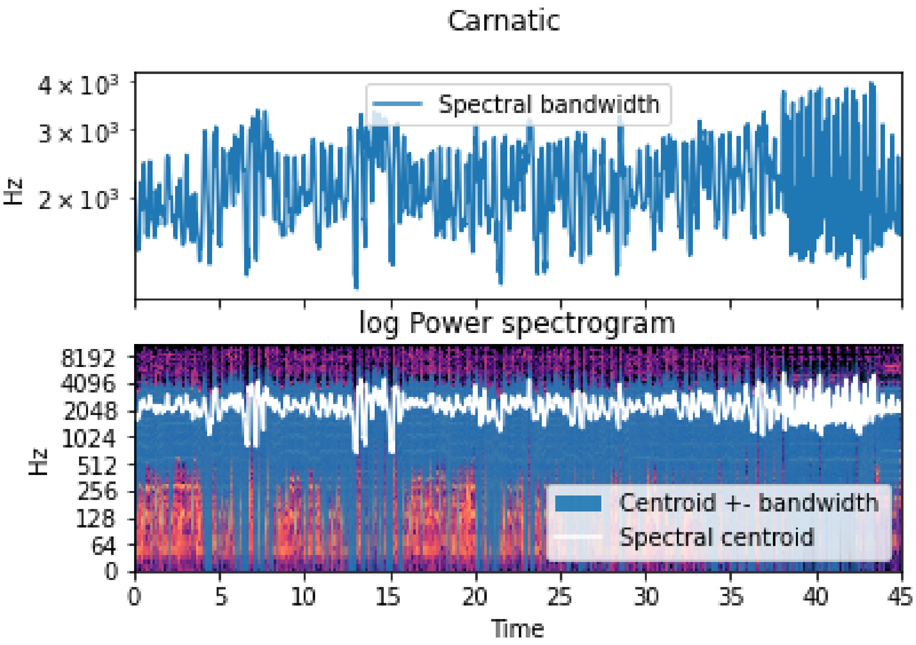

Figure 3: Spectral bandwidth of Carnatic song

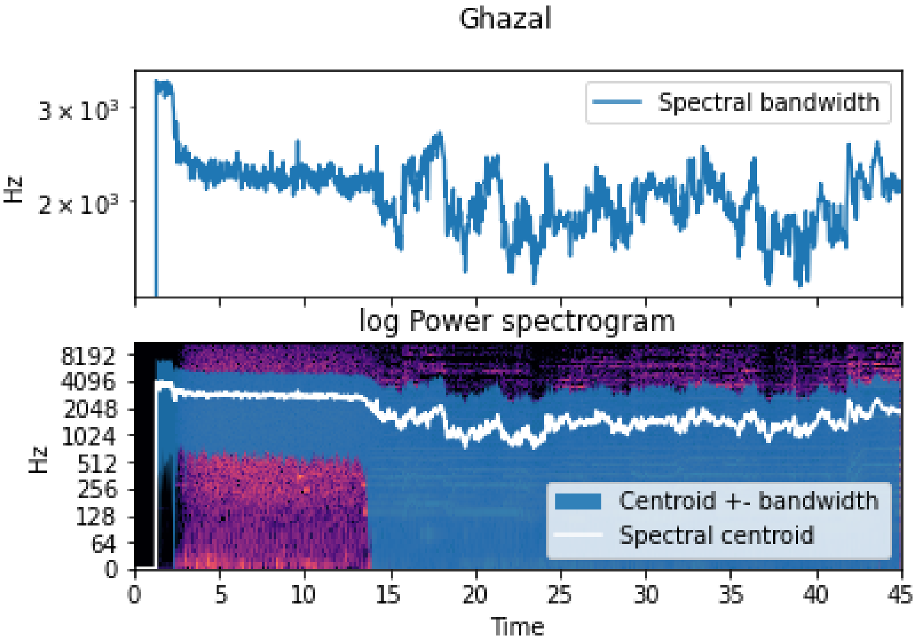

Figure 4: Spectral bandwidth of Ghazal song

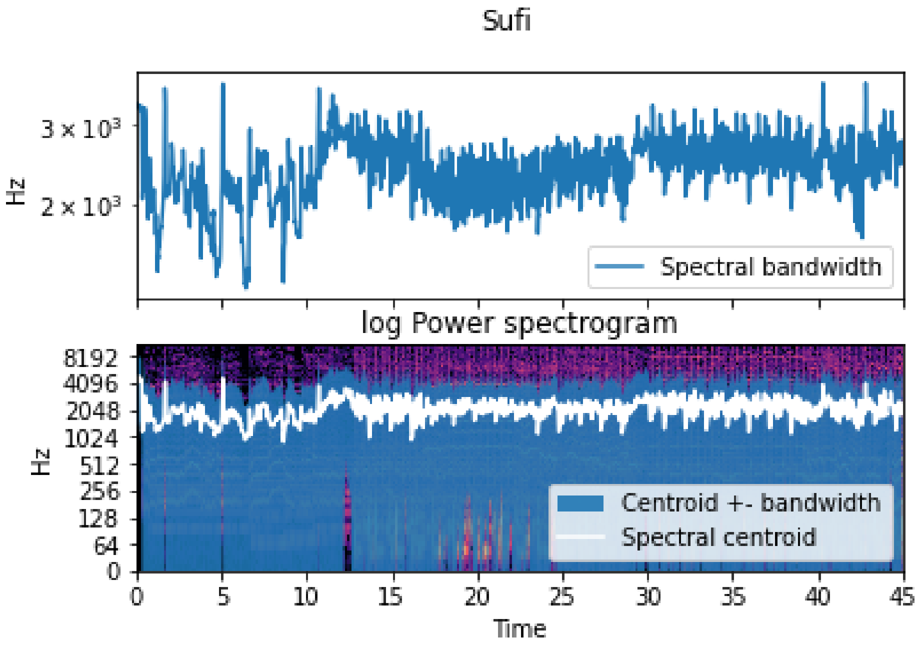

Figure 5: Spectral bandwidth of Sufi song

- **mel-spectrogram co-efficients**

  The mel-spectrogram co-efficients are generated by librosa.feature.melspectrogram(). The mean across all the audio frames of a file of first two co-efficients are treated as the two separate features.

- **MFCC co-efficients**

  'Most-frequently considered coefficients', MFCC is that one feature being used in any machine learning experiment involving audio files. For an each audio file, multiple MFCC co-coefficient's are generated by librosa.feature.mfcc(). The mean value of first two co-efficients across all audio time frames are considered as the two separate features.

- **Zero crossing rate**

  Zero-crossing rate is a measure of number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero. A large number of zero crossings implies that there is no dominant low-frequency oscillation. The mean of the zero crossing rates across all the frames (`librosa.feature.zero_crossing_rate()`) of each audio file is treated as a single feature.

- **N point FFT**

  The time-series samples of audio file is transformed to frequency domain using N-point FFT with the help of scipy.fft(). Maximum frequency component of each audio file is treated as single feature.

- **RMS**

  The root mean square value (librosa.feature.rms()) of each audio file time series is treated as a single feature.

2. Results and Findings

All the features are normalized with sklearn.preprocessing.StandardScaler() before training the model. Following models are employed with 70:30 percent train and test split feature set and accuracy of each model is evaluated with 10 fold cross validation. Each audio file consists of 8 features which thus forms 280x8 and 120x8 training and test data matrices respectively.

- **SVM**

  The feature set is not linearly separable and the Kernel trick is employed using radial basis function. The accuracy results are as follows:

Model	Training	CV	Testing
SVM	96%	67%	73%

Table 1: SVM Classifier Results

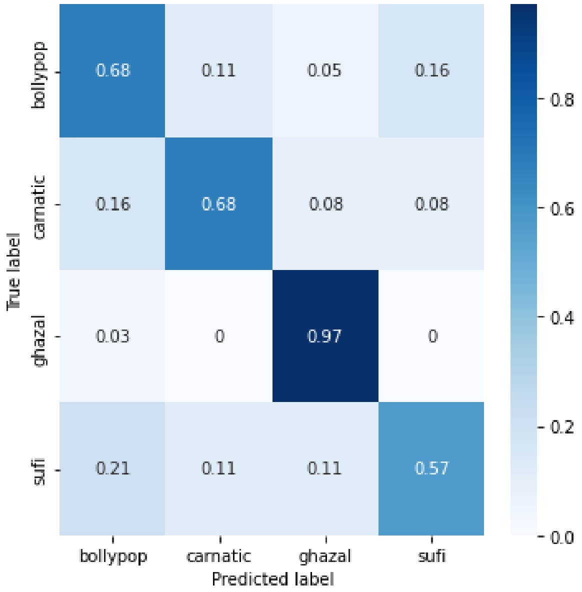

Figure 6: SVM-Confusion Matrix

- **k-Nearest Neighbour**

  kNN model is employed with neighbour count k=3. The accuracy results are as follows:

Model	Training	CV	Testing
kNN	87%	64%	72%

Table 2: kNN Classifier Results

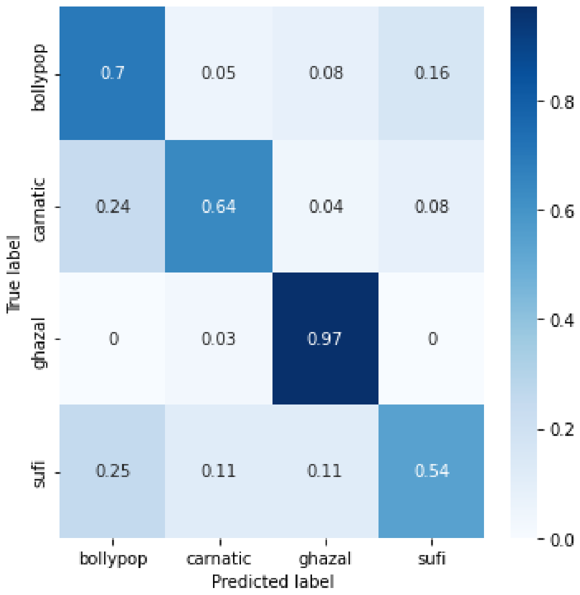

Figure 7: kNN-Confusion Matrix

- **Random Forest**

The Random Forest Classifier is employed with max depth=5 and the accuracy results are as follows:

Model	Training	CV	Testing
Random Forest	92%	67%	73%

Table 3: Random Forest Classifier Results

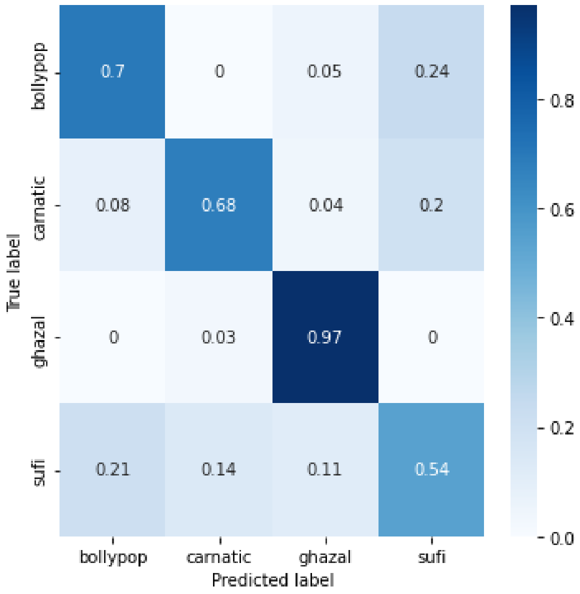

Figure 8: Random Forest-Confusion Matrix

- **Neural Network**

  A neural network with three hidden layers of size 512 with ’relu’, ’tanh’ and ’LeakyRelu’ respective activation functions is employed. The output layer with ’softmax’ activation function predicts the 4 separate classes of music genre. The entire model was trained with learning rate =0.0001, ’adam’ optimizer for 200 epochs. The accuracy results are as follows:

Model	Training	CV	Testing
Neural Network	98.5%	72%	82%

Table 4: Neural Network Results

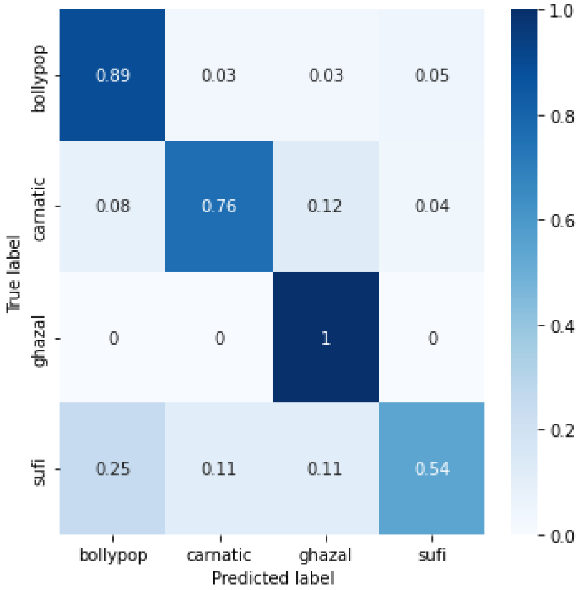

Figure 9: Neural Network-Confusion Matrix

2.4 Approach 2

1. **Feature extraction**

MFCCs (Mel-Frequency Cepstral Coefficients) are a short-time spectral decomposition of an audio signal that conveys the general frequency characteristics important to human hearing. MFCCs are commonly used in the field of speech recognition. Research shows that MFCCs are capable of capturing useful sound characteristics of music files as well. Our premise is that, MFCC’s contain enough information about the timbre of a song to perform genre classification. Generally, the first 13 coefficients(the lower dimensions) of MFCC are taken as features as they represent the envelope of spectra.

N MFCC coefficients are computed for 30s clip of each song and are stored in a (F x N) matrix, where F is the number of frames. In our case, F = 1292 and N = 13. In addition, the delta MFCCs of first and second orders are also computed, each of same dimensions (F x N). The delta MFCCs represent the derivatives of the MFCCs which describe the trajectories of the MFCC coefficients over time. In summary, we have 1292 x 39 matrix representing each song.

Now, calculating the mean, standard deviation, min and max values for each of the 39 columns gives a 4 x 39 = 156 features per song. Thus, each feature vector has a dimension of 1x156. These computations are done using librosa.feature.mfcc, librosa.features.delta. Further, the feature dimensions are reduced from 156 to 58 using PCA while retaining 90% of the variance in the data. The plots for the feature vectors of one song from each genre are shown below.

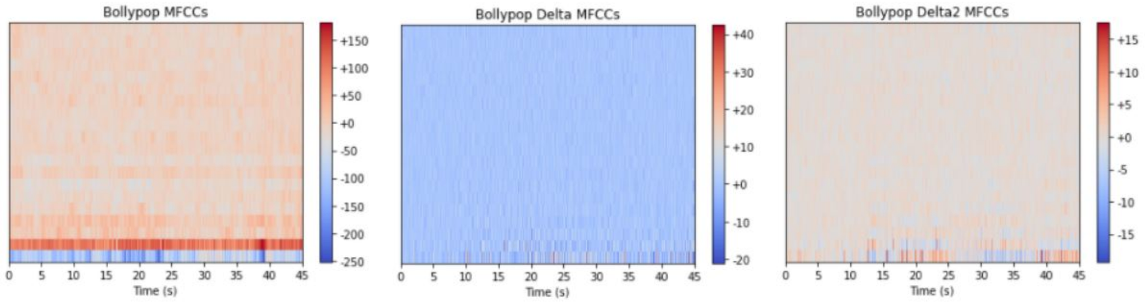

Figure 10: MFCC, Delta MFCC and Delta2 MFCC plots for an example from Bollypop genre

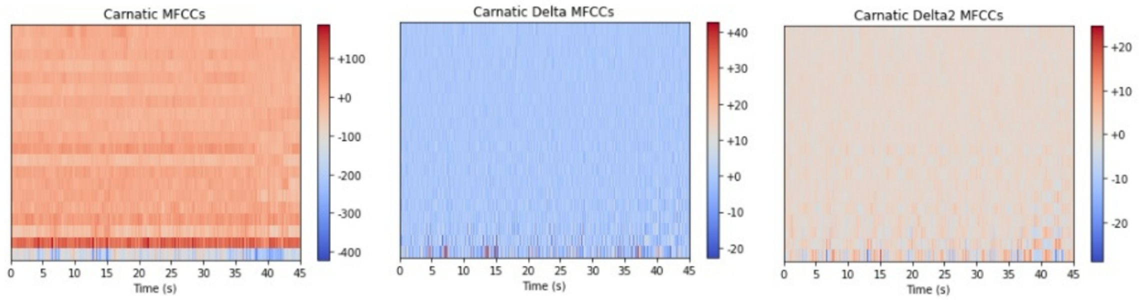

Figure 11: MFCC, Delta MFCC and Delta2 MFCC plots for an example from Carnatic genre

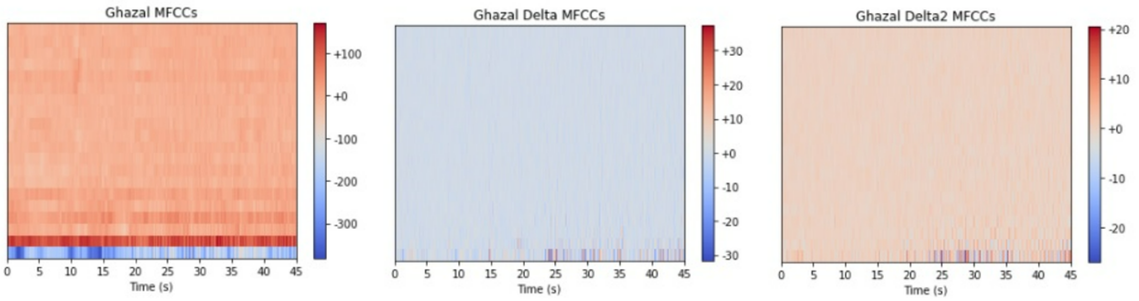

Figure 12: MFCC, Delta MFCC and Delta2 MFCC plots for an example from Ghazal genre

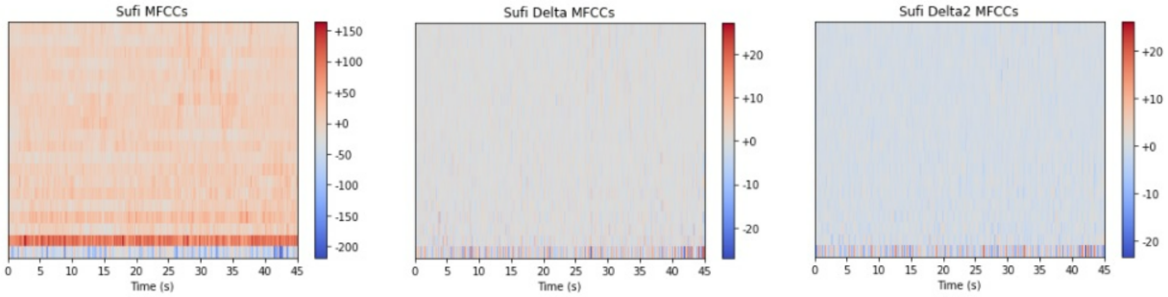

Figure 13: MFCC, Delta MFCC and Delta2 MFCC plots for an example from Sufi genre

2. **Results and Findings**

The dataset consists of 100 songs from each of the four genres (bollypop, carnatic, ghazal, sufi) i.e, 400 songs in total. The dataset is split into 70 percent training data and 30 percent testing data. In other words, the training data has 280 songs while the testing data has 120 songs. Following models are trained with 10-fold cross validation and the accuracy is evaluated using the test data.

- **SVM**

  For the SVM model, RBF kernel is used. The results of training and testing are as follows:

Model	Training	CV	Testing
SVM	97.5%	85.5%	85%

Table 5: SVM Results

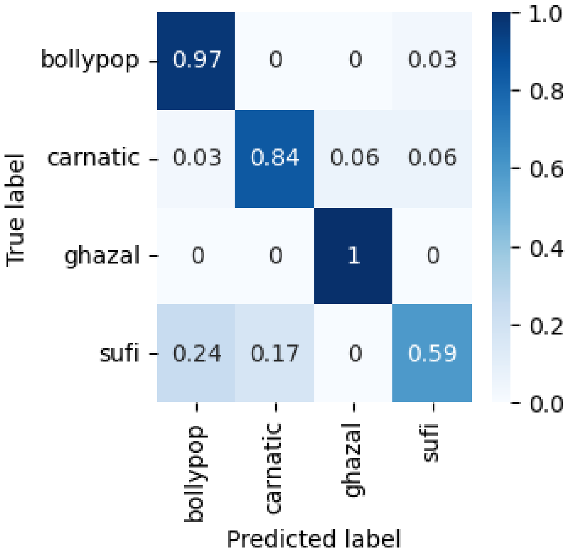

Figure 14: SVM Confusion Matrix

- **K-Nearest Neighbors**

  KNN Classifier with neighbour count k=4 is used. The results of training and testing are as follows:

Model	Training	CV	Testing
kNN	85.5%	77.85%	75%

Table 6: kNN Classifier Results

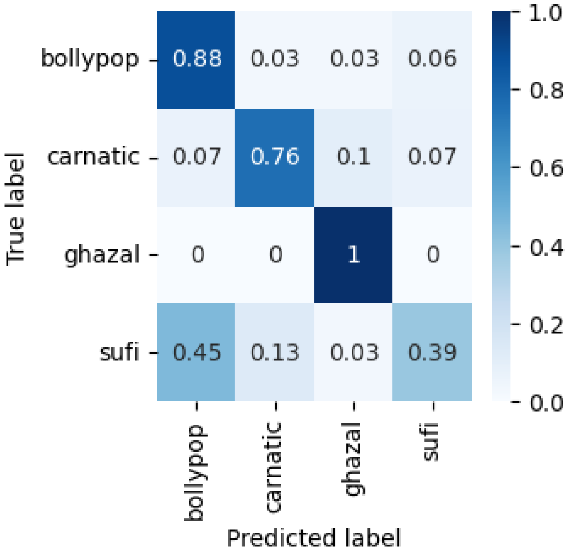

Figure 15: kNN Confusion Matrix

- **Random Forest**

  Random Forest Classifier with decision tree as the weak learner is used. The results of training and testing are as follows.

Model	Training	CV	Testing
Random Forest	98%	82%	76%

Table 7: Random Forest Classifier Results

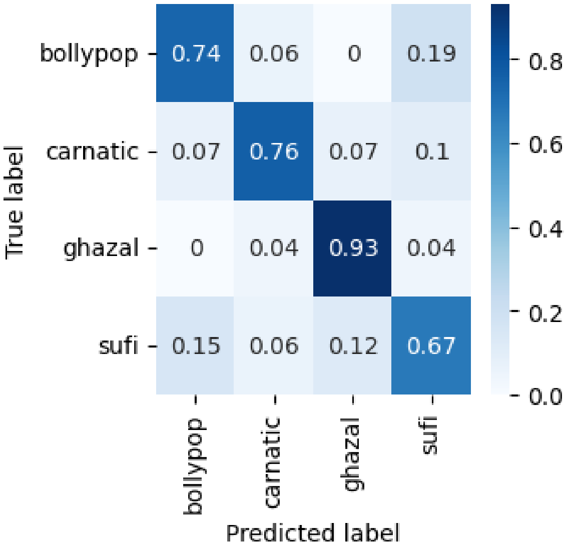

Figure 16: Random Forest Confusion Matrix

- **Neural Network**

A neural network with 3 hidden layers with 512, 256 and 64 nodes, each with ReLU activation is used. To reduce over-fitting, a dropout of 10% is used. Output layer with softmax activation with cross-entropy loss is used. Training was done for 300 epochs with learning rate=0.0001. The results of training and testing are as follows.

Model	Training	CV	Testing
Neural Network	98.21%	91.32%	90.83%

Table 8: Neural Network Results

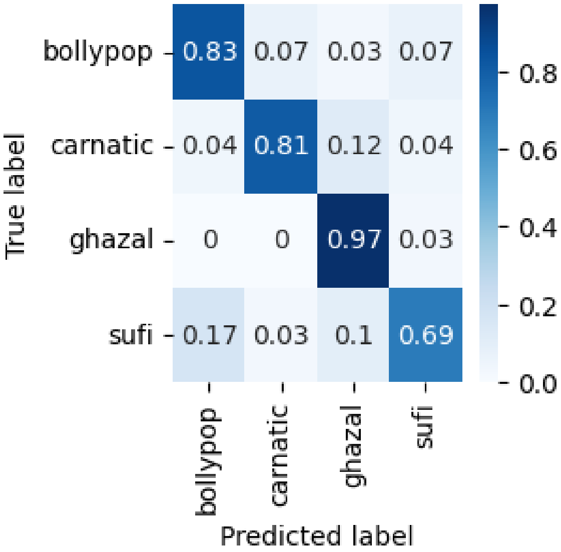

Figure 17: Neural Network Confusion Matrix

3 Summary of results

Below table summarizes the comparison of the two approaches presented in the previous section. We find that, when the models are tested with the test data, the second approach gives better performance compared to the first approach in all models.

We infer that, MFCC's carry more information about quality of music as compared to other features that helps in better classification of music genres. Of the four models that are trained for the given data set, we find that Neural Network outperforms the other three models in both the approaches.

.

Approach 1 vs Approach 2						
	Approach 1			Approach 2		
Model	Training	CV	Testing	Training	CV	Testing
SVM	96%	67%	73%	97.5%	85.5%	85%
kNN	87%	64%	72%	85.5%	77.85%	75%
Random Forest	92%	67%	73%	98%	82%	76%
Neural Network	98.5%	72%	82%	98.21%	91.32%	90.83%

Table 9: Comparison of results

4 Future Scope

The above approach is able to classify BollywoodPop, Carnatic and Ghazal genres with a good accuracy. However, the generalization error is significant in case of Sufi. Future work could be focused on mitigating this shortcoming and generalizing the models for multiple genres of Indian music. A possible direction could be feature selection specific to Indian music genres and corresponding application of Convolution and Recurrent Neural Network models for the classification problem.

References

[1] M. Haggblade et al., Music Genre Classification HaggbladeHongKao-MusicGenreClassification.pdf

[2] A Note on MFCC and delta features https://desh2608.github.io/2019-07-26-delta-feats/

[3] Robert Jang, Tutorial on music genre classification Mirlab tutorial

[4] Python Speech Features https://github.com/jameslyons/python$_{s}peech_{f}eatures$

[5] Hareesh Bahuleyan, Music Genre Classification https://arxiv.org/pdf/1804.01149.pdf

[6] Mel Frequency Cepstral Coefficient(MFCC) tutorial MFCC tutorial

[7] Indian Music Dataset https://www.kaggle.com/winchester19/indian-music-genre-dataset