A PROJECT REPORT

ON

MUSIC GENRE CLASSIFICATION

Submitted in partial fulfillment of

CS271 – TOPICS IN MACHINE LEARNING

by

Shruti Kothari - 013736463

Anand Vishwakarma - 013706329

Under the guidance of

Prof. Mark Stamp

Table of Contents

- Introduction
- Problem Description
- Dataset
- Audio Representation
- Feature Extraction
- Feature Elimination
- Methodology
- Hyperparameter Tuning
- Experiments & Results
- Conclusion
- Future Work
- References

Introduction

Music genre classification has been a widely studied area of research since the early days of the Internet. This is an interesting as well as challenging task in the field of Music Information Retrieval (MIR). Music genres are hard to systematically and consistently describe due to their inherent subjective nature. With the growth of online music databases and easy access to music content, people find it hard to manage the songs that they listen to. Being able to automatically classify and provide tags to the music present in a user's library, based on genre, can be beneficial for various audio streaming services.

Problem Description

In this project, we implemented our successive steps towards building different classifiers allowing us to identify a specific genre from an audio file. We will first describe how we collected data, and why this choice of data was pertinent. Then, we will discuss the feature extraction process by which we obtained our features and the techniques we used to perform feature selection. We will then progress onto presenting our various algorithms and machine learning techniques used for classification. The final output of our algorithms is the prediction of the genre of each input. In this project, we are using two machine learning models - Random Forest and Support Vector Machine, and two deep learning models - Multilayer Perceptron and Convolutional Neural Network.

Dataset

We initially started our work on a dataset that is a subset of the Lakh Midi Dataset (LMD). This subset contains about 45,129 files, out of which some of the files were corrupt. The dataset was not labelled and hence we had to separately download these labels and map them to the audio files. The labels we managed to find were for about 13,000 of these files. After creating this labelled dataset and doing some considerable amount of work we realized there was a large imbalance in the dataset. From 13,000 files about 9,000 files belonged to only one genre, and for some of the genres/ classes we had only about 30 files. Thus, the dataset had huge variations and none of the techniques to overcome this imbalance (oversampling, under-sampling, SMOTE) gave us good results. Our model was overfitting and was clearly biased to the larger class. It was difficult to find a good labelled dataset with a good number of audio files. Most of the datasets we looked at, had audio files but no labels. We, therefore, decided to use the GTZAN dataset which we initially thought was too small for our analysis.

The **GTZAN** dataset though small has been very widely used in the field of MIR. The reason being it is perfectly balanced and a good dataset to get started with. The dataset was taken from the Marsyas software [2]. It is an open source dataset. It consists of 1000 .au files (a file extension used for audio files), split in 10 different music genres (Disco, Metal, Pop, Hip Hop, Rock, Blues, Classical, Country, Jazz, Reggae), with 100 samples of 30 seconds for each genre. When downloaded the dataset folder contains 10 folders named by the respective genre, each of which has 100 audio files.

Audio Representation

Sound is represented in the form of an audio signal having parameters such as frequency, bandwidth, decibel etc. A typical audio signal can be expressed as a function of Amplitude and Time. This can be seen in the below figure [1]. Thus, an audio looks like a wave where the amplitude changes with respect to time as shown below.

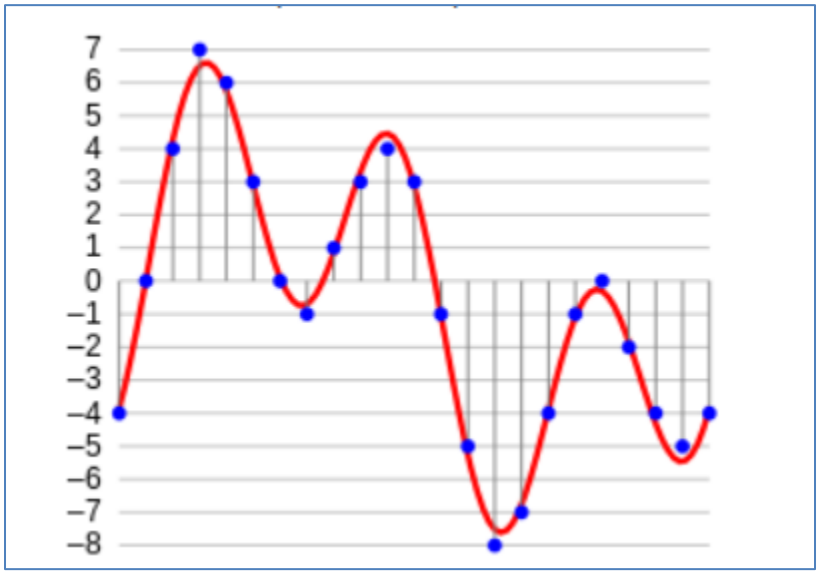

Figure 1: Audio Frequency Wave

Feature Extraction

We used LibROSA (a Python package for music and audio analysis) to convert raw music files into the above signals and extract main features from the dataset. The GTZAN dataset contains audio files in .au format, so our first step was to convert these files to .wav to extract features as .wav files can be easily read by the scipy library. There are a lot of features in an audio signal and LibROSA provides a lot of functions to extract these features, but we restricted our analysis to the ones mentioned in the below figure as a lot of research papers[3;see also 4] have shown that MFCC's (Mel-Frequency Cepstral Coefficients) and spectral features provide good results.

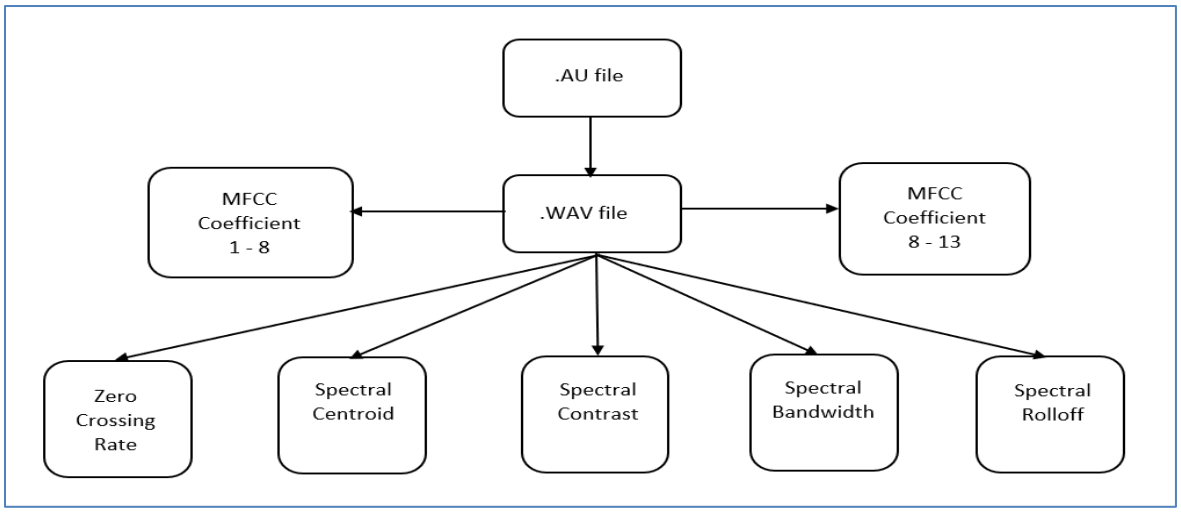

Figure 2: Feature Extraction Process

Below is a brief overview of these features -

1.) Zero Crossing Rate - It is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back [5]. For e.g., a closer view of a wave plot of an example audio signal gives the below plot which has a zero_crossing_rate of 5 as 5 times the signal has crossed 0.

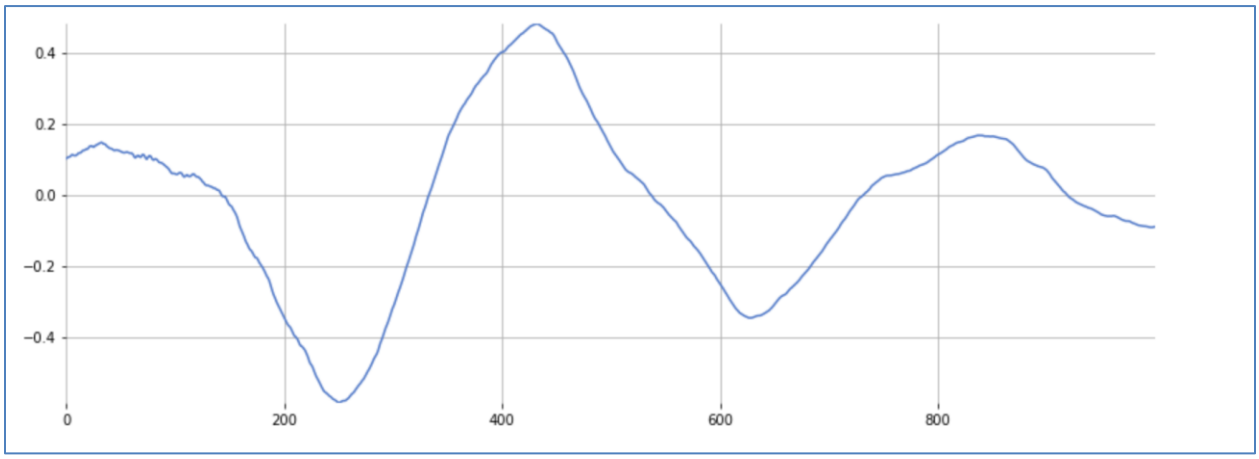

Figure 3:Zero Crossing Rate

2.) Spectral Centroid - It indicates where the" center of mass" for a sound is located and is calculated as the weighted mean of the frequencies present in the sound. If the frequencies in music are same throughout then spectral centroid would be around a center and if there are high frequencies at the end of sound, then the centroid would be towards its end.

3.) Spectral Bandwidth - It is the wavelength interval in which a radiated spectral quantity is greater than half its maximum value.

4.) Spectral Contrast - It considers the spectral peak, the spectral valley, and their difference in each frequency sub band.

5.) Spectral Rolloff - It is the frequency below which a specified percentage of the total spectral energy, e.g. 85%, lies. Spectral Rolloff is used to calculate rolloff for a given frame.

6.) MFCC- This is an important feature when working with audio signals. The MFCC's of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope. For our analysis we have considered 13 MFCC coefficients as various studies [6] pointed us to this.

As the data type of the values returned by these features in LibROSA is a multidimensional array, we took the mean and standard deviation of all these arrays and thus our final set of features was 36.

Feature Elimination

In order to increase the model accuracy, we used Principal Component Analysis (PCA) to reduce the number of features. PCA is a technique used to emphasize variation and bring out strong patterns in a dataset. The three models - SVM, MLP and RFC were run on all features starting from 1 feature to all 30 features. The accuracy plot for each model in case of 10 genres and 5 genres is shown below:

PCA Result on 10 genres

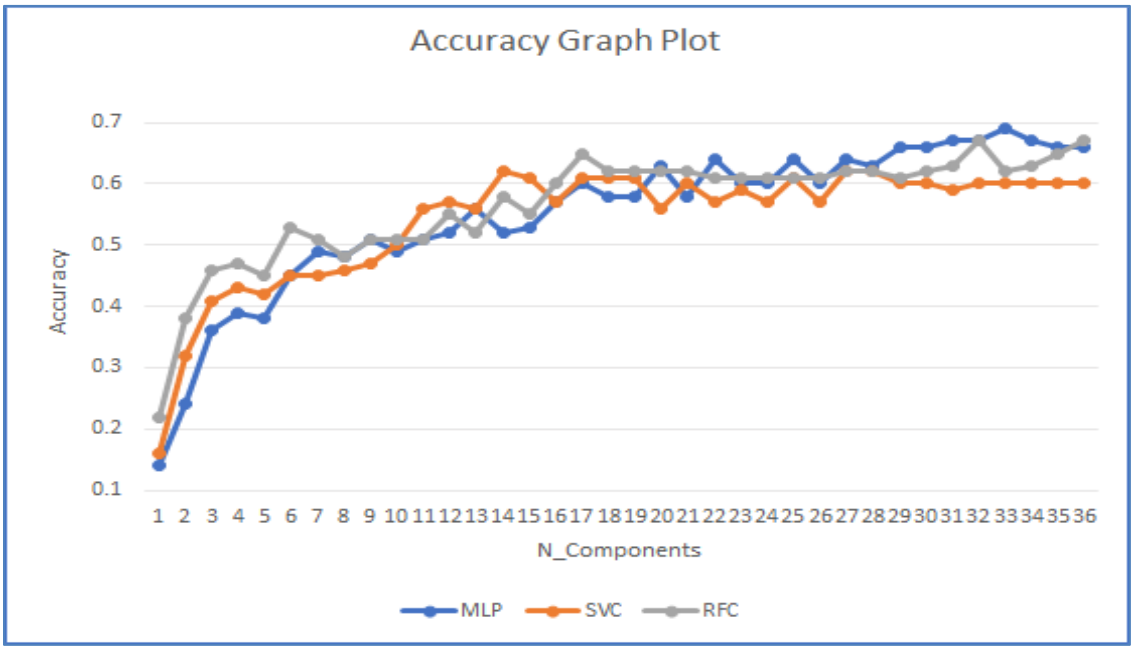

Figure 4: Accuracy Plot of MLP, SVC & RFC on 10 Genre Dataset for Reduced Features

The above graph shows the optimal number of features each model needs to obtain the maximum accuracy. MLP achieves maximum accuracy of 0.69 by using 33 features whereas SVM obtains accuracy of 0.62 with only 15 features. In case of RFC, 18 features are needed to achieve the highest accuracy of 0.65.

PCA Result on 5 genres:

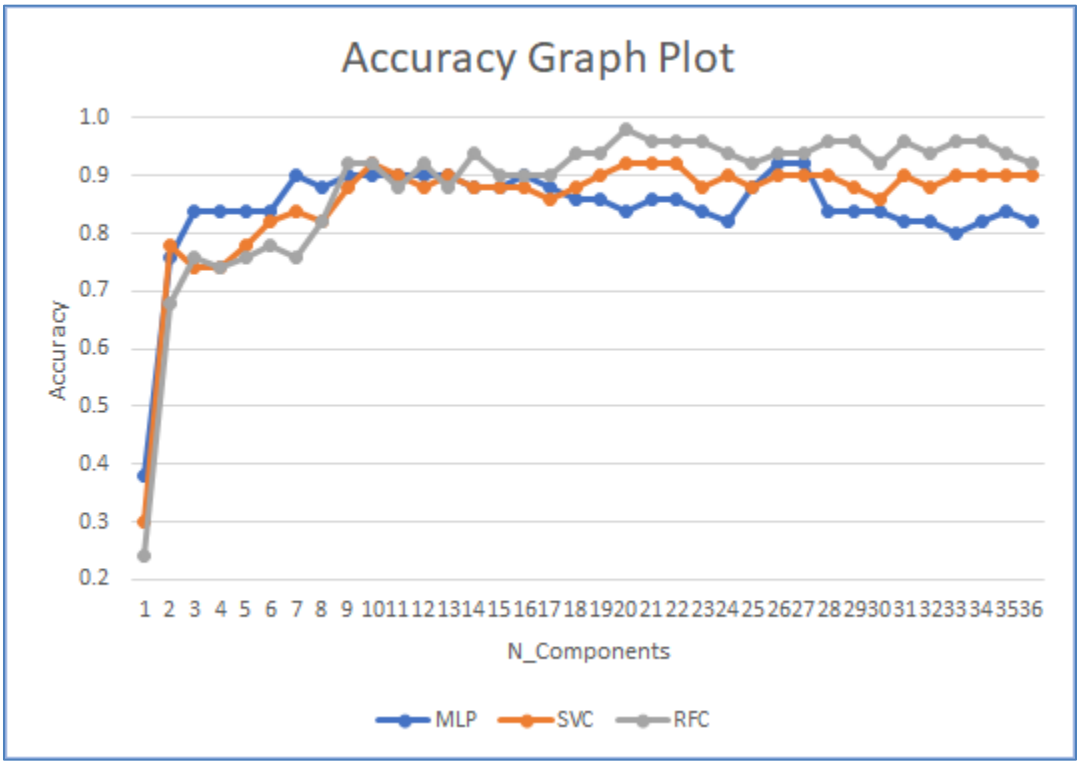

Figure 5: Accuracy Plot of MLP, SVC & RFC on 5 Genre Dataset for Reduced Features

In case of 5 genre dataset, spectacular results were found with application of PCA on each machine learning model. MLP achieved accuracy of 0.9 with only 7 features. In case of Support Vector Machine, 10 features resulted in accuracy of 0.92 whereas same number of features gave accuracy of 0.92 in case of Random Forest Classifier.

Methodology

We used the below machine learning classifiers for our project:

1) Random Forest (RF)

Random Forest is an ensemble learner that combines the prediction from a pre-specified number of decision trees. It works on the integration of two main principles: 1) each decision tree is trained with only a subset of the training samples which is known as bootstrap aggregation 2) each decision tree is required to make its prediction using only a random subset of the features. The final predicted class of the RF is determined based on the majority vote from the individual classifiers. We ran the Random Forest algorithm with a 100 decision trees.

2) Support Vector Machine (SVM)

SVM is a very useful technique used for classification. It is a classifier which performs classification methods by constructing hyper planes in a multidimensional space that separates different class labels based on statistical learning theory. We have used linear kernel and polynomial kernel SVM classifiers.

Multilayer Perceptron and Convolutional Neural Networks are the two deep learning models that we have considered.

1) Multilayer Perceptron (MLP)

It is a feedforward artificial neural network that is composed of more than one perceptron. It consists of an input layer to receive the signal, an output layer that decides or prediction about the input, and in between these two it has an arbitrary number of hidden layers which are the true computational engine of the MLP. Training involves adjusting the parameters, or the weights and biases of the model in order to minimize the error. Backpropagation is used to make these weight and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including root mean squared error (RMSE).

2) Convolutional Neural Network (CNN)

This deep learning model does not need any hand-crafted features which all the models discussed above require i.e. we do not need to provide any features to a CNN. CNNs have been known to work extremely well for the task of image classification, hence we converted each audio file into a spectrogram using LibROSA and then fed these spectrograms into our CNN. A spectrogram is a 2D representation of a signal, having time on the x-axis and frequency on the y-axis. A colormap is used to quantify the magnitude of a given frequency within a given time window. From the below figures we can see that different genres have noticeable differences in their Mel-spectrogram which gives us confidence in using a CNN to do the classification.

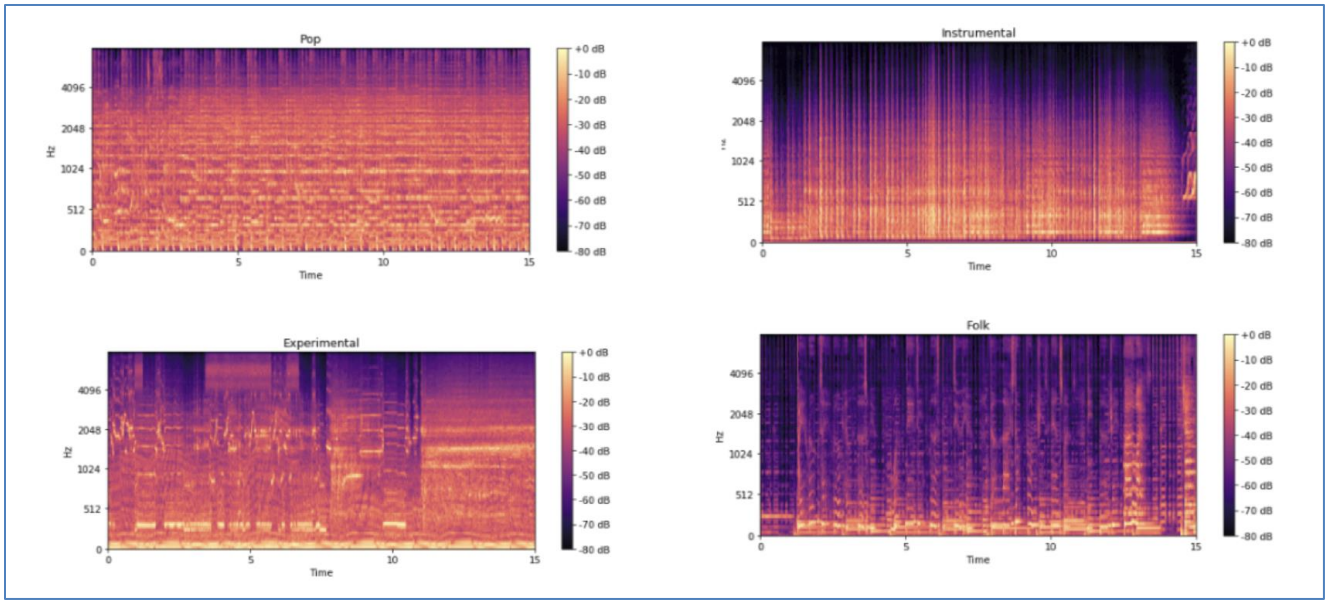

Figure 6: Spectrograms of Audio Files

A CNN consists of convolution and pooling layers. The convolution layers are paired with pooling layers in a convolution block. Our CNN model consists of 5 such blocks. Each conv block also has a dropout mechanism to prevent overfitting. The convolutional filters of our CNN model have the sizes of 16, 32, 64, 128 and last convolutional filter is of size 64. The max pooling filter is a matrix of size 2 * 2. The last layer in CNN is densely connected, using which the model outputs the probability that a given image belongs to each of the possible classes. The activation function chosen is relu.

Hyperparameter Tuning

In order to get the optimal parameters for our machine learning classifiers and MLP model we used sklearn GridSearchCV. Grid search is the process of performing hyper parameter tuning in order to determine the optimal values for a given model. This is significant as the performance of the entire model is based on the hyper parameter values specified. For example, below is the optimal parameter set returned by GridSearchCV for MLP.

{

'activation': 'identity',

'alpha': 0.05,

'hidden_layer_sizes': (15,),

'learning_rate': 'constant',

'solver': 'lbfgs'

}

Experiments and Results

We are presenting all our analysis by dividing it into 3 experiments. Experiment 1 includes the results we got and the analysis we did on all the 10 genres, thus our first set of experiment considers all the 10 genres and gives the comparison of the models - RFC, SVM and MLP. While it gives a comparison across the models, it also gives a comparison of the performance of these models individually i.e. their performance with and without feature reduction. Experiment 2 shows a similar analysis as above but by considering only 5 genres. Lastly, experiment 3 includes our results achieved from training and testing a CNN. For all the above experiments, we split our dataset into 80% train and 20% test and for experiment 1 and 2 we did a 5-fold cross validation on the training data to better train the models.

1) Experiment 1 - All 10 genres

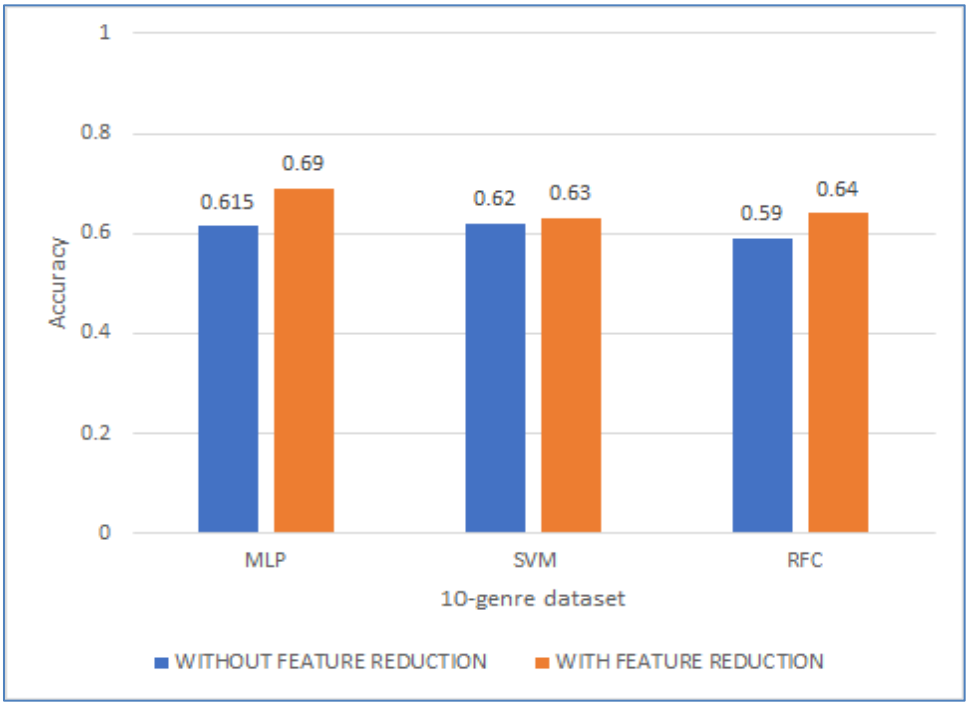

Figure 7: Accuracy on 10 Genre Dataset

Analysis:

1a) From the above bar chart, we can clearly see that for all the models the results obtained from PCA are better than the results when it is not applied. For SVM the results are pretty much the same but for in case of MLP and RFC there is a performance boost.

As accuracy is not an optimal parameter, we used confusion matrix and ROC curves as well as our metrics to determine the overall performance of the models. The actual genres were label encoded and the ROC curve shows the encoded labels. The encoding was done as below,

GENRE	LABEL
blues	0
classical	1
country	2
disco	3
hiphop	4
jazz	5
metal	6
pop	7
reggae	8
rock	9

Table 1: Genre and Label

The confusion matrix and ROC curve for the relatively best classifier MLP is shown below.

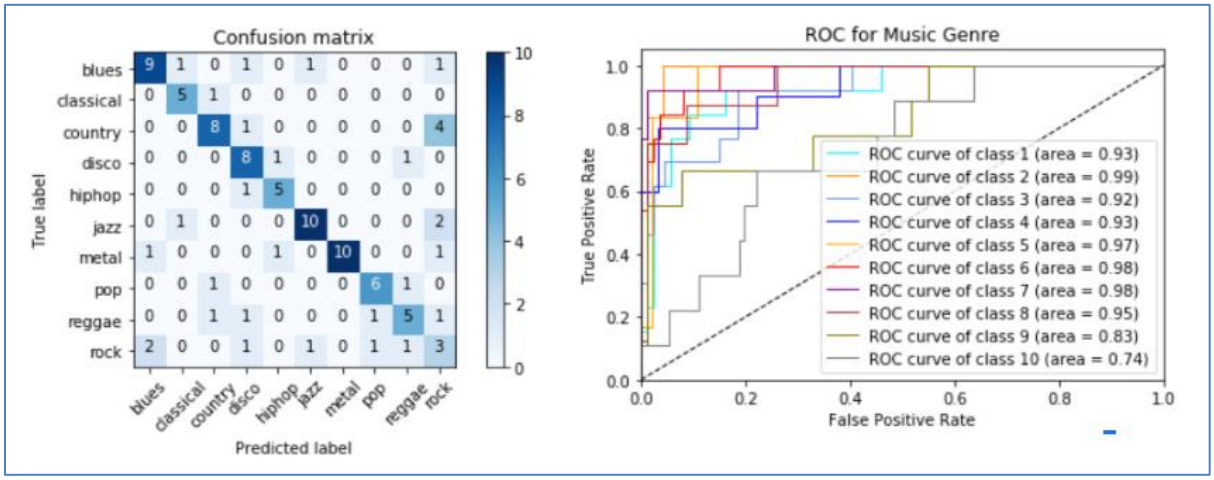

Figure 8: Confusion Matrix and ROC for MLP Classifier

1b) From the above we can see that the AUC value for each class is pretty good. The lowest value is for class 9 which corresponds to the rock genre. The good AUC results for majority of the classes was surprising as because we did not get good accuracy, we expected our AUC values to be low. But by looking at the confusion matrix we can see that the number of test files in each genre are less, which is because of the small size of the dataset. Thus, when considered individually the classifier seems to perform well on individual classes but when the average performance across all the genres is considered, the results are not good (accuracy is less).

Experiment 2 : On 5 genres

As the accuracy results in experiment 1 were not good we followed another approach which was used in a recent study [7] and that was to reduce the genres. For this experiment we only considered 5 genres - classical, hiphop, jazz, metal and pop. After reducing the genres, the accuracy obtained by the three models is shown below,

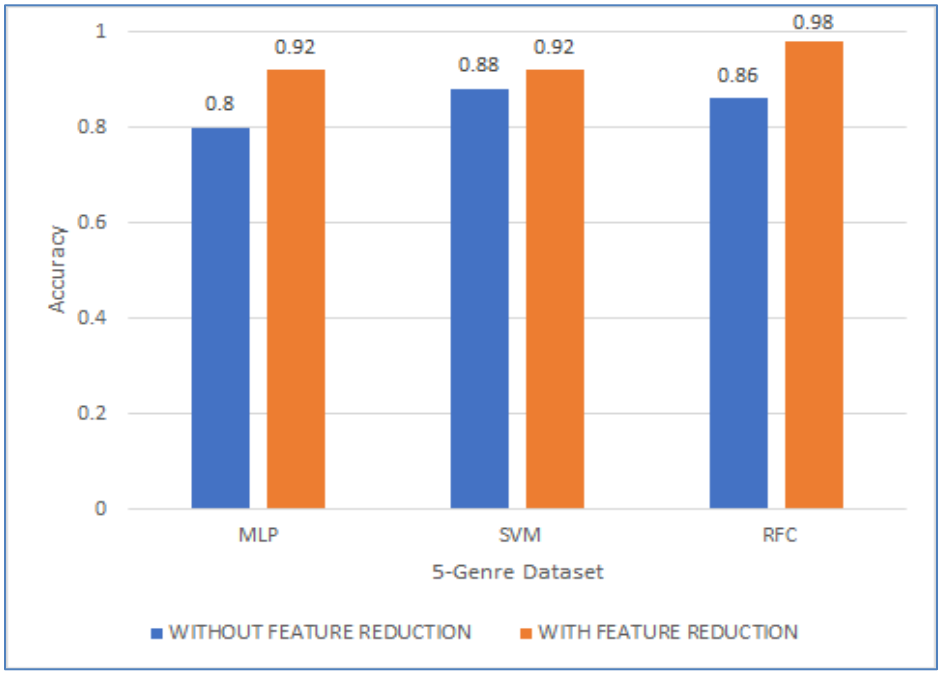

Figure 9: Accuracy on 5 Genre Dataset

Analysis:

2a) Comparing the above chart to the chart in experiment 1 we can see a significant rise in the accuracy of all the models. The below tables make it easier to see how the accuracy of all the models has improved when we reduced the genres from 10 to 5. The above analysis is applicable to both cases when feature reduction is applied and is not.

WITHOUT FEATURE REDUCTION			
MODEL	MLP	SVM	RFC
10-GENRE	0.615	0.62	0.59
5-GENRE	0.8	0.88	0.86

Table 2: Accuracy of models without PCA

WITH FEATURE REDUCTION			
MODEL	MLP	SVM	RFC
10-GENRE	0.69	0.63	0.64
5-GENRE	0.92	0.92	0.98

Table 3: Accuracy of models with PCA

2b) Analogous to the analysis in part 1a, where we said that PCA helped improve the accuracy, here as well we can see that using PCA the results obtained are higher for all the models. The confusion matrix and ROC curve for the best classifier RFC in case of 5 genres is shown below,

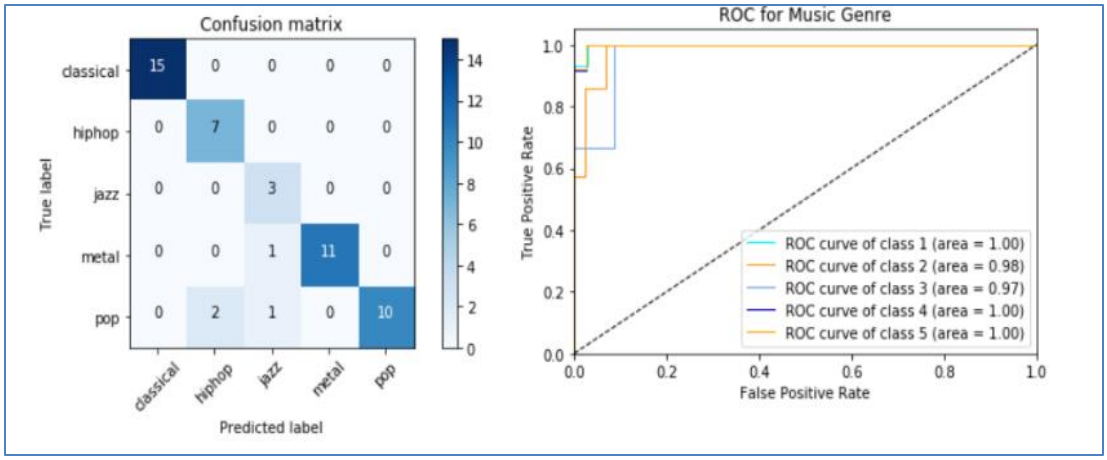

Figure 10: Confusion Matrix and ROC for RFC Classifier

The AUC obtained for each of the classes is high and the diagonal of the confusion matrix as well has higher values which indicates that the classifier performed well.

Experiment 3 - CNN model

The results we obtained from our CNN model are shown in the below chart. We can clearly see that CNN has outperformed all the other 3 models, in case of both 5 and 10 genres.

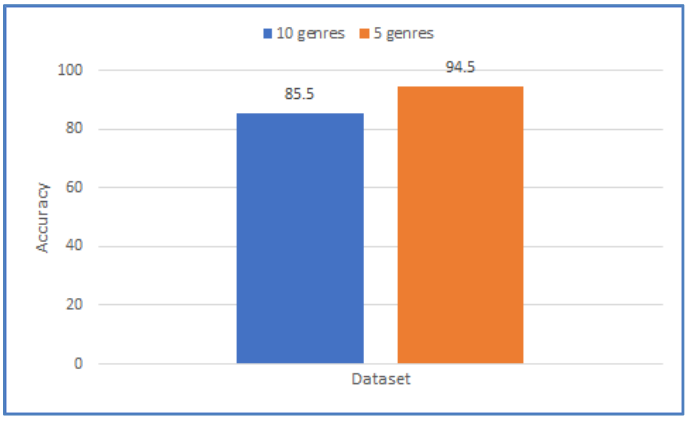

Figure 11: Accuracy of CNN on 10 genres and 5 genres

The accuracy and loss curve for CNN for both 10 and 5 genres are shown below,

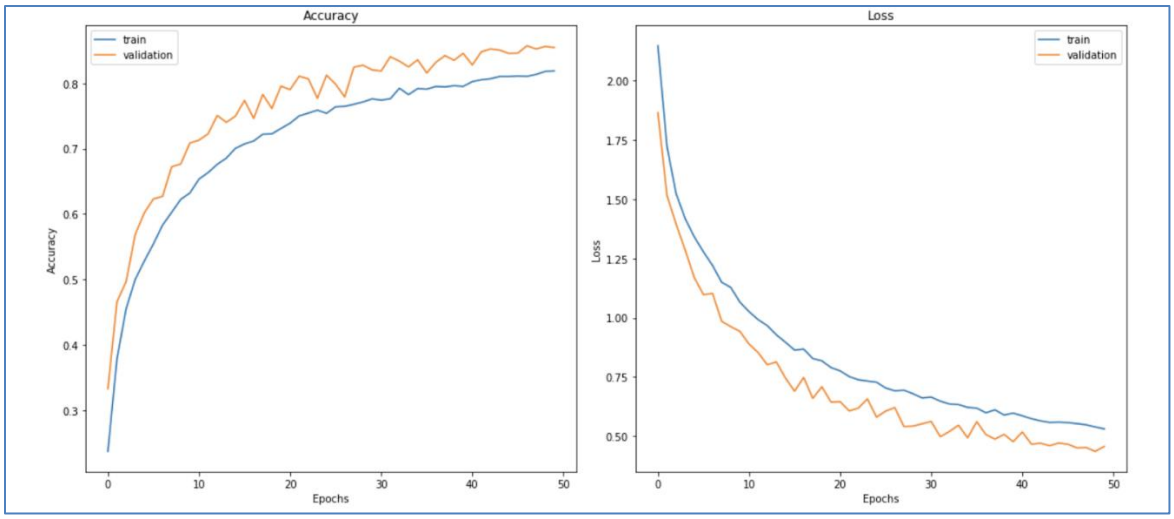

Figure 12: Accuracy and Loss curve for 10 genres

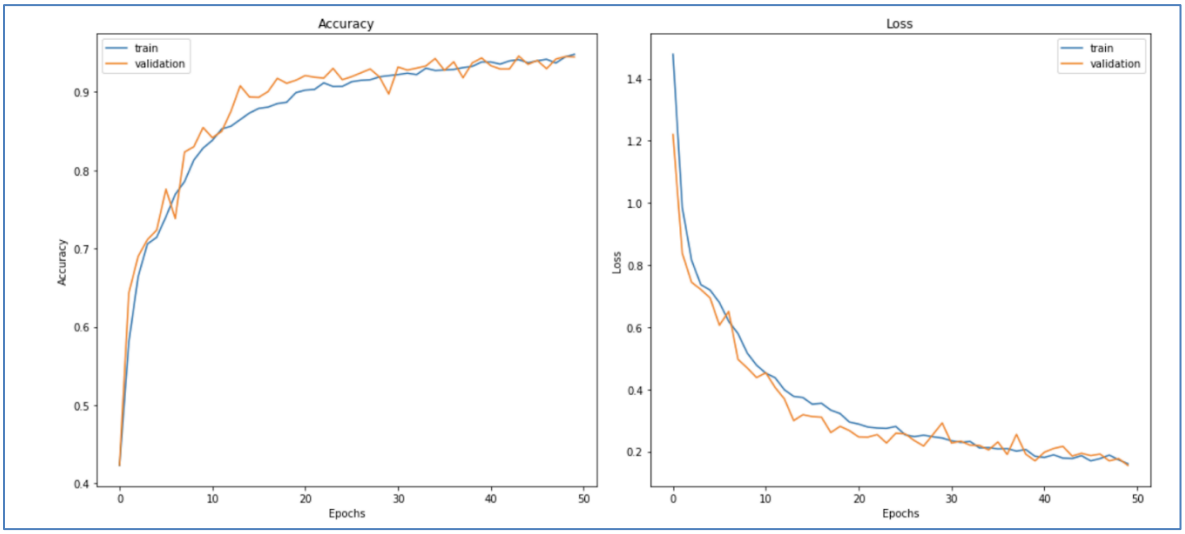

Figure 13: Accuracy and Loss curve for 5 genres

In both the cases we can see that the accuracy at the end of 50 epochs is maximum and the loss is minimum when the number of epochs reaches 50.

Conclusion

Convolutional Neural Network which is a deep learning model gave remarkable results and proved to be the best model for our dataset both in case of 10 and 5 genres. Traditional machine learning classifiers, RFC and SVM along with MLP did well when the genres were reduced to 5 but did not give good results when all the genres were considered. Feature reduction technique, PCA improved the results of RFC, SVM and MLP. The time domain-based music feature zero crossing rate and frequency domain based spectral features along with MFCC, were sufficiently good for our analysis. The advantage of training a CNN was that it did not require any hand-crafted features, but the drawback was the time it took to train. However, the outstanding results of CNN outweigh the time it took to train.

Future Work

As the dataset we used in this project was small it would be useful to perform this task of music genre classification on a bigger dataset where the number of audio files for each genre are considerably more. For our project we have used limited features, particularly MFCC's spectral features and zero crossing rate. Although these give a fair comparison of Machine Learning Algorithms, exploring the effectiveness of different features would help me to determine which machine learning stack does best in Music Classification.

REFERENCES

1. Analytics Vidhya, Getting Started with Audio Data Analysis using Deep Learning (with case study) [Online]] https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/
2. Masrsyas , GTZAN dataset [Online] http://marsyas.info/downloads/datasets.html
3. Logan, Mel Fequency Cepstral Coefficients for Music Modeling, 2000, http://ismir2000.ismir.net/papers/logan_paper.pdf
4. Banitalebi-Dehkordi, Mehdi & Banitalebi Dehkordi, Amin. (2014). Music Genre Classification Using Spectral Analysis and Sparse Representation of the Signals. Journal of Signal Processing Systems. 74. 10.1007/s11265-013-0797-4.
5. Towards Data Science, Music Feature Extraction in Python [Onlin] https://towardsdatascience.com/extract-features-of-music-75a3f9bc265d
6. Breebaart, Jeroen & McKinney, Martin. (2004). Features for Audio Classification. 10.1007/978-94-017-0703-9.
7. D. P. Kumar, B. J. Sowmya, Chetan and K. G. Srinivasa, "A comparative study of classifiers for music genre classification based on feature extractors," *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Mangalore, 2016, pp. 190-194. doi: 10.1109/DISCOVER.2016.7806258